Capturing Polynomial Time and Logarithmic Space using Modular Decompositions and Limited Recursion

DISSERTATION

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät der
Humboldt-Universität zu Berlin

von
Dipl.-Inf. Berit Grußien

Gefördert durch die Deutsche Forschungsgemeinschaft im Rahmen des Graduiertenkollegs
'Methods for Discrete Structures' (GRK 1408)

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Jan-Hendrik Olbertz

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:
Prof. Dr. Elmar Kulke

Gutachter/innen:
1. Prof. Dr. Martin Grohe
2. Prof. Dr. Nicole Schweikardt
3. Prof. Dr. Johannes Köbler

Eingereicht am: 23. März 2016

Tag der mündlichen Prüfung: 2. Dezember 2016

Abstract

Descriptive complexity theory is concerned with the characterization of complexity classes by means of suitable logics. A central open question is whether there exists a logic that characterizes, or *captures*, the complexity class polynomial time (PTIME) on the class of all graphs. A promising step towards finding an answer is the recent result of Grohe that fixed-point logic with counting (FP+C) captures PTIME on all classes of graphs with excluded minors. In this thesis we present another step. We consider classes of graphs with excluded induced subgraphs. We show that FP+C captures PTIME on the class of permutation graphs and on the class of chordal comparability graphs. The results are based on a graph decomposition, known as modular decomposition, which was introduced by Gallai in 1976. The graphs that are non-decomposable with respect to modular decomposition are called *prime*. For graph classes $\mathcal{C}$ that are closed under taking induced subgraphs, we prove the Modular Decomposition Theorem. It reduces (definable) canonization of $\mathcal{C}$ to (definable) canonization of the class of prime graphs of $\mathcal{C}$ that are colored with binary relations on a linearly ordered set. Our capturing results for permutation graphs and chordal comparability graphs follow from an application of the Modular Decomposition Theorem and reveal its strength. We also show that the modular decomposition of a graph is definable in symmetric transitive closure logic with counting (STC+C). As a side result, we obtain a logarithmic-space algorithm for computing the modular decomposition tree.

Further, we turn our attention to the complexity class logarithmic space (LOGSPACE), and introduce a new logic for LOGSPACE. We extend first-order logic with counting by a new operator that allows it to formalize a limited form of recursion which can be evaluated in logarithmic space. The data complexity of the resulting logic LREC is in LOGSPACE. Furthermore, LREC defines LOGSPACE-complete problems such as deterministic reachability and Boolean formula evaluation. We prove that LREC is strictly more expressive than deterministic transitive closure logic with counting (DTC+C) and that it is strictly contained in FP+C. Its expressive power is incomparable with symmetric transitive closure logic (STC) and transitive closure logic (TC) (with or without counting). We show that LREC captures LOGSPACE on the class of directed trees. We also study an extension $\mathsf{LREC}_=$ of LREC that has nicer closure properties and is more expressive than both LREC and STC. The data complexity of $\mathsf{LREC}_=$ is still in LOGSPACE, and $\mathsf{LREC}_=$ is contained in FP+C as well. We prove that $\mathsf{LREC}_=$ captures LOGSPACE on the class of interval graphs and the class of chordal claw-free graphs.

Zusammenfassung

Ziel der deskriptiven Komplexitätstheorie ist es, Komplexitätsklassen mit Hilfe von geeigneten Logiken zu charakterisieren. Eine auch aus praktischer Sicht besonders wichtige Komplexitätsklasse ist die Klasse der Polynomialzeit-Eigenschaften (PTIME). Wir beschäftigen uns mit der ungelösten Frage, ob es eine Logik gibt, welche PTIME auf der Klasse aller Graphen charakterisiert. Eine Herangehensweise an dieses Problem ist es, eingeschränkte Graphklassen zu betrachten. So wurde von Grohe gezeigt, dass PTIME auf allen Graphklassen mit verbotenen Minoren durch Fixpunktlogik mit Zählen (FP+C) charakterisiert wird. In dieser Arbeit betrachten wir Graphklassen mit verbotenen induzierten Teilgraphen. Wir beweisen, dass FP+C die Komplexitätsklasse PTIME auf der Klasse aller Permutationsgraphen und auf der Klasse aller chordalen Komparabilitätsgraphen charakterisiert. Unsere Resultate basieren auf der Zerlegung von Graphen in Module, welche 1976 von Gallai eingeführt wurde. Graphen, die durch modulare Zerlegung nicht zerlegbar sind, heißen *prim*. Für Graphklassen $\mathcal{C}$, die unter induzierten Subgraphen abgeschlossen sind, beweisen wir das Modulare Zerlegungstheorem. Dieses reduziert (definierbare) Kanonisierung der Graphklasse $\mathcal{C}$ auf (definierbare) Kanonisierung der Klasse aller primen Graphen aus $\mathcal{C}$, die mit binären Relationen auf einer linear geordneten Menge gefärbt sind. Unsere Resultate für Permutationsgraphen und chordale Komparabilitätsgraphen folgen aus dem Modularen Zerlegungstheorem. Wir zeigen zudem, dass die modulare Zerlegung eines Graphen in Symmetrisch-Transitive-Hüllen-Logik mit Zählen (STC+C) definiert werden kann. Als Folgerung erhalten wir einen Algorithmus, der mit logarithmischer Platzbeschränkung modulare Zerlegungsbäume berechnet.

Weiterhin definieren wir eine neue Logik für die Komplexitätsklasse Logarithmischer Platz (LOGSPACE). Die Logik LREC erweitert die Logik erster Stufe mit Zählen (FO+C) um einen Operator, der eine beschränkte und in logarithmischem Platz berechenbare Form der Rekursion erlaubt. Die Datenkomplexität von LREC liegt in LOGSPACE. Zudem können LOGSPACE-vollständige Probleme wie deterministische Erreichbarkeit und die Auswertung Boolescher Formeln in LREC definiert werden. Wir beweisen, dass LREC echt ausdrucksstärker als Deterministisch-Transitive-Hüllen-Logik mit Zählen (DTC+C) ist, und dass LREC echt in FP+C enthalten ist. Andererseits ist die Ausdrucksstärke von LREC weder mit der von Symmetrisch-Transitive-Hüllen-Logik (STC) noch mit der von Transitive-Hüllen-Logik (TC) vergleichbar (dies gilt auch für die Erweiterungen dieser Logiken mit Zählen). Wir zeigen, dass LREC die Komplexitätsklasse LOGSPACE auf gerichteten Bäumen charakterisiert. Zudem betrachten wir eine Erweiterung $\mathsf{LREC}_=$ von LREC. Die Datenkomplexität von $\mathsf{LREC}_=$ liegt in LOGSPACE, und wie LREC ist $\mathsf{LREC}_=$ auch in FP+C enthalten. Die Logik $\mathsf{LREC}_=$ zeichnet sich jedoch durch bessere Abschlusseigenschaften als LREC aus und ist ausdrucksstärker als LREC und STC. Wir beweisen, dass $\mathsf{LREC}_=$ die Komplexitätsklasse LOGSPACE sowohl auf Intervallgraphen als auch auf chordalen klauenfreien Graphen charakterisiert.

Acknowledgments

First and foremost, I would like to thank my supervisor Martin Grohe for his guidance and support, and Nicole Schweikardt for her contribution in supervising my thesis after Martin's move to Aachen. I am grateful to these two and Johannes Köbler for reviewing this thesis. Special thanks also go to my coauthors Martin Grohe, André Hernich and Bastian Laubner.

In addition, I would like to thank the following people for proofreading parts of this thesis and comments: Lucas Heimberg, Frank Fuhlbrück, Robert Prüfer, Pascal Lenzner, Oleg Verbitsky and Sebastian Kuhnert.

I would like to give particular thanks to Sven Strickroth and Volker Diels-Grabsch for keeping me company at a lot of weekends at the university and days in the library, Lucas Heimberg for ensuring a steady cookie supply, and my family for their kind support throughout all this time. Last but not least, I would like to thank Eva Sandig, Petra Kämpfer and my former and current colleagues for their support and for being there for all sort of concerns.

Contents

1. Introduction **1**

2. Preliminaries **5**
2.1. General Notation . . . 5
2.2. Relations and Orders . . . 5
2.3. Graphs and Structures . . . 6
2.3.1. Structures . . . 6
2.3.2. Graphs and Basic Graph Notions . . . 7
2.3.3. Forests and Trees . . . 8
2.3.4. Colorings and LO-Colored Graphs . . . 9
2.4. Logics . . . 10
2.5. Transductions . . . 12
2.5.1. Transductions . . . 13
2.5.2. Parameterized Transductions . . . 16
2.5.3. Composition . . . 20
2.5.4. Counting Transductions . . . 20
2.6. Canonization . . . 22
2.7. Descriptive Complexity . . . 25
2.7.1. Turing Machines and Complexity Classes . . . 25
2.7.2. Capturing Complexity Classes . . . 26
2.8. Graph-Theoretical Preliminaries . . . 27
2.8.1. Depth-First Tree Traversal . . . 27
2.8.2. Max Cliques and Spanning Vertices . . . 28
2.8.3. Centroids . . . 29

I. The Modular Decomposition Theorem and Applications **31**

3. STC+C-Definability of the Modular Decomposition **35**
3.1. Modules and their Basic Properties . . . 35
3.2. Modular Decomposition . . . 37
3.3. Spanned Modules and (W)edge Classes . . . 38
3.4. Defining the Modular Decomposition in STC+C . . . 41
3.5. A Logspace Algorithm for the Modular Decomposition Tree . . . 47

4. The Modular Decomposition Theorem **49**
4.1. Modular Contraction . . . 50
4.2. The Representation of a Graph . . . 51
4.3. The Modular Decomposition Theorem . . . 52
4.4. Proof of the Modular Decomposition Theorem . . . 53

5. Capturing PTIME on Permutation Graphs **62**
5.1. Permutation Graphs 62
5.2. Capturing Result 65

6. Capturing PTIME on Chordal Comparability Graphs **72**
6.1. Chordal Comparability Graphs 72
6.2. Modular Decomposition Theorem - Application 72
6.3. The Graph's Structure 73
6.3.1. Max Cliques 73
6.3.2. Ends and the Bundle Tree 78
6.3.3. Inner and Outer Ends and Max Cliques 79
6.3.4. Inner Ends 82
6.3.5. The Sets $S_1^{\prec}$ and $S_2^{\prec}$ 84
6.3.6. Sides of Inner Ends 87
6.3.7. Outer Ends 91
6.3.8. Sides of Outer Ends 93
6.3.9. Sides and the Middle of a Graph 95
6.3.10. Side Depth and Side Trees 98
6.4. The Bundle Extension and Extended Valid Subgraphs 102
6.4.1. The O-Extension 103
6.4.2. The Bundle Extension 108
6.4.3. Subbundle Pairs 114
6.4.4. Valid Triples and (Extended) Valid Subgraphs 117
6.5. The Genealogical Decomposition Tree 124
6.5.1. The Simplified Decomposition Tree 125
6.5.2. The Decomposition Tree of a Block 127
6.5.3. The Decomposition Tree of a Valid Subgraph 139
6.5.4. The Genealogical Decomposition Tree 145
6.6. Canonization 155
6.6.1. Preliminaries 155
6.6.2. The Set $\mathcal{W}(q)$ 157
6.6.3. Decomposing $\mathcal{W}(q)$ for Non-Minimal Affiliated Consistent Pairs q . 161
6.6.4. Decomposing $\mathcal{W}(q)$ for Trivial Affiliated Subbundle Pairs q 167
6.6.5. The Set $\mathcal{W}_{\texttt{anc}}(q)$ and its Decomposition 172
6.6.6. Canonization 175
6.6.7. Defining the Extended Copy 179

II. L-Recursion and a New Logic for Logarithmic Space **187**

7. The Logic LREC **191**

8. Capturing LOGSPACE on Directed Trees **202**
8.1. Directed Tree Isomorphism 202
8.2. Defining an Order on Directed Trees 206
8.3. Canonizing Directed Trees 210
8.4. Colored Directed Trees 211

9. Inexpressibility of Reachability in Undirected Graphs 213
9.1. The Logic $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ 214
9.2. Translation of LREC-Formulas into $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-Formulas 216

10. LREC$_=$ - An Extension of LREC 221

11. Capturing LOGSPACE on Interval Graphs 224
11.1. Background on Interval Graphs 224
11.2. Twinless Modular Decompositions 225
11.3. Extracting Information About the Order of Maximal Cliques 226
11.4. Modules $\mathcal{W}_G$ and the Graph L_G 228
11.5. The Colored Twinless Modular Decomposition Tree 233
11.6. Canonization 241

12. Capturing LOGSPACE on Chordal Claw-Free Graphs 245
12.1. Introduction of Clique Trees 246
12.2. Uniqueness of the Clique Tree for Connected Graphs 249
12.3. Structure of the Clique Tree 250
12.4. Defining the Clique Tree in FO 256
12.5. Directed Clique Trees 257
12.6. The Supplemented Clique Tree 258
12.7. Canonization 259
12.8. Algorithm for Computing the Max Cliques of the Canon 260

13. Conclusion 271

A. Appendix 273
A.1. Proofs of Properties of Transductions 273
A.1.1. Proof of the Transduction Lemma 273
A.1.2. Proof of Proposition 12 (Composition of Transductions) 277
A.2. Proof omitted in Section 9 279

Bibliography 281

1. Introduction

In computational complexity theory problems are classified into complexity classes based on the resources that an abstract machine needs to solve them. Descriptive complexity theory, in contrast, is concerned with classifying problems based on the expressive power that is needed in order to describe them. In particular, descriptive complexity theory aims to find logics that characterize the standard complexity classes of computational complexity theory. By this means, one hopes to gain insight into the inherent structure of the problems contained in a certain complexity class.

Computational complexity theory provides the following inclusions between the complexity classes logarithmic space (LOGSPACE), non-deterministic logarithmic space (NL), polynomial time (PTIME) and non-deterministic polynomial time (NP):

$$\mathsf{LOGSPACE} \subseteq \mathsf{NL} \subseteq \mathsf{PTIME} \subseteq \mathsf{NP}.$$

Until now, it is unknown whether any of these inclusions are strict. Deciding whether the rearmost inclusion is strict corresponds to the famous open question of whether PTIME is equal to NP. This question is not only interesting from a theoretical point of view, but also has consequences for practical computational problems. While PTIME is commonly accepted as a good theoretical model of what can be computed efficiently, the class NP contains a variety of problems of practical importance for which it is unknown whether they can be solved efficiently.

Besides of giving a new perspective on the constitution of complexity classes, characterizations of complexity classes by means of suitable logics provide new prospects on the comparison and, possibly, separation of these complexity classes. In 1974 Fagin showed that a problem is in NP if and only if it can be defined in existential second-order logic (∃SO) [20], that is, first-order logic extended by existential quantification over relations. In short, we say that ∃SO *captures* NP. Naturally, the question was raised whether there also is a logic that captures PTIME (Gurevich, 1984 [35]).[1] If such a logic exists, methods from logic and model theory could be applied to separate PTIME and NP or to show that they coincide. Clearly, PTIME is not equal to NP if such a logic does not exist. The same holds for the complexity classes LOGSPACE and NL. A logical characterization of LOGSPACE (or NL) could pave the way for separating LOGSPACE or NL from NP.

This thesis contributes to the quest for logical characterizations for the complexity classes PTIME and LOGSPACE.

[1] Note that Chandra and Harel asked a similar question from the perspective of database theory in 1982 [8]

Capturing PTIME

Independently of each other, Immerman [39] and Vardi [66] obtained an early result towards a logical characterization for PTIME. They proved that fixed-point logic (FP) captures PTIME on ordered structures,[2] that is, on structures that have a distinguished binary relation which linearly orders the elements of the structure. However, a simple counterexample shows that this does not generalize to arbitrary, that is, not necessarily ordered, structures: Although the class of structures whose universe has even cardinality can clearly be decided in PTIME, it cannot be defined in FP [17]. Thus, in 1987 Immerman proposed to add to FP the ability to count [40], and proved, together with Lander, that the resulting logic FP+C captures PTIME on the class of trees [44]. However, in 1992 Cai, Führer and Immerman showed that also FP+C does not suffices to capture PTIME on arbitrary structures [7].

There are two basic strategies for finding a logic that captures PTIME. The first strategy is to develop new logics whose expressive power converges towards PTIME. This is not an easy task and since 1992 only few more logics have been introduced towards this end. Examples of these are Choiceless Polynomial Time C̃PT [3] and rank logics [14, 26]. The second strategy is to find restricted graph classes on which already known logics capture PTIME. In the first part of this thesis, we follow the second approach.

It is known that FP+C captures PTIME on planar graphs [27], all classes of graphs of bounded tree width [34], as well as K_5-minor free graphs [28]. Note that all these classes can be defined by a list of forbidden minors. In fact, Grohe recently showed that FP+C captures PTIME on all classes of graphs with excluded minors [30]. This leads to the question whether a similar result can be obtained for classes of graphs that are characterized by a (finite or infinite) list of forbidden induced subgraphs. Grohe showed that capturing PTIME on the class of chordal graphs[3] is as hard as capturing PTIME on all graphs for any "reasonable" logic [29]. Thus, for FP+C a general result which captures PTIME on all graphs with forbidden induced subgraphs is not possible. However, there are partial results showing that FP+C captures PTIME on the class of interval graphs [49] and on the class of chordal line graphs [29].

This thesis, showing that FP+C captures PTIME also on the class of permutation graphs and on the class of chordal comparability graphs, adds to this line of research. Both results are based on modular decomposition, a graph decomposition which was introduced by Gallai in 1976 [21]. The modular decomposition of a graph partitions the vertex set of the graph into so called modules, that is, into subsets that share the same neighbors. A graph is *prime* if only the vertex set itself and all vertex sets of size 1 are modules of the graph. For every class $\mathcal{C}$ of graphs that is closed under induced subgraphs, we let $\mathcal{C}^*$ be the class of all prime graphs from $\mathcal{C}$ that are colored with binary relations on a linearly ordered set. Our Modular Decomposition Theorem states that there is an FP+C-canonization of $\mathcal{C}$ if there is an FP+C-canonization of the class $\mathcal{C}^*$. It follows that FP+C captures PTIME on $\mathcal{C}$ if FP+C captures PTIME on $\mathcal{C}^*$. To prove the Modular Decomposition Theorem, we show that the modular decomposition of a graph is definable

[2] More precisely, Immerman and Vardi's theorem holds for least fixed-point logic (LFP) and the equally expressive inflationary fixed-point logic (IFP). Our indeterminate FP refers to either of these two logics.

[3] For the class of chordal graphs the forbidden induced subgraphs are cycles of length at least 4.

in STC+C. This also proves that there exists a logarithmic-space algorithm that computes the modular decomposition of a graph. Note that the Modular Decomposition Theorem extends to all logics whose expressive power is as least as strong as FP+C's and which are closed under FP+C-reductions. Our capturing results for permutation graphs and chordal comparability graphs follow from an application of the Modular Decomposition Theorem.

Capturing LOGSPACE

In the second part of this thesis, we consider the complexity class LOGSPACE. Similar to PTIME, there is a capturing result for LOGSPACE on ordered structures: Immerman proved that deterministic transitive closure logic (DTC) captures LOGSPACE on ordered structures [41]. Much less is known for LOGSPACE on arbitrary structures. Recall that for PTIME, the logic FP, which captures PTIME on ordered structures, was equipped with counting operators to obtain the logic FP+C, which captures PTIME on certain interesting graph classes. Thus, an obvious idea is to capture LOGSPACE with the extension DTC+C of DTC by counting operators. However, Etessami and Immerman proved that (directed) tree isomorphism is not definable in DTC+C, and not even definable in the stronger transitive closure logic with counting (TC+C) [18]. Since Lindell [53] showed that tree isomorphism is decidable in logarithmic space, it follows that DTC+C does not even capture LOGSPACE on the class of all trees.

We introduce a new logic LREC and prove that it captures LOGSPACE on directed trees. Furthermore, we extend LREC to a logic $\mathsf{LREC}_=$ that captures LOGSPACE on the class of interval graphs and on the class of chordal claw-free graphs. As a consequence, we obtain the first logical characterizations of LOGSPACE on non-trivial natural classes of unordered structures.

The logic LREC extends first-order logic with counting (FO+C) by an operator that allows limited recursion. The limited recursion operator bounds the recursion depth by a "resource term", and thereby makes sure that the recursive definition can be evaluated in logarithmic space. It is easy to see that LREC is (semantically) contained in FP+C. Furthermore, we show that LREC contains DTC+C. Since LREC captures LOGSPACE on directed trees, its expressive power exceeds the one of DTC+C, and LREC is not contained in STC+C or TC+C. We also prove that undirected graph reachability is not definable in LREC. It follows that LREC does not contain STC or TC, and is strictly contained in FP+C.

Apart from the incapability of LREC to express undirected graph reachability, another weakness of LREC is that it is not closed under (first-order) logical reductions. To remedy this weakness, we enhance the limited recursion operator of LREC, and thus obtain the logic $\mathsf{LREC}_=$. Due to this enhancement, undirected graph reachability is definable in $\mathsf{LREC}_=$, and therefore, $\mathsf{LREC}_=$ strictly contains STC+C. We prove that $\mathsf{LREC}_=$ captures LOGSPACE on the class of interval graphs. Further, we show that the class of chordal claw-free graphs admits $\mathsf{LREC}_=$-definable canonization. This implies that there is a logarithmic-space algorithm for computing a canonization mapping for chordal claw-free graphs, and that $\mathsf{LREC}_=$ captures LOGSPACE on the class of chordal claw-free graphs. Since $\mathsf{LREC}_=$ is contained in FP+C, we also obtain that FP+C captures PTIME on the class of chordal claw-free graphs

Structure of this Thesis

After giving the necessary preliminaries in Chapter 2, this thesis consists of the two already outlined parts.

In the first part, we focus on PTIME. In Chapter 3, we introduce the modular decomposition and show its STC+C-definability. In Chapter 4, we prove the Modular Decomposition Theorem. Finally, we apply the Modular Decomposition Theorem in Chapters 5 and 6 to show that FP+C captures PTIME on the class of permutation graphs and on the class of chordal comparability graphs, respectively.

In the second part of the thesis, we focus on LOGSPACE. In Chapter 7, we introduce the logic LREC and show in Chapter 8 that LREC captures LOGSPACE on directed trees. We prove in Chapter 9 that undirected graph reachability is not definable in LREC. Chapter 10 defines the logic $\mathsf{LREC}_=$, and Chapters 11 and 12 show that $\mathsf{LREC}_=$ captures LOGSPACE on the class of interval graphs and on the class of chordal claw-free graphs, respectively.

Finally, Chapter 13 provides a conclusion of the thesis.

About this Thesis

The research presented in Chapters 7–11 was done in collaboration with Martin Grohe, André Hernich and Bastian Laubner. It has been published in [32] and [33].

2. Preliminaries

2.1. General Notation

$\mathbb{Z}$, $\mathbb{N}$ and $\mathbb{N}^+$ denote the sets of all integers, non-negative integers and positive integers, respectively. For all $n, n' \in \mathbb{Z}$, we define $[n, n'] := \{m \in \mathbb{Z} \mid n \leq m \leq n'\}$ and $[n] := [1, n]$. We often denote tuples $(a_1, \dots, a_k)$ by $\bar{a}$. Given a tuple $\bar{a} = (a_1, \dots, a_k)$, let $\tilde{a} := \{a_1, \dots, a_k\}$. By $|\bar{a}|$ we denote the length of the tuple $\bar{a}$. Let $n \geq 1$, and $\bar{a}^i = (a^i_1, \dots, a^i_{k_i})$ be a tuple of length k_i for each $i \in [n]$. We denote the tuple $(a^1_1, \dots, a^1_{k_1}, \dots, a^l_1, \dots, a^l_{k_l})$ by $(\bar{a}^1, \dots, \bar{a}^l)$. Mappings $f\colon A \to B$ are extended to tuples $\bar{a} = (a_1, \dots, a_k)$ over A via $f(\bar{a}) := (f(a_1), \dots, f(a_k))$. For a subset $A' \subseteq A$, we let $f(A') := \{f(a) \mid a \in A'\}$. For a set S, we denote the cardinality of S by $|S|$. A *singleton* is a set S with $|S| = 1$. We let $\mathcal{P}(S)$ be the set of all subsets of S and $\binom{S}{2}$ be the set of all subsets S' of S with $|S'| = 2$. If $\mathcal{S}$ is a set of sets, then $\bigcup \mathcal{S}$ and $\dot{\bigcup} \mathcal{S}$, respectively, denote the union and the disjoint union of all sets in $\mathcal{S}$. The disjoint union of two sets S and S' is denoted by $S \mathbin{\dot\cup} S'$. A *partition* of a set S is a set $\mathcal{S}$ of disjoint non-empty subsets of S such that $S = \bigcup \mathcal{S}$.

2.2. Relations and Orders

Let $n \in \mathbb{N}^+$. An n-ary *relation* on a set U is a subset R of U^n. The *restriction* $R_{|A}$ of an n-ary relation R on U to a subset $A \subseteq U$ is the relation $R' := R \cap A^n$. A binary relation R is

- *reflexive* if $(a, a) \in R$ for all $a \in U$,
- *irreflexive* if $(a, a) \notin R$ for all $a \in U$,
- *transitive* if $(a, b) \in R$ and $(b, c) \in R$ imply $(a, c) \in R$ for all $a, b, c \in U$,
- *symmetric* if $(a, b) \in R$ implies $(b, a) \in R$ for all $a, b \in U$,
- *asymmetric* if $(a, b) \in R$ implies $(b, a) \notin R$ for all $a, b \in U$,
- *antisymmetric* if $(a, b) \in R$ and $(b, a) \in R$ imply $a = b$ for all $a, b \in U$,
- *total* if $(a, b) \in R$ or $(b, a) \in R$ for all $a, b \in U$,
- *connex* if $(a, b) \in R$ or $(b, a) \in R$ or $a = b$ for all $a, b \in U$.

For binary relations R on U, we also denote $(a, b) \in R$ by $a \mathrel{R} b$.

Given a set U, a reflexive, transitive, symmetric binary relation $\approx$ is called an *equivalence relation* on U. For each $a \in U$ we denote the equivalence class $\{a' \in U \mid a' \approx a\}$ of a by $a/_{\approx}$. (We also use another notation, which we specify later.) For a k-ary relation $R \subseteq U^k$ we let $R/_{\approx}$ be the set $\{(a_1/_{\approx}, \dots, a_k/_{\approx}) \mid (a_1, \dots, a_k) \in R\}$. The set $U/_{\approx}$ of equivalence classes is a partition of U.

The *symmetric closure* of a binary relation R on U is the smallest (with respect to inclusion) relation R' on U such that $R \subseteq R'$ and R' is symmetric. Similarly, we define the *transitive closure* and the *reflexive, symmetric, transitive closure*. The reflexive, symmetric, transitive closure of a binary relation R on U is called the equivalence relation *generated* by R.

A *partial order* is a reflexive, transitive and antisymmetric binary relation $\preceq$. A binary relation $\preceq$ is a *linear order* if it is transitive, antisymmetric and total. For a subset $M \subseteq \mathbb{N}$ we denote the natural linear order on M by $\leq_M$. For each partial order and each linear order $\preceq$ there exists an associated irreflexive relation $\prec$, called a *strict partial order* and *strict linear order*, respectively, which is defined by $a \prec b$ if and only if $a \preceq b$ and $a \neq b$. Then $\prec$ is a strict partial order if and only if $\prec$ is irreflexive and transitive. Further, $\prec$ is a strict linear order if and only if $\prec$ is irreflexive, transitive and connex. A reflexive and transitive binary relation $\preceq$ is called a *preorder*. For a preorder $\preceq$ the associated irreflexive relation $\prec$, called a *strict preorder*, is defined by $a \prec b$ if and only if $a \preceq b$ and not $b \preceq a$. Then $\prec$ is a strict preorder if and only if $\prec$ is irreflexive and transitive.

Let $\preceq$ be a partial order on U. We call $a \in U$ $\preceq$*-minimal* if we have $a = b$ for all $b \in U$ with $b \preceq a$. Similarly, we define $\preceq$*-maximal*. For a strict partial order $\prec$, we define the $\prec$-minimal and $\prec$-maximal elements analogously. Thus, $a \in U$ is $\prec$*-minimal* if there is no $b \in U$ such that $b \prec a$. If it is clear what partial order $\preceq$ or strict partial order $\prec$ we are referring to, we simply call an element $a \in U$ *minimal* or *maximal*.

Let $\prec$ be a strict partial order on U. We say $a \in U$ and $b \in U$ are *comparable* with respect to $\prec$ if $a \prec b$ or $b \prec a$. If they are not comparable, we call them *incomparable*. A *strict weak order* is a strict partial order where incomparability is transitive. Moreover, in a strict weak order incomparability is an equivalence relation. Further, if a and b are incomparable with respect to strict weak order $\prec$, then $a \prec c$ implies $b \prec c$, and $c \prec a$ implies $c \prec b$. As a consequence, if $\prec$ is a strict weak order on U and $\approx$ is the equivalence relation defined by incomparability, then $\prec$ induces a strict linear order on $U/_{\approx}$.

The reverse of a binary relation $\preceq$ is the relation $\preceq^R := \{(b, a) \mid (a, b) \in \preceq\}$. Occasionally, we denote the reverse of a binary relation $\preceq$ by $\succeq$, that is, by mirroring the relation symbol. For (strict) partial or (strict) linear orders, the reverse remains a (strict) partial or (strict) linear order.

For $k \in \mathbb{N}$ and a set U with a linear order $\preceq$ on it, we define the *lexicographic extension* $\preceq_{\text{lex}}$ of $\preceq$ on U^k, that is, on all k-tuples of elements of U, as follows. For $\bar{a}, \bar{b} \in U^k$ with $\bar{a} = (a_1, \ldots, a_k)$ and $\bar{b} = (b_1, \ldots, b_k)$ we let $\bar{a} \preceq_{\text{lex}} \bar{b}$ if, and only if, $\bar{a} = \bar{b}$ or there is an $i \in [k]$ such that $a_i \prec b_i$ and $a_j = b_j$ for all $j < i$. For sets $A, B \subseteq U^k$ we let $A \preceq_{\text{lex}} B$ if and only if $A = B$ or there is a $\bar{b} \in B \setminus A$ such that for all $\bar{a} \in U^k$ with $\bar{a} \prec_{\text{lex}} \bar{b}$ we have $\bar{a} \in A \iff \bar{a} \in B$. It should be clear how to extend $\preceq$ to a linear order $\preceq_{\text{lex}}$ on tuples of subsets of U^k. We also call $\preceq_{\text{lex}}$ the *lexicographic order* if it is apparent from the context what linear order $\preceq$ we refer to.

2.3. Graphs and Structures

2.3.1. Structures

A *signature* or *vocabulary* is a finite set τ of relation symbols $R_1, R_2, \ldots$. Each relation symbol $R \in \tau$ has a fixed arity $\text{ar}(R) \in \mathbb{N}$. A structure A of signature τ, also called a

τ-structure, consists of a non-empty finite set $U(A)$, its *universe* or *domain*, and for each relation symbol $R \in \tau$ of a relation $R(A) \subseteq U(A)^{\mathrm{ar}(R)}$. We also denote the universe $U(A)$ and the relations $R(A)$ by U and R for $R \in \tau$ if it is clear which structure we are referring to. Let σ and τ be vocabularies such that $\sigma \subseteq \tau$. For a τ-structure A the σ-reduct is the σ-structure $A|_\sigma$ where $U(A|_\sigma) := U(A)$ and $R(A|_\sigma) := R(A)$ for each relation symbol $R \in \sigma$.

An *isomorphism* between τ-structures A and B is a bijection $f\colon U(A) \to U(B)$ such that for all $R \in \tau$ and all $a_1, \dots, a_{\mathrm{ar}(R)} \in U(A)$ we have $(a_1, \dots, a_{\mathrm{ar}(R)}) \in R(A)$ if and only if $(f(a_1), \dots, f(a_{\mathrm{ar}(R)})) \in R(B)$. We call τ-structures A and B *isomorphic* if there exists an isomorphism between them. We write $A \cong B$ to indicate that A and B are isomorphic. When we consider a class $\mathcal{C}$ of structures we always assume it to be *closed under isomorphism*, that is, whenever $A \in \mathcal{C}$ and B is isomorphic to A, then we also have $B \in \mathcal{C}$.

2.3.2. Graphs and Basic Graph Notions

In the following we introduce graphs and the basic graph notions. More on graphs can be found in [15].

A *directed graph* (or short a *digraph*) is a pair $G = (V, E)$ where V is a non-empty finite set and E is a subset of V^2. An *(undirected) graph* G is a pair (V, E) consisting of a non-empty finite set V and a set $E \subseteq \binom{V}{2}$ of 2-element subsets of V. We call V the vertices and E the edges of the directed or undirected graph G. Generally, the term graph refers to an undirected graph. However, when it is clear that the structure referred to is a directed graph, we occasionally also omit the term directed.

Let $\tau_E := \{E\}$ be a signature with binary relation symbol E. Each digraph corresponds to a τ_E-structure, where the universe is the vertex set and relation E the edge relation. We also understand every (undirected) graph as a τ_E-structure $G = (V, E)$ where E is an irreflexive and symmetric binary relation. We do not distinguish between the set of all edges (a subset of $\binom{V}{2}$) of an undirected graph and the corresponding edge relation (a subset of V^2).

If $\{v, w\} \in E$ is an edge of an undirected graph $G = (V, E)$, then v and w are *adjacent*, and w is a *neighbor* of v. The *degree* of a vertex $v \in V$ is the number of neighbors of v. For an edge $(v, w) \in E$ of a directed graph we say the edge is *directed* from v to w. An edge $(v, w) \in E$ is an *incoming edge* of w and an *outgoing edge* of v. Let $(v, w) \in E$. Then $v \in V$ is an *in-neighbor* of w, and w an *out-neighbor* of v. The *in-degree* (*out-degree*) of a vertex $v \in V$ is the number of in-neighbors (out-neighbors) of v.

The following definitions only refer to undirected graphs. We partly omit the analogous definitions for directed graphs, as (if necessary) they can be easily transferred from the given definitions. We only add definitions for directed graphs if they differ from the ones for undirected graphs.

Let $G = (V, E)$ and $H = (V', E')$ be graphs. The *union* $G \cup H$ of G and H is the graph $(V \cup V', E \cup E')$. If $V \cap V' = \emptyset$, then $G \cup H$ is called the *disjoint union* of G and H. The graph H is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E \cap \binom{V'}{2}$. For a subset $W \subseteq V$ of vertices, $G[W]$ denotes the *induced subgraph* $(W, E \cap \binom{W}{2})$ of G on W, and $G \setminus W$ denotes the induced subgraph $G[V \setminus W]$. The *complement graph* of G is $\overline{G} := (V, \overline{E})$ where $\overline{E} = \binom{V}{2} \setminus E$. The graph G is *complete* if $E = \binom{V}{2}$. A *clique* of G is a subset A of vertices such that $G[A]$ is complete. The graph G is called *edgeless* if G is the complement graph of a complete graph. A subset A of vertices of G is an *independent set* of G if $G[A]$ is edgeless.

A *simple path* P in $G = (V, E)$ is a sequence $v_0, v_1, \ldots, v_k$ of distinct vertices where $\{v_{i-1}, v_i\} \in E$ for all $i \in [k]$. Non-simple paths may contain vertices repeatedly. Unless stated otherwise, the paths we consider are simple paths. We call v_0 and v_k the *ends* of the path P. We also say $P = v_0, v_1, \ldots, v_k$ is a path from v_0 to v_k. Note that for undirected graphs a path from v_0 to v_k is also a path from v_k to v_0. The *length* of P is the number k of edges of P. We also understand a simple path $P = v_0, v_1, \ldots, v_k$ in G as a subgraph (V', E') of G where $V' = \{v_0, v_1, \ldots, v_k\}$ and $E' = \{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}\}$. A *subpath* of path P is a subgraph of P that is a path. A *cycle* C in G is a non-simple path $v_0, v_1, \ldots, v_k$ with $k \geq 3$ where $v_0 = v_k$ and $v_0, \ldots, v_k$ are distinct vertices. (For directed graphs we allow $k \geq 1$ in the definition of cycle. If $k = 1$, we call C a *loop*.) Again, we also regard cycles $C = v_0, v_1, \ldots, v_k$ in G as subgraphs of G. The *length* of C is the number k of edges of C. A graph G that does not contain a cycle is called *acyclic*.

Let $G = (V, E)$ be a graph. If there is a path in G from v to w, we say v and w are *connected* in G. The graph G is *connected* if v and w are connected for all vertices $v, w \in V$. (A directed graph G is connected if the graph that we obtain from G by deleting loops and interpreting every directed edge as an undirected edge is connected.) If $\overline{G}$ is connected, we call G *co-connected*. A set $W \subseteq V$ is *connected* (or *co-connected*) if $G[W]$ is connected (or co-connected). A *connected component* is a maximal (with respect to inclusion) connected subset $W \subseteq V$. Let G be connected. Then a *separator* is a set $S \subseteq V$ such that $G \setminus S$ is not connected. A separator S *separates* vertices $v, w \in V$ in G if v and w are in different connected components of $G \setminus S$.

2.3.3. Forests and Trees

A *forest* is an acyclic undirected graph. A connected forest is a *tree*. A *rooted tree* is a triple $T = (V, E, r)$ where (V, E) is a tree and $r \in V$ is a distinguished node called the *root*.

A *directed forest* is an acyclic directed graph where every vertex has an in-degree of at most 1. Often we will call the vertices of forests and directed forests *nodes*. A directed tree is a connected directed forest. There is a one-to-one correspondence between rooted trees and directed trees. For every rooted tree $T = (V, E, r)$ we let the corresponding directed tree T' be the directed graph (V, E') where $E' = \{(v, w) \mid \{v, w\} \in E$ and v lies on a path from r to $w\}$. Then the corresponding rooted tree T for a directed tree $T' = (V', E')$ is (V', E, r) where $r \in V'$ is the node of T' with in-degree 0 and $E = \{\{v, w\} \mid (v, w) \in E'\}$. We will often switch back and forth between rooted and directed trees. We will also transfer the terminology we introduce for directed trees to rooted trees and if possible to trees in general.

Let $T = (V, E)$ be a directed tree. A *subtree* of T is a subgraph T' of T that is a directed tree. If $(v, w) \in E$, then w is a *child* of v, and v is the *parent* of w. Let w, w' be children of a node v. Then w is a *sibling* of w' if $w \neq w'$. A node of out-degree 0 is a *leaf* or *outer* *node*, and a node of out-degree at least 1 is an *inner node*. If there is a path from $v \in V$ to $w \in V$ in T, then v is an *ancestor* of w, and w is a *descendant* of v. If additionally $v \neq w$, then v is called a *proper ancestor* of w, and w a *proper descendant* of v. The *depth* of $v \in V$ is the length of the path from the root r to v. The *height* of a node v is the length of the longest path from v to a leaf in the subtree of T rooted by v. For example, for the directed tree $T = (\{v\}, \emptyset)$ that consists only of one node v, the height of v is 0.

2.3.4. Colorings and LO-Colored Graphs

In this section we define colored and LO-colored graphs. We will also direct our attention to the representation of such graphs as logical structures.

Let $G = (V, E)$ be a (directed or undirected) graph and $f \colon V \to C$ be a mapping from the vertices of G to a finite set C. Then f is a *coloring* of G, and the elements of C are called *colors*. A *colored graph* is a triple (V, E, f) where (V, E) is a directed or undirected graph and f is a coloring of the graph (V, E). A coloring f of G defines a partition $\{f^{-1}(c) \mid c \in C, f^{-1}(c) \neq \emptyset\}$ of the vertex set V into *color classes*. Furthermore, for a partition $\mathcal{S} = \{A_1, \ldots, A_k\}$ of the vertex set V the mapping $f \colon V \to \mathcal{S}$ where $f(v) := A_i$ if $v \in A_i$ is a coloring of G. Thus, we can also say a coloring of G is a partition of the vertex set V of G.

Let $G = (V, E)$ be an undirected) graph, and $k > 0$. A *k-coloring* of G is a coloring f where $|f(V)| \leq k$ and $f(v) \neq f(w)$ if $\{v, w\} \in E$. The graph G is *bipartite* if there exists a 2-coloring of G. It is commonly known that G is bipartite if, and only if, each cycle of G has an even length (see e.g. [15]). Thus, for example, forests are bipartite graphs.

Throughout this thesis we often color the vertices of a graph with binary relations on a linearly ordered set.[1] We call graphs with such a coloring LO-*colored graphs*. More precisely, an LO-colored graph is a tuple $G = (V, E, M, \trianglelefteq, L)$ with the following properties:

- The pair (V, E) is an undirected graph. We call (V, E) the *underlying graph* of G.
- The set of *basic color elements* $M \neq \emptyset$ is a finite set with $M \cap V = \emptyset$.
- Further, the binary relation $\trianglelefteq \subseteq M^2$ is a linear order on M.
- Finally, the *color relation* $L \subseteq V \times M^2$ is a ternary relation that assigns to each vertex $v \in V$ a color $L_v := \{(d, d') \mid (v, d, d') \in L\}$.

Let $d_0, \ldots, d_{|M|-1}$ be the enumeration of the basic color elements in M according to their linear order $\trianglelefteq$. Then we call $L_v^{\mathbb{N}} := \{(i, j) \in \mathbb{N}^2 \mid (d_i, d_j) \in L_v\}$ the *natural color* of $v \in V$.

We can use the linear order $\trianglelefteq$ on M to obtain a linear order $\trianglelefteq_{\text{lex}}$ on the colors $\{L_v \mid v \in V\}$ of G. For all $v, w \in V$ we let

$$L_v \trianglelefteq_{\text{lex}} L_w \quad \Longleftrightarrow \quad L_v \text{ is lexicographically less than or equal to } L_w.$$

Clearly, $\trianglelefteq_{\text{lex}}$ is a linear order on the colors of G. Thus, an LO-colored graph is a special kind of colored graph with a linear order on its colors.

[1] In particular, we color graphs with representations of ordered copies of graphs on the number sort (defined in Section 4.2).

In order to represent $G = (V, E, M, \trianglelefteq, L)$ as a logical structure we extend the 5-tuple with its properties by a set U to a 6-tuple $(U, V, E, M, \trianglelefteq, L)$ and additionally require that $U = V \dot{\cup} M$. Clearly, the set U serves as the universe of the structure, and $V, E, M, \trianglelefteq, L$ are relations on U. For convenience, we can omit V and M from the 6-tuple, as they are implicitly given by $\trianglelefteq$ and U: Since $\trianglelefteq$ is a linear order on M, we have $M = \{d \mid d \trianglelefteq d\}$, and $V = U \setminus M$. Hence, we can regard an LO-colored graph as a τ'-structure $(U, E, \trianglelefteq, L)$, where $\tau' := \tau_E \cup \{\trianglelefteq, L\}$ is a signature with binary relation symbols E and $\trianglelefteq$ and a ternary relation symbol L, with the following properties:

- The binary relation $\trianglelefteq$ is a linear order on a non-empty subset $M \subseteq U$.
- Structure (V, E) is an undirected graph where $V := U \setminus M$.
- We have $L \subseteq V \times M^2$.

Similarly, we can also represent colored graphs $G = (V, E, f)$ with $f \colon V \to C$ as logical structures (U, V, E, R_f) where $U := V \cup C$ and $R_f := \{(v, f(v)) \mid v \in V\}$.

In this thesis we usually do not distinguish between the graphs defined in this section and their representation as logical structures. It will be clear from the context which form we are referring to.

2.4. Logics

We assume basic knowledge in logic, in particular we suppose that the reader is familiar with *first order logic* (FO) and *inflationary fixed-point logic* (IFP), and their counting extensions FO+C and IFP+C, respectively. Further, we occasionally use *simultaneous (inflationary) fixed-point logic*, which has the same expressive power as IFP. For the syntax and semantics of these logics we refer the reader to [31]. The notation used in this thesis corresponds to the one in [31]. A detailed introduction of inflationary fixed-point logic and simultaneous inflationary fixed-point logic can be found in [17]. In this thesis we simply call inflationary fixed-point logic *fixed-point logic* (FP) and *fixed-point logic with counting* (FP+C).

In many places throughout this thesis we also refer to various transitive closure logics, that is, DTC, STC, TC, and their counting extensions. Note that these logics are semantically contained in FP and its counting extension FP+C, respectively. These logics are relevant for a reader familiar with descriptive complexity theory to put our results in context. The main results of the first part of this thesis refer to FP+C. Although we might refer to a transitive closure logic when defining or describing a formula, it will always be easy (and sufficient) to see that the formula is definable in FP+C. In the second part of this thesis, we use these logics to point out the relation between them and our new logics LREC and $\mathsf{LREC}_=$, but they are not essential to follow the technical core. Therefore, we omit the definitions and refer the reader to the textbooks [17, 25, 43, 52] and the paper [41].

In the following we introduce notations and conventions.

For logics L, L' we write $\mathsf{L} \leq \mathsf{L}'$ if L is semantically contained in L', and $\mathsf{L} < \mathsf{L}'$ if this containment is strict.

Let L be a logic in $\{\mathsf{FO}, \mathsf{DTC}, \mathsf{STC}, \mathsf{TC}\}$. The counting extension L+C of L extends L by a counting operator that allows for counting the cardinality of L+C-definable relations.

It lives in a two-sorted context, where structures A are equipped with a *number sort* $N(A) := [0, |U(A)|]$. L+C has two types of *individual variables*: L+C-variables are either *structure variables* that range over the universe $U(A)$ of a structure A, or *number variables* that range over the number sort $N(A)$. Let u be an individual variable. Then the *type* $\mathrm{t}(u)$ is s if u is a structure variable and n if u is a number variable. For each variable u, let $A^u := U(A)$ if u is a structure variable, and $A^u := N(A)$ if u is a number variable. Usually, we use x, y, z and variants like x_1, y', z^* of these letters for structure variables, and o, p, q, r and variants for number variables. If the type of an (individual) variable does not matter, we use the letters u, v and variants of it.

FP+C also has *relational variables*. Let X be a relational variable of arity k that ranges over relations $R \subseteq W_1 \times \cdots \times W_k$ where $W_i \in \{U(A), N(A)\}$ for all $i \in [k]$. We let $A^X := \mathcal{P}(W_1 \times \cdots \times W_k)$. Further, we let the *type* $\mathrm{t}(X)$ of X be the tuple $(\mathrm{t}_1, \ldots, \mathrm{t}_k)$ where $\mathrm{t}_i = \mathrm{s}$ if $W_i = U(A)$ and $\mathrm{t}_i = \mathrm{n}$ if $W_i = N(A)$. Usually, we denote relational variables by X, Y, Z and variants of these letters.

Let $\bar{u} = (u_1, \ldots, u_k)$ be a tuple of (individual or relational) variables. The *type* $\mathrm{t}(\bar{u})$ of $\bar{u}$ is the tuple $(\mathrm{t}(u_1), \ldots, \mathrm{t}(u_k))$. We let $A^{\bar{u}} := A^{u_1} \times \cdots \times A^{u_k}$. The tuples $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_\ell)$ of variables are *compatible* if $k = \ell$, and for every $i \in [k]$ the variables u_i and v_i have the same type.

An *assignment* in A is a mapping α where for each variable u we have $\alpha(u) \in A^u$. For tuples $\bar{u} = (u_1, \ldots, u_k)$ of variables and $\bar{a} = (a_1, \ldots, a_k) \in A^{\bar{u}}$, the assignment $\alpha[\bar{a}/\bar{u}]$ maps u_i to a_i for each $i \in [k]$, and each variable $v \notin \tilde{u}$ to $\alpha(v)$.

We write $\varphi(u_1, \ldots, u_k)$ to denote a formula φ with $\mathrm{free}(\varphi) \subseteq \{u_1, \ldots, u_k\}$, where $\mathrm{free}(\varphi)$ denotes the set of free variables in φ. Given a formula $\varphi(u_1, \ldots, u_k)$, a structure A and $(a_1, \ldots, a_k) \in A^{(u_1, \ldots, u_k)}$, we write $A \models \varphi[a_1, \ldots, a_k]$ if φ holds in A with u_i assigned to a_i, for each $i \in [k]$. We use a similar notation for substitution: For a tuple $(v_1, \ldots, v_k)$ of variables that is compatible with $(u_1, \ldots, u_k)$, we let $\varphi(v_1, \ldots, v_k)$ be the result of substituting v_i for u_i for every $i \in [k]$. We write $\varphi[A, \alpha; \bar{u}]$ for the set of all tuples $\bar{a} \in A^{\bar{u}}$ with $(A, \alpha[\bar{a}/\bar{u}]) \models \varphi$. For a formula $\varphi(\bar{u})$ (with $\mathrm{free}(\varphi) \subseteq \tilde{u}$) we also denote $\varphi[A, \alpha; \bar{u}]$ simply by $\varphi[A; \bar{u}]$. For a formula $\varphi(\bar{v}, \bar{u})$ and $\bar{a} \in A^{\bar{v}}$, we denote $\varphi[A, \alpha[\bar{a}/\bar{v}]; \bar{u}]$ by $\varphi[A, \bar{a}; \bar{u}]$.

For structures A and tuples $\bar{n} = (n_1, \ldots, n_k) \in N(A)^k$ we let $\langle \bar{n} \rangle_A$ be the number

$$\langle \bar{n} \rangle_A \ := \ \sum_{i=1}^{k} n_i \cdot (|U(A)| + 1)^{i-1}.$$

If A is understood from the context, we write $\langle \bar{n} \rangle$ instead of $\langle \bar{n} \rangle_A$.

We use the following notational conventions and basic arithmetics:

We write $u \neq v$ for $\neg u = v$ and $p < q$ for $p \leq q \wedge \neg p = q$. Let $k \geq 1$, and $\bar{u} = (u_1, \ldots, u_k)$ and $\bar{v} = (v_1, \ldots, v_k)$. We abbreviate $\exists u_1 \ldots \exists u_k$ by $\exists \bar{u}$ and $\forall u_1 \ldots \forall u_k$ by $\forall \bar{u}$. We use $\bar{u} = \bar{v}$ to abbreviate $u_1 = v_1 \wedge \cdots \wedge u_k = v_k$, and $\bar{u} \neq \bar{v}$ to abbreviate $\neg \bar{u} = \bar{v}$. We write $\top$ for $\forall x\, x = x$ and $\bot$ for $\neg \top$.

There exist FO+C-formulas $\mathrm{zero}(p)$, $\mathrm{one}(p)$ and $\mathrm{largest}(p)$ that define the numbers 0, 1 and $|A|$ for all structures A (cf. Example 2.3.5(1) in [31]). We will occasionally use 0 and 1 as constants within formulas. It is not hard to see that these formulas can be modified so that they do not use these constants with the help of formulas zero(p) and one(p).

Let $\bar{p}, \bar{p}', \bar{q}$ be tuples of number variables. There is a DTC+C-formula plus($\bar{p}, \bar{p}', \bar{q}$) that defines the addition function, that is, for all structures A and $\bar{n} \in A^{\bar{p}}$, $\bar{n}' \in A^{\bar{p}'}$ and $\bar{m} \in A^{\bar{q}}$ we have

$$A \models \text{plus}[\bar{n}, \bar{n}', \bar{m}] \quad \iff \quad \langle \bar{n} \rangle_A + \langle \bar{n}' \rangle_A = \langle \bar{m} \rangle_A .$$

Example 2.3.5(3) in [31] shows an FP+C-formula for the addition of unary tuples. We write $\bar{p} + \bar{p}' = \bar{q}$ instead of plus($\bar{p}, \bar{p}', \bar{q}$).

Let L be a logic with DTC+C $\leq$ L. Let $\bar{u}$ be a tuple of variables, $\bar{p}$ be a tuple of number variables, and ψ be an L-formula. By $\#\bar{u}\,\psi = \bar{p}$ we denote the L-formula which holds in a structure A under an assignment α in A if $|\{\bar{a} \in A^{\bar{u}} \mid (A, \alpha[\bar{a}/\bar{u}]) \models \psi\}| = \langle \alpha(\bar{p}) \rangle_A$. As simple arithmetics like addition and multiplication are definable in DTC+C, there is an L-formula equivalent to $\#\bar{u}\,\psi = \bar{p}$.

Let $k > 0$, and let $\bar{p} = (p_1, \ldots, p_k)$ and $\bar{q} = (q_1, \ldots, q_k)$ be tuples of number variables. Then

$$\varphi_{<_{\text{lex}}}(\bar{p}, \bar{q}) := \bigvee_{i \in [k]} \Big(p_i < q_i \wedge \bigwedge_{j \in [i-1]} p_j = q_j \Big)$$

is an FO+C-formula that defines the lexicographic order on $N(A)^k$ for all structures A. Thus, for all structures A and $\bar{n}, \bar{m} \in A^{\bar{p}}$, we have

$$A \models \varphi_{<_{\text{lex}}}[\bar{n}, \bar{m}] \iff \bar{n} <_{\text{lex}} \bar{m}.$$

For LO-colored graphs $G^* = (U, V, E, M, \trianglelefteq, L)$ the lexicographic order $\trianglelefteq_{\text{lex}}$ is a linear order on the colors of the vertices of G^*. Similarly, $\triangleleft_{\text{lex}}$ is a strict linear order on the colors of the vertices of G^*. We define FO-formulas $\psi_{\trianglelefteq_{\text{lex}}}(x, x')$ and $\psi_{\triangleleft_{\text{lex}}}(x, x')$, respectively, which define the total preorder and the strict preorder on V induced by the linear order $\trianglelefteq_{\text{lex}}$ and strict linear order $\triangleleft_{\text{lex}}$ on the colors of the vertices. Thus, for all $a, a' \in U$, we have

$$\begin{aligned} G^* &\models \psi_{\trianglelefteq_{\text{lex}}}[a, a'] \iff a, a' \in V \text{ and } L_a \trianglelefteq_{\text{lex}} L_{a'}, \\ G^* &\models \psi_{\triangleleft_{\text{lex}}}[a, a'] \iff a, a' \in V \text{ and } L_a \triangleleft_{\text{lex}} L_{a'}. \end{aligned} \tag{2.1}$$

Let $\varphi_{\triangleleft_{\text{lex}}}$ be the formula we obtain from $\varphi_{<_{\text{lex}}}$ by replacing $p_i < q_i$ by $\trianglelefteq (p_i, q_i) \wedge p_i \neq q_i$ for all $i \in [k]$. Then

$$\begin{aligned} \psi_{\triangleleft_{\text{lex}}}(x, x') &:= \exists \bar{y} \Big(L(x', \bar{y}) \wedge \neg L(x, \bar{y}) \wedge \forall \bar{z} \Big(\varphi_{\triangleleft_{\text{lex}}}(\bar{z}, \bar{y}) \to \big(L(x, \bar{z}) \leftrightarrow L(x', \bar{z}) \big) \Big) \Big), \\ \psi_{\trianglelefteq_{\text{lex}}}(x, x') &:= \psi_{\triangleleft_{\text{lex}}}(x, x') \vee \forall \bar{z} \big(L(x, \bar{z}) \leftrightarrow L(x', \bar{z}) \big). \end{aligned}$$

2.5. Transductions

Transductions (also known as *syntactical interpretations*) define certain structures within other structures. We will be using different notions of transductions throughout this thesis.

In order to introduce transductions, we will first present a simple form of transductions (that uses only structure variables) and use it to illustrate the application of transductions. Gradually, we will turn to more general definitions by adding number variables and parameter variables to the transductions, until we finally present a notion which includes all aspects that are necessary in this thesis. Along the way, we introduce the Transduction Lemma, which shows us how to apply transductions. Throughout this thesis we will always only use those aspects of transductions that are necessary for our purpose and leave out the ones that would complicate the presentation. Subsequent to the introduction of (parameterized) transductions, we look at the composition of two transductions, which is again a transduction. A proof of this result can be found in the Appendix. In the last part of this section, we introduce a new form of transductions, called counting transductions. They allow a shorter presentation of specific kinds of transductions. Finally, we show that each counting transduction can be rephrased as a transduction.

In Section 2.6 transductions will be used to define canonizations, and we will need them on many other occasions in this thesis. More on transductions can be found in [31], [60] and [25]. For further examples and applications of transductions see [31] and [17].

Throughout this section we let $\mathcal{L}$ be the following set of logics:

$$\mathcal{L} = \{\mathsf{STC}, \mathsf{STC{+}C}, \mathsf{TC}, \mathsf{TC{+}C}, \mathsf{FP}, \mathsf{FP{+}C}\}.$$

2.5.1. Transductions

We start by introducing *transductions* that use only structure variables.

Definition 1 (Transduction)**.** Let $\mathsf{L} \geq \mathsf{STC}$ be a logic, and let τ_1, τ_2 be vocabularies.

1. An L-*transduction from* τ_1 *to* τ_2 (short: $\mathsf{L}[\tau_1, \tau_2]$-*transduction*) is a tuple
$$\Theta \;=\; \Big(\theta_{\mathrm{dom}}, \theta_U(\bar{u}), \theta_\approx(\bar{u}, \bar{u}'), \big(\theta_R(\bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)})\big)_{R\in\tau_2}\Big)$$
of $\mathsf{L}[\tau_1]$-formulas, where $\bar{u}, \bar{u}'$ and $\bar{u}_{R,i}$ for every $R \in \tau_2$ and $i \in [\mathrm{ar}(R)]$ are tuples of structure variables of the same length.
2. The *domain* of transduction Θ is the class $\mathrm{Dom}(\Theta)$ of all τ_1-structures A where $A \models \theta_{\mathrm{dom}}$ and $\theta_U[A; \bar{u}]$ is not empty. The variables appearing in $\bar{u}$ are called the *domain variables*.
3. Let A be in the domain of Θ. We define a τ_2-structure $\Theta[A]$ as follows. We let $\approx$ be the binary relation generated by $\theta_\approx[A; \bar{u}, \bar{u}']$, i.e. the reflexive, symmetric, transitive closure of $\theta_\approx[A; \bar{u}, \bar{u}']$, and call it the *equivalence relation of A under* Θ.[2] We let
$$U(\Theta[A]) \;:=\; \theta_U[A; \bar{u}]/_{\approx}$$
be the universe of $\Theta[A]$. Further, for each $R \in \tau_2$, we let
$$R(\Theta[A]) \;:=\; \Big(\theta_R[A; \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)}] \cap \theta_U[A; \bar{u}]^{\mathrm{ar}(R)}\Big)\Big/_{\approx}$$

[2] The traditional notion of transduction (or congruence closure) requires $\theta_\approx$ to actually define an equivalence relation and not only to generate one. However, for logics L as least as strong as STC the equivalence relation generated by an L-definable binary relation is also L-definable. Since our main results involve logics that are at least as strong as STC, we can use this more general definition of transduction.

Informally, an L-transduction from τ_1 to τ_2 defines a mapping from structures over the first vocabulary τ_1 into structures over the second vocabulary τ_2 via $\mathsf{L}[\tau_1]$-formulas.

Occasionally, it will occur that we do not need to exploit the capabilities of formulas θ_{dom} or $\theta_{\approx}$, and we simply let θ_{dom} be $\top$ and $\theta_{\approx}$ be $\bot$. If $\theta_{\text{dom}} := \top$, the domain is the class of all τ_1-structures with a non-empty universe. If $\theta_{\approx} := \bot$, the equivalence relation $\approx$ is the reflexive closure of the empty set, that is, the equivalence classes are the 1-element subsets of $A^{\bar{u}}$. As a convention, we identify each 1-element equivalence class $\{\bar{a}\}$ with its element $\bar{a}$, in this case. We can omit the respective formula when presenting the transduction. Thus, in each transduction that is given without formula θ_{dom}, formula θ_{dom} is to be interpreted as $\top$, and if formula $\theta_{\approx}$ is missing, it has to be interpreted as $\bot$.

Example 2. On the class of all graphs let us consider the $\mathsf{STC}[\{E\}, \emptyset]$-transduction $\Theta_1 = (\theta_U(x), \theta_{\approx}(x, x'))$, where

$$\theta_U(x) := \top$$
$$\theta_{\approx}(x, x') := E(x, x').$$

The domain of transduction Θ_1 is the class of all graphs. For a graph G the universe of $\Theta_1[G]$ is the set C of connected components of G. Since $\Theta_1[G]$ does not contain any relations, we have $\Theta_1[G] = (C)$. Therefore, the transduction Θ_1 maps every graph G to its set of connected components $\Theta_1[G]$. ⌟

If L is a counting logic, we can extend the definition of L-transductions above to not only allow structure variables but also number variables as domain variables. In this case we do not only need that the tuples of variables occurring within the transduction are of the same length but we also need that they are compatible. More precisely, we require that $\bar{u}, \bar{u}'$ and $\bar{u}_{R,i}$ for every $R \in \tau_2$ and $i \in [\text{ar}(R)]$ are compatible tuples of individual variables in the first part of the definition.

An important property of transductions from τ_1 to τ_2 is that, for suitable logics, they allow to *pull back* τ_2-formulas, which means that for each τ_2-formula there exists a τ_1-formula that expresses essentially the same. This property is the core of the Transduction Lemma. First we present the Transduction Lemma restricted to sentences, which makes it easier to perceive the key idea.

Proposition 3. *Let L be a logic in $\mathcal{L}$, let τ_1, τ_2 be vocabularies and let Θ be an $\mathsf{L}[\tau_1, \tau_2]$-transduction. Then for every $\mathsf{L}[\tau_2]$-sentence ψ there is an $\mathsf{L}[\tau_1]$-sentence $\psi^{-\Theta}$ such that for all $A \in \text{Dom}[\Theta]$*

$$A \models \psi^{-\Theta} \iff \Theta[A] \models \psi.$$

Thus, if ψ is an L-sentence, which defines a certain property of τ_2-structures, then $\psi^{-\Theta}$ is an L-sentence which defines the property of τ_1-structures that ψ holds after applying Θ, for logics L with $\mathsf{L} \in \mathcal{L}$.

Example 4. In this example we want to use Proposition 3 to show that there is an $\mathsf{STC}[\{E\}]$-sentence φ_{conn} such that for all graphs G we have

$$G \models \varphi_{\text{conn}} \iff G \text{ is connected.}$$

Consider the transduction Θ_1 from Example 2. In order to obtain sentence φ_{conn}, we pull back the $\mathsf{STC}[\emptyset]$-sentence $\psi := \exists x \forall x'\, x = x'$ under transduction Θ_1. Sentence ψ is satisfied if and only if there exists only one element in the universe of a given structure.

Thus, if sentence ψ is satisfied after Θ_1 has been applied to a graph G, then this means that there exists only one connected component in G, and vice versa. Hence, there exists a sentence $\psi^{-\Theta_1}$ such that for all graphs $G \in \text{Dom}(\Theta_1)$ we have

$$G \models \psi^{-\Theta_1} \iff \Theta_1[G] \models \psi \iff \text{G is connected},$$

and we define $\varphi_{\text{conn}} := \psi^{-\Theta_1}$. ⌟

In the following proposition the Transduction Lemma is formulated for formulas with free structure variables. Afterwards we explain how it extends to formulas with free structure and number variables for the counting logics in $\mathcal{L}$.

Proposition 5. *Let* L *be a logic in* $\mathcal{L}$*, let* τ_1, τ_2 *be vocabularies and let* Θ *be an* $\mathsf{L}[\tau_1, \tau_2]$*-transduction, where* $\bar{u}$ *is the tuple of domain variables. Further, let* $\psi(x_1, \ldots, x_\kappa)$ *be an* $\mathsf{L}[\tau_2]$*-formula where* $x_1, \ldots, x_\kappa$ *are structure variables. Then there exists an* $\mathsf{L}[\tau_1]$*-formula* $\psi^{-\Theta}(\bar{u}_1, \ldots, \bar{u}_\kappa)$*, where* $\bar{u}_1, \ldots, \bar{u}_\kappa$ *are compatible with* $\bar{u}$*, such that for all* $A \in \text{Dom}(\Theta)$ *and all* $\bar{a}_1, \ldots, \bar{a}_\kappa \in A^{\bar{u}}$

$$\begin{aligned} A \models \psi^{-\Theta}[\bar{a}_1, \ldots, \bar{a}_\kappa] \iff\ & \bar{a}_1/_{\approx}, \ldots, \bar{a}_\kappa/_{\approx} \in U(\Theta[A]) \text{ and} \\ & \Theta[A] \models \psi[\bar{a}_1/_{\approx}, \ldots, \bar{a}_\kappa/_{\approx}], \end{aligned}$$

where $\approx$ *is the equivalence relation of* A *under* Θ*.*

Similarly we can pull back $\mathsf{L}[\tau_2]$-formulas with free structure and number variables if L is one of the counting logics in $\mathcal{L}$. Whenever we have a free number variable p in a τ_2-formula, the pulled-back τ_1-formula contains a tuple of free number variables $\bar{q}$ of the same length as $\bar{u}$. The tuple of variables $\bar{q}$ is then used to represent the number associated with variable p in the different numerical system. In the next part of this section, where we introduce parameters to transductions, we will also cover the proceeding with free number variables in more detail.

Example 6. Now we use Proposition 5 to show that there exists an $\mathsf{STC}[\{E\}]$-sentence $\chi_{\text{conn}}(x, x')$ such that for all graphs G we have

$$G \models \chi_{\text{conn}}[v, v'] \iff v \text{ and } v' \text{ are connected in } G.$$

Again we consider the transduction Θ_1 from Example 2. The equivalence relation $\approx$ generated by $\theta_{\approx}[G; x, x']$ partitions the vertex set of G into connected components. This time we pull back $\emptyset$-formula $\psi(y, y') := y = y'$. Thus, there exists an $\{E\}$-formula $\psi^{-\Theta_1}(x, x')$ such that for all graphs G and all vertices v, v' of G we have

$$\begin{aligned} G \models \psi^{-\Theta_1}[v, v'] &\iff v/_{\approx}, v'/_{\approx} \in U(\Theta_1[G]) \text{ and } \Theta_1[G] \models \psi[v/_{\approx}, v'/_{\approx}] \\ &\iff v/_{\approx}, v'/_{\approx} \text{ are connected components of } G \text{ and } v/_{\approx} = v'/_{\approx} \\ &\iff v \text{ and } v' \text{ are connected in } G. \end{aligned}$$

We let $\chi_{\text{conn}}(x, x') := \psi^{-\Theta_1}(x, x')$. ⌟

Let $\mathcal{C}_1$ be a class of τ_1-structures and $\mathcal{C}_2$ be a class of τ_2-structures. We call a mapping f from $\mathcal{C}_1$ to $\mathcal{C}_2$ L-*definable*, if there exists an $\mathsf{L}[\tau_1, \tau_2]$-transduction Θ such that $\mathcal{C}_1 \subseteq \mathrm{Dom}(\Theta)$ and for all τ_1-structures $A \in \mathcal{C}_1$ we have $f(A) = \Theta[A]$. An $\mathsf{L}[\tau_1, \tau_2]$-transduction Θ is called an L-*reduction from* $\mathcal{C}_1$ *to* $\mathcal{C}_2$ if for all τ_1-structures A we have $A \in \mathcal{C}_1$ if and only if $\Theta[A] \in \mathcal{C}_2$. Notice that transductions and logical reductions use the same formalism.

Let L and L' be logics with $\mathsf{L}' \leq \mathsf{L}$. We say $\mathsf{L}'[\tau_1, \tau_2]$-transduction Θ *allows to pull back* L-*formulas* if each $\mathsf{L}[\tau_2]$-formula ψ can be pulled back under Θ to an $\mathsf{L}[\tau_1]$-formula $\psi^{-\Theta}$. Logic L is *closed under* L'-*reductions* if for all (relational) vocabularies τ_1, τ_2 each $\mathsf{L}'[\tau_1, \tau_2]$-transduction Θ allows to pull back L-formulas. If L is closed under L-reductions, then we say that L is *closed under logical reductions*. Each logic $\mathsf{L} \in \mathcal{L}$ is closed under logical reductions (see Exercise 11.2.4 in [17] or Lemma 1.49 in [60]). For FP+C this is shown in the Appendix in Section A.1.1 by proving the Transduction Lemma for parameterized FP+C-transductions.

2.5.2. Parameterized Transductions

In this part, we further generalize transductions and the Transduction Lemma. We introduce parameterized transductions for FP+C and generalize the Transduction Lemma in two steps so that we can pull back arbitrary FP+C-formulas under parameterized transductions.

In the following we consider parameterized transductions for FP+C. As parameter variables of these transductions, we allow individual variables as well as relational variables. The domain variables are individual variables. The definition of parameterized transduction for other logics that are at least as strong as STC can be obtained from the given definition of parameterized FP+C-transduction by leaving out all variables of types that do not occur in the particular logic.

Definition 7 (Parameterized Transduction)**.** Let τ_1, τ_2 be vocabularies.

1. A *parameterized* $\mathsf{FP{+}C}[\tau_1, \tau_2]$-*transduction* is a tuple

$$\Theta(\bar{X}) \;=\; \Big(\theta_{\mathrm{dom}}(\bar{X}), \theta_U(\bar{X}, \bar{u}), \theta_\approx(\bar{X}, \bar{u}, \bar{u}'), \big(\theta_R(\bar{X}, \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)})\big)_{R \in \tau_2}\Big)$$

   of $\mathsf{L}[\tau_1]$-formulas, where $\bar{X}$ is a tuple of individual or relational variables, and $\bar{u}, \bar{u}'$ and $\bar{u}_{R,i}$ for every $R \in \tau_2$ and $i \in [\mathrm{ar}(R)]$ are compatible tuples of individual variables.
2. The *domain* of parameterized transduction $\Theta(\bar{X})$ is the class $\mathrm{Dom}(\Theta(\bar{X}))$ of all pairs $(A, \bar{P})$, where A is a τ_1-structure and $\bar{P} \in A^{\bar{X}}$, with $A \models \theta_{\mathrm{dom}}[\bar{P}]$ and $\theta_U[A, \bar{P}; \bar{u}]$ is not empty. The variables occurring in tuple $\bar{X}$ are called *parameter variables*, and the ones occurring in $\bar{u}$ are referred to as *domain variables*. The elements in $\bar{P}$ are called *parameters*.
3. Let $(A, \bar{P})$ be in the domain of $\Theta(\bar{X})$. We define a τ_2-structure $\Theta[A, \bar{P}]$ as follows. We let $\approx$ be the equivalence relation generated by $\theta_\approx[A, \bar{P}; \bar{u}, \bar{u}']$, and call it the *equivalence relation of* $(A, \bar{P})$ *under* Θ. We let

$$U(\Theta[A, \bar{P}]) \;:=\; \theta_U[A, \bar{P}; \bar{u}]/_{\approx}$$

be the universe of $\Theta[A, \bar{P}]$. Further, for each $R \in \tau_2$, we let

$$R(\Theta[A, \bar{P}]) := \Big(\theta_R[A, \bar{P}; \bar{u}_{R,1}, \dots, \bar{u}_{R,\mathrm{ar}(R)}] \cap \theta_U[A, \bar{P}; \bar{u}]^{\mathrm{ar}(R)}\Big)\Big/_{\approx}$$

A parameterized L-transduction from τ_1 to τ_2 basically defines a parameterized mapping[3] from τ_1-structures into τ_2-structures via $\mathsf{L}[\tau_1]$-formulas. An L-transduction is a parameterized L-transduction were the tuple of parameter variables is empty. We occasionally drop the word "parameterized" in parameterized transduction if it is clear from the content that we deal with parameterized transductions. As for transductions, omitting the formula θ_{dom} in a parameterized transduction means that formula θ_{dom} is equal to $\top$ and omitting $\theta_{\approx}$ means that formula $\theta_{\approx}$ is equal to $\bot$.

Example 8. In the following we present a parameterized $\mathsf{STC}[\{E\}, \{E\}]$-transduction $\Theta_2(x_r)$ that maps a tree T to a directed version of this tree. It uses a node r of T as a parameter to root T at r. Transduction $\Theta_2(x_r)$ maps T and its parameter r to the directed tree that corresponds to the rooted tree $(r, V(T), E(T))$. We let

$$\Theta_2(x_r) := (\theta_{\mathrm{dom}}(x_r), \theta_U(x_r, x), \theta_E(x_r, x, x'))$$

where

$$\begin{aligned} \theta_{\mathrm{dom}}(x_r) &:= \top, \\ \theta_U(x_r, x) &:= \top, \\ \theta_E(x_r, x, x') &:= E(x, x') \wedge \vartheta_{\mathrm{conn}}(x', x_r, x). \end{aligned}$$

We let $\vartheta_{\mathrm{conn}}(y, x_1, x_2)$ be an STC-formula such that for each graph G and $w, v_1, v_2 \in V(G)$ we have

$$G \models \vartheta_{\mathrm{conn}}[w, v_1, v_2] \iff v_1 \text{ and } v_2 \text{ are connected in } G \setminus \{w\}.$$

The existence of $\vartheta_{\mathrm{conn}}(y, x_1, x_2)$ will be shown in Example 10.

Now, the domain of parameterized transduction $\Theta_2(x_r)$ consists of all pairs (T, r) where T is a tree and $r \in V(T)$. The universe of $\Theta[T, r]$ is the set of vertices of T. We use $\theta_E(x_r, x, x')$ to direct each edge such that it points to the vertex that is farther away from the root. Therefore, $\Theta[T, r]$ is the directed tree that corresponds to the rooted tree $(r, V(T), E(T))$. ⌟

In the following we present a version of the Transduction Lemma for parameterized $\mathsf{FP{+}C}$-transductions that allows us to pull back $\mathsf{FP{+}C}$-formulas with free individual variables.

[3] Usually, parameterized mappings are considered for fixed parameters. Here, we cannot do this since for each τ_1-structure A, valid parameters must belong to the universe of A. Therefore, we use the formula $\theta_{\mathrm{dom}}(\bar{X})$ to define possible parameters for each structure. They often have a special property that distinguishes them from other elements of the universe. That is why the term parameter makes sense in this context.

Proposition 9 (Transduction Lemma)**.** *Let τ_1, τ_2 be vocabularies. Let $\Theta(\bar{X})$ be a parameterized $\mathsf{FP{+}C}[\tau_1, \tau_2]$-transduction, where ℓ-tuple $\bar{u}$ is the tuple of domain variables. Further, let $\psi(x_1, \dots, x_\kappa, p_1, \dots, p_\lambda)$ be an $\mathsf{FP{+}C}[\tau_2]$-formula where $x_1, \dots, x_\kappa$ are structure variables and $p_1, \dots, p_\lambda$ are number variables. Then there exists an $\mathsf{FP{+}C}[\tau_1]$-formula $\psi^{-\Theta}(\bar{X}, \bar{u}_1, \dots, \bar{u}_\kappa, \bar{q}_1, \dots, \bar{q}_\lambda)$, where $\bar{u}_1, \dots, \bar{u}_\kappa$ are compatible with $\bar{u}$ and $\bar{q}_1, \dots, \bar{q}_\lambda$ are ℓ-tuples of number variables, such that for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$, all $\bar{a}_1, \dots, \bar{a}_\kappa \in A^{\bar{u}}$ and all $\bar{n}_1, \dots, \bar{n}_\lambda \in N(A)^\ell$,*

$$\begin{aligned} A \models \psi^{-\Theta}[\bar{P}, \bar{a}_1, \dots, \bar{a}_\kappa, \bar{n}_1, \dots, \bar{n}_\lambda] \\ \iff\ & \bar{a}_1/_{\approx}, \dots, \bar{a}_\kappa/_{\approx} \in U(\Theta[A, \bar{P}]), \\ & \langle \bar{n}_1 \rangle_A, \dots, \langle \bar{n}_\lambda \rangle_A \in N(\Theta[A, \bar{P}]) \text{ and} \\ & \Theta[A, \bar{P}] \models \psi[\bar{a}_1/_{\approx}, \dots, \bar{a}_\kappa/_{\approx}, \langle \bar{n}_1 \rangle_A, \dots, \langle \bar{n}_\lambda \rangle_A], \end{aligned}$$

where $\approx$ is the equivalence relation of $(A, \bar{P})$ under Θ.

Example 10. We show that there exists an STC-formula $\vartheta_{\mathrm{conn}}(y, x_1, x_2)$ such that for each graph $G = (V, E)$ and $w, v_1, v_2 \in V$ we have

$$G \models \vartheta_{\mathrm{conn}}[w, v_1, v_2] \iff v_1 \text{ and } v_2 \text{ are connected in } G \setminus \{w\}.$$

We pull back the STC-formula $\chi_{\mathrm{conn}}(x, x')$ from Example 6 under parameterized transduction

$$\Theta_3(y) := (\theta_{\mathrm{dom}}(y), \theta_U(y, x), \theta_E(y, x, x'))$$

where

$$\begin{aligned} \theta_{\mathrm{dom}}(y) &:= \top, \\ \theta_U(y, x) &:= x \neq y, \\ \theta_E(y, x, x') &:= E(x, x'). \end{aligned}$$

Transduction $\Theta_3(y)$ maps every (directed) graph G and vertex w (the parameter) to the induced subgraph $G[V \setminus \{w\}]$. Hence, there is a formula $\vartheta_{\mathrm{conn}}(y, x_1, x_2) := \chi_{\mathrm{conn}}^{-\Theta_3}(y, x_1, x_2)$ such that for all pairs (G, w) where G is a (directed) graph and $w \in V$, and all $v_1, v_2 \in V$ we have

$$\begin{aligned} G \models \chi_{\mathrm{conn}}^{-\Theta_3}[w, v_1, v_2] &\iff v_1/_{\approx}, v_2/_{\approx} \in U(\Theta[G, w]) \text{ and } \Theta_3[G, w] \models \chi_{\mathrm{conn}}[v_1/_{\approx}, v_2/_{\approx}] \\ &\iff v_1, v_2 \text{ are connected vertices in } G[V \setminus \{w\}]. \end{aligned}$$ ⌟

The Transduction Lemma for parameterized $\mathsf{FP{+}C}$-transductions in its most general form allows us to pull back arbitrary $\mathsf{FP{+}C}$-formulas. In order to present it, we need the subsequent definitions. These definitions allow us to formulate the Transduction Lemma in Proposition 11 in a nice, short way. Further, we use these definitions in Proposition 12 and its proof.

Let $\Theta(\bar{X})$ be a parameterized $\mathsf{FP{+}C}[\tau_1, \tau_2]$-transduction where ℓ-tuple $\bar{u}$ is the tuple of domain variables, and let v be an individual or relational variable. In the following we define $v^{\ltimes \bar{u}}$. If v is a variable in a τ_2-formula ψ, then $v^{\ltimes \bar{u}}$ is the variable or tuple of variables, we need in order to represent the values for variable v after pulling back ψ to a τ_1-formula $\psi^{-\Theta}$. For the following recall the definition of the type of (tuples of) variables defined in Section 2.4. We define $v^{\ltimes \bar{u}}$ as follows:

If v is a

- structure variable, we let $v^{\ltimes \bar{u}}$ be a tuple of individual variables of type $\mathrm{t}(\bar{u})$.
- number variable, we let $v^{\ltimes \bar{u}}$ be an ℓ-tuple of number variables.
- relational variable of type $\mathrm{t}(v) = (\mathrm{t}_1, \dots, \mathrm{t}_k)$, we let $v^{\ltimes \bar{u}}$ be a relational variable of type $t(v^{\ltimes \bar{u}}) = (\bar{\mathrm{t}}'_1, \dots, \bar{\mathrm{t}}'_k)$ where for all $i \in [k]$ tuple $\bar{\mathrm{t}}'_i$ is an ℓ-tuple, and $\bar{\mathrm{t}}'_i = \mathrm{t}(\bar{u})$ if $\mathrm{t}_i = \mathrm{s}$ and $\bar{\mathrm{t}}'_i = (\mathrm{n}, \dots, \mathrm{n})$ if $\mathrm{t}_i = \mathrm{n}$.

For a tuple $\bar{v} = (v_1, \dots, v_k)$ of (arbitrary) variables, we let $\bar{v}^{\ltimes \bar{u}} := (v_1^{\ltimes \bar{u}}, \dots, v_k^{\ltimes \bar{u}})$. Thus, if $\bar{v}$ is the tuple of variables occurring in a τ_2-formula ψ, then $\bar{v}^{\ltimes \bar{u}}$ is the tuple of variables in τ_1-formula $\psi^{-\Theta}$.

Now, let $(A, \bar{P})$ be in the domain of $\Theta(\bar{X})$ and let $\approx$ be the equivalence relation of $(A, \bar{P})$ under Θ. Again, let v be an individual or relational variable. Let $S \in A^{v^{\ltimes \bar{u}}}$. Then, S is a tuple of elements from $U(A) \cup N(A)$ if v is an individual variable, and a relation on $U(A) \cup N(A)$ if v is a relational variable. By forming the equivalence classes and interpreting the number tuples we can define the value $\langle S \rangle^v_{A,\approx}$ each S represents.

If v is a

- structure variable, we let
$$\langle S \rangle^v_{A,\approx} := S/_{\approx}.$$
- number variable, we let
$$\langle S \rangle^v_{A,\approx} := \langle S \rangle_A .$$
- relational variable of type $\mathrm{t}(v) = (\mathrm{t}_1, \dots, \mathrm{t}_k)$, we let
$$\langle S \rangle^v_{A,\approx} := \{ (\langle \bar{a}_1 \rangle^{t_1}_{A,\approx}, \dots, \langle \bar{a}_k \rangle^{t_k}_{A,\approx}) \mid (\bar{a}_1, \dots, \bar{a}_k) \in S \},$$
where $\langle \bar{a}_i \rangle^{t_i}_{A,\approx} := \bar{a}_i/_{\approx}$ if $t_i = \mathrm{s}$ and $\langle \bar{a}_i \rangle^{t_i}_{A,\approx} := \langle \bar{a}_i \rangle_A$ if $t_i = \mathrm{n}$ for all $i \in \{1, \dots k\}$.

For a tuple $\bar{v} = (v_1, \dots, v_k)$ of (arbitrary) variables, and $\bar{S} = (S_1, \dots, S_k) \in A^{\bar{v}^{\ltimes \bar{u}}}$, we let
$$\langle \bar{S} \rangle^{\bar{v}}_{A,\approx} := (\langle S_1 \rangle^{v_1}_{A,\approx}, \dots, \langle S_k \rangle^{v_k}_{A,\approx}).$$

Now we can state the Transduction Lemma for parameterized FP+C-transductions in its general form. In Section A.1.1 in the Appendix the Transduction Lemma for parameterized FP+C-transductions (Proposition 11) is repeated in more detail, and proved afterwards. There, we distinguish explicitly between the types of the variables that are used and their assigned values.

Proposition 11 (Transduction Lemma)**.** *Let τ_1, τ_2 be vocabularies. Let $\Theta(\bar{X})$ be a parameterized* FP+C$[\tau_1, \tau_2]$*-transduction where $\bar{u}$ is the tuple of domain variables. Further, let $\psi(\bar{v})$ be an* FP+C$[\tau_2]$*-formula where $\bar{v}$ is a tuple of (individual and relational) variables. Then there exists an* FP+C$[\tau_1]$*-formula $\psi^{-\Theta}(\bar{X}, \bar{v}')$, where $\bar{v}' = \bar{v}^{\ltimes \bar{u}}$, such that for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$, and all $\bar{S} \in A^{\bar{v}'}$,*
$$A \models \psi^{-\Theta}[\bar{P}, \bar{S}] \iff \langle \bar{S} \rangle^{\bar{v}}_{A,\approx} \in \Theta[A, \bar{P}]^{\bar{v}} \text{ and } \Theta[A, \bar{P}] \models \psi[\langle \bar{S} \rangle^{\bar{v}}_{A,\approx}],$$
where $\approx$ is the equivalence relation of $(A, \bar{P})$ under Θ.

The proof of the Transduction Lemma for parameterized FP+C-transductions (Proposition 11) can be found in Section A.1.1 in the Appendix. By leaving out number variables and everything concerning them, it yields a proof for the corresponding result for FP. The proofs for STC, TC and their counting variants are analogous.

2.5.3. Composition

The following proposition shows that the composition of two parameterized transductions is again a parameterized transduction.

We let L be any logic in $\mathcal{L}$.

Proposition 12. *Let τ_1, τ_2 and τ_3 be vocabularies. Let $\Theta_1(\bar{X}_1)$ be a parameterized $\mathsf{L}[\tau_1,\tau_2]$-transduction, and let $\Theta_2(\bar{Y})$ be a parameterized $\mathsf{L}[\tau_2,\tau_3]$-transduction where $\bar{u}_1$ and $\bar{u}_2$ are the respective tuples of domain variables. Then there exists a parameterized $\mathsf{L}[\tau_1,\tau_3]$-transduction $\Theta(\bar{X})$ with $\bar{X} = (\bar{X}_1, \bar{X}_2)$ such that $\bar{X}_2 = \bar{Y}^{\ltimes \bar{u}_1}$, the tuple $\bar{u}_2^{\ltimes \bar{u}_1}$ is the tuple of domain variables, and for all τ_1-structures A and all $\bar{P} \in A^{\bar{X}}$ with $\bar{P} = (\bar{P}_1, \bar{P}_2)$,*

$$\begin{aligned}(A,\bar{P}) \in \mathrm{Dom}(\Theta(\bar{X})) \iff\ & (A,\bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)),\\ & \bar{Q} := \langle \bar{P}_2 \rangle^{\bar{Y}}_{A,\approx_1} \in \Theta_1[A,\bar{P}_1]^{\bar{Y}} \text{ and}\\ & (\Theta_1[A,\bar{P}_1],\bar{Q}) \in \mathrm{Dom}(\Theta_2(\bar{Y})),\end{aligned}$$

where $\approx_1$ is the equivalence relation of $(A,\bar{P}_1)$ under Θ_1, and for all $(A,\bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$,

$$\Theta[A,\bar{P}] \cong \Theta_2[\Theta_1[A,\bar{P}_1],\bar{Q}].$$

Section A.1.2 in the Appendix contains the proof of Proposition 12.

2.5.4. Counting Transductions

We introduce the new notion of counting transductions in this section. For a structure A (and a tuple of parameters) from the domain of the counting transduction, a counting transduction automatically includes the number sort of A in the universe of the structure defined by the counting transduction. For this reason, counting transductions sometimes allow a shorter formulation of the transduction. Counting transductions are as powerful as transductions, which we will show in the end of this section. Often giving a counting transduction instead of a transduction will contribute to an easier and clearer presentation.

In the following we assume L is a counting logic.

Definition 13 (Parameterized Counting Transduction)**.** Let τ_1, τ_2 be vocabularies.

1. A *parameterized* $\mathsf{L}[\tau_1,\tau_2]$*-counting transduction* is a tuple

$$\Theta^{\#}(\bar{X}) = \Big(\theta^{\#}_{dom}(\bar{X}), \theta^{\#}_{U}(\bar{X},\bar{u}), \theta^{\#}_{\approx}(\bar{X},\bar{u},\bar{u}'), \big(\theta^{\#}_{R}(\bar{X},\bar{u}_{R,1},\ldots,\bar{u}_{R,\mathrm{ar}(R)})\big)_{R\in\tau_2}\Big)$$

   of $\mathsf{L}[\tau_1]$-formulas, where $\bar{X}$ is a tuple of individual or relational variables, $\bar{u}, \bar{u}'$ are compatible tuples of individual variables, $\bar{u}, \bar{u}'$ are not tuples of number variables of length 1, and for every $R \in \tau_2$ and $i \in [\mathrm{ar}(R)]$, $\bar{u}_{R,i}$ is a tuple of variables that is compatible to $\bar{u}$ or a tuple of number variables of length 1.
2. The *domain* of counting transduction $\Theta^{\#}(\bar{X})$ is the class $\mathrm{Dom}(\Theta^{\#}(\bar{X}))$ of all pairs $(A,\bar{P})$ such that $A \models \theta^{\#}_{dom}[\bar{P}]$ where A is a τ_1-structure and $\bar{P} \in A^{\bar{X}}$.

3. Let $(A, \bar{P})$ be in the domain of $\Theta^{\#}(\bar{X})$. We define a τ_2-structure $\Theta^{\#}[A, \bar{P}]$ as follows. We let $\approx$ be the equivalence relation generated by $\theta^{\#}_{\approx}[A, \bar{P}; \bar{u}, \bar{u}']$. We let

$$U(\Theta^{\#}[A, \bar{P}]) \ := \ \theta^{\#}_U[A, \bar{P}; \bar{u}]/_{\approx} \ \dot{\cup} \ N(A)$$

be the universe of $\Theta^{\#}[A, \bar{P}]$.[4]
For a tuple $\bar{a} \in A^{\bar{u}} \,\dot{\cup}\, N(A)$, let $\bar{a}^{\approx} := \bar{a}/_{\approx}$ if $\bar{a}$ is compatible to $\bar{u}$, and $\bar{a}^{\approx} := \bar{a}$ if $\bar{a}$ is a tuple of number variables of length 1. For each $R \in \tau_2$, let

$$R^{\#}_{A,\bar{P}} \ := \ \theta^{\#}_R[A, \bar{P}; \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)}] \cap \Big(\theta^{\#}_U[A, \bar{P}; \bar{u}] \,\dot{\cup}\, N(A)\Big)^{\mathrm{ar}(R)}.$$

Then,

$$R(\Theta^{\#}[A, \bar{P}]) := \big\{(\bar{a}_1^{\approx}, \ldots, \bar{a}_{\mathrm{ar}(R)}{}^{\approx}) \ \big| \ (\bar{a}_1, \ldots, \bar{a}_{\mathrm{ar}(R)}) \in R^{\#}_{A,\bar{P}}\big\}.$$

Proposition 14. *Let $\Theta^{\#}(\bar{X})$ be an $\mathsf{L}[\tau_1, \tau_2]$-counting transduction. Then there exists a parameterized $\mathsf{L}[\tau_1, \tau_2]$-transduction $\Theta(\bar{X})$ such that*

- $\mathrm{Dom}(\Theta(\bar{X})) = \mathrm{Dom}(\Theta^{\#}(\bar{X}))$ *and*
- $\Theta[A, \bar{P}] \cong \Theta^{\#}[A, \bar{P}]$ *for all* $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$

Proof. Let

$$\Theta^{\#}(\bar{X}) \ = \ \Big(\theta^{\#}_{dom}(\bar{X}), \theta^{\#}_U(\bar{X}, \bar{u}), \theta^{\#}_{\approx}(\bar{X}, \bar{u}, \bar{u}'), \big(\theta^{\#}_R(\bar{X}, \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)})\big)_{R \in \tau_2}\Big)$$

be a parameterized $\mathsf{L}[\tau_1, \tau_2]$-counting transduction. Let $\bar{u}_0$ be a tuple of individual variables compatible to $\bar{u}$, and $\bar{u}_1$ be a tuple of number variables of length 1. Further, let o_0, o_1 be number variables. Now let $\bar{t} := (o_0, o_1, \bar{u}_0, \bar{u}_1)$. We present an $\mathsf{L}[\tau_1, \tau_2]$-transduction

$$\Theta(\bar{X}) \ = \ \Big(\theta_{dom}(\bar{X}), \theta_U(\bar{X}, \bar{t}), \theta_{\approx}(\bar{X}, \bar{t}, \bar{t}'), \big(\theta_R(\bar{X}, \bar{t}_{R,1}, \ldots, \bar{t}_{R,\mathrm{ar}(R)})\big)_{R \in \tau_2}\Big),$$

where $\mathrm{Dom}(\Theta(\bar{X})) = \mathrm{Dom}(\Theta^{\#}(\bar{X}))$ and for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$ structures $\Theta[A, \bar{P}]$ and $\Theta^{\#}[A, \bar{P}]$ are isomorphic. Within this transduction the variables o_0 and o_1 help us to construct the desired universe, the union of $\theta^{\#}_U[A, \bar{P}; \bar{u}]$ and $[0, |A|]$. We will only allow values for the variables o_0 and o_1, where one variable is assigned to value 1 and the other variable obtains value 0. Now, if the value assigned to variable o_0 is 1, only tuples where the part corresponding to variable tuple $\bar{u}_0$ belongs to different equivalence classes of equivalence relation $\theta^{\#}_{\approx}[A, \bar{P}; \bar{u}, \bar{u}']$ are distinguished by our equivalence relation. If the

[4] We can also define counting transductions $\Theta^{\#}(\bar{X})$ such that for $(A, \bar{P}) \in \mathrm{Dom}(\Theta^{\#}(\bar{X}))$ the universe of $\Theta^{\#}[A, \bar{P}]$ is

$$U(\Theta^{\#}[A, \bar{P}]) \ := \ \theta^{\#}_U[A, \bar{P}; \bar{u}]/_{\approx} \ \dot{\cup} \ \bigcup_{m \in M} N(A)^m$$

for a finite set $M \subset \mathbb{N}$. Then we let $\bar{u}_{R,i}$ be a tuple of variables that is compatible to $\bar{u}$ or a tuple of number variables of length ℓ with $\ell \in M$ in part 1 of the definition, and forbid that $\bar{u}$ is a tuple of number variables of length ℓ with $\ell \in M$. The proof of Proposition 14 works analogous in this case. We simply use number variables o_m and variable tuples $\bar{u}_m$ for $m \in M$ in the same way we use o_1 and $\bar{u}_1$ in the proof of Proposition 14.

value assigned to variable o_1 is 1, then the equivalence relation distinguishes only between tuples where the part of the tuple corresponding to tuple $\bar{u}_1$ of number variables differs.

More precisely, we let $\Theta(\bar{X})$ be the parameterized transduction where

$$\begin{aligned} \theta_{dom}(\bar{X}) &:= \theta^{\#}_{dom}(\bar{X}), \\ \theta_U(\bar{X}, \bar{t}\,) &:= (o_0 = 1 \wedge o_1 = 0) \vee (o_0 = 0 \wedge o_1 = 1), \\ \theta_{\approx}(\bar{X}, \bar{t}, \bar{t}') &:= \big(o_0 = 1 \wedge o'_0 = 1 \wedge\ \theta^{\#}_{\approx}(\bar{X}, \bar{u}_0, \bar{u}'_0)\big) \vee \big(o_1 = 1 \wedge o'_1 = 1 \wedge \bar{u}_1 = \bar{u}'_1\big). \end{aligned}$$

Let T be the set of all tuples $\bar{u}_{R,i}$ where $R \in \tau_2$ and $i \in [\mathrm{ar}(R)]$. Let h be a function that maps every $\bar{u}_* \in T$ to 0 if $\bar{u}_*$ is compatible to $\bar{u}$ and to 1 otherwise. Then

$$\theta_R(\bar{X}, \bar{t}_{R,1}, \ldots, \bar{t}_{R,\mathrm{ar}(R)}) := \theta^{\#}_R\Big(\bar{X}, \bar{u}^{R,1}_{h(\bar{u}_{R,1})}, \ldots, \bar{u}^{R,\mathrm{ar}(R)}_{h(\bar{u}_{R,\mathrm{ar}(R)})}\Big) \wedge \bigwedge_{i \in [\mathrm{ar}(R)]} o^{R,i}_{h(\bar{u}_{R,i})} = 1$$

for all $R \in \tau_2$, where $\bar{t}_{R,i} = (o^{R,i}_0, o^{R,i}_1, \bar{u}^{R,i}_0, \bar{u}^{R,i}_1)$.

It is not hard to see, that structure $\Theta[A, \bar{P}]$ is isomorphic to structure $\Theta^{\#}[A, \bar{P}]$ for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$. □

2.6. Canonization

In this section we introduce ordered structures, the notion of canonization and definable canonization. Further, we present two important results regarding canonization from [31], which we will need in Section 4.3 and Chapter 12. A detailed introduction of (definable) canonization and more examples can be found in [31].

In the following let τ be a signature with $\leq \notin \tau$, and let L be an arbitrary logic.

We call a $\tau \cup \{\leq\}$-structure A' *ordered* if the relation symbol $\leq$ is interpreted as a linear order on the universe of A'. Ordered structures A' and B' are *order isomorphic* if they are isomorphic. (We use the formulation “order isomorphic” to emphasize the presence of the ordering.) Let A be a τ-structure. We say a $\tau \cup \{\leq\}$-structure A' is an *ordered copy* of A if $A'|_\tau \cong A$.

Let $\mathcal{C}$ be a class of τ-structures. A mapping f is a *canonization mapping* of $\mathcal{C}$ if it assigns every structure $A \in \mathcal{C}$ to an ordered copy $f(A) = (A_f, \leq_f)$ of A such that for isomorphic structures $A, B \in \mathcal{C}$ the ordered copies $f(A)$ and $f(B)$ are order isomorphic. We call the ordered structure $f(A)$ the *canon* of A. Notice that when we talk about the canon of a structure we implicitly assume a specific canonization mapping. For every canonization mapping f of $\mathcal{C}$, let $f_{\mathbb{N}}$ be the unique canonization mapping of $\mathcal{C}$ that maps every structure $A \in \mathcal{C}$ with $|U(A)| = n$ to the ordered copy $f_{\mathbb{N}}(A) = (A_{f_{\mathbb{N}}}, \leq_{f_{\mathbb{N}}})$ where the universe $U(A_{f_{\mathbb{N}}})$ is $[n]$, the linear order $\leq_{f_{\mathbb{N}}}$ is the natural order on $[n]$, and $f_{\mathbb{N}}(A)$ and $f(A)$ are order isomorphic. We say that $f_{\mathbb{N}}$ defines an *ordered copy* of A *on the number sort* for all $A \in \mathcal{C}$.

Let $\bar{x}$ be a tuple of individual variables, and let $\Theta(\bar{x})$ be a parameterized $\mathsf{L}[\tau, \tau \cup \{\leq\}]$-transduction. We say $\Theta(\bar{x})$ *canonizes* a τ-structure A if there is at least one tuple $\bar{p} \in A^{\bar{x}}$ such that $(A, \bar{p}) \in \mathrm{Dom}(\Theta(\bar{x}))$, and for all tuples $\bar{p} \in A^{\bar{x}}$ with $(A, \bar{p}) \in \mathrm{Dom}(\Theta(\bar{x}))$, the

$\tau \cup \{\leq\}$-structure $\Theta[A, \bar{p}]$ is an ordered copy of A.[5] A parameterized transduction $\Theta(\bar{x})$ *canonizes* a class $\mathcal{C}$ of τ-structures if it canonizes all $A \in \mathcal{C}$. A *parameterized* L*-canonization* of a class $\mathcal{C}$ of τ-structures is a parameterized $\mathsf{L}[\tau, \tau \cup \{\leq\}]$-transduction that canonizes $\mathcal{C}$. Like for transductions, we refer to parameterized L-canonization as L*-canonizations* if the tuple $\bar{x}$ is empty. A class $\mathcal{C}$ of structures *admits* L*-definable canonization* if for all vocabularies τ the class $\mathcal{C}[\tau]$ has a parameterized L-canonization.

The following proposition shows the connection between canonization and parameterized canonization.

Proposition 15 ([31], Lemma 3.3.18[6])**.** *Let* $\mathsf{L} \geq \mathsf{STC{+}C}$ *be a logic that is closed under logical reductions, and let* $\mathcal{C}$ *be a class of* τ*-structures. If there is a parameterized* L*-canonization of* $\mathcal{C}$*, then there exists an* L*-canonization of* $\mathcal{C}$ *without parameter variables.*

Proposition 15 shows that each parameterized L-canonization of $\mathcal{C}$ yields a canonization mapping of $\mathcal{C}$, because for each parameterized L-canonization $\Theta(\bar{x})$ of $\mathcal{C}$ there is an L-canonization Θ' of $\mathcal{C}$ without parameter variables, and the mapping $A \mapsto \Theta'[A]$ is a canonization mapping of $\mathcal{C}$.

Example 16. In the following we present an STC+C-canonization Θ_4 of the class $\mathcal{K}$ of complete graphs. We let

$$\Theta_4 := (\theta_U(p), \theta_E(p, p'), \theta_{\leq}(p, p')),$$

where

$$\begin{aligned} \theta_U(p) &:= p > 0 \\ \theta_E(p, p') &:= p \neq p', \\ \theta_{\leq}(p, p')) &:= p \leq p'. \end{aligned}$$

Θ_4 is an $\mathsf{STC{+}C}[\{E\}, \{E, \leq\}]$-transduction that uses only tuples of number variables of length $\ell = 1$. The relation $\theta[G; p]$ is non-empty for complete graphs G. Thus, the domain of Θ_4 contains the class of all complete graphs. For $G \in \mathrm{Dom}(\Theta_4)$ where n is the number of vertices in G, we have $\Theta_4[G] = ([n], \binom{[n]}{2}, \leq_{[n]})$, that is, structure $\Theta_4[G]$ is the complete graph on the vertex set $[n]$ together with the natural linear order $\leq_{[n]}$ on the numbers in $[n]$. Thus, $\Theta_4[G]$ is an ordered copy of G and Θ_4 canonizes $\mathcal{K}$. ⌟

Example 17. Next we present an STC+C-canonization Θ^* of the class $\mathcal{K}^*$ of LO-colored graphs $K^* = (U, V, E, M, \trianglelefteq, L)$ where the underlying graph (V, E) is a complete graph. We let formulas $\psi_{\triangleleft_{\mathrm{lex}}}(x, x')$ and $\psi_{\trianglelefteq_{\mathrm{lex}}}(x, x')$ be the FO-formulas from (2.1). In order to define the canon of LO-colored graph K^* we do the following: We order all vertices of K^* according to the lexicographic order of their colors. Note that this is a total preorder. We assign each vertex to every position the vertex can obtain within a linear order that

[5] Note that if the tuple $\bar{x}$ of parameter variables is the empty tuple, $\mathsf{L}[\tau, \tau \cup \{\leq\}]$-transduction Θ *canonizes* a τ-structure A if $A \in \mathrm{Dom}(\Theta)$ and the $\tau \cup \{\leq\}$-structure $\Theta[A]$ is an ordered copy of A.

[6] As the main result in [31] is based on IFP+C, Lemma 3.3.18 is only shown for IFP+C in [31]. However, addition, multiplication and all arithmetics (e.g. Fact 3.3.14) that are necessary to show Lemma 3.3.18 can be defined in DTC+C. Further, Lemma 3.3.12 and 3.3.17, which are used to prove Lemma 3.3.18, can be shown by pulling back simple FO+C-formulas under STC+C-transductions. Hence, it suffices if our logic L is closed under logical reductions and is as least as strong as STC+C.

extends the total preorder. Thus, each vertex is assigned to numbers in $[1, |V|]$. We order all basic color elements by the given linear order $\trianglelefteq$ on M and assign them to the numbers in $[|V|+1, |V|+|M|]$. The formulas $\varphi_V(y,p)$ and $\varphi_M(y,p)$ define these assignments.

$$\varphi_V(y,p) := \exists q\, \exists q' \big(\#x\; \psi_{\triangleleft_{\text{lex}}}(x,y) = q \;\wedge\; \#x\; \psi_{\trianglelefteq_{\text{lex}}}(x,y) = q' \;\wedge\; \text{``} q < p \leq q' \text{''}\big)$$
$$\varphi_M(y,p) := \exists q\, \exists q' \big(\#x\; V(x) = q \;\wedge\; \#x\; x \trianglelefteq y = q' \;\wedge\; p = q + q'\big)$$

For $K^* \in \mathcal{K}^*$ the pair $(v,n) \in U(K^*) \times N(K^*)$ satisfies $\varphi_V(y,p)$ if, and only if, after ordering all vertices of K^* according to the lexicographic order of their colors there exist less than n vertices smaller than v and at least n vertices larger than v. The pair $(v,n) \in U(K^*) \times N(K^*)$ satisfies $\varphi_M(y,p)$, if v is the mth vertex of the linear order $\trianglelefteq$ on the basic color elements M and $n = |V| + m$.

We let $\Theta^* := (\theta^*_U(p), \theta^*_V(p), \theta^*_E(p,p'), \theta^*_M(p), \theta^*_{\trianglelefteq}(p,p'), \theta^*_L(p,p',p''), \theta^*_{\leq}(p,p'))$, where

$$\begin{aligned}
\theta^*_U(p) &:= p > 0,\\
\theta^*_V(p) &:= \exists y\, \varphi_V(y,p),\\
\theta^*_E(p,p') &:= \theta^*_V(p) \wedge \theta^*_V(p') \wedge p \neq p',\\
\theta^*_M(p) &:= \exists y\, \varphi_M(y,p),\\
\theta^*_{\trianglelefteq}(p,p') &:= \theta^*_M(p) \wedge \theta^*_M(p') \wedge p \leq p',\\
\theta^*_L(p,p',p'') &:= \exists y, z', z'' \big(L(y,z',z'') \wedge \varphi_V(y,p) \wedge \varphi_M(z',p') \wedge \varphi_M(z'',p'')\big), \text{ and}\\
\theta^*_{\leq}(p,p')) &:= p \leq p'.
\end{aligned}$$

Then Θ^* is a canonization of $\mathcal{K}^*$. ⌟

Remark 18. If we also allowed parameter variables that are relational variables in the definition of canonization, then every class of τ-structures would admit FP-definable canonization. To verify this, consider the following parameterized transduction where parameter variable X is a binary relational variable:

$$\Theta(X) := (\theta_{\text{dom}}(X), \theta_U(X,x), (\theta_R(X, x_1, \ldots, x_{\text{ar}(R)}))_{R \in \tau}, \theta_{\leq}(X,x,x'))$$

with

$$\begin{aligned}
\theta_{\text{dom}}(X) &:= \varphi_{\text{refl}}(X) \wedge \varphi_{\text{antisym}}(X) \wedge \varphi_{\text{trans}}(X) \wedge \varphi_{\text{connex}}(X),\\
\theta_U(X,x) &:= \top,\\
\theta_R(X, x_1, \ldots, x_{\text{ar}(R)}) &:= R(x_1, \ldots, x_{\text{ar}(R)}) \text{ for all } R \in \tau, \text{ and}\\
\theta_{\leq}(X,x,x') &:= X(x,x').
\end{aligned}$$

where $\varphi_{\text{refl}}(X)$, $\varphi_{\text{antisym}}(X)$, $\varphi_{\text{trans}}(X)$ and $\varphi_{\text{connex}}(X)$ are FO-definable formulas that decide if X is interpreted by a reflexive, antisymmetric, transitive and connex binary relation, respectively. ⌟

Next let us consider graphs. The following proposition shows that if we can canonize the graphs induced by the connected components of a graph, then we can canonize the whole graph.

Proposition 19 ([31], Corollary 3.3.21[7]). *Let* $\mathsf{L} \geq \mathsf{STC{+}C}$ *be a logic that is closed under logical reductions. Let* $\mathcal{C}$ *be a class of graphs, and let* $\mathcal{C}_{conn}$ *be the class of all graphs induced by the connected components of the graphs in* $\mathcal{C}$*. If* $\mathcal{C}_{conn}$ *admits* L*-definable canonization, then* $\mathcal{C}$ *does as well.*

2.7. Descriptive Complexity

To learn more about descriptive complexity we recommend Chapter 7 and 11 in [17] and Chapter 2 in [31]. In this section we introduce descriptive complexity only briefly.

2.7.1. Turing Machines and Complexity Classes

We assume that the reader has basic knowledge in complexity theory; see, e. g., [48, 64] for an introduction into complexity theory. The main complexity classes used in this thesis are PTIME and LOGSPACE. With the following definitions, which are mostly adopted from [50], we shortly introduce these complexity classes.

A Turing machine M *decides* a language L, that is, a class L of finite strings, if for any finite string x,

$$M(x) \begin{cases} \text{accepts} & \text{if } x \in L \\ \text{rejects} & \text{if } x \notin L. \end{cases}$$

Let C be an additional language. A Turing machine M *decides* a language L *on* C if M decides a language L' such that $L \cap C = L' \cap C$.

A *polynomial-time Turing machine* M is a Turing machine for which there exists a constant $c \in \mathbb{N}$ such that all computation paths of M terminate within $\mathcal{O}(|x|^c)$ steps on all input strings x. A language L is *polynomial-time decidable* if there is a deterministic polynomial-time Turing machine that decides L. We denote the class of all polynomial-time decidable languages by PTIME. We let NP denote the class of all languages decidable by a non-deterministic polynomial-time Turing machine.

A *logarithmic-space Turing machine* is a Turing machine whose input tape is read-only, and which accesses at most $\mathcal{O}(\log(|x|))$ different work tape cells for any computation path for all input strings x. LOGSPACE is the class of all languages L for which there exists a deterministic logarithmic-space Turing machine that decides L. We also use the word *logspace* as an abbreviation for logarithmic space. We denote the class of all languages decidable by a non-deterministic logarithmic-space Turing machine by NL.

A *logspace transducer* is a deterministic logspace Turing machine which has, in addition to its read-only input tape and its worktapes, one write-only output tape. Note that the length of the output string may be polynomial. Logspace transducers can be concatenated,

[7] Similar to Lemma 3.3.18 (see footnote of Proposition 15), Corollary 3.3.21 is only shown for IFP+C in [31]. The proof of Corollary 3.3.21 uses Lemma 3.3.18, the definability of simple arithmetics, connectivity and the Transduction Lemma. Hence, Corollary 3.3.21 also holds for all logics L that are closed under logical reductions and are as least as strong as STC+C.

that is, if one logspace transducer uses the output of another as its input, these two transducers can be combined into a single logspace transducer (cf. [48]).

We use the terms Turing machine and algorithm interchangeably.

2.7.2. Capturing Complexity Classes

We can associate with an ordered structure $(A, \leq)$ a binary string $\text{enc}(A, \leq)$ that represents this ordered structure (see [31], Section 3.1.2, for possible representation schemes). For structures A, in general, we let the set $A_{\leq}$ of all ordered representations of A be

$$A_{\leq} := \{\text{enc}(A, \leq) \mid \leq \text{ is a linear order on } U(A)\}.$$

For a class $\mathcal{A}$ of structures, let $\mathcal{A}_{\leq} := \bigcup_{A \in \mathcal{A}} A_{\leq}$.

Let L be a logic, τ be a vocabulary and $\mathcal{A}$ be a class of τ-structures. A τ-sentence φ of L *defines* the class $\mathcal{A}$ if for all τ-structures A we have $A \in \mathcal{A} \iff A \models \varphi$. The class $\mathcal{A}$ is L-*definable* if there exists a τ-sentence φ of L that defines $\mathcal{A}$. Let K be a complexity class. The language $\mathcal{A}_{\leq}$ is K-*decidable* if $\mathcal{A}_{\leq} \in \mathsf{K}$. Logic L *captures* the complexity class K if for every vocabulary τ and every class $\mathcal{A}$ of τ-structures, $\mathcal{A}$ is L-definable if and only if $\mathcal{A}_{\leq}$ is K-decidable .

Let L be a logic and $\mathcal{C}$ be a class of structures. Further, let τ be a vocabulary and $\mathcal{A}$ be a class of τ-structures. A τ-sentence φ of L *defines* the class $\mathcal{A}$ *on* $\mathcal{C}$ if for all τ-structures $A \in \mathcal{C}$ we have $A \in \mathcal{A} \iff A \models \varphi$. The class $\mathcal{A}$ is L-*definable on* $\mathcal{C}$ if there exists a τ-sentence φ of L that defines $\mathcal{A}$ on $\mathcal{C}$. Let K be a complexity class. The language $\mathcal{A}_{\leq}$ is K-*decidable on* $\mathcal{C}_{\leq}$ if there is a language $L \in \mathsf{K}$ such that $L \cap \mathcal{C}_{\leq} = \mathcal{A}_{\leq} \cap \mathcal{C}_{\leq}$. Logic L *captures* K *on* $\mathcal{C}$ if for every vocabulary τ and each class $\mathcal{A}$ of τ-structures, $\mathcal{A}$ is L-definable on $\mathcal{C}$ if and only if $\mathcal{A}_{\leq}$ is K-decidable on $\mathcal{C}_{\leq}$.

Remark 20. The definition of capturing actually has to be refined to exclude pathological examples of logics (cf. [17]). We have to pose restrictions on what constitutes a "logic". Further, we need an effective procedure that maps any τ-sentence that defines a class of τ-structures $\mathcal{A}$ (on $\mathcal{C}$) to a Turing machine that decides $\mathcal{A}_{\leq}$ (on $\mathcal{C}_{\leq}$). To include this (cf. [17]), we say a logic L *effectively captures* a complexity class K (on $\mathcal{C}$) if L captures K (on $\mathcal{C}$) (as defined above), and for every vocabulary τ, the set of τ-sentences of L is decidable and there is a recursive procedure which assigns to every τ-sentence φ of L a Turing machine M and (the code of) a function f such that M decides $\{A \mid A \models \varphi\}_{\leq}$ (on $\mathcal{C}_{\leq}$) and f witnesses that M is resource-bounded according to K. Within all capturing results in this thesis the respective logic effectively captures the respective complexity class (on the respective class of structures). ⌟

On ordered structures there already exist capturing results for the complexity classes PTIME and $\mathsf{LOGSPACE}$. One fundamental result in descriptive complexity theory is the Immerman-Vardi Theorem.

Theorem 21 (Immerman-Vardi Theorem, [39, 66])**.** IFP *captures* PTIME *on the class of all ordered structures.*

Further, Immerman showed that there also is a descriptive characterization of the complexity classes LOGSPACE and NL.

Theorem 22 (Immerman [41, 42])**.** *On the class of all ordered structures*

- DTC *captures* LOGSPACE*, and*
- TC *captures* NL*.*

Since DTC ≤ STC and STC-formulas can be evaluated in logarithmic space [62], Theorem 22 implies the following corollary.

Corollary 23. STC *captures* LOGSPACE *on the class of all ordered structures.*

Let L be a logic that captures a complexity class on ordered structures. If L is closed under logical reductions, then in order to show that L captures a complexity class on a class $\mathcal{C}$ of structures, it suffices to find a (parameterized) L-canonization of $\mathcal{C}$. As we can pull back each sentence of L under this canonization, the capturing result transfers from ordered structures to the class $\mathcal{C}$.

Proposition 24. *Let* $\mathcal{L}_{poly} := \{\mathsf{FP}, \mathsf{FP{+}C}\}$ *and* $\mathcal{L}_{log} := \{\mathsf{DTC}, \mathsf{DTC{+}C}, \mathsf{STC}, \mathsf{STC{+}C}\}$.

1. *Let* $\mathsf{L} \in \mathcal{L}_{poly}$. *If a class* $\mathcal{C}$ *of structures admits* L*-definable canonization, then* L *captures polynomial time on* $\mathcal{C}$.
2. *Let* $\mathsf{L} \in \mathcal{L}_{log}$. *If a class* $\mathcal{C}$ *of structures admits* L*-definable canonization, then* L *captures logarithmic space on* $\mathcal{C}$.

2.8. Graph-Theoretical Preliminaries

2.8.1. Depth-First Tree Traversal

There are different methods to traverse a graph, that is, to visit each vertex at least once. One method that is commonly known is depth-first search (see e.g. [63]). For trees depth-first search is also called *depth-first tree traversal*. In [53], for example, it is shown that depth-first tree traversal is possible in logarithmic space.[8] Note that the representation of the nodes of the tree in the input string of the algorithm induces a linear order on the nodes of the tree, and therefore on the children of each node. We assume that the children of a node are given in this order.

In the following we summarize depth-first traversal as it is described by Lindell in [53]. It is illustrated in Figure 2.1. We start at the root. For every node of the tree we have three possible moves:

- **down**: go down to the first child, if it exists
- **over**: move over to the next sibling, if it exists
- **up**: buck up to the parent, if it exists

At each step we only need to remember our last move and the current node. If our last move was **down**, **over** or there was no last move, which means we are visiting a new node, then we perform the first move out of **down**, **over** or **up** that succeeds. If our last move was **up**, then we are backtracking, and we call **over** if it is possible or else **up**.

[8] In [53] Lindell proves that tree canonization is in LOGSPACE.

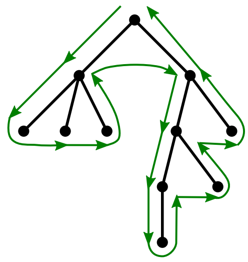


Figure 2.1.: Depth-first tree traversal

2.8.2. Max Cliques and Spanning Vertices

Let $G = (V, E)$ be an arbitrary graph. A maximal clique, or *max clique*, of G is a clique of G that is not properly contained in another clique of G. The set of max cliques of G is denoted by $\mathcal{M}_G$. We omit the index if it is clear what graph we are referring to. Let $k > 0$. We say k vertices $v_1, \dots, v_k \in V$ *span* a max clique $A \in \mathcal{M}$, if A is the only max clique that contains vertices $v_1, \dots, v_k$.

Let $\mathcal{C}$ be a class of graphs, and $k > 1$. For the remainder of this section, we assume the following.

Assumption 25. For every graph $G \in \mathcal{C}$, each max clique of G is spanned by k vertices.

We show that the max cliques of graphs in graph class $\mathcal{C}$ are FO-definable.

Lemma 26. *Let A be a max clique of a graph $G = (V, E)$, and $v_1, \dots, v_k \in A$. Then $v_1, \dots, v_k$ span A if, and only if, for all $v, w \in V \setminus \{v_1, \dots, v_k\}$ with $v \neq w$ there is an edge between v and w if $\{v_1, \dots, v_k\} \times \{v, w\}$ is a subset of E.*

Proof. Let $S := \{v_1, \dots, v_k\}$ be a subset of max clique A of G. Let us assume there exist vertices $v, w \in V \setminus S$ with $v \neq w$ such that there is no edge between v and w but $S \times \{v, w\} \subseteq E$. Then $S \cup \{v\}$ and $S \cup \{w\}$ are cliques but $S \cup \{v, w\}$ is not a clique. Thus, $S \cup \{v\}$ is a subset of a max clique C with $w \notin C$, and $S \cup \{w\}$ is a subset of a max clique D with $v \notin D$. Consequently, vertices $v_1, \dots, v_k$ are contained in more than one max clique and therefore do not span A.

Next, let us suppose $v_1, \dots, v_k$ do not span A. Then there must exist another max clique B with $v_1, \dots, v_k \in B$. As A cannot be a subset of B, there exists a vertex $v \in A \setminus B$. Now, $B \cup \{v\}$ cannot be a clique. Thus, there must exist a vertex $w \in B$ that is not adjacent to v. Since v is adjacent to all vertices in $A \setminus \{v\}$, we have $w \in B \setminus A$. Consequently, v and w are vertices in $V \setminus S$ with $v \neq w$ that are not adjacent although $S \times \{v, w\}$ is a subset of E. □

Let $v_1, \dots, v_k \in V$. Vertices $v_1, \dots, v_k$ are jointly contained in a max clique precisely if, $v_i = v_j$ or v_i and v_j are adjacent, for all $i, j \in [k]$. Therefore, vertices $v_1, \dots, v_k \in V$ span a max clique if, and only if, for all $i, j \in [k]$ we have $v_i = v_j$, or v_i and v_j are adjacent, and for all $v, w \in V \setminus \{v_1, \dots, v_k\}$ with $v \neq w$ vertices v and w are adjacent if $\{v_1, \dots, v_k\} \times \{v, w\}$ is a subset of edge relation E. This characterization of spanning vertices allows us to define max cliques in FO. The following formula is satisfied by a graph $G \in \mathcal{C}$ and vertices $v_1, \ldots, v_k \in V$ if and only if vertices $v_1, \ldots, v_k$ span a max clique.

$$\begin{aligned}\varphi_{\text{span}}(x_1, \ldots, x_k) := \bigwedge_{i,j \in [k]} & (x_i = x_j \vee E(x_i, x_j)) \\ & \wedge \forall x \forall y \Big(\Big(\bigwedge_{i \in [k]} E(x, x_i) \wedge \bigwedge_{i \in [k]} E(y, x_i) \Big) \rightarrow E(x, y) \Big) \end{aligned} \tag{2.2}$$

If vertices $v_1, \ldots, v_k$ span a max clique, then according to Lemma 26 vertices $v_1, \ldots, v_k$ and all vertices w with $\{v_1, \ldots, v_k\} \times \{w\} \subseteq E$ form a clique. It is not hard to see that this clique is maximal. The next formula allows us to define max cliques, and for graphs $G \in \mathcal{C}$ we have $G \models \varphi_{\mathcal{M}}(v_1, \ldots, v_k, w)$ exactly if $v_1, \ldots, v_k$ span a max clique A of G and $w \in A$.

$$\varphi_{\mathcal{M}}(x_1, \ldots, x_k, y) := \varphi_{\text{span}}(x_1, \ldots, x_k) \wedge \Big(\bigvee_{i \in [k]} y = x_i \vee \bigwedge_{i \in [k]} E(y, x_i) \Big). \tag{2.3}$$

In order to decide whether the vertices $v_1, \ldots, v_k$ and $v'_1, \ldots, v'_k$ span the same max clique, we use formula $\varphi_{\text{span},\approx}$. It is satisfied for a graph $G \in \mathcal{C}$ and $v_1, \ldots, v_k, v'_1, \ldots, v'_k, \in V$ if and only if the vertices $v_1, \ldots, v_k$ and the vertices $v'_1, \ldots, v'_k$ both span max cliques of G and the max cliques are equal.

$$\begin{aligned}\varphi_{\text{span},\approx}(x_1, \ldots, x_k, x'_1, \ldots, x'_k) := & \varphi_{\text{span}}(x_1, \ldots, x_k) \wedge \varphi_{\text{span}}(x'_1, \ldots, x'_k) \\ & \wedge \forall y \left(\varphi_{\mathcal{M}}(x_1, \ldots, x_k, y) \leftrightarrow \varphi_{\mathcal{M}}(x'_1, \ldots, x'_k, y) \right). \end{aligned} \tag{2.4}$$

Clearly, $\varphi_{\text{span},\approx}$ defines an equivalence relation on the set of all k-tuples of spanning vertices. We use these equivalence classes to represent the max cliques, and obtain the following corollary.

Corollary 27. *For graphs $G \in \mathcal{C}$ the max cliques of G are* FO*-definable.*

2.8.3. Centroids

Let $T = (V, E)$ be a tree and $w \in V$. Let $C_1, \ldots, C_k$ be the connected components of $T \setminus \{w\}$. The *weight* $\text{wg}(w)$ of w in T is the maximum of $|C_1|, \ldots, |C_k|$. A node $w \in V$ of minimal weight is called a *centroid* of T.

Lemma 28 ([46], p. 387)**.** *There are at most two centroids in a tree, and if two centroids exist, they are adjacent.*

Observation 29. *Let T be a tree with at least one inner node. Then all centroids of T are inner nodes.*

Proof. Let $T = (V, E)$ be a tree with at least one inner node w. Let $C_1, \ldots, C_k$ be the connected components of $T \setminus \{w\}$. Since w is an inner node, it holds that $k \geq 2$. Let us consider a connected component C_i. Each connected component C_j with $j \neq i$ contains at least one vertex. Therefore, $|C_i| \leq |V| - 1 - (k-1)$. Thus, the weight of w in T is at most $|V| - k \leq |V| - 2$. Now, let us suppose the outer node w is a centroid of T. As w is a leaf, $T \setminus \{w\}$ has one connected component C and $|C| = |V| - 1$. Consequently, $|V| - 1$ is the weight of w in T. It follows that w is not a node of minimal weight. □

In order to define the weight of a vertex in STC+C, we use the STC-formula $\vartheta_{\text{conn}}(y, x_1, x_2)$ from Example 10. For each tree T and $w, v_1, v_2 \in V(T)$ we have

$$T \models \vartheta_{\text{conn}}[w, v_1, v_2] \iff v_1 \text{ and } v_2 \text{ are connected in } T \setminus \{w\}.$$

The following STC+C-formula φ_{wg} uses ϑ_{conn} to determine the sizes of the connected components of the tree T after removing a vertex w. Formula $\varphi_{\text{wg}}(y, p)$ is satisfied for a tree T and a pair $(v, n) \in V(G) \times N(G)$ if and only if the weight of v in T is n.

$$\varphi_{\text{wg}}(y, p) = \exists x \, \#z \, \vartheta_{\text{conn}}(y, x, z) = p \ \wedge \ \forall x' \, \forall p' \big(\#z \, \vartheta_{\text{conn}}(y, x', z) = p' \to p' \leq p\big)$$

We can use this formula to define the centroids of a tree in STC+C. We let

$$\varphi_{\text{cen}}(y) = \exists p \Big(\varphi_{\text{wg}}(y, p) \wedge \forall y' \, \forall p' \big(\varphi_{\text{wg}}(y', p') \to p \leq p'\big)\Big) \tag{2.5}$$

For each tree $T = (V, E)$ and $w \in V$ we have

$$T \models \vartheta_{\text{cen}}[w] \iff w \text{ is a centroid of } T.$$

Part I.

The Modular Decomposition Theorem and Applications

Introduction

The (unique) modular decomposition of a graph partitions the vertex set of the graph into modules, that is, into subsets that share the same neighbors. By recursively constructing the modular decompositions of the subgraphs induced by these modules, one eventually obtains the modular decomposition tree of a graph.

Modular decomposition (also called substitution decomposition) was introduced in 1967 by Gallai [21] as a tool for the structural analysis of comparability graphs. In the following it found a variety of applications in graph theory [58, 36]. For example, modular decomposition is often employed in recognition algorithms for classes of graphs that are well-structured with respect to the modular decomposition, like permutation graphs [61], interval graphs [37] and cographs [10]. The modular decomposition tree of a graph can be constructed in linear time [55, 11, 13]. It is useful to solve many combinatorial optimization problems on graphs efficiently [58]. Another application of modular decomposition is graph canonization. In particular, modular decomposition has been used to show that there exists a logarithmic-space algorithm for computing a canonization mapping for interval graphs [47]. In descriptive complexity theory, modular decomposition was first used by Laubner [49, 50][9], showing that the class of interval graphs admits $\mathsf{FP{+}C}$-definable canonization. This implies that $\mathsf{FP{+}C}$ captures PTIME on interval graphs.

In this thesis, we use modular decomposition to prove that $\mathsf{FP{+}C}$ also captures PTIME on permutation graphs and chordal comparability graphs. To this end, we show that (under certain conditions that have to be made more precise) a logic captures PTIME on a class $\mathcal{C}$ of graphs if it captures PTIME on the class $\mathcal{C}^*$ of LO-colored graphs with prime underlying graphs from $\mathcal{C}$.[10] A graph is *prime* if only the vertex set itself and all vertex sets of size 1 are modules of the graph. For each class $\mathcal{C}$ of graphs that is closed under induced subgraphs and every logic L that is as least as expressive as $\mathsf{FP{+}C}$ and closed under $\mathsf{FP{+}C}$-reductions, we prove the Modular Decomposition Theorem, which says that there is an L-canonization of $\mathcal{C}$ if there is an L-canonization of the class $\mathcal{C}^*$. It follows from the Modular Decomposition Theorem that L captures PTIME on $\mathcal{C}$ if L captures PTIME on $\mathcal{C}^*$.

To prove the Modular Decomposition Theorem, we show that the modular decomposition of a graph is definable in $\mathsf{STC{+}C}$. This also proves that there exists a logspace algorithm that computes the modular decomposition of a graph, which we extend to an algorithm that computes the modular decomposition tree in logarithmic space.

This part of the thesis is organized as follows: In Chapter 3 we first introduce modules and the modular decomposition of a graph. Furthermore, we show that there exists

[9] The canonization result in [47] is actually based on observations of Laubner in [49, 50].

[10] Note that an LO-coloring is a coloring of the vertices of the graph where the colors are linearly ordered. Often it is easy to extend an L-canonization of a class $\mathcal{D}$ of (prime) graphs to an L-canonization of the class of LO-colored graphs with underlying graphs from $\mathcal{D}$ (cf. Chapters 5 and 6).

an STC+C-formula that defines the modular decomposition. We outline the logspace algorithm behind this formula and extend it to a logspace algorithm that computes the modular decomposition tree of a graph in Section 3.5. In Chapter 4, we prove the Modular Decomposition Theorem. Finally, in Chapters 5 and 6 we apply the Modular Decomposition Theorem to show that FP+C captures PTIME on the class of permutation graphs and the class of chordal comparability graphs, respectively.

3. STC+C-Definability of the Modular Decomposition

The aim of this chapter is to show that the modular decomposition of a graph is definable in symmetric transitive closure logic with counting, and therefore, computable in logarithmic space.

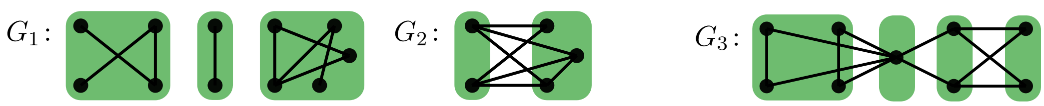
First, we introduce modules and the modular decomposition of a graph in this chapter. Then in order to show that the modular decomposition is definable in STC+C, we consider modules that are spanned by two vertices, that is, modules that contain the two vertices and are minimal with this property. We use the concept of edge classes introduced by Gallai in [21] to show that these spanned modules are definable in STC+C. Afterwards, we show how the spanned modules are related to the modules occurring in the modular decomposition, and exploit the STC+C-definability of the spanned modules to define the modules of the modular decomposition. As a result, we obtain that the modular decomposition is definable in STC+C. Consequently, it is computable in logarithmic space. In the last part of this chapter we outline the logspace algorithm behind the STC+C-formulas. Finally, we introduce the modular decomposition tree and show that it is computable in logarithmic space as well.

We utilize the STC+C-definability (actually we only require FP+C-definability) of the modular decomposition in order to prove the Modular Decomposition Theorem in Chapter 4.

3.1. Modules and their Basic Properties

Let $G = (V, E)$ be a graph. By $\overline{G}$ we denote the *complement graph* of G which has vertex set V and edge set $\overline{E}$ where $e \in \overline{E}$ if and only if $e \notin E$. A non-empty subset $M \subseteq V$ is a *module* of a graph G if for all vertices $v \in V \setminus M$ either v is adjacent to all vertices in M or v is adjacent to no vertex in M. Thus, a non-empty subset $M \subseteq V$ is a module if and only if for all $v \in V \setminus M$ and all $w, w' \in M$ we have

$$\{v, w\} \in E \iff \{v, w'\} \in E.$$



(a) The connected components of G_1 and $\overline{G_2}$ are modules.

(b) The highlighted sets, for example, are modules of graph G_3.

Figure 3.1.: Modules

All vertex sets of size 1 are modules. We call them *singleton modules*. Further, the vertex set V is a module. We also refer to the module V and the singleton modules as *trivial modules*. The connected components and unions of connected components of G or $\overline{G}$ are modules as well (see Figure 3.1a). Figure 3.1b shows a further example of modules in a graph.

A module M is a *proper module* if $M \subset V$. We call a graph *prime* if it does not contain any non-singleton proper modules. Thus, a graph is prime if all its modules are trivial modules. Figure 3.2 shows a prime graph. Notice that if M is a module of a graph G, then M is also a module of $\overline{G}$. Therefore, a graph G is prime if and only if $\overline{G}$ is prime.

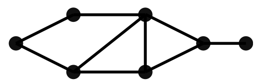


Figure 3.2.: A prime graph

In the following we show three fundamental properties of modules.

Observation 30. *If M_1 and M_2 are modules of a graph G with $M_1 \cap M_2 = \emptyset$, then either there exist no edges between vertices in M_1 and vertices in M_2, or every vertex in M_1 is adjacent to each vertex in M_2.*

Proof. Let there be an edge between vertices $v_1 \in M_1$ and $v_2 \in M_2$. Then v_1 must be adjacent to all vertices in M_2 as M_2 is a module. Since each vertex v in M_2 is adjacent to v_1, there must also be an edge between v and each vertex in M_1 as M_1 is a module. Consequently, there is an edge between each vertex in M_1 and every vertex in M_2. □

Observation 31. *If M_1 and M_2 are modules of a graph G and have a non-empty intersection, then $M_1 \cap M_2$ and $M_1 \cup M_2$ are modules as well.*

Proof. Let $M := M_1 \cap M_2$ be non-empty. In order to prove that M is a module, we need to show that for all vertices $v \in V \setminus M$ and all vertices $w, w' \in M$ there is an edge between v and w if and only if there is one between v and w'. Thus, let $v \in V \setminus M$ and $w, w' \in M$ be arbitrary vertices (see Figure 3.3a). There exists an $i \in [2]$ such that v is contained in $V \setminus M_i$. Further, we have $w, w' \in M_i$ since M is a subset of M_i. As M_i is a module it follows that v and w are adjacent if and only if v and w' are. Consequently, M is a module.

(a) $M_1 \cap M_2$ is a module (here, $v \in V \setminus M_2$).

(b) $M_1 \cup M_2$ is a module (here, $w \in M_1, w' \in M_2$).

Figure 3.3.: Illustrations for the proof of Observation 31

Now let us consider the union of two modules M_1, M_2 with non-empty intersection, and let $M' := M_1 \cup M_2$. Let $v \in V \setminus M'$ and $w, w' \in M'$ be arbitrary vertices (see Figure 3.3b). Suppose that w is in M_j and w' is in $M_{j'}$ with $j, j' \in \{1, 2\}$. Further, let z be a vertex in $M_1 \cap M_2$. As v is in $V \setminus M_j$, and M_j is a module, v and w are adjacent if and only if there is an edge between v and z. Equivalently, there is an edge between v and w' if and only if v and z are adjacent. Thus, there is an edge between v and w if and only if there is one between v and w'. Hence, $M_1 \cup M_2$ is a module. □

Observation 32. *Let M' be a module of G, and M be a subset of M'. Then M is a module of G if and only if it is a module of $G[M']$.*

Proof. If M is a module of G, then clearly it must be a module of $G[M']$. Thus, we only need to show the other direction, and we let M be a module of $G[M']$. To show that M is a module of G, let us consider arbitrary vertices $v \in V \setminus M$ and $w, w' \in M$. If v is in M', then v and w are adjacent if and only if v and w' are, since M' is a module. If $v \in V \setminus M'$, then we can use that $w, w' \in M'$ and M' is a module of G. Again we obtain that there is an edge between v and w if and only if there is one between v and w'. □

3.2. Modular Decomposition

In the following we present the modular decomposition of a graph, which was introduced by Gallai in 1967 [21]. The modular decomposition decomposes a graph, and can be applied recursively.

Let $G = (V, E)$ be an arbitrary graph. We let n be the number of vertices in G. If G (or $\overline{G}$) is not connected, then every connected component of G (or $\overline{G}$) is a module. Thus, if we partition the vertices of G (or $\overline{G}$) into its connected components, we have a partition of V into proper modules. For graphs G where G and $\overline{G}$ are connected there also exists a unique partition of V into proper modules. Gallai showed that for those graphs G, the maximal proper modules of G form a partition of V if $n > 1$ (Satz 2.9 and 2.11 in [21]). Actually, Figure 3.1b does not only depict arbitrary modules, but the maximal proper modules of a connected and co-connected graph. A graph is *co-connected* if the complement of the graph is connected.

Consequently, we can partition each graph G with $n > 1$ into proper modules. For a vertex v of graph G we let $D_G(v)$ be the respective proper module containing vertex v. Thus, for a vertex v of a graph G with at least two vertices, $D_G(v)$ is[1]

- the connected component of G that contains v if G is not connected,
- the connected component of $\overline{G}$ that contains v if $\overline{G}$ is not connected, or
- the maximal proper module of G that contains v if G and $\overline{G}$ are connected.

If the graph G has only one vertex v, we let $D_G(v) := \{v\}$.

[1] We can also say $D_G(v)$ is the maximal strong proper module of G that contains v. (A module M is *strong*, if we have $M \cap M' = \emptyset$, $M \subseteq M'$ or $M' \subseteq M$ for all other modules M'.) Gallai proved that the maximal strong proper modules partition the vertex set of G (Satz 2.11 in [21]), and that for each graph G they coincide with the sets $D_G(v)$ as they are defined here (Satz 2.9 and 2.10 in [21]).

We define the *(recursive) modular decomposition*[2] of G as the following family of subsets $D_{i,v} \subseteq V$ with $i \in [0,n]$, $v \in V$. We let $D_{0,v} := V$ for all $v \in V$, and for $i \in [0,n]$ we define $D_{i+1,v}$ for all $v \in V$ recursively:

$$D_{i+1,v} := D_{G[D_{i,v}]}(v).$$

As an example, a graph and its modular decomposition is illustrated in Figure 3.4.

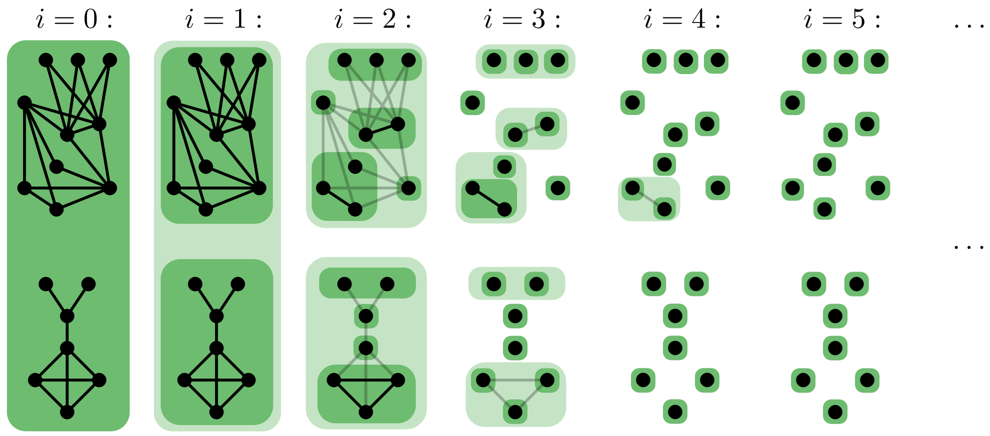


Figure 3.4.: Modular decomposition of a graph

It is easy to see that there exists a $k \in [0,n]$ such that $V = D_{0,v} \supset D_{1,v} \supset \cdots \supset D_{k,v} = \{v\}$ and that $D_{i,v} = \{v\}$ for all $i \geq k$. Thus, $D_{n,v} = \{v\}$ for all $v \in V$. For all $i \in [0,n]$ and all $v \in V$ the set $D_{i,v}$ is a module of G as we can apply Observation 32 inductively. Further, an easy induction shows that the set $\{D_{i,v} \mid v \in V\}$ is a partition of the vertex set V for all $i \in [0,n]$. Hence, we can conclude the following.

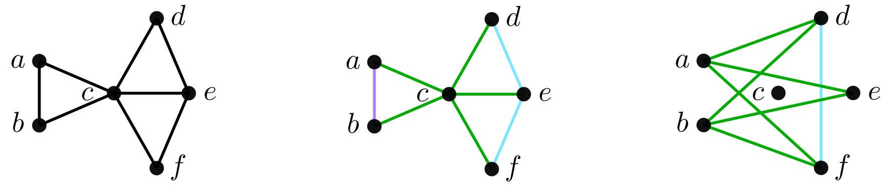
Observation 33. *For all $v, w \in V$ and all $i \in [0,n]$, the modules $D_{i,v}$ and $D_{i,w}$ are equal if and only if $w \in D_{i,v}$.*

3.3. Spanned Modules and (W)edge Classes

Let $v, w \in V$ be vertices of G. We let $M_{v,w}$ be the intersection of all modules of G that contain v and w. Notice that V is a module containing v and w. Therefore, the set $M_{v,w}$ exists. Further, $M_{v,w}$ is non-empty, since v and w are contained in $M_{v,w}$, and as there exist only finitely many modules containing v and w, Observation 31 implies that $M_{v,w}$ is a module. Consequently, $M_{v,w}$ is the smallest module containing v and w. We say vertices $v, w \in V$ *span* module M if $M = M_{v,w}$, that is, if M is the smallest module containing v and w. We call M a *spanned* module if there exists $v, w \in V$ that span M. Trivially, $M_{v,v} = \{v\}$.

[2] Note that usually the term modular decomposition denotes only the decomposition of the graph into the modules $D_G(v)$, $v \in V$.

Let $e, e' \in E$ be two edges of G. We say e and e' *form a wedge* in G if there exist three distinct vertices $u, v, w \in V$ such that $e = \{u, v\}$, $e' = \{u, w\}$ and there is no edge between v and w. We also write $e \wedge e'$ if the edges e and e' form a wedge in G. Clearly, $e \wedge e'$ implies $e' \wedge e$. We call the relation $\wedge$ the *wedge relation* on E. We say two edges e and e' are *wedge connected* if their exists a $k \geq 1$ and a sequence of edges $e_1, \ldots, e_k$, such that $e = e_1$, $e' = e_k$ and $e_i \wedge e_{i+1}$ for all $1 \leq i < k$. It is not hard to see that wedge connectivity is an equivalence relation on the set of edges of the graph. We call the equivalence classes the *edge classes* of G.[3] Thus, the edge classes partition the set of edges of a graph. The same way, we can partition the set of edges of the complement graph $\overline{G}$ of G. We define the *wedge class* of a binary set $\{v, w\}$ of vertices as the edge class of G that contains $\{v, w\}$ if $\{v, w\}$ is an edge of G, or as the edge class of $\overline{G}$ that contains $\{v, w\}$ otherwise. For distinct vertices v and w we let $W_{v,w}$ be the set of vertices occurring in binary subsets in the wedge class of $\{v, w\}$. Hence, $W_{v,w}$ is the union of all elements in the wedge class of $\{v, w\}$. Clearly, we have $v, w \in W_{v,w}$.



(a) Graph H (b) Edge classes of H (c) Edge classes of $\overline{H}$

Figure 3.5.

Example 34. Consider the graph H, depicted in Figure 3.5a.

The following edges form a wedge in H:

- $\{b, c\}$ and $\{c, d\}$,
- $\{b, c\}$ and $\{c, e\}$,
- $\{b, c\}$ and $\{c, f\}$,
- $\{a, c\}$ and $\{c, d\}$,
- $\{a, c\}$ and $\{c, e\}$,
- $\{a, c\}$ and $\{c, f\}$,
- $\{c, d\}$ and $\{c, f\}$,
- $\{d, e\}$ and $\{e, f\}$.

Thus, for example, $\{c, e\}$ is wedge connected to $\{b, c\}$, $\{a, c\}$, $\{c, d\}$, $\{c, f\}$ and itself. The edge classes of H, which are illustrated in Figure 3.5b, are

$$\{\{a, b\}\}, \ \{\{a, c\}, \{b, c\}, \{c, d\}, \{c, e\}, \{c, f\}\} \text{ and } \{\{d, e\}, \{e, f\}\}.$$

The complement of H and its edge classes are shown in Figure 3.5c. Finally, we list some examples of the set $W_{v,w}$ for distinct vertices $v, w \in V(H)$:

- $W_{e,f} = \{d, e, f\}$,
- $W_{c,f} = V(H)$,
- $W_{d,f} = \{d, f\}$,
- $W_{b,f} = V(H) \setminus \{c\}$. ⌟

[3] Edge classes (or Kantenklassen) are defined in [21]. We extend this definition to wedge classes.

Lemma 35 ([21], Satz 1.5)**.** *Let $v, w \in V$ with $v \neq w$. Then $W_{v,w}$ is a module.*

Proof. Let $v \neq w$. For a contradiction, let us assume that there exist $a, b \in W_{v,w}$ and $z \notin W_{v,w}$ with $\{a, z\} \in E$ and $\{b, z\} \notin E$. Let a' and b' be vertices in $W_{v,w}$ such that $\{a, a'\}$ and $\{b, b'\}$ are in the wedge class of $\{v, w\}$. Without loss of generality let v and w be adjacent. Then a and a', and b and b' are also adjacent. Since $\{a, a'\}$ and $\{b, b'\}$ are wedge connected, there exists a $k \geq 1$ and a sequence $e_1, \ldots, e_k$ of edges with $e_1 = \{a, a'\}$ and $e_k = \{b, b'\}$ and $e_i \curlywedge e_{i+1}$ for all $1 \leq i < k$. Next we show that there exists an edge $e_j = \{c_j, c'_j\}$ with $1 \leq j \leq k$ in this sequence such that $\{c_j, z\} \in E$ and $\{c'_j, z\} \notin E$. Then $\{c_j, c'_j\}$ and $\{c_j, z\}$ form a wedge, which means that $\{c_j, z\}$ belongs to the wedge class of $\{v, w\}$ as well, and it follows that $z \in W_{v,w}$, which is a contradiction to the choice of z. So let us assume that for all $i \in [k]$ vertex z is adjacent to both vertices of e_i or to none. As a and z are adjacent, z is adjacent to both vertices of e_1. Further, for all $i < k$ edges e_i and e_{i+1} share a vertex. Thus, it follows inductively that z is adjacent to both vertices of all edges e_i with $i \in [k]$. We obtain a contradiction, since $\{z, b\} \notin E$. □

Lemma 36 ([21], Satz 1.5)**.** *$W_{v,w} \subseteq M_{v,w}$ for all vertices $v, w \in V$ with $v \neq w$.*

Proof. Let $v \neq w$, and without loss of generality let v and w be adjacent. Let $z \in W_{v,w}$. Then there exists an edge e' with $z \in e'$ that is wedge connected to $e := \{v, w\}$. In the following we prove that each edge that is wedge connected to e is a subset of $M_{v,w}$. As a result $z \in M_{v,w}$, and therefore $W_{v,w} \subseteq M_{v,w}$. Let e and e' be wedge connected, then there exist $e_1, \ldots, e_k$ such that $e = e_1$, $e' = e_k$ and $e_i \curlywedge e_{i+1}$ for all $1 \leq i < k$. We show inductively that $e_j \subseteq M_{v,w}$ for all $j \in [k]$. Clearly, $e_1 \subseteq M_{v,w}$. Now let $e_j \subseteq M_{v,w}$ for $j \in [k-1]$. We show that $e_{j+1} \subseteq M_{v,w}$ as well. Since $e_j \curlywedge e_{j+1}$, there exist distinct vertices $a, b, c \in V$ such that $e_j = \{a, b\}$, $e_{j+1} = \{a, c\}$ and $\{b, c\} \notin E$. As $a, b \in M_{v,w}$, the assumption that $c \notin M_{v,w}$ directly implies that $M_{v,w}$ is not a module. Thus, c has to be contained in $M_{v,w}$, and $e_{j+1} \subseteq M_{v,w}$. □

Lemma 35 and 36 yield the following corollary.

Corollary 37. *$M_{v,w} = W_{v,w}$ for all vertices $v, w \in V$ with $v \neq w$.*

In the following Lemma we show that spanned modules are definable in symmetric transitive closure logic. As a consequence, they are also computable in logarithmic space.

Lemma 38. *There exists an STC-formula $\varphi_M(x_1, x_2, y)$ such that for all pairs $(v_1, v_2) \in V^2$ of vertices of G, the set $\varphi_M[G, v_1, v_2; y]$ is the module spanned by v_1 and v_2.*

Proof. To prove this lemma we apply Corollary 37, which allows us to use the definition of $W_{v,w}$ in order to define the module spanned by two distinct vertices.

First of all, we need a formula for the wedge relation, that is, a formula which is satisfied for vertices $v_1, v_2, w_1, w_2 \in V$ if, and only if, $\{v_1, v_2\} \curlywedge \{w_1, w_2\}$ in G. Clearly, this is precisely the case if there exist $i, j \in [2]$ such that

$$v_i = w_j,\ v_{3-i} \neq w_{3-j},\quad \text{and}$$
$$\{v_1, v_2\}, \{w_1, w_2\} \in E,\ \{v_{3-i}, w_{3-j}\} \notin E.$$

Thus, we obtain an FO-formula for the wedge relation by taking the disjunction of the above statement over all $i, j \in [2]$. Since the wedge relation is symmetric, we can use the STC-operator to express wedge connectivity. Hence, there exists an STC-formula that expresses wedge connectivity in G, and similarly we obtain one for wedge connectivity in $\overline{G}$, as well. Using these formulas we are able to define the wedge classes of a graph. Consequently, we can also define the set $W_{v,w}$ for distinct vertices $v, w \in V$ in symmetric transitive closure logic.

Now it remains for φ_M to distinguish between the cases of whether the spanning vertices are equal or not and define the spanned module accordingly. ☐

3.4. Defining the Modular Decomposition in STC+C

In this section we prove that the modular decomposition of a graph is definable in symmetric transitive closure logic with counting.

Let us fix a $v \in V$. Our goal is to define the sets $D_{i,v}$ for $i \in [0, n]$. We use the modules $M_{v,w}$ with $w \in V$ for it. It is possible to construct $D_{i,v}$ out of certain modules $M_{v,w}$ with $w \in V$. In order to do that, we first need to gain a better understanding of the connection between $D_{i,v}$ and the sets $M_{v,w}$.

Lemma 39 ([21], Satz 2.9 and 2.11 in connection with Satz 1.2 (3b)[4])**.** *Let G and $\overline{G}$ be connected and let $M', M'' \in \mathcal{M}$ be maximal proper modules of G with $M' \neq M''$. Further let $v \in M'$ and $w \in M''$. Then $M_{v,w} = V$.*

Proof. The set of all maximal proper modules $\mathcal{M}$ of G is a partition of V if G and $\overline{G}$ are connected and $n > 1$. Since this is a fact which is commonly known, we only show that Lemma 39 follows directly from this fact, although their actual proofs in [21] are linked.

Let us assume $M_{v,w}$ is a proper module. Then there must exist a maximal proper module $M \in \mathcal{M}$ such that $M_{v,w} \subseteq M$. Since $v \in M' \cap M$ and $w \in M'' \cap M$, and $\mathcal{M}$ is a partition of V into maximal proper modules, we have $M' = M$ and $M'' = M$. Thus, $M' = M''$, a contradiction. ☐

Corollary 40. *Let $i \in [0, n]$ and $v \in V$. If $G[D_{i,v}]$ and its complement are connected and $|D_{i,v}| > 1$, then for all vertices $w \in D_{i,v} \setminus D_{i+1,v}$ we have $D_{i,v} = M_{v,w}$.*

Lemma 41 ([21], Satz 1.2 (2)[4])**.** *Let G be not connected and v and w be in different connected components of G. Let C_v and C_w be the respective connected components. Then $M_{v,w} = C_v \cup C_w$.*

Proof. We know that $C_v \cup C_w$ is a module containing v and w. Thus, $M_{v,w}$ has to be a subset of $C_v \cup C_w$. To show that $M_{v,w} = C_v \cup C_w$ we assume that there exists a vertex in $C_v \cup C_w$ that is not contained in $M_{v,w}$. Without loss of generality, let this vertex be in C_v. As C_v is connected, there must exist a vertex $x \in C_v \setminus M_{v,w}$ such that x is adjacent to a vertex $y \in M_{v,w} \cap C_v$ (see Figure 3.6). Now x and $y \in M_{v,w}$ are adjacent, but there is no edge between x and $w \in C_w \cap M_{v,w}$. Since $M_{v,w}$ is a module, we have a contradiction. ☐

[4] In [21] Gallai showed this lemma for the set $W_{v,w}$ instead of $M_{v,w}$.

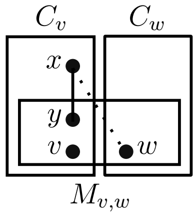


Figure 3.6.: Illustration for the proof of Lemma 41

Corollary 42. *Let $i \in [0, n-1]$ and $v \in V$. If $G[D_{i,v}]$ or its complement is not connected, then for all $w \in D_{i,v} \setminus D_{i+1,v}$ we have $M_{v,w} = D_{i+1,w} \cup D_{i+1,v}$.*

From Corollary 40 and 42 we can conclude that in some cases there exists a vertex $w \in V$ such that $D_{i,v} = M_{v,w}$. As shown, this is the case if $G[D_{i,v}]$ and its complement are connected, or if $G[D_{i,v}]$ or its complement consist of two connected components. If $G[D_{i,v}]$ or its complement consists of more than two connected components, then for each $w \in D_{i,v}$ we have $M_{v,w} \neq D_{i,v}$. However, $D_{i,v}$ is the union of all connected components $D_{i+1,w}$ with $w \in D_{i,v}$. Thus, Corollary 42 shows that $D_{i,v}$ is the union of all $M_{v,w}$ where $w \in D_{i,v}$ is in a connected component different from the one containing v.

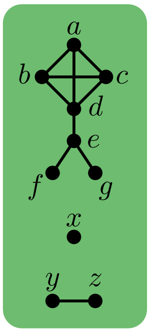
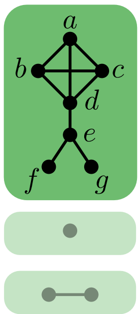
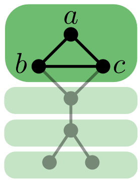
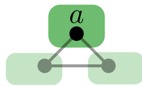
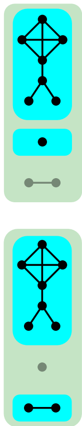
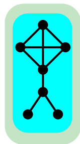
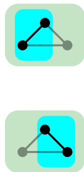
Let $v \in V$ be fixed. So far we have seen, that we obtain each set $D_{i,v}$ by taking the union of certain submodules $M_{v,w}$ of $D_{i,v}$. We show in the following that we can partition the vertex set V into $A_0^v, \dots, A_k^v$ such that

$$D_{i,v} = \bigcup_{w \in A_i^v} M_{v,w},$$

where k is minimal with $D_{k,v} = \{v\}$. In order to obtain this partition we order the modules $M_{v,w}$ with $w \in V$ with respect to proper inclusion. This order is a strict weak order (Lemma 43). Hence, incomparability is an equivalence relation. If we identify each module $M_{v,w}$ with its vertex w, the incomparability relation leads to an equivalence relation on the vertex set V. The resulting equivalence classes form the partition $\{A_0^v, \dots, A_k^v\}$. Consequently, we obtain the sets $D_{i,v}$ by taking the union of all sets $M_{v,w}$ that are incomparable with respect to proper inclusion. An example showing the connection between $D_{i,v}$, $M_{v,w}$ for $w \in V$ and the sets $A_0^v, \dots, A_k^v$ for a specific vertex $v \in V$ is given in Figure 3.7a and c. We define the relation $\prec_v$ by letting $w_1 \prec_v w_2$ if and only if the module spanned by v, w_2 is a proper subset of the module spanned by v, w_1. Further, w_1 and w_2 are *incomparable* with respect to $\prec_v$ if neither $w_1 \prec_v w_2$ nor $w_2 \prec_v w_1$. Figure 3.7b depicts the relation $\prec_v$ for the example in Figure 3.7a.

Lemma 43. *For every v the relation $\prec_v$ is a strict weak order, that is, a strict partial order, where incomparability is transitive.*

Proof. It is easy to see that $\prec_v$ is transitive and irreflexive. Let us show that incomparability is transitive. Thus, let w_1 and w_2, and w_2 and w_3 be incomparable with respect to $\prec_v$, and let us assume that w_1 and w_3 are comparable, that is, without loss of generality we have $w_1 \prec_v w_3$, which means $M_{v,w_1} \supset M_{v,w_3}$. Let $i \in \{0, \dots, n\}$ be maximal such that $D_{i,v}$ contains M_{v,w_1}, M_{v,w_2} and M_{v,w_3}.

	$i = 0$:	$i = 1$:	$i = 2$:	$i = 3$:
$D_{i,a}$:				
Sets $M_{a,w}$ for $w \in V$: (ordered according to set inclusion)	$M_{a,x}$ $M_{a,y}$ $M_{a,z}$ 	$M_{a,d}$ $M_{a,e}$ $M_{a,f}$ $M_{a,g}$ 	$M_{a,b}$ $M_{a,c}$ 	$M_{a,a}$ 
Partition $A^a_0, \dots, A^a_3$:	$A^a_0 = \{x, y, z\}$	$A^a_1 = \{d, e, f, g\}$	$A^a_2 = \{b, c\}$	$A^a_3 = \{a\}$

(a) The sets $D_{i,a}$, $M_{a,w}$ and $A^a_0, \dots, A^a_3$ for a given graph with vertex a

Relation $\prec_a$:	x, y, z	$\prec_a$	d, e, f, g	$\prec_a$	b, c	$\prec_a$	a

(b) The relation $\prec_a$

$G[D_{i,a}]$ is	not connected	connected	connected	connected
$\overline{G}[D_{i,a}]$ is	connected	connected	not connected	connected
Result:	$D_{0,a} = M_{a,x} \cup M_{a,w}$ for $w \in \{y, z\}$	$D_{1,a} = M_{a,w}$ for $w \in \{d, e, f, g\}$	$D_{2,a} = M_{a,b} \cup M_{a,c}$	$D_{3,a} = M_{a,a}$

(c) Connection between the sets $D_{i,a}$ and $M_{a,w}$

Figure 3.7.

First of all, we show that $M_{v,w_j} \neq D_{i,v}$ for all $j \in \{1,2,3\}$. M_{v,w_3} cannot be equal to $D_{i,v}$ as M_{v,w_3} is a proper subset of M_{v,w_1}. If the module M_{v,w_2} was equal to $D_{i,v}$, then $M_{v,w_3} \subset M_{v,w_2}$, and w_3 and w_2 would be comparable with respect to $\prec_v$. Thus, $M_{v,w_2} \neq D_{i,v}$. Finally, M_{v,w_1} cannot be equal to $D_{i,v}$ either since $M_{v,w_1} = D_{i,v}$ implies that $M_{v,w_2} \subset M_{v,w_1}$ and then w_2 and w_1 would be comparable. Consequently, neither of M_{v,w_1}, M_{v,w_2} and M_{v,w_3} is equal to $D_{i,v}$, and $|D_{i,v}| > 1$.

Now, if $G[D_{i,v}]$ is connected and co-connected, we can partition $D_{i,v}$ into maximal proper modules, and for all $j \in [3]$ we obtain that M_{v,w_j} is a subset of module $D_{i+1,v}$ if $w_j \in D_{i+1,v}$ or equal to $D_{i,v}$ if $w_j \in D_{i,v} \setminus D_{i+1,v}$ (Corollary 40). As we have shown above that $M_{v,w_j} \neq D_{i,v}$ for all $j \in [3]$, we have $M_{v,w_1}, M_{v,w_2}, M_{v,w_3} \subseteq D_{i+1,v}$, which is a contradiction to the choice of i.

If $G[D_{i,v}]$ is not connected, we can partition $D_{i,v}$ into its connected components. The case of $\overline{G}[D_{i,v}]$ being not connected can be treated analogously. For every $u \in D_{i,v}$, the set $D_{i+1,u}$ is the connected component of $G[D_{i,v}]$ containing u. Let us denote $D_{i+1,u}$ by C_u. Since i has been chosen maximal, there has to be a $j \in \{1,2,3\}$ such that M_{v,w_j} is not contained in C_v. For this j, vertex w_j must be a vertex in $M_{v,w_j} \setminus C_v$, and by Corollary 42 we obtain that $M_{v,w_j} = C_v \dot\cup C_{w_j}$. As w_1 and w_2 are incomparable and w_2 and w_3 are incomparable, independent from our choice of j, there exists an index $k \in \{1,2,3\} \setminus \{j\}$ such that w_j and w_k are incomparable. Thus, M_{v,w_k} cannot be a proper subset of $C_v \dot\cup C_{w_j}$, and consequently, $M_{v,w_k} \setminus C_v \neq \emptyset$. As above, we obtain that module M_{v,w_k} is equal to $C_v \dot\cup C_{w_k}$. Let us assume $j = 3$ or $k = 3$. Module $M_{v,w_3} = C_v \dot\cup C_{w_3}$ is a proper subset of module M_{v,w_1}. Thus, $M_{v,w_1} \setminus C_v \neq \emptyset$ and we can deduce $M_{v,w_1} = C_v \dot\cup C_{w_1}$ as we did before. Since both M_{v,w_1} and M_{v,w_3} are the union of two connected components, M_{v,w_3} cannot be a proper subset of M_{v,w_1}. Therefore, $j = 1$ and $k = 2$, or $j = 2$ and $k = 1$. As a consequence, we have $M_{v,w_1} = C_v \dot\cup C_{w_1}$ and $M_{v,w_2} = C_v \dot\cup C_{w_2}$. Now, if $M_{v,w_3} \setminus C_v \neq \emptyset$, then M_{v,w_3} is the disjoint union of the connected components C_v and C_{w_3}, a contradiction to $M_{v,w_3} \subset M_{v,w_1}$. If M_{v,w_3} is a subset of C_v, then M_{v,w_3} is a proper subset of M_{v,w_2}, which yields that w_2 and w_3 are comparable, a contradiction. Hence, incomparability is transitive. □

There exists an STC-formula $\varphi_\prec(x, y_1, y_2)$ such that for all $v, w_1, w_2 \in V$ we have $G \models \varphi_\prec[v, w_1, w_2]$ if, and only if, $w_1 \prec_v w_2$, that is, the module spanned by v, w_2 is a proper subset of the module spanned by v, w_1. Let φ_M be the formula from Lemma 38. Then we let

$$\begin{aligned}\varphi_\prec(x, y_1, y_2) := \quad & \forall z\big(\varphi_M(x, y_2, z) \to \varphi_M(x, y_1, z)\big) \\ & \wedge \exists z\big(\varphi_M(x, y_1, z) \wedge \neg\varphi_M(x, y_2, z)\big). \end{aligned} \tag{3.1}$$

According to Lemma 43 incomparability with respect to $\prec_v$ is transitive. Hence, incomparability is an equivalence relation. We denote the incomparability of two vertices w and w' by $w \sim_v w'$. We let $[w]_v$ be the equivalence class of w, and $V/_{\sim_v}$ be the set of all equivalence classes. Then $V/_{\sim_v} = \{A_0^v, \dots, A_k^v\}$. We let $[z]_v \prec_v [w]_v$ if there exist $z' \in [z]_v$ and $w' \in [w]_v$ such that $z' \prec_v w'$. Notice that if w and w', and z and z' are incomparable with respect to strict weak order $\prec_v$, then $z \prec_v w$ implies $z' \prec_v w'$, and $\prec_v$ induces a strict linear order on $V/_{\sim_v}$.

We use the strict linear order on the equivalence classes of the incomparability relation induced by $\prec_v$ to assign numbers to the equivalence classes, which match their position within the strict linear order. We assign 0 to the smallest equivalence class regarding $\prec_v$. The largest equivalence class regarding $\prec_v$ is $[v]_v = \{v\}$. Let $p_v\colon V/_{\sim_v} \to \mathbb{N}$ be this assignment. Then $p_v([z]_v) < p_v([w]_v)$ if and only if $[z]_v \prec_v [w]_v$, for all $z, w \in V$. We let

$$S_{i,v} := \{v\} \cup \bigcup\{M_{v,w} \mid p_v([w]_v) = i,\ w \in V\}$$

for all $i \in [0, n]$. Thus, $S_{i,v}$ is the union of $\{v\}$ and all modules $M_{v,w}$ where w belongs to the equivalence class at position i regarding $\prec_v$. If $k+1$ is the number of equivalence classes of $\sim_v$, then

$$S_{i,v} = \begin{cases} \bigcup\{M_{v,w} \mid p_v([w]_v) = i,\ w \in V\} & \text{if } i \leq k \\ \{v\} & \text{if } i \geq k \end{cases}$$

Lemma 44. *For all $i \in \{0, \ldots, n\}$ and $v \in V$, we have $D_{i,v} = S_{i,v}$.*

Proof. We prove this by induction on i. First, let us consider $i = 0$. The set $D_{0,v}$ is equal to V for all $v \in V$. We show that $S_{0,v}$ is equal to V as well. Let G and $\overline{G}$ be connected. We prove that there exists a $w \in V$ such that $M_{v,w} = V$: If $n = 1$, then $M_{v,v} = V$ and $w = v$ serves the purpose. If $n > 1$, we can apply Lemma 39, and let w be from a different maximal proper module than the one containing v. We obtain $M_{v,w} = V$, and $p_v([w]_v) = 0$. Thus, $S_{0,v}$ is equal to V if G and $\overline{G}$ are connected. Let G or $\overline{G}$ be not connected, and for each $u \in V$ let C_u be the connected component of G or $\overline{G}$ containing u. For all $w \in V$ we have either $M_{v,w} \subseteq C_v$ if $w \in C_v$, or if $w \notin C_v$, we have $M_{v,w} = C_v \cup C_w$ according to Lemma 41. Thus, for all $w \notin C_v$ we have $p_v([w]_v) = 0$, and $S_{0,v}$ is the union of $\{v\}$ and all sets $C_v \cup C_w$ with $w \notin C_v$ and therefore equal to V.

Let $i > 0$. By inductive assumption, we have $D_{i-1,v} = S_{i-1,v}$. Let us first consider the case where $D_{i-1,v}$ and $S_{i-1,v}$ are equal to $\{v\}$. Then clearly, $D_{i,v} = \{v\}$. Further, if there exists an $w \in V$ with $p_v([w]_v) = i-1$, then $w = v$. Thus, $[v]_v$, the largest equivalence class regarding $\prec_v$, is situated at position at most $i-1$. Therefore, $S_{i,v} = \{v\}$ as well. Hence $D_{i,v} = S_{i,v}$.

Now we let $D_{i-1,v}$ and $S_{i-1,v}$ contain more than one vertex. Then, $i-1$ is smaller than the position of the largest equivalence class within the strict linear order $\prec_v$. In the following suppose $G[D_{i-1,v}]$ is connected and co-connected. Then there is a w' such that $D_{i-1,v} = M_{v,w'}$ (Corollary 40). First, we show that $p_v([w']_v) = i-1$: Since w' is contained in $M_{v,w'} = S_{i-1,v}$, there must exist a $w'' \in V$ such that $p_v([w'']_v) = i-1$ and $w' \in M_{v,w''}$. Consequently, $M_{v,w'} \subseteq M_{v,w''}$, and $p_v([w']_v) \geq p_v([w'']_v)$. Further, we have $M_{v,w''} \subseteq M_{v,w'}$ as $M_{v,w'} = S_{i-1,v}$ and $M_{v,w''} \subseteq S_{i-1,v}$. Therefore, $p_v([w'']_v) \geq p_v([w']_v)$. It follows that $p_v([w'']_v)$ and $p_v([w']_v)$ are equal, and we obtain $p_v([w']_v) = i-1$. As a consequence, $p_v([w]_v) = i-1$ for all $w \in V$ with $M_{v,w} = D_{i-1,v}$.

Next we prove that $D_{i,v} = S_{i,v}$ in the case where $G[D_{i-1,v}]$ and its complement are connected: The set $D_{i,v}$ is a maximal proper module of $G[D_{i-1,v}]$ containing v. According to Corollary 40, we have $M_{v,w} = D_{i-1,v}$ and $p_v([w]_v) = i-1$ for all $w \in D_{i-1,v}$ with $w \notin D_{i,v}$. Therefore, for all $w \in V$ vertex w is contained in module $D_{i,v}$ if and only if $p_v([w]_v) > i-1$. It follows that $M_{v,w} \subseteq D_{i,v}$ for all $w \in V$ with $p_v([w'']_v) = i-1$. Hence,$S_{i,v} \subseteq D_{i,v}$. Now, if $G[D_{i,v}]$ is connected and co-connected, there exists a vertex $z \in V$ such that $D_{i,v} = M_{v,z}$ (Corollary 40). Further, $p_v([z]_v) = i$ as $D_{i,v}$ is a proper module of $G[D_{i-1,v}]$. Thus, $D_{i,v} \subseteq S_{i,v}$, and therefore, $D_{i,v} = S_{i,v}$ in this case. If $G[D_{i,v}]$ or its complement is not connected, then $i < n$ and for all $w \in D_{i,v} \setminus D_{i+1,v}$ we have $M_{v,w} = D_{i+1,w} \cup D_{i+1,v}$ (Corollary 42). Thus, for all $w \in D_{i,v}$, either $M_{v,w} \subseteq D_{i+1,v}$ if $w \in D_{i+1,v}$ or $M_{v,w} = D_{i+1,w} \cup D_{i+1,v}$ if $w \notin D_{i+1,v}$. Hence, we have $p_v([w]_v) = i$ for all $w \in D_{i,v} \setminus D_{i+1,v}$, and $D_{i,v} = \bigcup\{M_{v,w} \mid w \in D_{i,v} \setminus D_{i+1,v}\} \subseteq S_{i,v}$. We obtain $D_{i,v} = S_{i,v}$.

Now, let $G[D_{i-1,v}]$ be not connected. The case of $\overline{G}[D_{i-1,v}]$ being not connected can be treated analogously. By Corollary 42 we know $M_{v,w} = D_{i,w} \cup D_{i,v}$ for all $w \in D_{i-1,v} \setminus D_{i,v}$. As for all $w_1, w_2 \in D_{i-1,v} \setminus D_{i,v}$ modules M_{v,w_1} and M_{v,w_2} are incomparable, $p_v([w_1]_v) = p_v([w_2]_v)$. Let us show that $p_v([w]_v) = i - 1$ for all $w \in D_{i-1,v} \setminus D_{i,v}$. Let $w \in D_{i-1,v} \setminus D_{i,v}$. Since $D_{i-1,v} = S_{i-1,v}$ by inductive assumption, there exists a $w' \in V$ such that $p_v([w']_v) = i - 1$ and $w \in M_{v,w'}$. Consequently, $M_{v,w} \subseteq M_{v,w'}$ and $p_v([w]_v) \geq p_v([w']_v)$. On the other hand we have that $D_{i-1,v}$ is equal to the union of all $M_{v,w}$ with $w \in D_{i-1,v} \setminus D_{i,v}$. As $w' \in D_{i-1,v}$, there exists a $w'' \in D_{i-1,v} \setminus D_{i,v}$ with $w' \in M_{v,w''}$. It follows that $M_{v,w'} \subseteq M_{v,w''}$ and $p_v([w']_v) \geq p_v([w'']_v)$. Since modules $M_{v,w}$ and $M_{v,w''}$ are incomparable, we have $p_v([w'']_v) = p_v([w]_v)$. Therefore, $p_v([w]_v) = i - 1$.

The set $D_{i,v}$ is the connected component of $G[D_{i-1,v}]$ containing v, and every module $M_{v,w}$ with $p_v([w]_v) \geq i$ is a subset of $D_{i,v}$. Similar to the case where $G[D_{i-1,v}]$ and its complement are connected, we can use Corollary 40 and 42 to obtain $D_{i,v} = S_{i,v}$. □

Theorem 45. *There is an* STC+C*-formula* $\varphi_D(p, x, z)$ *such that for all graphs* G*, all* $i \in N(G)$ *and all vertices* $v \in V(G)$ *the set* $\varphi_D[G, i, v; z]$ *is the set* $D_{i,v}$ *of the modular decomposition of* G*.*

Proof. First we define a formula φ_{ord} that assigns to each vertex $w \in W$ the position $p_v([w]_v)$ of $[w]_v$ within the strict linear order of the equivalence classes of the incomparability relation induced by $\prec_v$ for $v \in V$. More precisely, $G \models \varphi_{ord}[v, w, n]$ if and only if $p_v([w]_v) = n$, for all $v, w \in V(G)$ and $n \in N(G)$. Clearly, φ_{ord} is satisfied for $v, w \in V, n \in N(V)$ exactly if n is the number of equivalence classes that are smaller than $[w]_v$ regarding $\prec_v$. Thus, we need an STC+C-formula which counts the number of equivalence classes smaller than $[w]_v$. We obtain such an STC+C-formula by an easy application of the Transduction Lemma for parameterized STC+C-transductions. We use a transduction $\Theta(x) = (\theta_U(x, y), \theta_\approx(x, y, y'), \theta_\prec(x, y, y'))$ where for each $v \in V$, θ_U and $\theta_\approx$ define the equivalence classes of $\sim_v$, and $\theta_\prec$ defines the strict linear order on them induced by $\prec_v$. More precisely, we let

$$\begin{aligned}\theta_U(x, y) &= \top,\\ \theta_\approx(x, y, y') &= \neg\varphi_\prec(x, y, y') \wedge \neg\varphi_\prec(x, y, y'),\\ \theta_\prec(x, y, y') &= \varphi_\prec(x, y, y'),\end{aligned}$$

where $\varphi_\prec$ is the formula defined in (3.1). Then we obtain φ_{ord} by applying the Transduction Lemma to the formula $\#y'(y' \prec y) = p$.

Now, we let φ_M be the formula from Lemma 38, and we apply Lemma 44, that is, we use that $D_{i,v} = S_{i,v}$, Then it is easy to see that the following formula is as desired:

$$\varphi_D(p, x, z) := \exists y\big(\varphi_{ord}(x, y, p) \wedge \varphi_M(x, y, z)\big) \vee x = z. \qquad \square$$

As STC+C-formulas can be evaluated in logarithmic space [62], we obtain the following corollary.

Corollary 46. *There exists a Turing machine, which given a graph $G = (V, E)$, a number $i \leq |V|$ and vertices $v, w \in V$ decides in* LOGSPACE *whether $w \in D_{i,v}$.*

3.5. A Logspace Algorithm for the Modular Decomposition Tree

In the previous section we showed that the modular decomposition is definable in STC+C, and therefore, computable in logarithmic space. The aim of this section is to briefly outline the logspace algorithm behind the STC+C-formulas. Further, we define the modular decomposition tree, and extend the algorithm to a logspace algorithm for the modular decomposition tree.

Logspace Algorithm for Corollary 46

According to Corollary 46 there exists a logspace algorithm that decides for a graph G, vertices $v, w \in V$ and $i \in [0, n]$ whether vertex w is contained in $D_{i,v}$. In the following we shortly describe such an algorithm. It is based on the fact that $w \in D_{i,v}$ if, and only if, there exist (at least) $i + 1$ vertices $u_i, \dots, u_0$ such that $u_i = w$ and $M_{v,u_k} \subset M_{v,u_{k-1}}$ for all $k \in [i]$.

First of all, we need a logspace algorithm that tests for two vertices u, u' whether module $M_{v,u}$ is a proper subset of module $M_{v,u'}$. Clearly, this is not the case if $u' = v$, and this is always the case if $u = v$ and $u' \neq v$. Thus, let $u \neq v$ and $u' \neq v$. According to Corollary 37 the spanned modules are equal to their respective wedge classes. Hence, we have to decide whether $W_{v,u} \subset W_{v,u'}$. A vertex a is contained in set $W_{v,z}$ for $z \neq v$ if and only if there exists a vertex b with $b \neq a$ such that $\{v, z\}$ is wedge connected to $\{a, b\}$ in G or $\overline{G}$. Now wedge connectivity amount to (undirected) connectivity in the graph G_{wedge} with vertex set $\binom{V}{2}$ where there is an edge between $e, e' \in \binom{V}{2}$ if and only if $e \wedge e'$ in G or $\overline{G}$. Clearly, there is a logspace transducer that constructs G_{wedge}, and as (undirected) graph connectivity is in LOGSPACE [62], there exists a logspace transducer for wedge connectivity. Hence, it is possible to determine in logspace whether $a \in W_{v,u}$. In order to find out if $W_{v,u} \subset W_{v,u'}$, we check whether $a \in W_{v,u}$ implies $a \in W_{v,u'}$ for all vertices $a \in V$.

Now that we have a logspace algorithm that decides proper inclusion for two spanned modules $M_{v,u}$ and $M_{v,u'}$ with $u, u' \in V$, we can present an algorithm that finds out if $w \in D_{i,v}$ for $v, w \in V$ and $i \in [0, n]$. As already mentioned, we use that $w \in D_{i,v}$ if, and only if, there are (at least) $i + 1$ vertices $u_i, \dots, u_0$ such that $u_i = w$ and $M_{v,u_k} \subset M_{v,u_{k-1}}$ for all $k \in [i]$. The algorithm starts with $u_i = w$. In each step it determines for a vertex u_k with $k \in [i]$ a vertex u_{k-1} such that M_{v,u_k} is a proper subset of $M_{v,u_{k-1}}$ and there exists no vertex u'_{k-1} with $M_{v,u_k} \subset M_{v,u'_{k-1}} \subset M_{v,u_{k-1}}$. The algorithm counts the number of possible steps. If there are at least i steps, then $w \in D_{i,v}$.

Modular Decomposition Tree

Next, let us turn to the modular decomposition tree. Let the family of subsets $D_{i,v}$ with $i \in [0, n]$, $v \in V$ be the modular decomposition of graph $G = (V, E)$. The *modular* *decomposition tree* of G is the directed tree $T = (V_T, E_T)$ with

$$V_T := \{D_{i,v} \mid i \in [0,n], v \in V\}$$
$$E_T := \{(D_{i,v}, D_{i+1,v'}) \in V_T^2 \mid D_{i+1,v'} \subset D_{i,v}\}$$

Hence, the vertex set V is the root of the modular decomposition tree of G. The children of each vertex $D_{i,v}$ with $|D_{i,v}| > 1$ are the maximal proper modules of $G[D_{i,v}]$, and the singleton sets $\{v\}$ for $v \in V$ are the leaves of the tree.

Corollary 46 implies the following Corollary.

Corollary 47. *There exists a Turing machine, which given a graph $G = (V, E)$, outputs the modular decomposition tree of G in logarithmic space.*

Proof. First, there is a logspace Turing machine T_1 that outputs the sets $D_{i,v}$ for all $i \in [0,n]$ and $v \in V$ in the following way: For each $i \in [0,n]$, we set a marker for i, and we output the sets $D_{i,v}$ for all $v \in V$. To output a set $D_{i,v}$, we first mark the beginning of a new set $D_{i,v}$. Then we go through all vertices $w \in V$ in the lexicographic order of their representation in the input string, and we output $w \in V$ if $w \in D_{i,v}$. This is possible in logspace by Corollary 46.

We use this output as an input for a logspace Turing machine T_2 that eliminates duplicates. We keep the markers for $i \in [0,n]$ and we only keep the first occurrence of a set $D_{i,v}$ with $i \in [0,n]$, $v \in V$. Hence, for each set $D_{i,v}$ listed at marker i, there is no $i' > i$ such that $D_{i',v} = D_{i,v}$.

Again we use this output as an input for a logspace transducer, T_3, which now creates the modular decomposition tree. To output the vertex set we only have to remove the markers for $i \in [0,n]$. We obtain the (directed) edges by going through all sets $D_{i,v}$ and finding the edges outgoing at $D_{i,v}$. Thus, for every set $D_{i,v}$ we go to the next marker and check for each set $D_{i+1,w}$ at this next marker whether $D_{i+1,w}$ is included in $D_{i,v}$. If this is the case, we output the edge $\{D_{i,v}, D_{i+1,w}\}$.

The composition of T_1, T_2 and T_3 yields a logspace transducer for the modular decomposition tree. ☐

4. The Modular Decomposition Theorem

The Modular Decomposition Theorem is a tool for showing that certain classes of graphs admit FP+C-definable canonization. It also extends to logics L that are stronger than fixed-point logic with counting and are closed under parameterized FP+C-transductions. Let $\mathcal{C}$ be a class of graphs that is closed under induced subgraphs. The Modular Decomposition Theorem states that there is an L-canonization for $\mathcal{C}$ if there is one for the class of LO-colored graphs (defined in Section 2.3.4) with prime underlying graphs from class $\mathcal{C}$. We use the Modular Decomposition Theorem in Chapters 5 and 6 to prove that there exists an FP+C-canonization for the class of permutation graphs and the class of chordal comparability graphs, respectively. As a result, FP+C captures polynomial time on these graph classes. In [49] (also [50]) Laubner used modular decompositions to prove that FP+C captures polynomial time on interval graphs. The application of the Modular Decomposition Theorem in [49] (and [50]) would lead to a shorter proof.

Overview

Let us shortly elaborate on how the Modular Decomposition Theorem exploits the properties of the modular decomposition of a graph and why we consider LO-colored graphs with prime underlying graphs.

Given a graph we can contract all modules to vertices. The resulting graph is called a modular contraction and is introduced in Section 4.1. Given a modular contraction and the graphs induced by the modules we can recreate the original graph by replacing each vertex with the corresponding module. We can restore the original edge relation as modules are either completely connected with edges or not at all.

Now in order to compute a canon we could simply take the canon of the modules and the canon of the modular contraction and combine them. However, we need to be able to reassign each vertex in the canon of the modular contraction to the canon of the module it represents. That is why we do not use the canon of the modular contraction, but the canon of a colored version of it. Basically, we color the vertices of the modular contraction with the canons of the graphs induced by the corresponding modules. To realize this coloring we encode the canons into a binary relation on the numbers, which we call the representation of the canon. We introduce this representation in Section 4.2. As a result we obtain an LO-colored graph.

Hence, to be able to construct the canon of a graph, we need the canonization of LO-colored modular contractions to be definable. Modular contractions are either complete graphs, edgeless graphs or prime graphs. As complete graphs and edgeless graphs are easy to handle, we only require that the canonization of LO-colored graphs with prime underlying graphs is definable.

On a first glance it seems rather inconvenient to have to find a canonization for a class of

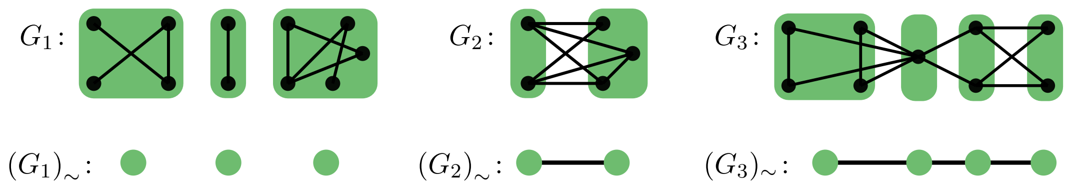
LO-colored graphs. However, LO-colored graphs have a linear order on its colors. Often it is not hard to extend a canonization of a class of (prime) graphs to a canonization of a class of (prime) graphs with ordered colors (see Chapters 5 and 6).

4.1. Modular Contraction

In this section we introduce the modular contraction of a graph and some basic properties of it. The modular contraction is basically the graph that we obtain by contracting the maximal proper modules of a graph to vertices.

For a graph $G = (V, E)$ let $\sim_G$ be the equivalence relation on V given by the partition $\{D_G(v) \mid v \in V\}$. $D_G(v)$ is defined on page 37. We let $G_\sim$ be the graph consisting of vertex set $V/_{\sim_G} = \{v/_{\sim_G} \mid v \in V\}$, where there is an edge between vertices $w/_{\sim_G}$ and $w'/_{\sim_G}$ if and only if there is one between w and w' in G. Since $w/_{\sim_G}$ and $w'/_{\sim_G}$ are the modules $D_G(w)$ and $D_G(w')$, edges are well-defined (Observation 30). We call $G_\sim$ the *modular contraction* of G. Thus, the modular contraction of a graph G is

- an edgeless graph with as many vertices as there are connected components in G if G is not connected,
- a complete graph with as many vertices as there are connected components in $\overline{G}$ if $\overline{G}$ is not connected, or
- if G and $\overline{G}$ are connected and $|V(G)| > 1$, a set of vertices one for each maximal proper module where there is an edge between two vertices exactly if the corresponding modules are (completely) connected with edges; or a single vertex if $|V(G)| = 1$.



(a) Graphs G_1 (not connected) and G_2 (not co-connected) and their modular contractions $(G_1)_\sim$ and $(G_2)_\sim$

(b) Graph G_3 (connected and co-connected) and its modular contraction $(G_3)_\sim$

Figure 4.1.: Modular contractions

Figure 4.1 depicts the graphs from Figure 3.1 together with their modular contractions.

Observation 48 ([21], Satz 1.8)**.** *If G and $\overline{G}$ are connected, then the modular contraction $G_\sim$ of G is prime.*

Proof. Let G and $\overline{G}$ be connected, and let us assume the modular contraction $G_\sim$ of G is not prime. Then there exists a non-trivial module $M_\sim$ in $G_\sim$. We show that

$M := \bigcup\{v/_{\sim_G} \in M_\sim\} \subset V$ is a module of G, which is a contradiction to $v/_{\sim_G} \in M_\sim$ being maximal proper modules of G. Let $v \in V \setminus M$ and $w, w' \in M$. We have

$$\begin{aligned}\{v, w\} \in E(G) &\iff \{v/_{\sim_G}, w/_{\sim_G}\} \in E(G_\sim)\\ &\iff \{v/_{\sim_G}, w'/_{\sim_G}\} \in E(G_\sim) \iff \{v, w'\} \in E(G)\end{aligned}$$

The second equivalence follows from $M_\sim$ being a module. □

As a consequence of Observation 48 we obtain that $G_\sim$ and also the modular contraction $\overline{G}_\sim$ of the complement graph $\overline{G}$ are prime if G and $\overline{G}$ are connected. Further, notice that for a prime graph H with more than 2 vertices, H and its complement are connected.

Lemma 49. *For every graph G, the modular contraction of G is isomorphic to an induced subgraph of G.*

Proof. For every vertex $w \in V(G_\sim)$ in the modular contraction of G, we can find a representative $v \in V(G)$ such that $w = v/_{\sim_G}$. Let U be the set of representatives. Then $v \in U \mapsto v/_{\sim_G} \in V(G_\sim)$ is an isomorphism between the induced subgraph $G[U]$ and the modular contraction $G_\sim$. □

For all modules $D_{i,v}$ of G, we denote the modular contraction $G[D_{i,v}]_\sim$ of $G[D_{i,v}]$ for all $i \leq n$ and $v \in V$ by $G_{i,v}$. Notice that $G_{0,v}$ is the modular contraction $G_\sim$ of G.

4.2. The Representation of a Graph

In the following we introduce the representation of a graph. As we only need to represent canons of graphs, we suppose our graph G has the vertex set $[|V(G)|]$. We use the representation to encode the given graph in a binary relation. Later, when we want to color vertices with graphs, we use these representations as colors instead. As a result we obtain an LO-colored graph.

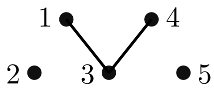
Let G be a graph with vertex set $[|V(G)|]$. We encode graph G in a symmetric binary relation $g_{\text{rep}}(G) \subseteq [0, |V(G)|]^2$. Relation $g_{\text{rep}}(G)$ contains all edges of the graph and the pair (n, n) where n is the number of vertices in G. More precisely, we let

$$g_{\text{rep}}(G) = \{(m, m') \mid \{m, m'\} \in E(G)\} \cup \{(n, n) \mid n = |V(G)|\}.$$

We call $g_{\text{rep}}(G)$ the *representation* of G. An example of a graph and its representation can be found in Figure 4.2. Conversely, we can interpret every symmetric binary relation $R \subseteq N(G)^2$ as a graph $g_{\text{graph}}(R)$. We call $g_{\text{graph}}(R)$ the *graph* of relation R. Let n' be minimal with the property that $(n', n') \in R$, or 0 if such an n' does not exist. We let

$$\begin{aligned}&V(g_{\text{graph}}(R)) := [1, n'] \quad \text{and}\\ &E(g_{\text{graph}}(R)) := \{\{m_1, m_2\} | (m_1, m_2) \in R \cap [1, n']^2\} \setminus \{\{n', n'\}\}.\end{aligned}$$

Notice that we obtain the empty graph if $n' = 0$. It is easy to see that $g_{\text{graph}}(g_{\text{rep}}(G)) = G$.

G: 

$g_{\text{rep}}(G) = \{(1,3),(3,1),(3,4),(4,3),(5,5)\}$

Figure 4.2.: A graph G and its representation $g_{\text{rep}}(G)$

4.3. The Modular Decomposition Theorem

In this section we present the Modular Decomposition Theorem, the main theorem of Part I. Let $\mathcal{C}$ be a class of graphs that is closed under induced subgraphs. Further, let L be a logic with $\mathsf{FP{+}C} \leq \mathsf{L}$ that is closed under parameterized $\mathsf{FP{+}C}$-transductions. The Modular Decomposition Theorem proves that in order to show the existence of an L-canonization of graph class $\mathcal{C}$ it is sufficient to find a (parameterized) L-canonization for all LO-colored graphs with prime underlying graphs from class $\mathcal{C}$.

LO-colored graphs are graphs that are colored with binary relations on an ordered set. They were introduced in Section 2.3.4. We call an LO-colored graph $H^* = (U,V,E,M,\trianglelefteq,L)$ *prime* if the underlying graph (V,E) is prime. For a class $\mathcal{C}$ of graphs that is closed under induced subgraphs, we let $\mathcal{C}^*$ be the class of all LO-colored graphs $H^* = (U,V,E,M,\trianglelefteq,L)$ where the underlying graph (V,E) is a prime graph in $\mathcal{C}$ and $|V| \geq 4$.

Theorem 50 (Modular Decomposition Theorem)**.** *Let* L *be a logic with* $\mathsf{L} \geq \mathsf{FP{+}C}$ *that is closed under parameterized* $\mathsf{FP{+}C}$*-transductions. Further, let* $\mathcal{C}$ *be a class of graphs which is closed under induced subgraphs. If* $\mathcal{C}^*$ *admits* L*-definable (parameterized) canonization, then* $\mathcal{C}$ *admits* L*-definable canonization.*

If $\mathsf{L} \geq \mathsf{FP{+}C}$ and there is a parameterized L-canonization of $\mathcal{C}^*$, then there also exists an L-canonization of $\mathcal{C}^*$ by Proposition 15. Hence, it suffices to prove the Modular Decomposition Theorem under the assumption that $\mathcal{C}^*$ admits L-definable canonization.

Remember that $\mathcal{C}^*$ admits L-definable canonization if, and only if, for $\tau = \{V,E,M,\trianglelefteq,L\}$ there is an $\mathsf{L}(\tau,\tau \cup \{\leq\})$-transduction Θ^{c} such that for every LO-colored graph $H^* \in \mathcal{C}^*$ the LO-colored graph $\Theta^{\mathrm{c}}[H^*]$ is an ordered copy of H^*. Let $\mathcal{K}^*$ be the class of all LO-colored graphs where the underlying graph is complete, and $\mathcal{I}^*$ be the class of all LO-colored graphs where the underlying graph is edgeless. Without loss of generality we can assume that Θ^{c} defines not only a canonization mapping for all prime LO-colored graphs in $\mathcal{C}^*$ but also for all LO-colored graphs in $\mathcal{K}^* \cup \mathcal{I}^*$. It is easy to describe in $\mathsf{FP{+}C}$ whether the underlying graph H of an LO-colored graph H^* is complete or edgeless. Also, it is not hard to define the canon of an LO-colored graph $H^* \in \mathcal{K}^* \cup \mathcal{I}^*$ in $\mathsf{FP{+}C}$ (see Example 17). We only need to assign the vertices of H^* to numbers according to the lexicographical order of the vertices' natural colors. (The natural color of a vertex of an LO-colored graph is defined in Section 2.3.4.) Thus, we can extend Θ^{c} in such a way that it first detects whether LO-colored graph H^* is in $\mathcal{K}^* \cup \mathcal{I}^*$ or not. If $H^* \in \mathcal{K}^* \cup \mathcal{I}^*$, then Θ^{c} defines the canon as explained above. If $H^* \notin \mathcal{K}^* \cup \mathcal{I}^*$, then Θ^{c} behaves as originally intended. Thus, from now on we assume that Θ^{c} is an L-canonization for the class $\mathcal{C}^*_{\mathcal{K}\mathcal{I}} := \mathcal{C}^* \cup \mathcal{K}^* \cup \mathcal{I}^*$. Notice that $\mathcal{C}^*_{\mathcal{K}\mathcal{I}}$ contains all prime LO-colored graphs where the underlying graph is in $\mathcal{C}$, because every prime graph with less than 4 vertices is complete or edgeless. Further, we let f^* be the canonization mapping defined by Θ^{c}.

In order to show the Modular Decomposition Theorem the idea is to construct the canon of each $G \in \mathcal{C}$ recursively using the modular decomposition. Let n be the number of vertices of G. Then for all $i \in \{n, \dots, 0\}$, starting with $i = n$, we inductively define the canons of the induced subgraphs $G[D_{i,v}]$ for all $v \in V$. We can trivially define the canon for each module that is a singleton. For the inductive step we consider the modular contraction $G_{i,v}$ of $G[D_{i,v}]$ for all $i < n$ and $v \in V$. For all $i < n$ and $v \in V$ the graph $G_{i,v}$ is prime if $G[D_{i,v}]$ and $\overline{G}[D_{i,v}]$ are connected, complete if $\overline{G}[D_{i,v}]$ is not connected or edgeless if $G[D_{i,v}]$ is not connected. We transform $G_{i,v}$ into an LO-colored graph $G^*_{i,v}$ by coloring every vertex $w/_{\sim_{G[D_{i,v}]}}$ of $G_{i,v}$ with the representation of the canon of graph $G[D_{i+1,w}]$. The canon of $G[D_{i+1,w}]$ is definable by inductive assumption. Then $G^*_{i,v} \in \mathcal{C}^*$ or $G^*_{i,v} \in \mathcal{K}^* \cup \mathcal{I}^*$. Thus, we can apply f^* to get $G^*_{i,v}$'s canon $K^*_{i,v}$. Now each vertex in $K^*_{i,v}$ stands for a module, and the color of every vertex is the representation of the canon of the graph induced by this module. Therefore, we can use the color to replace each vertex of $K^*_{i,v}$ by the module the vertex represents. Further, the order on the vertices of $K^*_{i,v}$ induces an order on the vertices of the resulting graph.

4.4. Proof of the Modular Decomposition Theorem

In the following we will give a detailed proof of the Modular Decomposition Theorem. We start by recursively defining the canonization mapping f which maps each graph $G \in \mathcal{C}$ to its canon $f(G)$. Afterwards we show that this canonization mapping is L-definable.

Canonization Mapping

In this section we define the canonization mapping f, which maps each graph $G \in \mathcal{C}$ to the canon $f(G) = \big(V_{f(G)}, E_{f(G)}, \leq_{f(G)}\big)$. We let the vertex set of canon $f(G)$ be $[|V(G)|]$. The linear order on the vertex set is the natural order on $[|V(G)|]$.

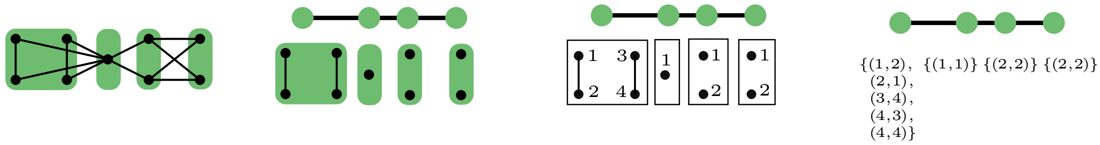
If $|V(G)| = 1$, then the canon of G is $f(G) := (\{1\}, \emptyset, \leq_{\{1\}})$. Now in order to define the canonization mapping f on graphs G with $|V(G)| > 1$, we use their decomposition into modules to recursively construct the canon of a graph from its modules' canons. In a first step we define $G^*_\sim$, the LO-*colored graph of* G, which has $G_\sim$, the modular contraction of G, as underlying graph. To obtain $G^*_\sim$ we color every vertex $w/_{\sim_G}$ in $G_\sim$ with the representation of the canon $f(G[D_G(w)])$ of $G[D_G(w)]$. More precisely, we let

$$G^*_\sim := (U_{G^*_\sim}, V_{G^*_\sim}, E_{G^*_\sim}, M_{G^*_\sim}, \trianglelefteq_{G^*_\sim}, L_{G^*_\sim})$$

where

$$\begin{aligned} U_{G^*_\sim} &:= V_{G^*_\sim} \mathbin{\dot\cup} M_{G^*_\sim}, \\ (V_{G^*_\sim}, E_{G^*_\sim}) &:= G_\sim, \\ M_{G^*_\sim} &:= [0, |V(G)|], \\ \trianglelefteq_{G^*_\sim} &:= \leq_{[0,|V(G)|]} \text{ and} \\ L_{G^*_\sim} &:= \{(v, i, j) \in V_{G^*_\sim} \times M^2_{G^*_\sim} \mid (i, j) \in g_{\mathrm{rep}}(f(G[D_G(v)]))\}. \end{aligned}$$

The construction of $G^*_\sim$ is illustrated in Figure 4.3.



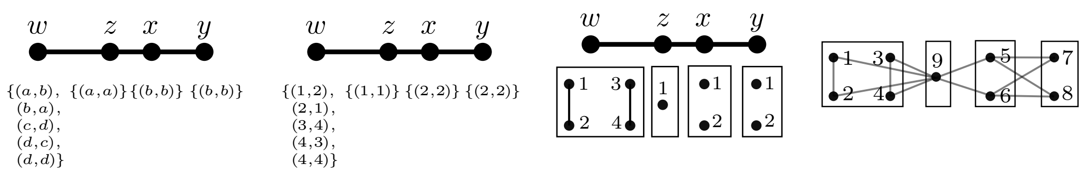
Graph G | Modular contraction $G_\sim$ and the maximal proper modules $D_G(w)$, $w \in V$ | Modular contraction $G_\sim$ and the canons $f(D_G(w))$, $w \in V$ | The graph $G^*_\sim$

Figure 4.3.: Construction of $G^*_\sim$

As $G_\sim$, the underlying graph of $G^*_\sim$, is a modular contraction, $G_\sim$ is a prime graph, complete or edgeless. Therefore, we can use the given canonization mapping f^* to obtain the canon of $G^*_\sim$:

$$K^*_\sim = (U_{K^*_\sim}, V_{K^*_\sim}, E_{K^*_\sim}, M_{K^*_\sim}, \trianglelefteq_{K^*_\sim}, L_{K^*_\sim}, \leq_{K^*_\sim}).$$

To get the canon of G we replace each vertex $w \in V_{K^*_\sim}$ of the ordered LO-colored graph $K^*_\sim$ by the graph represented by w's natural color (which is defined in Section 2.3.4). Since each LO-colored graph consists of a linear order on the basic color elements, the natural colors of isomorphic LO-colored graphs are equal. Hence, the natural colors of $K^*_\sim$ match the (natural) colors of $G^*_\sim$, which again encode the canons of the subgraphs induced by the modules the vertices represent. Thus, we replace the vertices of $K^*_\sim$ by the corresponding canons. We use the linear order on the vertices (given by the linear order $\leq_{K^*_\sim}$ restricted to the vertex set $V_{K^*_\sim}$) to replace one vertex after the other. We name the new vertices consecutively according to the time of their installment (and their order in the respective canon). Figure 4.4 shows the construction of $f(G)$.



A possible canon $K^*_\sim$ of graph $G^*_\sim$ where $w <_{K^*_\sim} x <_{K^*_\sim} y <_{K^*_\sim} z$, $a \trianglelefteq_{K^*_\sim} b \trianglelefteq_{K^*_\sim} c \trianglelefteq_{K^*_\sim} d$ | The underlying graph and the natural colors of $K^*_\sim$ | The underlying graph and the graphs represented by the natural colors of $K^*_\sim$ | Graph $f(G)$

Figure 4.4.: Construction of $f(G)$

In the following we describe the construction of the canon $f(G)$ more precisely. For all vertices $w \in V_{K^*_\sim}$, let $L^{\mathbb{N}}_w$ be the natural color of w, and let n_w be the only element with $(n_w, n_w) \in L^{\mathbb{N}}_w$. Since the module that w stands for (and whose induced subgraph's canon's representation is the natural color of w) consists of at least one vertex, such an n_w exists and $0 < n_w \leq |M_{K^*_\sim}|$. To construct the canon we assign each vertex n of the graph $g_{\text{graph}}(L_w^{\mathbb{N}})$ of representation $L_w^{\mathbb{N}}$ to the number

$$\text{nb}(w,n) := n \;+ \sum_{\substack{w' <_{K^*_\sim} w,\\ w' \in V_{K^*_\sim}}} n_{w'}, \tag{4.1}$$

where $w' <_{K^*_\sim} w$ if and only if $w' \leq_{K^*_\sim} w$ and $w' \neq w$. Clearly, mapping nb is a bijection, that maps (n, v), where n is a vertex in the graph represented by vertex v's natural color, to $m \in [|V(G)|]$.

We add a pair of numbers to the edges of $f(G)$ if they represent vertices from different modules, and the modules are completely connected; or they represent vertices from the same module that are connected by an edge. Thus, we add $\{m_1, m_2\}$ to the edges of $f(G)$ if

1. there exist an edge $\{w_1, w_2\} \in E_{K^*_\sim}$ and numbers $n_1, n_2 \in [|M_{K^*_\sim}|]$ such that $n_1 \leq n_{w_1}$, $n_2 \leq n_{w_2}$ and $(m_1, m_2) = \big(\text{nb}(w_1, n_1), \text{nb}(w_2, n_2)\big)$, or
2. there exist a vertex $w \in V_{K^*_\sim}$ and a pair $(n_1, n_2) \in L_w^{\mathbb{N}}$ such that $n_1 \neq n_2$ and $(m_1, m_2) = \big(\text{nb}(w, n_1), \text{nb}(w, n_2)\big)$.

Clearly, the ordered graph $f(G)$ is an ordered copy of G on the number sort. In the Observation 51 we show that for isomorphic graphs G_1 and G_2 we have $f(G_1) = f(G_2)$. Hence, f is a canonization mapping.

Observation 51. *For isomorphic graphs G and G' from class $\mathcal{C}$, we have $f(G) = f(G')$.*

Proof. Let h be an isomorphism between G and G'. We show that $f(G) = f(G')$ by induction. Clearly, we have $f(G) = f(G')$ if G and G' consist of only one vertex. Therefore, let $|V(G)| = |V(G')| > 1$. As the modular decomposition of a graph is unique, isomorphism h maps every maximal proper module of G to a maximal proper module of G'. Hence, the respective graphs induced by the maximal proper modules of G and G' are isomorphic, and by inductive assumption f maps them to the same canon. Further, h induces an isomorphism $h_\sim$ between $G_\sim$ and $G'_\sim$. Consequently, the graphs $G^*_\sim$ and $G'^*_\sim$ are isomorphic. They are mapped to order isomorphic copies $K^*_\sim$ and $K'^*_\sim$ by f^* . Let g be an isomorphism between them. Clearly, for each vertex $v \in V_{K^*_\sim}$, vertices v and $g(v)$ have the same natural color. Further, we have $v_1 \leq_{K^*_\sim} v_2$ if and only if $g(v_1) \leq_{K'^*_\sim} g(v_2)$. As a consequence, $f(G) = f(G')$. ☐

In the following five steps we show that f is L-definable.

Step 1: Transduction from the Graphs to the LO-Colored Graphs

For all modules $D_{i,v}$ of G with $i \leq V(G)$ and $v \in V(G)$, we denote the LO-colored graph $(G[D_{i,v}])^*_\sim$ of $G[D_{i,v}]$ by $G^*_{i,v}$. Notice that the underlying graph of $G^*_{i,v}$ is $G_{i,v}$. It is not hard to see that within the recursive definition of $f(G)$ we need the canons $f(G[D_{i,v}])$ of the subgraphs induced by all modules $D_{i,v}$ for $i \leq V(G)$ and $v \in V(G)$ of the modular decomposition of G.

The first step in constructing an L-formula that defines f is to define the colored version $G^*_{i,v}$ of the modular contraction $G_{i,v}$ for all $G \in \mathcal{C}$ and all $i \in N(G)$ and $v \in V(G)$.

For this purpose, we define a counting transduction $\Theta^{\#}(o, z, X)$, where o is a number variable, z is a structure variable, and X is a relational variable of type (n, s, n, n). It is a parameterized $\mathsf{L}[\{E\}, \{V, E, M, \trianglelefteq, L\}]$-counting transduction, which maps every graph G to an LO-colored graph $G^R_{i,v} := \Theta^{\#}[G, i, v, R]$ for $(G, i, v, R) \in \mathrm{Dom}(\Theta^{\#}(o, z, X))$. Thus, $\Theta^{\#}(o, z, X)$ defines a parameterized mapping from the graphs to the LO-colored graphs. For some triples $(i, v, R) \in G^{(o,z,X)}$ where R is a specific relation depending on i and v, the LO-colored graph $G^R_{i,v}$ is isomorphic to $G^*_{i,v}$. We let

$$\begin{aligned}\Theta^{\#}(o, z, X) = \big(&\theta_{\mathrm{dom}}(o, z, X), \theta_U(o, z, X, y), \theta_{\approx}(o, z, X, y, y'),\\ &\theta_V(o, z, X, y), \theta_E(o, z, X, y, y'),\\ &\theta_M(o, z, X, p), \theta_{\trianglelefteq}(o, z, X, p, p'), \theta_L(o, z, X, y, p, p')\big),\end{aligned}$$

where

$$\begin{aligned}\theta_{\mathrm{dom}}(o, z, X) &:= \neg\, \mathrm{largest}(o) \quad \text{(see Section 2.4)},\\ \theta_U(o, z, X, y) &:= \varphi_D(o, z, y),\\ \theta_{\approx}(o, z, X, y, y') &:= \exists o' \big(o + 1 = o' \wedge \varphi_D(o', y, y')\big),\\ \theta_V(o, z, X, y) &:= \top,\\ \theta_E(o, z, X, y, y') &:= E(y, y'),\\ \theta_M(o, z, X, p) &:= \top,\\ \theta_{\trianglelefteq}(o, z, X, p, p') &:= p \le p' \quad \text{and}\\ \theta_L(o, z, X, y, p, p') &:= \exists o' \big(o + 1 = o' \wedge X(o', y, p, p')\big).\end{aligned}$$

As a reminder, formula $\varphi_D(o, z, y)$, which was introduced in Theorem 45, defines the set $D_{i,v}$ of the modular decomposition, i.e. for all $i \in N(G)$ and all vertices $v \in V(G)$ we have $\varphi_D[G, i, v; y] = D_{i,v}$.

Let $G \in \mathcal{C}$. We say a triple $(i, v, R) \in G^{(o,z,X)}$ is *suitable* for $G \in \mathcal{C}$ if it satisfies $i < |V(G)|$ and the following property:

> For all $w \in D_{i,v}$ the relation $R_{i+1,w} := \{(n_1, n_2) \mid (i + 1, w, n_1, n_2 \in R)\}$ is the representation of the canon of $G[D_{i+1,w}]$ defined by f.

We let $\mathrm{Suit}(G)$ be the set of all suitable triples for G.

Lemma 52. *Let G be a graph in $\mathcal{C}$ and let $(i, v, R) \in G^{(o,z,X)}$ be a suitable triple for G. Then $(G, i, v, R) \in \mathrm{Dom}(\Theta^{\#}(o, z, X))$ and $G^R_{i,v} = G^*_{i,v}$.*

Proof. Let $G \in \mathcal{C}$ and let $i \in N(G)$, $v \in V(G)$ and $R \subseteq N(G) \times V(G) \times N(G)^2$ be such that $(i, v, R) \in \mathrm{Suit}(G)$. Then, $i < |V(G)|$. Therefore, $(G, i, v, R) \in \mathrm{Dom}(\Theta^{\#}(o, z, X))$. Clearly, $\theta_U[G, i, v, R; y]$ is the set $D_{i,v}$. Further, $\theta_{\approx}[G, i, v, R; y, y']$ is the equivalence relation $\{D_{i+1,w} \mid w \in V(G)\}$. Let $\approx$ denote this equivalence relation. Then the universe of $G^R_{i,v}$ is the set $D_{i,v}/_{\approx} \,\dot\cup\, [0, |V(G)|]$. The vertex set $V(G^R_{i,v})$ is $D_{i,v}/_{\approx}$, and it is not hard to see that the formulas θ_V, $\theta_{\approx}$ and θ_E of transduction $\Theta^{\#}(o, z, X)$ define the graph $G_{i,v}$. Further, $M(G^R_{i,v}) = [0, |V(G)|]$ and $\trianglelefteq(G^R_{i,v})$ is the natural order on $[0, |V(G)|]$. Finally, formula θ_L defines the color relation. As $(i, v, R) \in \mathrm{Suit}(G)$, relation $\{(m_1, m_2) \mid (i + 1, w, m_1, m_2) \in R\}$ is the representation of the canon of $G[D_{i+1,w}]$ for all $w \in V(G)$, and we obtain that $G^R_{i,v}$, that is, $\Theta^{\#}[G, i, v, R]$, is equal to $G^*_{i,v}$ for all $(i, v, R) \in \mathrm{Suit}(G)$. ☐

Later, we will make sure that the triple of parameters (o, z, X) is always interpreted by a suitable triple.

Step 2: Transduction from the Graphs to the ordered LO-Colored Graphs

Since $\Theta^{\#}(o, z, X)$ is a counting transduction, there exists a parameterized L-transduction $\Theta^{*}(o, z, X)$ with the same domain such that $\Theta^{\#}[G, i, v, R]$ and $\Theta^{*}[G, i, v, R]$ are isomorphic for all (G, i, v, R) in the domain (Proposition 14). As a consequence, Lemma 52 holds for L-transduction $\Theta^{*}(o, z, X)$ in a similar way: For a graph $G \in \mathcal{C}$ and a suitable triple $(i, v, R) \in G^{(o,z,X)}$ the tuple (G, i, v, R) is in the domain of $\Theta^{*}(o, z, X)$, and the LO-colored graph $\Theta^{*}[G, i, v, R]$ is isomorphic to $G^{*}_{i,v}$.

Let $G \in \mathcal{C}$ and let (i, v, R) be a suitable triple for G. Then $\Theta^{*}[G, i, v, R]$, as it is isomorphic to $G^{*}_{i,v}$, is an LO-colored graph in $\mathcal{C}^{*}_{\mathcal{KI}}$. We know that there exists an L-canonization Θ^{c} for the class $\mathcal{C}^{*}_{\mathcal{KI}}$ of LO-colored graphs. According to Proposition 12 we can compose transductions $\Theta^{*}(o, z, X)$ and Θ^{c}. We obtain an L$[\{E\}, \{V, E, M, \trianglelefteq, L, \leq\}]$-transduction $\Theta^{*c}(o, z, X)$ where $(G, i, v, R) \in \mathrm{Dom}(\Theta^{*c}(o, z, X))$ for all $G \in \mathcal{C}$ and $(i, v, R) \in \mathrm{Suit}(G)$.

As $\Theta^{*}[G, i, v, R]$ and $G^{*}_{i,v}$ are isomorphic for $G \in \mathcal{C}$ and suitable triples (i, v, R) for G, and Θ^{c} is a canonization, the ordered LO-colored graph $\Theta^{c}[\Theta^{*}[G, i, v, R]]$ is an ordered copy of $G^{*}_{i,v}$. Further, for all $(G, i, v, R) \in \mathrm{Dom}(\Theta^{*c}[o, z, X])$ the ordered LO-colored graphs $\Theta^{*c}[G, i, v, R]$ and $\Theta^{c}[\Theta^{*}[G, i, v, R]]$ are isomorphic. Thus, $\Theta^{*c}[G, i, v, R]$ also is an ordered copy of $G^{*}_{i,v}$ for $G \in \mathcal{C}$ and suitable triples (i, v, R) for G. We denote the ordered copy $\Theta^{*c}[G, i, v, R]$ of $G^{*}_{i,v}$ by $K^{*}_{i,v}$.

The relations between the different transductions used in Step 2 are illustrated in Figure 4.5.

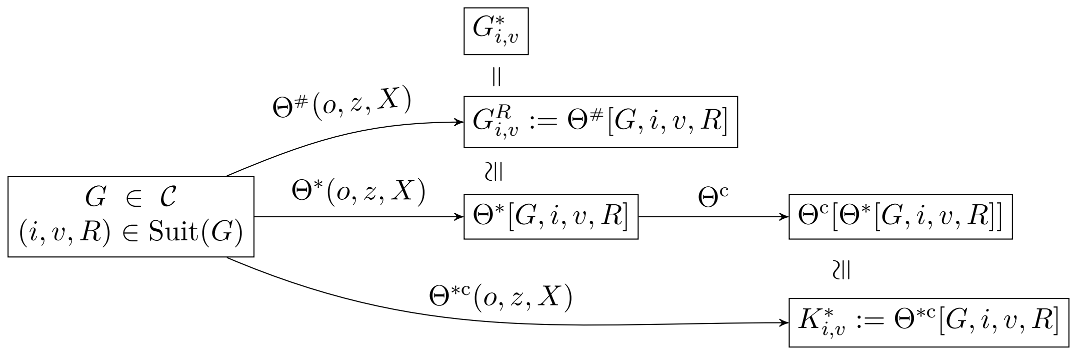


Figure 4.5.: Overview of the transductions

Step 3: Defining the Edge Relation

In the following we construct a $\{V, E, M, \trianglelefteq, L, \leq\}$-formula that given an ordered LO-colored graph $K^*_{i,v}$ defines the edge relation of $f(G[D_{i,v}])$ as it is defined in the first part of this section.

In order to do this, we have to define the function $\mathrm{nb}(w, n)$ from (4.1) in logic L. For every vertex w of $K^*_{i,v}$ and each vertex n occurring in the graph of the natural color of vertex w, $\mathrm{nb}(w, n)$ is the number that vertex n is assigned to in the canon of $G[D_{i,v}]$.

Function $\mathrm{nb}(w, n)$ depends on the values $n_{w'}$ for certain vertices w'. Value n_w is the number of vertices in the graph represented by the natural color of vertex w. We can determine this number by finding the only vertex u for which (u, u) belongs to the color of w. Then n_w is the number of vertices that are smaller than u with respect to the linear order $\trianglelefteq(K^*_{i,v})$ of the basic color elements. We define n_w in formula φ_{n_w}:

$$\varphi_{n_w}(x, p) := \exists y\Big(L(x, y, y) \ \wedge\ p = \#y'\big(\trianglelefteq(y', y) \wedge y' \neq y\big)\Big).$$

Then we have $K^*_{i,v} \models \varphi_{n_w}[w, n_w]$ if, and only if, the graph represented by the natural color $L^{\mathbb{N}}_w$ of vertex w has n_w vertices, where $L^{\mathbb{N}}_w$ is the natural color of w in $K^*_{i,v}$. Notice that formula φ_{n_w} cannot be satisfied if w is a basic color element.

To define function $\mathrm{nb}(w, n)$, we first check whether $n \in [n_w]$. Then we count the vertices occurring in the graphs of the natural colors of all vertices w' that are smaller than w with respect to the linear order $\leq(K^*_{i,v})$, and the vertices n' in the graph of the natural color of w with $0 < n' \leq n$. Thus, we let

$$\begin{aligned}\varphi_{\mathrm{nb}}(x, r, s) := \exists p\ \big(\varphi_{n_m}(x, p) \wedge \text{“}0 < r \leq p\text{”}\big)\\ \wedge \Big(s = \#(x', r')\ \Big(\exists p'\ \big(\varphi_{n_m}(x', p') \wedge \leq(x', x) \wedge x' \neq x \wedge \text{“}0 < r' \leq p'\text{”}\big)\\ \vee \big(x' = x \wedge \text{“}0 < r' \leq r\text{”}\big)\Big)\Big).\end{aligned}$$

Then $K^*_{i,v} \models \varphi_{\mathrm{nb}}[w, n, m]$ if and only if w is a vertex, $n \in [n_w]$ and $\mathrm{nb}(w, n) = m$ in $K^*_{i,v}$.

With formula φ_{nb} we are able to define the edge relation of the canon of $G[D_{i,v}]$. We let

$$\varphi_E(s_1, s_2) := \varphi_{E,1}(s_1, s_2) \vee \varphi_{E,2}(s_1, s_2)$$

where

$$\begin{aligned}\varphi_{E,1}(s_1, s_2) &:= \exists x_1, x_2, r_1, r_2\Big(E(x_1, x_2) \wedge \bigwedge_{j\in\{1,2\}} \varphi_{\mathrm{nb}}(x_j, r_j, s_j)\Big),\\ \varphi_{E,2}(s_1, s_2) &:= \exists x, y_1, y_2, r_1, r_2\Big(L(x, y_1, y_2) \wedge r_1 \neq r_2\\ &\qquad \wedge \bigwedge_{j\in\{1,2\}} \Big(r_j = \#y(\trianglelefteq(y, y_j) \wedge y \neq y_j) \wedge \varphi_{\mathrm{nb}}(x, r_j, s_j)\Big)\Big)\end{aligned}$$

It is not hard to see that $\varphi_{E,1}[K^*_{i,v}; s_1, s_2]$ and $\varphi_{E,2}[K^*_{i,v}; s_1, s_2]$ are exactly the edges of the canon of $G[D_{i,v}]$ obtained by Rule 1 and Rule 2 from the first part of this section.

Step 4: Pulling Back the Formula for the Edge Relation

Now we have a $\{V, E, M, \trianglelefteq, L, \leq\}$-formula $\varphi_E(s_1, s_2)$ that defines the edge relation of the canon $f(G[D_{i,v}])$ for a given ordered LO-colored graph $K_{i,v}$. In order to use the formula to construct an $\{E\}$-formula for the canonization mapping f, we need to pull it back under transduction $\Theta^{*c}(o, z, X)$. Hence, we apply the Transduction Lemma to the formula $\varphi_E(s_1, s_2)$. We obtain an $\{E\}$-formula $\varphi_E^{-\Theta^{*c}}(o, z, X, \bar{q}_1, \bar{q}_2)$. Let $G \in \mathcal{C}$ and let (i, v, R) be a suitable triple for G. Then $(G, i, v, R) \in \mathrm{Dom}(\Theta^{*c}(o, z, X))$. Thus, for all tuples $\bar{m}_1, \bar{m}_2$ of numbers in $N(G)$, we have

$$G \models \varphi_E^{-\Theta^{*c}}[i, v, R, \bar{m}_1, \bar{m}_2] \iff \begin{array}{l} \langle \bar{m}_1 \rangle_G, \langle \bar{m}_2 \rangle_G \in N(K_{i,v}^*) \text{ and} \\ K_{i,v}^* \models \varphi_E[\langle \bar{m}_1 \rangle_G, \langle \bar{m}_2 \rangle_G]. \end{array} \tag{4.2}$$

The length of tuples $\bar{q}_1, \bar{q}_2$, and therefore also of $\bar{m}_1, \bar{m}_2$, is the same and depends on the length of the tuple of domain variables of the canonization Θ^c. Let ℓ be the length of the listed tuples. Let $\bar{m}_1 = (m_1^1, \ldots, m_1^\ell)$ and let the other tuples be defined analogously. In the following we show that in each variable tuple we only need the first number variable, as the others are always assigned to 0.

Let G be again a graph in $\mathcal{C}$ and let (i, v, R) be a suitable triple for G. Since the vertex set of $f(G)$ is $[|V(G)|]$, we have $\langle \bar{m}_1 \rangle_G, \langle \bar{m}_2 \rangle_G \in [|V(G)|]$ for all $\bar{m}_1, \bar{m}_2 \in N(G)^\ell$ with $K_{i,v}^* \models \chi_E[\langle \bar{m}_1 \rangle_G, \langle \bar{m}_2 \rangle_G]$. Now remember that for a tuple $\bar{n} = (n_1, \ldots, n_\ell) \in N(G)^\ell$,

$$\langle \bar{n} \rangle_G = \sum_{i=1}^{\ell} n_i \cdot (|V(G)| + 1)^{i-1}.$$

Consequently, we have $m_1^j = 0$, and also $m_2^j = 0$, for all $j > 1$, which means that $m_1^1 = \langle \bar{m}_1 \rangle_G$ and $m_2^1 = \langle \bar{m}_2 \rangle_G$.

Now we define formula ϕ_E as follows:

$$\phi_E(o, z, X, q_1, q_2) := \varphi_E^{-\Theta^{*c}}(o, z, X, (q_1, 0, \ldots, 0), (q_2, 0, \ldots, 0)).$$

Then, for a graph $G \in \mathcal{C}$, $(i, v, R) \in \mathrm{Suit}(G)$ and $m_1, m_2 \in N(G)$ we have

$$G \models \phi_E[i, v, R, m_1, m_2] \iff \begin{array}{l} \text{There is an edge between vertices } m_1 \\ \text{and } m_2 \text{ in } f(G[D_{i,v}]). \end{array}$$

Step 5: Inductive Definition of the Canon $f(G)$

We are now able to inductively define the edge relation of the canon $f(G)$ of $G \in \mathcal{C}$. We let

$$\phi_K(s_1, s_2) := \exists o', z' \Big(o' = 0 \land s_1 \neq s_2 \land \mathrm{ifp}\big(X(o, z, q_1, q_2) \leftarrow \phi\big)(o', z', s_1, s_2) \Big)$$

where

$$\phi := \phi_1 \lor \big(\phi_2 \land (\phi_E \lor \phi_{n_w})\big)$$

and

$$\begin{aligned}\phi_1(o,z,q_1,q_2) &:= \text{ largest}(o) \wedge q_1 = 1 \wedge q_2 = 1,\\ \phi_2(o,z,X,q_1,q_2) &:= \neg \,\text{largest}(o) \wedge \exists o',z',q_1',q_2'(o+1=o' \wedge X(o',z',q_1',q_2')),\\ \phi_{n_w}(o,z,q_1,q_2) &:= q_1 = q_2 \wedge q_1 = \#y\ \varphi_D(o,z,y).\end{aligned}$$

The relational variable X within the inflationary fixed-point operator of formula ϕ_K is of type $(\mathrm{n},\mathrm{s},\mathrm{n},\mathrm{n})$. Let X^∞ be the relation assigned to variable X after the fixed-point is reached. We show in Lemma 53 that for each $i \in N(G)$ and $v \in V(G)$ the set of pairs $\{(n_1,n_2) \mid (i,v,n_1,n_2) \in X^\infty\}$ is the representation of the canon $f(G[D_{i,v}])$. For $i = 0$ and any vertex $v \in V(G)$ we have $D_{i,v} = V(G)$. Therefore, for all $G \in \mathcal{C}$ and all $n_1, n_2 \in N(G)$,

$$G \models \phi_K[n_1,n_2] \iff \{n_1,n_2\} \text{ is an edge in the canon } f(G).$$

Formula ϕ of the inflationary fixed-point operator is constructed such that ϕ_1 defines the basis of the inductive definition. For $i = |V(G)|$ and all vertices $v \in V(G)$, it ensures that the tuples describing the representation of the canon of $G[D_{i,v}]$, which consists only of one vertex, are added to the fixed-point relation in the first step. Thus, all tuples in $\{(|V(G)|, v, 1, 1) \mid v \in V(G)\}$ are added in the first step. Formulas ϕ_2 and $\phi_E \vee \phi_{n_w}$ take effect in the inductive step. In step k we add all tuples $(i,v,n_1,n_2) \in X^\infty$ to the fixed-point relation with $i = |V(G)| - k + 1$. Formula ϕ_2 ensures that we add only tuples (i,v,n_1,n_2) if the tuples for $i+1$ have already been included to the fixed-point relation. This way, i, v and the fixed-point relation form a suitable triple. Then, formula $\phi_E \vee \phi_{n_w}$ defines the representation of the canon of $G[D_{i,v}]$.

In the following lemma we show inductively that formula ϕ_K uses an inflationary fixed-point operator which in stage k of its iteration defines the representation of the canons of all $G[D_{i,v}]$ with $v \in V(G)$ and $i \geq |V(G)| - k + 1$.

Lemma 53. *Let X^k be the fixed-point relation that we get at stage k of the iteration of the inflationary fixed-point operator in formula ϕ_K. Further, let S^k be the set of all tuples $(i,v,n_1,n_2) \in N(G) \times V(G) \times N(G)^2$ where $i \geq |V(G)| - k + 1$ and (n_1,n_2) is in $g_{rep}(f(G[D_{i,v}]))$, the representation of the canon of $G[D_{i,v}]$. Then $X^k = S^k$.*

Proof. Of course, for $k = 0$ we have $X^k = \emptyset$ and $S^k = \emptyset$. For $k = 1$ it is easy to see that there is no tuple that satisfies ϕ_2 since $X^0 = \emptyset$. Thus, X^1 is the set $\phi_1[G; o,z,q_1,q_2] = \{(|V(G)|, v, 1, 1) \mid v \in V(G)\}$. Further, for all $v \in V(G)$ the representation of the canon of $G[D_{|V(G)|,v}]$ is $\{(1,1)\}$, and therefore, $X^1 = S^1$. Now let $k \geq 1$, and let $X^k = S^k$. In the following we prove that $X^{k+1} = S^{k+1}$ by showing that $X_j^{k+1} = S_j^{k+1}$ for all $j \in N(G)$, where S_j^{k+1} is the set of all tuples $(j,v,n_1,n_2) \in S^{k+1}$ and X_j^{k+1} is the set of tuples $(j,v,n_1,n_2) \in X^{k+1}$.

It is easy to see that $X_j^{k+1} = S_j^{k+1}$ for $j = |V(G)|$: We have already shown that $\phi_1[G; o,z,q_1,q_2] = X^1$ and that $X_1 = S_1$. Further, relation $\phi_2[G, \alpha[X^k/X]; o,z,q_1,q_2]$ cannot contain any tuple (i,v,n_1,n_2) with $i = |V(G)|$. Consequently, $X_j^{k+1} = S_1$. Since $S_1 = S_j^{k+1}$ for $j = |V(G)|$, we have $X_j^{k+1} = S_j^{k+1}$.

Next, let us consider $j < |V(G)| - k$. Then $j < |V(G)|$, and there does not exist a tuple $(i, v, n_1, n_2) \in \phi_1[G; o, z, q_1, q_2]$ with $i = j$. Further, by inductive assumption we have $X^k = S^k$, and by definition we know that S^k does not contain any tuples (j', v, n_1, n_2) with $j' < |V(G)| - k + 1$. Consequently, there cannot be a tuple (i, v, n_1, n_2) in $\phi_2[G, \alpha[X^k/X]; o, z, q_1, q_2]$ with $i = j$. Thus, for $j < |V(G)| - k$ we have $X_j^{k+1} = \emptyset$, and since S_j^{k+1} is also empty, we obtain $X_j^{k+1} = S_j^{k+1}$.

Now, let $|V(G)| - k \leq j < |V(G)|$. Then the relation $\phi_1[G; o, z, q_1, q_2]$ does not contain any tuple (i, v, m_1, m_2) with $i = j$. However, there exist a vertex $v \in V(G)$ and numbers $n_1, n_2 \in N(G)$ such that $(j, v, n_1, n_2) \in \phi_2[G, \alpha[X^k/X]; o, z, q_1, q_2]$ because $X^k = S^k$, by inductive assumption, and $S_{j'}^k$ is non-empty for all $j' \geq |V(G)| - k + 1$, by definition. Since we have $X^k = S^k$, and $j + 1 \geq |V(G)| - k + 1$ and $j < |V(G)|$, the relation $\{(n_1, n_2) \mid (j+1, w, n_1, n_2) \in X^k\}$ is the representation of the canon of $G[D_{j+1,w}]$ for all $w \in V(G)$. Therefore, (j, v, X^k) is a suitable triple for all $v \in V(G)$. As shown in Step 3 and 4, the relation $\phi_E[G, j, v, X^k; q_1, q_2]$ is the edge relation of the canon of $G[D_{j,v}]$ for suitable triples (j, v, X^k). Further, $\phi_{n_w}[G, j, v; q_1, q_2] = \{|D_{j,v}|, |D_{j,v}|\}$. Thus, the relation $\phi_{\mathrm{rep}}[G, j, v, X^k; q_1, q_2]$, where $\phi_{\mathrm{rep}} := \phi_E \vee \phi_{n_w}$, is the representation of the canon of $G[D_{j,v}]$ for all vertices $v \in V(G)$, and it follows that $X_j^{k+1} = S_j^{k+1}$. □

Proof of Theorem 50. As a direct consequence of Lemma 53 we obtain that L-formula ϕ_K defines the edge relation of the canon $f(G)$ for all $G \in \mathcal{C}$. Therefore, we conclude that there exists an L-canonization for the class $\mathcal{C}$ of graphs. □

5. Capturing PTIME on Permutation Graphs

In this chapter we use the Modular Decomposition Theorem to prove that fixed-point logic with counting captures polynomial time on the class of permutation graphs.

We begin this chapter with an introduction to permutation graphs, where we present different properties, characterizations and applications of permutation graphs. (A detailed introduction can also be found in [24].) Afterwards we show that there exists an FP+C-canonization of the class of permutation graphs by applying the Modular Decomposition Theorem. As a result, fixed-point logic with counting captures polynomial time on permutation graphs.

5.1. Permutation Graphs

In this section we consider a class of graphs with many interesting properties and characterizations. Let us take two parallel lines, and draw straight line segments between these parallel lines. Now we associate each line segment with a vertex, and let two vertices be adjacent if and only if their corresponding line segments intersect. This way we obtain a class of intersection graphs. The graphs of this class are called permutation graphs.

Permutation graphs arise, for example, in a problem of memory allocation in system programming [19]. Further, they occur in circuit design as an abstract representation of a special case of wire routing, known as two-terminal channel routing (see [19], and [59] for a survey on VLSI design). Given a set of nets, the goal in wire routing is to (only) connect terminals that are contained in the same net. In two-terminal channel routing the routing area is a rectangular grid, all nets consist of only two terminals and the terminals are placed on the lower and upper boundary of the grid. For an arbitrary fine grid, the intersection graph of the connecting routes of the two-terminal nets is a permutation graph. Applications similar to the mentioned ones can also be found in [24].

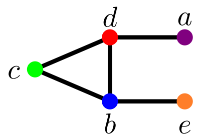
Definition

Let $G = (V, E)$ be a graph, and let $<_1$ and $<_2$ be two strict linear orders on the vertex set V. We call $(<_1, <_2)$ a *realizer* of G if two vertices u, v are adjacent in G if and only if they occur in different order in $<_1$ and $<_2$, that is,

- $u <_1 v$ and $v <_2 u$, or
- $v <_1 u$ and $u <_2 v$.

A graph G is a *permutation graph* if there exists a realizer of G. In the following we present an example of a permutation graph and its realizer.

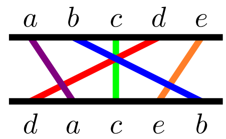
Example 54. Figure 5.1a and b show a graph H and a realizer $(<_1, <_2)$ for H. Let us verify that $(<_1, <_2)$ is indeed a realizer of H: Vertex a is adjacent to d but to no further vertex of graph H. Thus, vertices a and d have to occur in different order in $<_1$ and $<_2$, and the order of a and (each of) b, c, e has to be the same in $<_1$ and $<_2$. In strict linear order $<_1$ vertex a is the first vertex, and in strict linear order $<_2$ vertex d is smaller than a, and b, c and e are larger than a. Consequently, the realizer is correct for all pairs of vertices involving a. Next let us consider vertex e, the last vertex of $<_1$. For e we analogously observe that b, the only vertex adjacent to e, is larger than e with respect to $<_2$, and that the remaining vertices are smaller than e regarding $<_2$. It remains to consider vertices b, c and d. These three vertices form a clique, and we can verify in Figure 5.1b that their occurrence in $<_1$ is in reverse order to their occurrence in $<_2$. Thus, $(<_1, <_2)$ is in fact a realizer of permutation graph H.



(a) Graph H

$a <_1 b <_1 c <_1 d <_1 e$
$d <_2 a <_2 c <_2 e <_2 b$

(b) Realizer $(<_1, <_2)$ of H



(c) Diagram of $(<_1, <_2)$

Figure 5.1.: A (prime) permutation graph and its realizer

⌟

The following example illustrates the connection of permutation graphs with mathematical permutations.

Example 55. Let $<_{[n]}$ be the natural strict linear order on $[n]$. For each permutation π of the numbers from 1 to n, we let $<_\pi$ be the strict linear order defined by permutation π. Then $(<_{[n]}, <_\pi)$ is a realizer of a graph G_π with vertex set $V = [n]$. We call the graph G_π the *inversion graph* of π. Thus, each permutation π defines a permutation graph: the inversion graph G_π. Conversely, suppose we have given a realizer $(<_1, <_2)$ of a permutation graph G. Let us assign each vertex of graph G to the position of its occurrence in $<_1$, and let $h\colon V \to [|V|]$ be the corresponding mapping. If we rename the vertices of G according to h, we obtain a graph G' which is isomorphic to G and which is the inversion graph of a permutation. Thus, each permutation graph also is isomorphic to an inversion graph of a permutation. ⌟

Properties and Characterizations

As already mentioned in the introduction of this section, permutation graphs are intersection graphs. In order to see this, let $(<_1, <_2)$ be a realizer of the graph, and let us take two horizontal parallel lines. Now, above the first parallel line we write the vertices of the graph ordered by strict linear order $<_1$, and below the second parallel line we write them ordered by $<_2$. Now we draw straight line segments between the two parallel lines. Each line segment connects the two occurrences of one vertex, that is, the vertex on the upper line and the same vertex on the lower line are the two endpoints of a line segment. We call this the *diagram* of realizer $(<_1, <_2)$. Figure 5.1c shows the diagram of the realizer in Figure 5.1b. It is not hard to see that the corresponding line segments for two vertices intersect if and only if the two vertices occur in different order in $<_1$ and $<_2$. Therefore, we obtain that two line segments intersect if and only if the corresponding vertices are adjacent.

If we allow a circle instead of parallel lines in a diagram, then the intersection graph of the line segments, which are now chords of the circle, is called a circle graph. This graph class properly contains the class of permutation graphs. More precisely, permutation graphs are circle graphs that admit an equator [24, p. 252].

Let G be a permutation graph with realizer $(<_1, <_2)$. Each pair of strict linear orders that we obtain by exchanging or reversing both, $<_1$ and $<_2$, is a realizer of G as well. More precisely, $(<_2, <_1)$, $(<_1^R, <_2^R)$ and $(<_2^R, <_1^R)$ are further realizers of G, where $<^R$ denotes the reverse order of strict linear order $<$. If we reverse only, for example, $<_2$, and keep $<_1$ unchanged, then each pair of vertices that originally occurs in the same order now occurs inverted, and vice versa. Hence, the pair $(<_1, <_2^R)$ is a realizer of the complement graph of G. This shows that the complement of a permutation graph also is a permutation graph.

Another property of permutation graphs is that their edge relation can be oriented in such a way that the corresponding binary relation is transitive. In other words, a permutation graph is a comparability graph (defined in Section 6.1). To show this, let us direct each edge towards the vertex that is larger with respect to $<_1$. Then the resulting binary relation E' is transitive:

Observation 56. *Let $G = (V, E)$ be a permutation graph with realizer $(<_1, <_2)$. Then $E' := \{(v, w) \mid \{v, w\} \in E$ and $v <_1 w\}$ is transitive.*

Proof. Let $(u, v) \in E'$ and $(v, w) \in E'$. Then by definition of E' we know that $u <_1 v$ and $v <_1 w$, and that there are edges between u and v, and v and w. We obtain $v <_2 u$ and $w <_2 v$. Now transitivity of $<_1$ and $<_2$ implies that $u <_1 w$ and $w <_2 u$. Thus, there also is an edge between u and w. Since $u <_1 w$, this edge is directed towards w, and $(u, w) \in E'$. □

So permutation graphs are comparability graphs. As the complement of a permutation graph is a permutation graph, it is a comparability graph as well. Pnueli, Lempel and Even showed in [61] that also the other direction holds, that is, a graph G is a permutation graph if, and only if, the graph G and its complement are comparability graphs.

In the following section we want to apply the Modular Decomposition Theorem to permutation graphs. As it only holds for graph classes that are closed under induced subgraphs, we make sure that this is the case for permutation graphs.

Observation 57. *The class of permutation graphs is closed under induced subgraphs*

Proof. Let H be an induced subgraph of G. If the pair $(<_1, <_2)$ of strict linear orders is a realizer of G, then clearly we obtain a realizer of H by restricting the strict linear orders $<_1$ and $<_2$ to the vertices of H. Therefore, every induced subgraph of a permutation graph is a permutation graph as well. □

5.2. Capturing Result

Now we want to show that fixed-point logic with counting captures polynomial time on permutation graphs. In order to do this we first consider prime permutation graphs. It is known that the realizer of a prime permutation graph is unique up to reversal and exchange [21]. Thus, a prime permutation graph has at most 4 different realizers. We show that these realizers are definable in fixed-point logic. We use the strict linear orders of the realizers to construct a linear order on the universe of prime LO-colored permutation graphs. This way, we obtain an FP-canonization of the class of prime LO-colored permutation graphs. Then we can apply the Modular Decomposition Theorem, which yields that the class of all permutation graphs admits FP+C-definable canonization. As a consequence, we obtain the capturing result.

Defining the Realizers

First we prove that the realizers of prime permutation graphs are definable in fixed-point logic with counting. We start with defining certain properties of pairs of relations. These properties enable us to construct the realizers recursively.

Let $\lhd_1$ and $\lhd_2$ be two binary relations. We call the pair $(\lhd_1, \lhd_2)$ *transitive* if each of the binary relations $\lhd_1$ and $\lhd_2$ is transitive. Further, we let the transitive closure $(\lhd_1, \lhd_2)^T$ of $(\lhd_1, \lhd_2)$ be the pair $(\lhd_1^T, \lhd_2^T)$ where $\lhd_1^T$ and $\lhd_2^T$ is the transitive closure of $\lhd_1$ and $\lhd_2$, respectively. Let $G = (V, E)$ be a graph and $(\lhd_1, \lhd_2)$ be a pair of binary relations on V. The pair $(\lhd_1, \lhd_2)$ is *closed under edge relation E* if for all vertices $u, v \in V$ and all $i \in [2]$ the following holds:

- If $u \lhd_i v$ and $\{u, v\} \in E$, then $v \lhd_{3-i} u$.
- If $u \lhd_i v$ and $\{u, v\} \notin E$, then $u \lhd_{3-i} v$.

Notice that for a permutation graph $G = (V, E)$, each realizer of G is closed under edge relation E. Moreover, we observe the following.

Observation 58. *Let $G = (V, E)$ be a permutation graph. Then a pair of binary relations $(\lhd_1, \lhd_2)$ is a realizer of G if, and only if, $\lhd_1$ and $\lhd_2$ are strict linear orders and $(\lhd_1, \lhd_2)$ is closed under edge relation E.*

Now for all $i \in [2]$ we let

$$D_{3-i}^{E} := \{(v, u) \mid u \lhd_i v \text{ and } \{u, v\} \in E\} \text{ and}$$
$$D_{3-i}^{\not E} := \{(u, v) \mid u \lhd_i v \text{ and } \{u, v\} \notin E\},$$

and we let $(\lhd_1, \lhd_2)^E$ be the pair $(\lhd_1^E, \lhd_2^E)$ of relations where for all $i \in [2]$ we have

$$\lhd_i^E = \lhd_i \cup D_i^E \cup D_i^{\not E}.$$

Let $G = (V, E)$ be a permutation graph, and let R be a set of pairs of binary relations on vertex set V. We call a pair $(\lhd_1, \lhd_2) \in R$ the *minimum* of R if we have $\lhd_1 \subseteq \lhd_1'$ and $\lhd_2 \subseteq \lhd_2'$ for all pairs $(\lhd_1', \lhd_2') \in R$. There cannot exist more than one minimum of R, as the respective relations of two minimums would be contained in each other, and therefore be equal. For binary relations $\lhd_1$ and $\lhd_2$ on the vertex set V of permutation graph $G = (V, E)$, we let $R^E(\lhd_1, \lhd_2)$ be the set of all pairs $(\lhd'_1, \lhd'_2)$ of binary relations on V where $\lhd_1 \subseteq \lhd'_1$, $\lhd_2 \subseteq \lhd'_2$ and $(\lhd'_1, \lhd'_2)$ is closed under edge relation E. In Lemma 59 we show that $R^E(\lhd_1, \lhd_2)$ always has a minimum, which is $(\lhd_1, \lhd_2)^E$. We call the minimum $(\lhd_1, \lhd_2)^E$ of $R^E(\lhd_1, \lhd_2)$ the *closure* of $(\lhd_1, \lhd_2)$ *under edge relation* E.

Lemma 59. *For each permutation graph $G = (V, E)$, the pair $(\lhd_1, \lhd_2)^E$ is the minimum of $R^E(\lhd_1, \lhd_2)$.*

Proof. First we show that $(\lhd_1, \lhd_2)^E \in R^E(\lhd_1, \lhd_2)$. We know that $\lhd_1 \subseteq \lhd_1^E$ and $\lhd_2 \subseteq \lhd_2^E$. Hence, it remains to prove that $(\lhd_1, \lhd_2)^E$ is closed under edge relation E. In order to do this, let us consider arbitrary $u, v \in V$ and $i \in [2]$ with $u \lhd_i^E v$. Without loss of generality, let u and v be adjacent. Thus, we have to show that $v \lhd_{3-i}^E u$. Since $u \lhd_i^E v$, one of the following cases is satisfied: $u \lhd_i v$, $(u, v) \in D_i^E$ or $(u, v) \in D_i^{\not E}$; where $(u, v) \in D_i^{\not E}$ can be excluded as u and v are adjacent. Now if $u \lhd_i v$ is the case, then $(v, u) \in D_{3-i}^E$, and we obtain that $v \lhd_{3-i}^E u$. If $(u, v) \in D_i^E$, then $v \lhd_{3-i} u$, and we also have $v \lhd_{3-i}^E u$. Therefore, we obtain $v \lhd_{3-i}^E u$ in both cases. As a consequence, $(\lhd_1, \lhd_2)^E$ is closed under edge relation E.

In order to prove that $(\lhd_1, \lhd_2)^E$ is the minimum of $R^E(\lhd_1, \lhd_2)$, we need to show that $\lhd_1^E \subseteq \lhd'_1$ and $\lhd_2^E \subseteq \lhd'_2$ for all $(\lhd'_1, \lhd'_2) \in R^E(\lhd_1, \lhd_2)$. Let $(\lhd'_1, \lhd'_2)$ be a pair in relation $R^E(\lhd_1, \lhd_2)$. Then we have $\lhd_i \subseteq \lhd'_i$ for all $i \in [2]$. Further, the pair $(\lhd'_1, \lhd'_2)$ is closed under edge relation E. Thus, relation $\lhd'_i$ contains the pairs in $D_i^E \cup D_i^{\not E}$ for $i \in [2]$. It follows that $\lhd_i^E$ is a subset of $\lhd'_i$, which proves that $(\lhd_1, \lhd_2)^E$ is the minimum of $R^E(\lhd_1, \lhd_2)$. □

Let $G = (V, E)$ be a prime permutation graph. For each $w \in V$ we define two binary relations $\lhd_1^w$ and $\lhd_2^w$ on the vertex set V. We call w the *initial vertex* of $\lhd_1^w$ and $\lhd_2^w$. If there exists a realizer $(<_1, <_2)$ of G where w is the first vertex of the first strict linear order $<_1$, then it will turn out that $(\lhd_1^w, \lhd_2^w) = (<_1, <_2)$.

In order to construct the binary relations $\lhd_1^w$ and $\lhd_2^w$, we recursively define relations $\lhd_{1,k}^w$ and $\lhd_{2,k}^w$ on the vertex set V for all $k \geq 0$. To increase readability we often separately indicate the vertex $w \in V$ that the relations are referring to, and omit w in the notation of $\lhd_1^w$ and $\lhd_2^w$, or $\lhd_{1,k}^w$ and $\lhd_{2,k}^w$. So let us fix an initial vertex $w \in V$. We begin with defining the relations for $k = 0$. As w is the first element of the first strict linear order of the realizer that we want to reconstruct, we let

$$\begin{aligned} \lhd_{1,0} &:= \{(w, v) \mid v \in V, v \neq w\} \quad \text{and} \\ \lhd_{2,0} &:= \emptyset. \end{aligned}$$

Thus, we have $a \lhd_{1,0} b$ if, and only if, a is the initial vertex w and b is a vertex distinct from w. Further, there do not exist vertices a and b such that $a \lhd_{2,0} b$.

Now, we recursively define $\lhd_{1,k+1}$ and $\lhd_{2,k+1}$ for all $k > 0$ as follows:

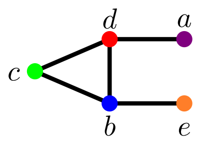
$$(\lhd_{1,k+1}, \lhd_{2,k+1}) := ((\lhd_{1,k}, \lhd_{2,k})^E)^T.$$

Clearly, for all initial vertices w and all $k \geq 0$ the relations satisfy the property that

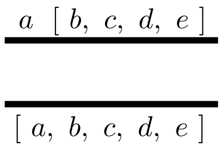
$$\lhd_{1,k} \subseteq \lhd_{1,k+1} \quad \text{and}$$
$$\lhd_{2,k} \subseteq \lhd_{2,k+1} \ .$$

Since the vertex set is finite, there exists an m such that $\lhd_{i,m} = \lhd_{i,m+1}$ for all $i \in [2]$. Let m be minimal with that property. We define $\lhd_i := \lhd_{i,m}$ for $i \in [2]$. Example 60 illustrates the construction of $(\lhd_1, \lhd_2)$.

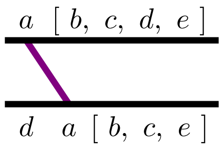
Example 60. Let us consider the prime permutation graph $H = (V, E)$ depicted in Figure 5.2a. We let $a \in V$ be the initial vertex. Figure 5.2b shows an illustration of $(\lhd_{1,0}, \lhd_{2,0})$. We use square brackets to mark incomparable vertices[1]. Now let us determine $(\lhd_{1,1}, \lhd_{2,1})$. We know that $a \lhd_{1,0} v$ for all $v \neq a$. Further, a and d are adjacent but there is no edge between a and the remaining vertices. Thus, for the closure of $(\lhd_{1,0}, \lhd_{2,0})$ under edge relation E we have $d \lhd^E_{2,0} a$ and $a \lhd^E_{2,0} v$ for $v \in \{b, c, e\}$. Now we take the transitive closure and additionally obtain that d is smaller than b, c and e with respect to $(\lhd^E_{2,0})^T = \lhd_{2,1}$. The resulting relations $\lhd_{1,1}$ and $\lhd_{2,1}$ are depicted in Figure 5.2c. For $k \in \{2, 3, 4\}$ the pairs of relations $(\lhd_{1,k}, \lhd_{2,k})$ are shown in Figure 5.2d-f. No new pairs of vertices are added to any of the relations for $k > 4$. ⌟



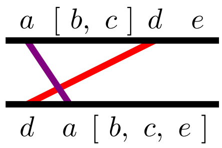
(a) Graph H



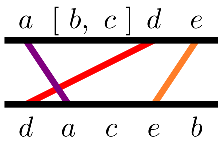
(b) $(\lhd_{1,0}, \lhd_{2,0})$



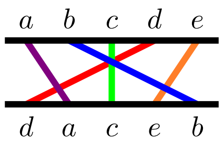
(c) $(\lhd_{1,1}, \lhd_{2,1})$



(d) $(\lhd_{1,2}, \lhd_{2,2})$



(e) $(\lhd_{1,3}, \lhd_{2,3})$



(f) $(\lhd_{1,4}, \lhd_{2,4})$

Figure 5.2.: Construction of $(\lhd_1, \lhd_2)$

In the following we let $(<_1, <_2)$ be a realizer of permutation graph G, and we let w be the first element of $<_1$. We show that the pair of relations $\lhd^w_1$ and $\lhd^w_2$ is the realizer $(<_1, <_2)$. We fix w as initial vertex and let $\lhd_1 := \lhd^w_1$ and $\lhd_2 := \lhd^w_2$. By definition of $(\lhd_{1,0}, \lhd_{2,0})$ we have $\lhd_{1,0} \subseteq <_1$ and $\lhd_{2,0} \subseteq <_2$. Further, we obtain $(\lhd_1, \lhd_2)$ from $(\lhd_{1,0}, \lhd_{2,0})$ by recursively taking the closure under edge relation E and the transitive closure. Since the realizer $(<_1, <_2)$ is closed under both, the following observation holds.

[1] For all $k \geq 0$ the relations $\lhd_{1,k}$ and $\lhd_{2,k}$ are strict weak orders (Lemma 62). This makes it possible to arrange the equivalence classes expressing incomparability into a strict linear order, and enables us to visualize the relations like this.

Observation 61. *For all $k \geq 0$, we have $\lhd_{1,k} \subseteq <_1$ and $\lhd_{2,k} \subseteq <_2$, and thus, $\lhd_1 \subseteq <_1$ and $\lhd_2 \subseteq <_2$.*

We use the following lemma to prove in Theorem 64 that $\lhd_1$ and $\lhd_2$ are strict linear orders. Then it follows with Observation 61 that $\lhd_1 = <_1$ and $\lhd_2 = <_2$.

Lemma 62. *Relations $\lhd_{1,k}$ and $\lhd_{2,k}$ are strict weak orders for all $k \geq 0$.*

Proof. In order to show that a relation is a strict weak order, we have to prove that it is a strict partial order and that incomparability is transitive. Let $k \geq 0$. As $<_1$ and $<_2$ are irreflexive, it follows from $\lhd_{1,k} \subseteq <_1$ and $\lhd_{2,k} \subseteq <_2$ that $\lhd_{1,k}$ and $\lhd_{2,k}$ are irreflexive as well. Further, relations $\lhd_{1,k}$ and $\lhd_{2,k}$ are transitive. Hence, $\lhd_{1,k}$ and $\lhd_{2,k}$ are strict partial orders. It remains to show that incomparability is transitive. We denote two vertices x and y that are incomparable with respect to $\lhd_{i,k}$ by $x \sim_{i,k} y$. Let us consider $k = 0$. With respect to $\lhd_{1,0}$, all elements in $V \setminus \{w\}$ are pairwise incomparable and w is incomparable to itself. Further, all elements in V are pairwise incomparable with respect to $\lhd_{2,0}$. Thus, for $\lhd_{1,0}$ and $\lhd_{2,0}$ incomparability is transitive. To show that incomparability is transive for $k > 0$ we need the following claims.

Claim 1. Let $\kappa \geq 0$, $i \in [2]$ and $x, y \in V$. If x and y are incomparable with respect to $\lhd_{i,\kappa+1}$, then x and y are incomparable with respect to $\lhd_{1,\kappa}$ and $\lhd_{2,\kappa}$.

Proof. Without loss of generality let $i = 1$, and let x and y be incomparable with respect to $\lhd_{1,\kappa+1}$. For a contradiction let us assume that x and y are comparable with respect to $\lhd_{1,\kappa}$ or $\lhd_{2,\kappa}$. If x and y are comparable with respect to $\lhd_{1,\kappa}$, then this directly implies that x and y are comparable with respect to $\lhd_{1,\kappa+1}$, since $\lhd_{1,\kappa} \subseteq \lhd_{1,\kappa+1}$. Thus, let us suppose x and y are comparable with respect to $\lhd_{2,\kappa}$. Then x and y are also comparable with respect to $\lhd^E_{1,\kappa}$, and therefore also with respect to $(\lhd^E_{1,\kappa})^T = \lhd_{1,\kappa+1}$, a contradiction. ⌟

Claim 2. Let $\kappa \geq 0$, $i \in [2]$ and $y, z \in V$. Further, let $\lhd_{1,\kappa}$ and $\lhd_{2,\kappa}$ be strict weak orders, and let y and z be incomparable with respect to $\lhd_{i,\kappa+1}$. Then for all vertices $v \in V$ the following holds: If $v \lhd^E_{i,\kappa} z$, then $v \lhd^E_{i,\kappa} y$.

Proof. Without loss of generality let $i = 1$. Relation $\lhd^E_{1,\kappa}$ contains only pairs that are in $\lhd_{1,\kappa}$, in $D^E_{1,\kappa}$ or in $D^{\not E}_{1,\kappa}$. Therefore, $v \lhd^E_{1,\kappa} z$ implies that either $v \lhd_{1,\kappa} z$, $z \lhd_{2,\kappa} v$ or $v \lhd_{2,\kappa} z$. If we have $v \lhd_{1,\kappa} z$, then we also have $v \lhd_{1,\kappa} y$, as y and z are incomparable with respect to $\lhd_{1,\kappa}$ by Claim 1 and $\lhd_{1,\kappa}$ is a strict weak order. Analogously, $z \lhd_{2,\kappa} v$ and $v \lhd_{2,\kappa} z$ imply $y \lhd_{2,\kappa} v$ and $v \lhd_{2,\kappa} y$, respectively. Hence, in each of the cases we obtain $v \lhd^E_{1,\kappa} y$. ⌟

Now, let us assume there exists a $k > 0$ such that incomparability is not transitive for $\lhd_{1,k}$ or $\lhd_{2,k}$, and let k be minimal. Without loss of generality, let incomparability be not transitive for $\lhd_{1,k}$. Consequently, there exist vertices $x, y, z \in V$ such that $x \sim_{1,k} y$, $y \sim_{1,k} z$ and $x \not\sim_{1,k} z$. Hence, x and z are comparable, which means $x \lhd_{1,k} z$ or $z \lhd_{1,k} x$. Without loss of generality, let $x \lhd_{1,k} z$. Since $\lhd_{1,k}$ is the transitive closure of $\lhd^E_{1,k-1}$, there exists an $l \geq 0$ and $v_0, v_1, \ldots, v_{l+1}$ such that

$$x = v_0 \lhd^E_{1,k-1} \cdots \lhd^E_{1,k-1} v_l \lhd^E_{1,k-1} v_{l+1} = z.$$

As we know that $\lhd_{1,k-1}$ and $\lhd_{2,k-1}$ are strict weak orders, that $y \sim_{1,k} z$, and that $v_l \lhd^E_{1,k-1} z$, we can apply Claim 2, and obtain $v_l \lhd^E_{1,k-1} y$. Consequently, we have

$$x = v_0 \lhd^E_{1,k-1} \cdots \lhd^E_{1,k-1} v_l \lhd^E_{1,k-1} y,$$

and therefore, $x \lhd_{1,k} y$, a contradiction. □

Corollary 63. *Relations $\lhd_1$ and $\lhd_2$ are strict weak orders.*

Theorem 64. *Relations $\lhd_1$ and $\lhd_2$ are strict linear orders.*

Proof. Let us assume that $\lhd_1$ is not a strict linear order. Since $\lhd_1$ is a strict weak order by Corollary 63, there must exist two distinct vertices u, v such that $u \sim_1 v$, i.e. u and v are incomparable regarding $\lhd_1$. Hence, the equivalence class $u/_{\sim_1}$ contains at least two elements. In the following we prove that $u/_{\sim_1}$ is a module. Let us assume $u/_{\sim_1}$ is not a module. Then there exists a vertex $z \notin u/_{\sim_1}$ and vertices $x, y \in u/_{\sim_1}$ such that z and x are adjacent and z and y are not adjacent. As $\lhd_1$ is a strict weak order, we either have $z \lhd_1 x$ and $z \lhd_1 y$, or $x \lhd_1 z$ and $y \lhd_1 z$. Let us assume $z \lhd_1 x$ and $z \lhd_1 y$. The other case can be shown analogously. Since there is an edge between z and x and no edge between z and y, and $(\lhd_1, \lhd_2)$ is closed under edge relation E, we have $x \lhd_2 z$ and $z \lhd_2 y$. Therefore, we must also have $x \lhd_2 y$, by transitivity of $\lhd_2$. Then we use again that $(\lhd_1, \lhd_2)$ is closed under E, and obtain that $x \lhd_1 y$ or $y \lhd_1 x$. Hence, x and y are comparable with respect to $\lhd_1$, a contradiction. Consequently, $u/_{\sim_1}$ is a module with $|u/_{\sim_1}| \geq 2$. Clearly, $u/_{\sim_1}$ cannot be the vertex set V since we know $w \lhd_1 v$ for all $v \neq w$, where w is the initial vertex. Thus, $u/_{\sim_1}$ is a non-trivial module, a contradiction to G being prime.

Similarly we can prove that $\lhd_2$ is a strict linear order. To show that a module $u/_{\sim_2}$ with $|u/_{\sim_2}| \geq 2$ for $u \in V$ cannot be the vertex set V, we argue as follows: Since $w \lhd_1 v$ for all $v \in V$ with $v \neq w$ and $(\lhd_1, \lhd_2)$ is closed under E, vertex w is comparable to all $v \neq w$ with respect to $\lhd_2$. □

Corollary 65. *We have $\lhd_1 = <_1$ and $\lhd_2 = <_2$.*

Now we know that the pair $(\lhd^w_1, \lhd^w_2)$ of relations is the realizer $(<_1, <_2)$ if w is the first vertex of $<_1$. We use this to show that the realizers of prime permutation graphs are definable in fixed-point logic.

First of all, there are FP-formulas $\varphi_{\lhd_1}(x, y, y')$ and $\varphi_{\lhd_2}(x, y, y')$ such that for all prime permutation graphs $G = (V, E)$ and all $w, v, v' \in V$ we have

$$G \models \varphi_{\lhd_i}[w, v, v'] \iff v \lhd^w_i v'.$$

In order to define $\varphi_{\lhd_i}$ we use a simultaneous inflationary fixed-point operator. Within this fixed-point operator we need two binary relational variables X_1 and X_2 to create the strict linear orders $\lhd^w_1$ and $\lhd^w_2$. Let X^k_1 and X^k_2 be the relations that we get at the kth iteration of the simultaneous fixed-point operator. We can design the simultaneous fixed-point operator such that X^k_1 and X^k_2 are precisely $\lhd^w_{1,k}$ and $\lhd^w_{2,k}$. It is not hard to see that this is possible since the transitive closure and the closure under the edge relation are definable in transitive closure logic and first order logic, respectively.

We use formulas $\varphi_{\lhd_1}$ and $\varphi_{\lhd_2}$ to define a formula $\chi(x)$ where for prime permutation graphs $G = (V, E)$ and $w \in V$ we have

$$G \models \chi[w] \iff (\lhd_1^w, \lhd_2^w) \text{ is a realizer of } G.$$

We already know that $(\lhd_1^w, \lhd_2^w)$ is closed under edge relation E by Lemma 59. Therefore, we only have to check whether $\lhd_1^w$ and $\lhd_2^w$ are strict linear orders to find out if $(\lhd_1^w, \lhd_2^w)$ is a realizer of G. Since irreflexivity, transitivity and antisymmetry of relations can be tested in first order logic, formula χ is FP-definable.

Formulas $\varphi_{\lhd_1}$, $\varphi_{\lhd_2}$ and χ enable us to define the realizers of prime permutation graphs in fixed-point logic.

Applying the Modular Decomposition Theorem

Let $\mathcal{C}_{\text{perm}}$ be the class of permutation graphs. Then $\mathcal{C}^*_{\text{perm}}$ is the class of all prime LO-colored permutation graphs with at least 4 vertices. In the following we describe the construction of a parameterized FP-canonization of $\mathcal{C}^*_{\text{perm}}$. Afterwards we apply the Modular Decomposition Theorem to show that fixed-point logic with counting captures polynomial time on $\mathcal{C}_{\text{perm}}$.

Since we can define the realizers of prime permutation graphs, it is also possible to define the realizers of the underlying graphs of prime LO-colored permutation graphs. We simply pull back the formulas $\varphi_{\lhd_1}$, $\varphi_{\lhd_2}$ and χ under $(\{V, E, M, \unlhd, L\}, \{E\})$-transduction $\Theta_{\text{graph}} = (V(x), E(x, x'))$, which maps every LO-colored graph to (an isomorphic copy of) its underlying graph. In order to actually construct the canonization of $\mathcal{C}^*_{\text{perm}}$, we only need the pull-backs of formula $\chi(x)$ and of formula $\varphi_{\unlhd_1}(x, y, y') := \varphi_{\lhd_1}(x, y, y') \vee y = y'$, which defines the linear order $\unlhd_1^w$ associated with the strict linear order $\lhd_1^w$ on the vertex set.

If $(\lhd_1^w, \lhd_2^w)$ is a realizer of the underlying graph of $G^* \in \mathcal{C}^*_{\text{perm}}$, then $\lhd_1^w$ is a strict linear order on the vertex set of G^*. We can use this strict linear order on the vertex set to construct a linear order on the universe of the LO-colored graph G^*. We simply compose the linear version $\unlhd_1^w$ of the strict linear order $\lhd_1^w$ and the linear order $\unlhd$ on the basic color elements $M(G^*)$. More precisely,

$$\leq^w := \unlhd_1^w \cup \unlhd \cup \{(v, m) \mid v \in V(G^*),\, m \in M(G^*)\} \tag{5.1}$$

is a linear order on the universe of LO-colored graph G^*.

We now define a parameterized FP-canonization $\Theta(x)$, which maps each prime LO-colored permutation graph $G^* \in \mathcal{C}^*_{\text{perm}}$ to the ordered copy $(G^*, \leq^w)$. Valid parameters of this transduction are vertices $w \in V$ that are the first vertex of the first strict linear order of a realizer. We let $\Theta(x) = (\theta_{\mathrm{dom}}, \theta_U, \theta_V, \theta_E, \theta_M, \theta_{\trianglelefteq}, \theta_L, \theta_{\leq})$, where

$$\begin{aligned}
\theta_{\mathrm{dom}}(x) &:= \chi^{-\Theta_{\mathrm{graph}}}(x), \\
\theta_U(x, y) &:= \top, \\
\theta_V(x, y) &:= V(y), \\
\theta_E(x, y, y') &:= E(y, y'), \\
\theta_M(x, y) &:= M(y), \\
\theta_{\trianglelefteq}(x, y, y') &:= \trianglelefteq(y, y'), \\
\theta_L(x, y, y', y'') &:= L(y, y', y'') \quad \text{and} \\
\theta_{\leq}(x, y, y') &:= \varphi_{\trianglelefteq_1}^{-\Theta_{\mathrm{graph}}}(x, y, y') \vee \trianglelefteq(y, y') \vee \big(V(y) \wedge M(y')\big),
\end{aligned}$$

Formula θ_{dom}, that is, the pull-back of formula χ, defines the valid parameters, and formula $\theta_{\leq}$ defines the linear order $\leq^w$ from (5.1) by using the pull-back of formula $\varphi_{\trianglelefteq_1}$.

Now that we have proved that there exists a parameterized FP-canonization of the class $\mathcal{C}^*_{\mathrm{perm}}$ of prime LO-colored permutation graphs with at least 4 vertices, and we know that the class $\mathcal{C}_{\mathrm{perm}}$ of permutation graphs is closed under induced subgraphs (Observation 57), we can apply the Modular Decomposition Theorem. We obtain that the class of permutation graphs admits FP+C-definable canonization. As a consequence, FP+C captures polynomial time on permutation graphs.

6. Capturing PTIME on Chordal Comparability Graphs

We use the Modular Decomposition Theorem to show that fixed-point logic with counting captures polynomial time on the class of chordal comparability graphs in this chapter.

6.1. Chordal Comparability Graphs

A graph is called *chordal* if all of its induced cycles are of length 3. Thus, each cycle of length at least 4 has a *chord*, which is an edge that connects two non-consecutive vertices of the cycle. Alternately, chordal graphs can be characterized by the property that its maximal cliques can be arranged in a tree T, so that for every vertex of the graph the set of max cliques containing it is connected in T. The tree T is called a clique tree. Clique trees of chordal graphs are of fundamental use in Chapter 12 and are defined in Section 12.1. An elementary introduction to multiple characterizations of chordal graphs (and clique trees) can be found in [2].

A graph G is a *comparability graph* (also called transitively orientable graph, partially orderable graph or containment graph [4]) if there exists a strict partial order for G. A *strict partial order for* a graph $G = (V, E)$ is a strict partial order $\prec$ (irreflexive, transitive) on G's vertex set so that $\{u, v\} \in E$ if and only if $u, v \in V$ are comparable with respect to $\prec$. It follows that G is a comparability graph if and only if its edges can be oriented in such a way that the corresponding binary relation is transitive. Given a comparability graph G, it is possible to transitively orient the edges, that is, to find a strict partial order for G, in linear time [56].

Every strict partial order is the intersection of a set of strict linear orders [16]. A strict partial order $\prec$ has dimension k if there exist k strict linear orders whose intersection is $\prec$. It is not hard to see that a graph G is a permutation graph (see Section 5.1) if and only if there is a strict partial order for G of dimension at most 2.

Chordal comparability graphs have been investigated, e. g., in [54, 38, 12]. They can be recognized in linear time [38, 56]. For all chordal comparability graphs G there exists a strict partial order for G of dimension at most 4 [54, 65]. This bound is tight [45], but as there are permutation graphs that are not chordal, e. g., a cycle of length 4, a strict partial order for a graph G of dimension at most 4 does not imply that the graph G is a chordal comparability graph. For chordal comparability graphs, these four strict linear orders can be found in linear time [12].

6.2. Modular Decomposition Theorem - Application

Let $\mathcal{C}_{\mathrm{ChCo}}$ be the class of all chordal comparability graphs. It is not hard to see, that the class of chordal graphs and the class of comparability graphs are closed under induced subgraphs. Therefore, the class of chordal comparability graphs is closed under induced subgraphs, and we can apply the Modular Decomposition Theorem (Theorem 50) to obtain a capturing result for chordal comparability graphs.

The Modular Decomposition Theorem states that in order to prove that FP+C captures polynomial time on chordal comparability graphs, it suffices to show that there is an FP+C-definable canonization mapping for the class $\mathcal{C}^*_{\mathrm{ChCo}}$ of all LO-colored graphs $G^* = (U, V, E, M, \trianglelefteq, L)$ where the underlying graph $G := (V, E)$ is a prime chordal comparability graph with at least 4 vertices. The remainder of Chapter 6 will be devoted to the proof of the existence of such an FP+C-definable canonization mapping.

6.3. The Graph's Structure

In this section we consider the class $\mathcal{C}^{\mathrm{pr}}_{\mathrm{ChCo}}$ of prime chordal comparability graphs, that is, the class of underlying graphs of the LO-colored graphs that we need to canonize according to the Modular Decomposition Theorem. In the following, we introduce structural elements of prime chordal comparability graphs, for example, maximal cliques, ends and sides. In addition, we prove necessary properties of these structural elements and their logical definability.

Notice that to show the mentioned definability results we present $\{E\}$-formulas for the graphs in $\mathcal{C}^{\mathrm{pr}}_{\mathrm{ChCo}}$ although we will actually need $\{V, E, M, \trianglelefteq, L\}$-formulas for the LO-colored graphs in $\mathcal{C}^*_{\mathrm{ChCo}}$. Yet, it is not hard to find a transduction, that maps each LO-colored graph $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$ to its underlying graph. Hence, for the presented formulas in this section the Transduction Lemma guarantees us matching $\{V, E, M, \trianglelefteq, L\}$-formulas that refer to the underlying graph of the LO-colored graphs in $\mathcal{C}^*_{\mathrm{ChCo}}$.

We denote prime chordal comparability graphs by G, and chordal comparability graphs (that do not have to be prime) by H throughout this section.

6.3.1. Max Cliques

In the following we want to show that the max cliques of a chordal comparability graph are FO-definable. This result does not require a restriction on the number of vertices of the graph or the graph to be prime. Thus, let $H = (V, E)$ be a chordal comparability graph and let $\prec$ be an arbitrary strict partial order for H. Further, let $\mathcal{M}$ be the set of all max cliques of H.

Observation 66. *For every max clique $A \in \mathcal{M}$ of graph H, the restriction $\prec_{|A}$ of $\prec$ to A is a strict linear order.*

Proof. The binary relation $\prec_{|A}$ is irreflexive and transitive, and as A is a clique, relation $\prec_{|A}$ also is connex. Thus, $\prec_{|A}$ is a strict linear order on A. □

Let $A \in \mathcal{M}$ be a max clique with $|A| = m$. We also denote a max clique $A = \{a_1, \dots, a_m\}$ with $a_1 \prec \dots \prec a_m$ by $(a_1, \dots, a_m)_\prec$. We depict a max clique A as shown in Figure 6.1, where vertex a_i is drawn above vertex a_j if and only if $a_i \prec a_j$.

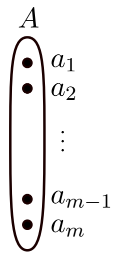


Figure 6.1.: Max clique A

Lemma 67. *Let A and B be max cliques of H and $v \in B$. If $\{b \in B \mid b \prec v\}$ is a subset of A, then there exists no vertex $a \in A \setminus B$ with $a \prec v$.*

Proof. Let $D := \{b \in B \mid b \prec v\}$ be a subset of A, and let us assume that there exists a vertex $a \in A \setminus B$ with $a \prec v$. We show that vertex a is adjacent to all vertices in $B = D \cup \{v\} \cup \{b \in B \mid v \prec b\}$ in graph H: Since $D \cup \{a\}$ is a subset of A, vertex a is adjacent to all vertices in D. Further, we have $a \prec v$, and for all $b \in B$ with $v \prec b$ transitivity implies that $a \prec b$. Consequently, there also exist edges between a and all vertices in $B \setminus D$. Hence, $B \cup \{a\}$ is a clique, which is a contradiction to B being a max clique. □

Lemma 68. *Let $A = (a_1, \ldots, a_m)_\prec, B = (b_1, \ldots, b_n)_\prec$ be intersecting max cliques of H, and $A \neq B$. Then there do not exist vertices $a_k, a_x, a_l \in A$ with $1 \leq k < x < l \leq m$ such that $a_x \in A \cap B$ and $a_k, a_l \in A \setminus B$.*

Proof. Now, let us assume there are $a_k, a_l \in A \setminus B$ and $a_x \in A \cap B$ with $1 \leq k < x < l \leq m$, and let $k > x$ be minimal and $l < x$ maximal such that $a_k, a_l \in A \setminus B$. Thus, for all i with $k < i < l$ we have $a_i \in A \cap B$. Let x' be the index such that $a_x = b_{x'}$. Further, let $k' < x'$ be the maximal index so that $b_{k'} \in B \setminus A$, and $l' > x'$ be the minimal index so that $b_{l'} \in B \setminus A$. A picture showing a_k, a_x, a_l and $b_{k'}, b_{x'}, b_{l'}$ can be found in Figure 6.2a. In order to show that such indices exist, assume there is no $k' < x'$ so that $b_{k'} \in B \setminus A$. Then, $\{b \in B \mid b \prec b_{x'}\}$ is a subset of A and a_k is a vertex in $A \setminus B$ with $a_k \prec a_x$, which is a contradiction to Lemma 67. Equivalently, we can show that l' exists.

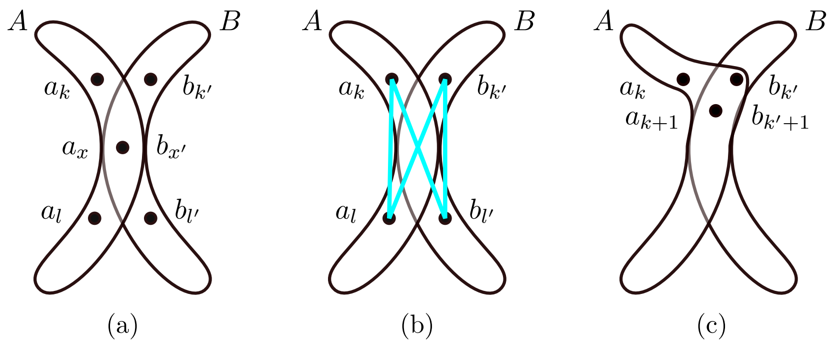


Figure 6.2.: Illustration for the proof of Lemma 68

We have $a_k \prec b_{l'}$ and $b_{k'} \prec a_l$, because $a_k \prec a_x$ and $b_{x'} \prec b_{l'}$, and $b_{k'} \prec b_{x'}$ and $a_x \prec a_l$. Hence, $\{a_k, b_{l'}\}, \{b_{k'}, a_l\}$ and, of course, $\{a_k, a_l\}, \{b_{k'}, b_{l'}\}$ are edges of the graph H. Thus, $a_k, a_l, b_{k'}, b_{l'}, a_k$ is a cycle of H as shown in Figure 6.2b. Since H is chordal, all of H's induced cycles are of length 3. Therefore, $\{a_k, b_{k'}\}$ or $\{a_l, b_{l'}\}$ is an edge in H. Without loss of generality, let us assume $\{a_k, b_{k'}\}$ is an edge of H. Then $a_k \prec b_{k'}$ or $b_{k'} \prec a_k$. If $a_k \prec b_{k'}$, we obtain by transitivity that $a_i \prec b_{k'}$ for all $i \leq k$, and $b_{k'} \prec a_i$ for all $i > l$, because $a_{k+1} = b_{k'+1}$. As illustrated in Figure 6.2c, the set $A \cup \{b_{k'}\}$ is a clique, which is a contradiction to the maximality of clique A. If $b_{k'} \prec a_k$, we obtain a contradiction by using the same argument. □

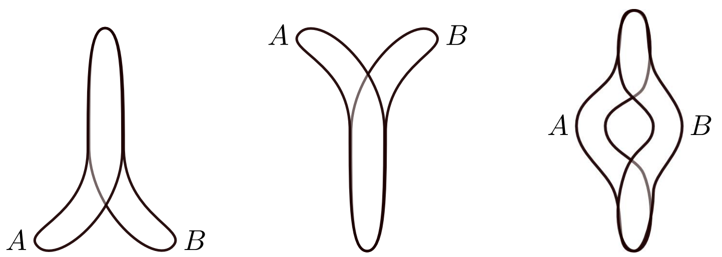
Lemma 69. *Let $A = (a_1, \ldots, a_m)_\prec, B = (b_1, \ldots, b_n)_\prec$ be intersecting max cliques, and $A \neq B$. If $a_x \in A \cap B$, then either $a_1 = b_1, \ldots, a_x = b_x$ or $a_x = b_{n-m+x}, \ldots, a_m = b_n$.*

Proof. Let $A \neq B$ and $a_x \in A \cap B$. Since A is a max clique different from B, there exists an $l \in [m]$ such that $a_l \in A \setminus B$. If $l > x$, then for all $k \in [1, x]$ we have $a_k \in A \cap B$ by Lemma 68. Let $x' \in [n]$ be such that $a_x = b_{x'}$. The existence of an index $k' < x'$ such that $b_{k'} \notin A$ contradicts Lemma 67. Thus, $\{a_1, \ldots, a_x\} = \{b_1, \ldots, b_{x'}\}$ and therefore, $a_1 = b_1, \ldots, a_x = b_x$. Equivalently, we can show that $a_x = b_{n-m+x}, \ldots, a_m = b_n$ if $l < x$. Finally, since there exists an $l \in [m]$ such that $a_l \in A \setminus B$ we cannot have both, $a_1 = b_1, \ldots, a_x = b_x$ and $a_x = b_{n-m+x}, \ldots, a_m = b_n$. □

The following corollary is an immediate consequence of Lemma 69.

Corollary 70. *Let $A = (a_1, \ldots, a_m)_\prec, B = (b_1, \ldots, b_n)_\prec$ be max cliques with $A \neq B$. Let $I := [\min\{m, n\} - 1]$. If A and B intersect, then they intersect in one of the following forms (see Figure 6.3):*

(a) $a_1 = b_1, \ldots, a_i = b_i$ *and* $\{a_{i+1}, \ldots, a_m\} \cap \{b_{i+1}, \ldots, b_n\} = \emptyset$ *for* $i \in I$,
(b) $\{a_1, \ldots, a_{m-j}\} \cap \{b_1, \ldots, b_{n-j}\} = \emptyset$ *and* $a_{m-j+1} = b_{n-j+1}, \ldots, a_m = b_n$ *for* $j \in I$,
(c) $a_1 = b_1, \ldots, a_i = b_i$, $\{a_{i+1}, \ldots, a_{m-j}\} \cap \{b_{i+1}, \ldots, b_{n-j}\} = \emptyset$ *and* $a_{m-j+1} = b_{n-j+1}, \ldots, a_m = b_n$ *for* $i, j \geq 1$ *and* $i + j \in I$.



(a) (b) (c)

Figure 6.3.: Max clique intersection types

For max cliques A and B with $A \neq B$, let

$$V_1^\prec(A, B) := \{a \in A \mid \forall a' \in A\colon (a' \preceq a \Rightarrow a' \in A \cap B)\} \quad \text{and}$$
$$V_2^\prec(A, B) := \{a \in A \mid \forall a' \in A\colon (a \preceq a' \Rightarrow a' \in A \cap B)\}.$$

It is easy to see, that for all $a' \in A$ with $a' \prec a$ we have $a' \in V_1^{\prec}(A,B)$ if $a \in V_1^{\prec}(A,B)$, and analogously, that we have $a' \in V_2^{\prec}(A,B)$ for all $a' \in A$ with $a \prec a'$ if $a \in V_2^{\prec}(A,B)$. Notice that $V_i^{\prec}(A,B)$ can be empty for an $i \in [2]$. Clearly, for all $i \in [2]$ we have $V_i^{\prec}(A,B) \subseteq A \cap B$. Moreover, if $a \in A \cap B$, then a is either in $V_1^{\prec}(A,B)$ or $V_2^{\prec}(A,B)$ according to Lemma 69. Thus, we obtain the following:

Observation 71. *Let $A, B \in \mathcal{M}$ be max cliques of H with $A \neq B$. Then $A \cap B$ is the disjoint union of $V_1^{\prec}(A,B)$ and $V_2^{\prec}(A,B)$.*

Corollary 70 implies that $V_i^{\prec}(A,B) = V_i^{\prec}(B,A)$ for all max cliques $A, B \in \mathcal{M}$ with $A \neq B$ and $i \in [2]$.

For a max clique A, let us define

$$V_i^{\prec}(A) := \bigcup_{B \in \mathcal{M} \setminus \{A\}} V_i^{\prec}(A,B).$$

An illustration of the set $V_i^{\prec}(A)$ can be found in Figure 6.4. Clearly, for a max clique $A = (a_1, \dots, a_m)_{\prec}$ we have $a_1, \dots, a_k \in V_1^{\prec}(A)$ if $a_k \in V_1^{\prec}(A)$, and $a_k, \dots, a_m \in V_2^{\prec}(A)$ if $a_k \in V_2^{\prec}(A)$. Further, the set $V_i^{\prec}(A)$ is a proper subset of A: Since $a_m \notin V_1^{\prec}(A,B)$ for any $B \in \mathcal{M} \setminus \{A\}$, we have $a_m \notin V_1^{\prec}(A)$. Equivalently we obtain $a_1 \notin V_2^{\prec}(A)$.

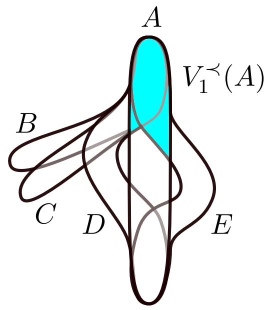
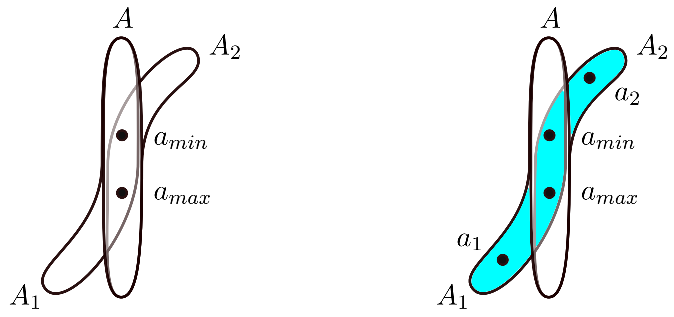


Figure 6.4.: $V_1^{\prec}(A)$

Observation 72. *Let $V_1^{\prec}(A) \neq \emptyset$, $V_2^{\prec}(A) \neq \emptyset$, and let a_{max} be the $\prec$-maximal element in $V_1^{\prec}(A)$ and a_{min} be the $\prec$-minimal element in $V_2^{\prec}(A)$. Then $a_{max} \prec a_{min}$ and $V_1^{\prec}(A) \cap V_2^{\prec}(A) = \emptyset$.*

Proof. Assume $a_{min} \preceq a_{max}$. Then there exists a max clique $A_1 \in \mathcal{M} \setminus \{A\}$ such that $a_{max} \in V_1^{\prec}(A, A_1)$ and a max clique $A_2 \in \mathcal{M} \setminus \{A\}$ such that $a_{min} \in V_2^{\prec}(A, A_2)$ as illustrated in Figure 6.5a. Since $a_{min} \preceq a_{max}$, we have $a_{max} \in V_2^{\prec}(A, A_2)$. As a consequence, $a_{max} \in A \cap A_1 \cap A_2$, and $\{a_1 \in A_1 \mid a_1 \preceq a_{max}\} = \{a \in A \mid a \preceq a_{max}\}$ and $\{a_2 \in A_2 \mid a_2 \succeq a_{max}\} = \{a \in A \mid a \succeq a_{max}\}$.

Let C be a max clique containing the clique $D := \{a \in A_1 \mid a \succeq a_{max}\} \cup \{a \in A_2 \mid a \prec a_{max}\}$ (see Figure 6.5b). As $A_1 \not\subseteq A$ and $\{a \in A_1 \mid a \preceq a_{max}\} = \{a \in A \mid a \preceq a_{max}\}$, there is an $a_1 \in A_1$ with $a_1 \succ a_{max}$ such that $a_1 \notin A$, analogously, there is an $a_2 \in A_2$ with $a_2 \prec a_{max}$ such that $a_2 \notin A$. Hence, $C \supseteq D$ is a maximal clique with $a_2 \prec a_{max} \prec a_1$ where $a_1, a_2 \notin A \cap C$ and $a_{max} \in A \cap C$, which is a contradiction to Lemma 68. □



(a) Max cliques A_1 and A_2 (b) Max clique C

Figure 6.5.: Illustration for the proof of Observation 72

Observation 73. *If $a \in V_1^{\prec}(A)$, then $v \in V_1^{\prec}(A)$ for all $v \in V$ with $v \prec a$. Equivalently, if $a \in V_2^{\prec}(A)$, then $v \in V_2^{\prec}(A)$ for all $v \in V$ with $a \prec v$.*

Proof. Without loss of generality, let $a \in V_1^{\prec}(A)$. Let v be an arbitrary vertex in V with $v \prec a$. Then $\{a, v\} \in E$ and there exists a max clique $B \in \mathcal{M}$ with $a, v \in B$. If $A = B$, then clearly $v \in V_1^{\prec}(A)$. Thus, let $A \neq B$. Since $a \in A \cap B$, there exists an $i \in [2]$ such that $a \in V_i^{\prec}(A, B) = V_i^{\prec}(B, A)$ (Observation 71). If $i = 1$, then $v \in V_1^{\prec}(B, A) \subseteq V_1^{\prec}(A)$. If $i = 2$, then $a \in V_2^{\prec}(A, B) \subseteq V_2^{\prec}(A)$, which is a contradiction to the previous lemma. □

The following lemma is used in Section 6.3.4.

Lemma 74. *Let $A = (a_1, \dots, a_m)_{\prec}, B, C$ be three max cliques such that $A \cap B \cap C \neq \emptyset$. Then a_1 or a_m is contained in $A \cap B \cap C$.*

Proof. Let us assume $A \cap B \cap C$ contains neither a_1 nor a_m, but as $A \cap B \cap C \neq \emptyset$ there is an $x \in [m]$ such that $a_x \in A \cap B \cap C$. According to Lemma 69 either $a_1, \dots, a_x \in A \cap B$ or $a_x, \dots, a_m \in A \cap B$. Without loss of generality let us assume $a_1, \dots, a_x \in A \cap B$. Since $a_1 \notin A \cap B \cap C$, we obtain $a_1 \notin C$. Then for $A \cap C$ Lemma 69 implies that $a_x, \dots, a_m \in A \cap C$. Therefore, a_x is contained in $V_1^{\prec}(A, B) \subseteq V_1^{\prec}(A)$, but also in $V_2^{\prec}(A, C) \subseteq V_2^{\prec}(A)$, which is a contradiction to Observation 72. □

Remember, two vertices $a_1, a_2 \in V$ *span* a max clique $A \in \mathcal{M}$, if A is the only max clique that contains vertices a_1, a_2. If $a_1, a_2 \in V$ span a max clique $A \in \mathcal{M}$, we call the two vertices a_1, a_2 a *spanning pair* of A.

Lemma 75. *Every max clique of H has a spanning pair.*

Proof. If there exists a vertex $a \in A$ that is contained in no other max clique of H, then obviously vertices a, a span max clique A. Thus, let A be a max clique where each vertex in A is also contained in another max clique. Let $A = (a_1, \dots, a_m)_{\prec}$. Both $V_1^{\prec}(A)$ and $V_2^{\prec}(A)$ are non-empty: For vertex a_1 there exists a max clique $B \neq A$ such that $a_1 \in B$. Thus, $a_1 \in V_1^{\prec}(A, B) \subseteq V_1^{\prec}(A)$. Equivalently, we can show that $a_m \in V_2^{\prec}(A)$.

Now, let a_{max} be the $\prec$-maximal element in $V_1^{\prec}(A)$ and a_{min} be the $\prec$-minimal element in $V_2^{\prec}(A)$. Let us assume a_{max}, a_{min} do not span A. Then there is a max clique

$B \in \mathcal{M} \setminus \{A\}$ such that $a_{max}, a_{min} \in B$. For a_{max} we have either $a_{max} \in V_1^{\prec}(A, B)$ or $a_{max} \in V_2^{\prec}(A, B)$ according to Observation 71. By Observation 72 we cannot have $a_{max} \in V_2^{\prec}(A, B) \subseteq V_2^{\prec}(A)$. Thus, $a_{max} \in V_1^{\prec}(A, B)$. Analogously, $a_{min} \in V_2^{\prec}(A, B)$. Therefore, $\{a_1, \ldots, a_{max}\}, \{a_{min}, \ldots, a_m\} \subseteq B$. As A is a max clique different from B, there has to be a vertex $a \in A \setminus B$ such that $a_{max} \prec a \prec a_{min}$. Since every vertex in A is also contained in another max clique, there exists a max clique $C \neq A$ with $a \in C$. Then Observation 71 implies that $a \in V_i^{\prec}(A, C) \subseteq V_i^{\prec}(A)$ for an $i \in [2]$, which is a contradiction to the maximality of a_{max} or the minimality of a_{min}. Hence, the vertices a_{max}, a_{min} span max clique A. □

A direct consequence of Lemma 75 is that there exist at most $|V|^2$ max cliques in a chordal comparability graph. Further, by using the spanning vertices we can define the max cliques in FO, which is shown in Section 2.8.2. Note that in the following sections we will represent max cliques as described in Section 2.8.2 and use the formulas defined in Section 2.8.2.

6.3.2. Ends and the Bundle Tree

In this section we consider connected chordal comparability graphs. Again, we do not require that the graphs are prime. Thus, in the following let $H = (V, E)$ be a connected chordal comparability graph with $|V| > 1$, and let $\prec$ be a strict partial order for H. We introduce $\prec$-ends and the $\prec$-bundle tree of H in this section. Let $\mathcal{M}$ be the set of max cliques of H.

Let $A = (a_1, \ldots, a_m)_{\prec}$ be a max clique of H. We call the vertices a_1 and a_m the *$\prec$-ends* of A. We also say $e \in V$ is a *$\prec$-end* of H if there exists a max clique $A \in \mathcal{M}$ such that e is a $\prec$-end of A. Since H is connected and $|V| > 1$, every max clique of H consists of at least two vertices and has therefore two distinct $\prec$-ends. As A is a max clique, $\prec$-end a_1 must be a minimal and $\prec$-end a_m a maximal element of V regarding $\prec$. Moreover, each $\prec$-minimal or $\prec$-maximal vertex e of V must be a $\prec$-end of every max clique containing it. Thus, we observe the following.

Observation 76. *Let $\prec$ be a strict partial order for H. Then vertex e is a $\prec$-end of H if, and only if, e is $\prec$-minimal or $\prec$-maximal.*

Observation 77. *Let $\prec$ be a strict partial order for H, and let A be a max clique of H. If vertex e is a $\prec$-end of H and $e \in A$, then e is a $\prec$-end of max clique A.*

Let $F^{\prec}$ be the set of $\prec$-ends of H. In the following we consider the subgraph of H induced by the set $F^{\prec}$.

Clearly, for all $\prec$-ends $e_1, e_2 \in F^{\prec}$ with $e_1 \neq e_2$, there is an edge between e_1 and e_2 in H if and only if there exists a max clique $A \in \mathcal{M}$ such that $e_1, e_2 \in A$. Further, notice that whenever two vertices $e, f \in F^{\prec}$ with $e \neq f$ satisfy $e, f \in A$ for a max clique $A \in \mathcal{M}$, the vertices e and f are the $\prec$-ends of A, according to Observation 77. Thus, an edge $\{e, f\}$ in $H[F^{\prec}]$ represents the set of all max cliques with $\prec$-ends e and f.

We show that the induced subgraph $H[F^{\prec}]$ is a tree (Lemma 80). We call $H[F^{\prec}]$ the *$\prec$-bundle tree* of H. We start by showing that $H[F^{\prec}]$ is connected and bipartite.

Lemma 78. *Let $H = (V, E)$ be connected and $|V| > 1$. Then $H[F^{\prec}]$ is connected.*

Proof. Let $H = (V, E)$ be a connected chordal comparability graph with $|V| > 1$. Then each max clique of H has exactly two $\prec$-ends. Let us assume $H[F^{\prec}]$ is not connected, and let F' be a connected component of $H[F^{\prec}]$. Since each $\prec$-end in F' is the $\prec$-end of a max clique, which has two $\prec$-ends, we have $|F'| \geq 2$. Let $\mathcal{M}'$ be the set of max cliques where both $\prec$-ends are in F', and let $W' := \bigcup \mathcal{M}'$. The set W' is a proper subset of V as $F^{\prec} \setminus F' \not\subseteq W'$. Further, $\mathcal{M}' \neq \emptyset$ implies $W' \neq \emptyset$. Since H is connected, there must exist vertices $v \in V \setminus W'$ and $w \in W'$ that are adjacent. Let $A \in \mathcal{M} \setminus \mathcal{M}'$ be a max clique containing v and w. Further, w is contained in a max clique $B \in \mathcal{M}'$. Notice that $A \neq B$. Thus, $w \in A \cap B$ and by Lemma 69 max cliques A and B share a $\prec$-end $e \in F'$. Let f be the other $\prec$-end of A. Then there is an edge between $e \in F'$ and $f \in F^{\prec} \setminus F'$ in $H[F^{\prec}]$, a contradiction. □

We use the strict partial order $\prec$ to define two subsets $F_1^{\prec}$ and $F_2^{\prec}$ of $F^{\prec}$. We let $F_1^{\prec}$ be the set of all $\prec$-ends that are $\prec$-minimal and $F_2^{\prec}$ be the set of all $\prec$-ends that are $\prec$-maximal.

Lemma 79. *Let $H = (V, E)$ be connected and $|V| > 1$. Then $\{F_1^{\prec}, F_2^{\prec}\}$ is a 2-coloring of the graph $H[F^{\prec}]$.*

Proof. Let $H = (V, E)$ be a connected chordal comparability graph with $|V| > 1$. According to Observation 76, we have $F^{\prec} = F_1^{\prec} \cup F_2^{\prec}$. Further, $F_1^{\prec} \cap F_2^{\prec}$ is empty, because a vertex that is $\prec$-minimal and $\prec$-maximal forms a max clique of size 1, which cannot exist in a connected graph with more than one vertex. Thus, $\{F_1^{\prec}, F_2^{\prec}\}$ is a partition of $F^{\prec}$.

Now, let $e, f \in F^{\prec}$ be adjacent in $H[F^{\prec}]$. Then $\{e, f\} \in E$, and either $e \prec f$ or $f \prec e$. Therefore, e is $\prec$-minimal if and only if f is $\prec$-maximal, which implies that $e \in F_1^{\prec}$ if and only if $f \in F_2^{\prec}$. Consequently, $\{F_1^{\prec}, F_2^{\prec}\}$ is a 2-coloring of $F^{\prec}$. □

Now we can prove that $H[F^{\prec}]$ is a tree.

Lemma 80. *Let H be connected. Then $H[F^{\prec}]$ is a tree.*

Proof. Let H be a connected chordal comparability graph. By Lemma 78, the graph $H[F^{\prec}]$ is connected. Let us assume $C = f_1, \ldots, f_m, f_1$ is a cycle in $H[F^{\prec}]$ of minimal length. Then C is an induced cycle in $H[F^{\prec}]$. Therefore, C is also an induced cycle in H. As there do not exist induced cycles of length greater than 3 in a chordal graph, cycle C must have length 3. This contradicts Lemma 79, since bipartite graphs cannot have cycles of length 3. □

6.3.3. Inner and Outer Ends and Max Cliques

In the subsequent sections we show that for prime chordal comparability graphs the $\prec$-ends of a max clique do not depend on the underlying strict partial order $\prec$ and that it is possible to define these $\prec$-ends in FP without knowing $\prec$. In order to do this we define two different kinds of $\prec$-ends, inner and outer $\prec$-ends, in this section. We also define two different kinds of max cliques, inner and outer max cliques, and show the connection between inner and outer max cliques and inner and outer $\prec$-ends.

From now on we consider prime chordal comparability graphs G with $|V| > 2$. Notice that a prime graph with at least 3 vertices is connected and co-connected, and it must have at least 2 maximal cliques. Further, each max clique contains at least 2 vertices. Let $\mathcal{M}$ be the set of max cliques of G.

Let $\prec$ be a strict partial order for G, and let $F^{\prec}$ be the set of $\prec$-ends of G. We distinguish between different types of $\prec$-ends. We call a $\prec$-end $e \in F^{\prec}$ an *inner $\prec$-end* if e is an inner node of the bundle tree $G[F^{\prec}]$ of G. If e is an outer node, that is, a leaf, of the bundle tree $G[F^{\prec}]$ we say $e \in F^{\prec}$ is an *outer $\prec$-end*. We let $F^{\prec}_{\text{in}}$ be the set of inner $\prec$-ends, and $F^{\prec}_{\text{out}}$ be the set of outer $\prec$-ends of G. Clearly, $F^{\prec}$ is the disjoint union of $F^{\prec}_{\text{in}}$ and $F^{\prec}_{\text{out}}$.

We also distinguish between different types of max cliques. Let the *neighborhood* $\mathcal{N}(A)$ of a max clique $A \in \mathcal{M}$ be the set of all max cliques $B \in \mathcal{M} \setminus \{A\}$ for which $B \cap A \neq \emptyset$. We say a max clique A is an *inner max clique* if there exist max cliques $A_1, A_2 \in \mathcal{N}(A)$ with $A_1 \cap A_2 = \emptyset$, and a max clique is an *outer max clique* otherwise. In Figure 6.6 you find an example for an inner max clique A. Notice that the definition of inner and outer max clique does not depend on the strict partial order for the graph. Further, the set of inner max cliques (and therefore also the set of outer max cliques) is FO-definable. It is easy to see, that there is an FO-formula that decides for each (spanning pair of) max clique A whether there exist (spanning pairs of) two max cliques A_1, A_2 that each have a non-empty intersection with A and that do not intersect with each other.

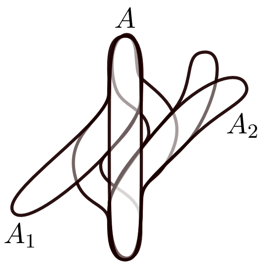


Figure 6.6.: Inner max clique A

Observation 81. *Let $\prec$ be an arbitrary strict partial order for G. Let $A = (a_1, \ldots, a_m)_{\prec}$ be an inner max clique of G, and let $A_1, A_2 \in \mathcal{N}(A)$ with $A_1 \cap A_2 = \emptyset$. Then either $a_1 \in A_1$ and $a_m \in A_2$, or $a_1 \in A_2$ and $a_m \in A_1$.*

Proof. For each $i \in [2]$ we have $a_1 \in A_i$ or $a_m \in A_i$ according to Corollary 70. Since $A_1 \cap A_2 = \emptyset$, we have either $a_1 \in A_1$ and $a_m \in A_2$, or $a_1 \in A_2$ and $a_m \in A_1$. ☐

Lemma 82. *Let $\prec$ be an arbitrary strict partial order for G, and let $A \in \mathcal{M}$ be an inner max clique of G. Then the two $\prec$-ends of A are inner $\prec$-ends.*

Proof. Let max clique $A \in \mathcal{M}$ be an inner max clique of G, and let $A_1, A_2 \in \mathcal{N}(A)$ be max cliques with $A_1 \cap A_2 = \emptyset$. Let us assume there is a strict partial order $\prec$ for G such that A has an outer $\prec$-end e. Let e' be the other $\prec$-end of A. According to Observation 81 we have $e \in A_i$ and $e' \in A_{3-i}$ for an $i \in [2]$. Let f be the $\prec$-end with $f \neq e$ of A_i. As $A_1 \cap A_2 = \emptyset$, we have $f \neq e'$. Hence, node e has two neighbors, e' and f, in the bundle tree $G[F^{\prec}]$ of G. Therefore, e is an inner $\prec$-end, a contradiction. □

Lemma 83. *Let $\prec$ be an arbitrary strict partial order for G, and let $A \in \mathcal{M}$ be an outer max clique of G. Then there is a $\prec$-end of A that is an outer $\prec$-end.*

Proof. Let us assume there is a strict partial order $\prec$ for G such that outer max clique $A \in \mathcal{M}$ has two inner $\prec$-ends e and f. Clearly, e and f are adjacent in $G[F^{\prec}]$. As e and f are inner $\prec$-ends, there exists a neighbor e' of e with $e' \neq f$ and a neighbor f' of f with $f' \neq e$ in $G[F^{\prec}]$. Since $G[F^{\prec}]$ is a tree (Lemma 80), we have $e' \neq f'$. Let A_1 be a max clique with $e, e' \in A_1$, and let A_2 be a max clique with $f, f' \in A_2$. Then $A_1, A_2 \in \mathcal{N}(A)$. By Observation 77, e and e' are the $\prec$-ends of A_1, and f and f' are the $\prec$-ends of A_2. Since $\{e, e'\} \cap \{f, f'\} = \emptyset$, it follows from Corollary 70 that $A_1 \cap A_2 = \emptyset$. Consequently, A is an inner max clique, which is a contradiction. □

Lemma 84. *Let $\prec$ be an arbitrary strict partial order for G and let $A = (a_1, \ldots, a_m)_\prec$ be an outer max clique of G. Then $a_1 \in B$ for all $B \in \mathcal{N}(A)$, or $a_m \in B$ for all $B \in \mathcal{N}(A)$.*

Proof. Let $\prec$ be an arbitrary strict partial order for G, and let $A = (a_1, \ldots, a_m)_\prec \in \mathcal{M}$ be an outer max clique. Let us assume there exist max cliques $B = (b_1, \ldots, b_n)_\prec$ and $B' = (b'_1, \ldots, b'_p)_\prec$ in $\mathcal{N}(A)$ with $a_1 \notin B'$ and $a_m \notin B$. Max cliques B and B' intersect with A. Therefore, we have $a_1 = b_1$ and $a_m = b'_p$ according to Corollary 70. Since A is an outer max clique, B and B' must intersect with each other. Thus, we have $b_1 = b'_1$ or $b_n = b'_p$ (Corollary 70). If $b_1 = b'_1$, then $a_1 \in B'$, and if $b_n = b'_p$, then $a_m \in B$, a contradiction. □

Lemma 85. *Let $A \in \mathcal{M}$ be an outer max clique. Then there exists an inner max clique $B \in \mathcal{M}$ with $A \cap B \neq \emptyset$.*

Proof. Let $\prec$ be an arbitrary strict partial order for G and let $A = (a_1, \ldots, a_m)_\prec$ be an outer max clique. Let $\mathcal{D}$ be the set $\mathcal{N}(A) \cup \{A\}$ of max cliques, and let us assume all max cliques in $\mathcal{D}$ are outer max cliques. By Lemma 84, $\prec$-end a_1 or $\prec$-end a_m is contained in $I := \bigcap\{D \in \mathcal{D}\}$. Without loss of generality let $a_1 \in I$. In the following we prove that all vertices $w \in V \setminus \{a_1\}$ are adjacent to a_1.

Let us assume the opposite, and let W be the non-empty set of vertices $w \in V \setminus \{a_1\}$ that are not adjacent to a_1. Since G is connected, there must exist a vertex $w \in W$ that is adjacent to a vertex $b \in V \setminus W$. Vertex b must be distinct from a_1 because w and a_1 are not adjacent. As there is an edge between b and a_1, there exist a max cliques B that contains the vertices b and a_1. Then $A \cap B \neq \emptyset$ and $B \in \mathcal{D}$. Further, there is a max clique C with $b, w \in C$ since b and w are adjacent. Clearly, $C \cap B \neq \emptyset$. As B is an outer max clique, we have $A \cap C \neq \emptyset$. Hence, $C \in \mathcal{D}$ and $a_1 \in C$. Consequently, there must be an edge between w and a_1, a contradiction.

Thus, all vertices in $V \setminus \{a_1\}$ are adjacent to a_1. Since $|V| > 2$, the set $V \setminus \{a_1\}$ is a non-trivial module. We obtain a contradiction because G is prime. □

Lemma 85 implies that there always exists an inner max clique of G. For every strict partial order $\prec$ for G, such an inner max clique has two inner $\prec$-ends by Lemma 82. From this we can infer the following for the $\prec$-bundle tree of G:

Corollary 86. *The $\prec$-bundle tree $G[F^{\prec}]$ of G has at least two inner nodes for each strict partial order $\prec$ for G.*

Lemma 87. *Let $\prec$ be an arbitrary strict partial order for G, and let $A \in \mathcal{M}$ be an outer max clique of G. Then there is a $\prec$-end of A that is an inner $\prec$-end.*

Proof. Let $\prec$ be an arbitrary strict partial order for G, and let $A \in \mathcal{M}$ be an outer max clique of G. Let $B \in \mathcal{M}$ be an inner max clique with $A \cap B \neq \emptyset$, which exists according to Lemma 85. By Corollary 70, the set $A \cap B$ contains a vertex e that is a $\prec$-end of A and of B. It follows from Lemma 82 that e is an inner $\prec$-end. □

The following corollary follows from Lemma 83 and Lemma 87.

Corollary 88. *Let $\prec$ be an arbitrary strict partial order for G, and let $A \in \mathcal{M}$ be an outer max clique of G. Then A has an outer $\prec$-end and an inner $\prec$-end.*

Lemma 89. *Let $\prec$ be an arbitrary strict partial order for G. Let $e \in F^{\prec}$ be an inner $\prec$-end of G. Then e is a $\prec$-end of an inner max clique.*

Proof. Let $\prec$ be an arbitrary strict partial order for G, and let $e \in F^{\prec}$ be an inner $\prec$-end of G. Let us assume e is not a $\prec$-end of any inner max clique. Then e must be a $\prec$-end of an outer max clique A. By Lemma 85 there exists an inner max clique $B \in \mathcal{M}$ with $A \cap B \neq \emptyset$. As shown in the proof of Lemma 87, there exists an inner $\prec$-end e' that is a $\prec$-end of A and of B. Since e is the only inner $\prec$-end of outer max clique A according to Corollary 88, we have $e = e'$, and e is a $\prec$-end of inner max clique B, a contradiction. □

Lemma 82 and Lemma 89 yield the following corollary.

Corollary 90. *Let $\prec$ be an arbitrary strict partial order for G, and let $e \in F^{\prec}$. Then, vertex e is an inner $\prec$-end if and only if vertex e is a $\prec$-end of an inner max clique.*

The following corollary is a direct consequence of Corollary 90 and Observation 81.

Corollary 91. *Let $\prec$ be an arbitrary strict partial order for G. Each inner $\prec$-end is contained in at least two max cliques.*

6.3.4. Inner Ends

In this section we show that the inner $\prec$-ends do not depend on the strict partial order $\prec$ for G and are definable in FO.

Let A be an inner max clique. Thus, there exist two max cliques $A_1, A_2 \in \mathcal{N}(A)$ such that $A_1 \cap A_2 = \emptyset$. For $i = 1, 2$ we let

$$E_{A_i}(A) := \bigcap \{B \in \mathcal{M} \mid (A_i \cap A) \cap B \neq \emptyset\}. \tag{6.1}$$

Note that $E_{A_i}(A) \subseteq A_i \cap A$ for each $i \in [2]$ since $(A_i \cap A) \cap A_i \neq \emptyset$ and $(A_i \cap A) \cap A \neq \emptyset$.

Example 92. Figure 6.7 shows an inner max clique A and its neighborhood $\mathcal{N}(A)$. Two max cliques A_1, A_2 in $\mathcal{N}(A)$ such that $A_1 \cap A_2 = \emptyset$ are indicated, and all max cliques B with $(A_1 \cap A) \cap B \neq \emptyset$ and the set $E_{A_1}(A)$ are highlighted. ⌟

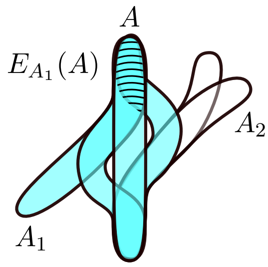


Figure 6.7.: The set $E_{A_1}(A)$ for an inner max clique A

Proposition 93. *Let A be an inner max clique and let $A_1, A_2 \in \mathcal{N}(A)$ be arbitrary max cliques such that $A_1 \cap A_2 = \emptyset$. Then $|E_{A_1}(A)| = |E_{A_2}(A)| = 1$. More precisely, for every strict partial order $\prec$ for G, which induces the order $a_1, \ldots, a_m$ on the elements of A, we have $E_{A_1}(A) = \{a_1\}$ and $E_{A_2}(A) = \{a_m\}$ if $a_1 \in A_1$ and $a_m \in A_2$, and $E_{A_2}(A) = \{a_1\}$ and $E_{A_1}(A) = \{a_m\}$ otherwise.*

Proof. Let $A \in \mathcal{M}$ and $A_1, A_2 \in \mathcal{N}(A)$ such that $A_1 \cap A_2 = \emptyset$. Let $\prec$ be an arbitrary strict partial order for G and $A = (a_1, \ldots, a_m)_\prec, A_i = (a_1^i, \ldots, a_n^i)_\prec$. According to Observation 81 either $a_1 \in A_1$ and $a_m \in A_2$, or $a_1 \in A_2$ and $a_m \in A_1$. Without loss of generality, let us assume $a_1 \in A_1$ and $a_m \in A_2$. Then $a_m \notin A_1$ and $a_1 \notin A_2$. We show that $E_{A_1}(A) = \{a_1\}$. The proof that $E_{A_2}(A) = \{a_m\}$ can be obtained the same way. First, let us suppose that $a_1 \notin E_{A_1}(A)$. Then there exists a max clique B with $(A_1 \cap A) \cap B \neq \emptyset$ and $a_1 \notin B$. By Lemma 74, it follows that $a_m \in (A_1 \cap A) \cap B \subseteq A_1$, a contradiction. Consequently, $a_1 \in E_{A_1}(A)$, and therefore, $|E_{A_1}(A)| \geq 1$. Further, we have $|E_{A_1}(A)| < |V|$ since $a_m \notin E_{A_1}(A) \subseteq A_1$. Next, we prove that $E_{A_1}(A)$ is a module. Then, it follows that $|E_{A_1}(A)| = 1$, and therefore $E_{A_1}(A) = \{a_1\}$, as there do not exist non-trivial modules in G. For a contradiction, let us assume $E_{A_1}(A)$ is not a module, that is, there are vertices $u, v \in E_{A_1}(A)$ and $w \notin E_{A_1}(A)$ such that u and w are adjacent and v and w are not adjacent. Since there is an edge between u and w, there exists a max clique C with $u, w \in C$. As u is contained in $E_{A_1}(A) \subseteq A_1 \cap A$ and in max clique C, we have $(A_1 \cap A) \cap C \neq \emptyset$. Thus, $E_{A_1}(A) \subseteq C$, and therefore, $v \in C$. Hence, $v, w \in C$ but v and w are not adjacent, a contradiction. □

A consequence of Proposition 93 is that the $\prec$-ends of an inner max clique A of G do not depend on the strict partial order $\prec$ for G, because the sets $E_{A_1}(A)$ and $E_{A_2}(A)$ defined in (6.1) do not depend on $\prec$. We obtain the following corollary.

Corollary 94. *Let A be an inner max clique of a graph G. Then there exist $e, f \in A$ with $e \neq f$ such that e and f are the $\prec$-ends of A for all strict partial orders $\prec$ for G.*

Let $\prec$ be an arbitrary strict partial order for G. By Corollary 90 we know that a $\prec$-end is an inner $\prec$-end if and only if it is a $\prec$-end of an inner max clique. Hence, the set $F_{\text{in}}^{\prec}$ of inner $\prec$-ends equals the set of $\prec$-ends of inner max cliques. Corollary 94 implies that the set $F_{\text{in}}^{\prec}$ does not depend on the the strict partial order $\prec$ for G.

Corollary 95. *There exists a set $F_{\text{in}} \subseteq V$ of vertices of G such that $F_{\text{in}} = F_{\text{in}}^{\prec}$ for all strict partial orders $\prec$ for G.*

Since inner $\prec$-ends do not depend on the strict partial order $\prec$ for G, we simply call them *inner ends*. We denote the set of all inner ends of G by F_{in}.

The set F_{in} of inner ends is definable in FO: Since F_{in} equals the set of $\prec$-ends of inner max cliques, it suffices to show that the set of $\prec$-ends of inner max cliques is FO-definable. We use the sets $E_{A_1}(A)$ and $E_{A_1}(A)$ as described in Proposition 93 to define the $\prec$-ends of inner max cliques A. First of all, we use spanning pairs to define max cliques in FO (see Section 2.8.2). Clearly, we can test in FO whether a max clique A, represented by a spanning pair, is an inner max clique. Moreover, there is an FO-formula that defines for each spanning pair of a max clique A spanning pairs of two arbitrary max cliques $A_1, A_2 \in \mathcal{N}(A)$ with $A_1 \cap A_2 = \emptyset$. Then we can check in FO whether a vertex e is a $\prec$-end of the inner max clique A by testing whether e is contained in all max cliques B (represented by a spanning pair) that have a non-empty intersection with $A \cap A_1$ or all max cliques B that have a non-empty intersection with $A \cap A_2$. Thus, there exists an FO-formula $\psi(x_1, x_2, x^*)$ that is satisfied for prime chordal comparability graphs G and vertices $a, b, c \in V$ exactly if a, b span an inner max clique A of G and c is a $\prec$-end of A. Then $\varphi_{F_{\text{in}}} := \exists x_1 \exists x_2\, \psi(x_1, x_2, x^*)$ defines the set of all $\prec$-ends of inner max cliques, and therefore, the set F_{in} of inner ends.

Corollary 96. *There exists an FO-formula $\varphi_{F_{\text{in}}}(x^*)$ that is satisfied for a vertex $e \in V$ in a prime chordal comparability graph G if, and only if, e is an inner end of A.*

6.3.5. The Sets $S_1^{\prec}$ and $S_2^{\prec}$

It is more difficult to show that the outer $\prec$-ends do not depend on the strict partial order $\prec$ for G and that they are definable in FP. We introduce the framework necessary to obtain these results in this and the next section.

Let $\mathcal{M}_v$ be the set of max cliques $A \in \mathcal{M}$ of G that satisfy $v \in A$. Further, let U_v be the set of all vertices that are contained in at least one max clique in $\mathcal{M}_v$, that is, $U_v := \bigcup \mathcal{M}_v$. It is not hard to see that the set U_v consists of vertex v and all vertices that are adjacent to v. Thus, for each $v \in V$ the set U_v is FO-definable. Let $\prec$ be a strict partial order for G. We consider sets $\mathcal{M}_e$ and U_e where $e \in F^{\prec}$. Observation 77 implies that vertex $e \in F^{\prec}$ is a $\prec$-end of all max cliques $A \in \mathcal{M}_e$.

For a strict partial order $\prec$ for G and $e \in F^{\prec}$, let

$$S_1^{\prec}(e) := \bigcup_{A \in \mathcal{M}_e} V_1^{\prec}(A) \quad \text{and} \quad S_2^{\prec}(e) := \bigcup_{A \in \mathcal{M}_e} V_2^{\prec}(A). \tag{6.2}$$

The sets $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$ are depicted in Figure 6.8. Clearly, we have $S_1^{\prec}(e) \subseteq U_e$ and $S_2^{\prec}(e) \subseteq U_e$.

As a direct consequence of Observation 73 we obtain the subsequent observation.

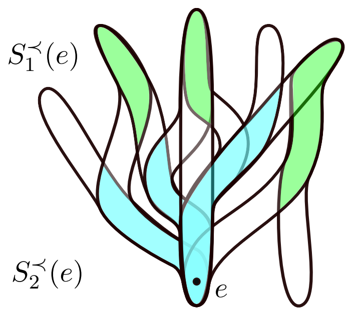


Figure 6.8.: The sets $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$

Observation 97. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. If $a \in S_1^{\prec}(e)$, then we have $b \in S_1^{\prec}(e)$ for all $b \in V$ with $b \prec a$. Equivalently, if $a \in S_2^{\prec}(e)$, then we have $b \in S_2^{\prec}(e)$ for all $b \in V$ with $a \prec b$.*

Observation 71 implies the next observation.

Observation 98. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $v \in V$. Then there exists an $i \in [2]$ such that $v \in S_i^{\prec}(e)$ if, and only if, there exist max cliques $A \in \mathcal{M}_e$ and $B \in \mathcal{M} \setminus \{A\}$ with $v \in A \cap B$.*

In the following we present some properties of $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$ for $e \in F^{\prec}$.

Observation 99. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. If e is $\prec$-minimal, then $e \in S_1^{\prec}(e)$ or $S_1^{\prec}(e) = \emptyset$. If e is $\prec$-maximal, then $e \in S_2^{\prec}(e)$ or $S_2^{\prec}(e) = \emptyset$.*

Proof. Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Without loss of generality, let e be $\prec$-minimal. Let $S_1^{\prec}(e) \neq \emptyset$. Then there exists a vertex $v \in S_1^{\prec}(e) \subseteq U_e$, and there has to be an edge between e and v. As e is $\prec$-minimal we have $e \prec v$. It follows from Observation 97 that $e \in S_1^{\prec}(e)$. □

Observation 100. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Then $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$ are disjoint sets of vertices of U_e.*

Proof. Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let us assume there exists a vertex $v \in S_1^{\prec}(e) \cap S_2^{\prec}(e)$. It follows that there are max cliques $A, B \in \mathcal{M}_e$ such that $v \in V_1^{\prec}(A) \cap V_2^{\prec}(B) \subseteq A \cap B$. According to Observation 71, we have $v \in V_1^{\prec}(A, B)$ or $v \in V_2^{\prec}(A, B)$. Now, $v \in V_1^{\prec}(A, B)$ implies $v \in V_1^{\prec}(B)$, and $v \in V_2^{\prec}(A, B)$ yields $v \in V_2^{\prec}(A)$. In both cases we have a contradiction to Observation 72. □

For $e \in F^{\prec}$, let $O_e^{\prec} := U_e \setminus (S_1^{\prec}(e) \cup S_2^{\prec}(e))$. The set $O_e^{\prec}$ is the set of vertices of U_e that are contained in exactly one max clique according to Observation 98. Thus, we obtain the following observation.

Observation 101. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $v \in U_e$. Then $v \in O_e^{\prec}$ if, and only if, v is contained in only one max clique.*

This implies that the set $O_e^{\prec}$ does not depend on the strict partial order $\prec$ for G. Therefore, we denote $O_e^{\prec}$ by O_e. We call O_e the *middle* of e. Note that there cannot exist two vertices in O_e that are contained in the same max clique as these would form a non-trivial module.

Since we can express inner ends and max cliques in first order logic, the middle O_e of e is definable in FO for inner ends $e \in F_{\text{in}}$ by applying Observation 101.

Lemma 102. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $C = (c_1, \dots, c_p)_{\prec} \in \mathcal{M}_e$ be a max clique with $\prec$-end $e \in \{c_1, c_p\}$. Then, $c_1 \notin S_2^{\prec}(e)$ and $c_p \notin S_1^{\prec}(e)$.*

Proof. Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $C = (c_1, \dots, c_p)_{\prec} \in \mathcal{M}_e$. If C is the only max clique containing c_p, then $c_p \in O_e$ by Observation 101. If c_p is also contained in a max clique A different from C, then $c_p \in V_2^{\prec}(C, A) \subseteq V_2^{\prec}(C) \subseteq S_2^{\prec}(e)$, and $c_p \notin S_1^{\prec}(e)$ since $S_1^{\prec}(e) \cap S_2^{\prec}(e) = \emptyset$ by Observation 100. For c_1 we analogously obtain that c_1 is either in $S_1^{\prec}(e)$ or in O_e. Thus, $c_1 \notin S_2^{\prec}(e)$ and $c_p \notin S_1^{\prec}(e)$. □

Lemma 102 directly implies that there is no $i \in [2]$ such that both $\prec$-ends of a max clique are contained in the same set $S_i^{\prec}(e)$.

Observation 103. *Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $A \in \mathcal{M}$, $v \in A$, and let $i \in [2]$. If $v \in S_i^{\prec}(e)$, then $v \in V_i^{\prec}(A)$.*

Proof. Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $A \in \mathcal{M}$ and $i \in [2]$. Let v be a vertex in A that is in $S_i^{\prec}(e)$. Then there exists a max clique $B \in \mathcal{M}_e$ with $v \in V_i^{\prec}(B)$. If $B = A$, we are done. Thus, let $B \neq A$. Then, $v \in A \cap B$. According to Observation 71 either $v \in V_i^{\prec}(A, B)$ or $v \in V_{3-i}^{\prec}(A, B)$. If $v \in V_{3-i}^{\prec}(A, B)$, then $v \in V_{3-i}^{\prec}(B)$, a contradiction to Observation 72. Consequently, we have $v \in V_i^{\prec}(A, B) \subseteq V_i^{\prec}(A)$. □

Lemma 104. *Let $\prec$ be a strict partial order for G and let $e \in F^{\prec}$. Further, let $B \in \mathcal{M}$ and $C, M, N \in \mathcal{M}_e$ be max cliques, and let v be a vertex in $(B \cap C \cap M) \setminus N$. Let $i \in [2]$. If $v \in S_i^{\prec}(e)$, then $(M \cap N) \setminus S_{3-i}^{\prec}(e) \subset (B \cap C) \setminus S_{3-i}^{\prec}(e)$.*

A picture illustrating Lemma 104 can be found in Figure 6.9.

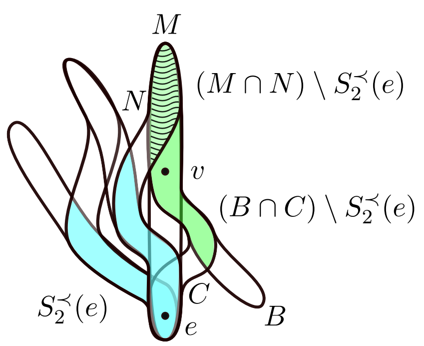


Figure 6.9.: Illustration for Lemma 104

Proof. Let $\prec$ be a strict partial order for G and $e \in F^{\prec}$. Let $B \in \mathcal{M}$ and $C, M, N \in \mathcal{M}_e$. Further, let $v \in (B \cap C \cap M) \setminus N$ and $v \in S_i^{\prec}(e)$ for an $i \in [2]$. First we show that $(M \cap N) \setminus S_{3-i}^{\prec}(e) \subseteq (B \cap C) \setminus S_{3-i}^{\prec}(e)$. Let $w \in (M \cap N) \setminus S_{3-i}^{\prec}(e)$. Since $w \in M \cap N$, we have $w \in V_j^{\prec}(M, N)$ for a $j \in [2]$ according to Observation 71. Now $V_j^{\prec}(M, N) \subseteq V_j^{\prec}(M) \subseteq S_j^{\prec}(e)$ implies that $w \in S_j^{\prec}(e)$. As $w \notin S_{3-i}^{\prec}(e)$, it follows that $i = j$ and $w \in S_i^{\prec}(e)$. Let us show that $w \in B \cap C$: By Observation 103 we have $v \in V_i^{\prec}(B) \cap V_i^{\prec}(C)$ and $w \in V_i^{\prec}(N)$. Since $v, w \in M$, either $v = w$, then $w \in B \cap C$, or there is an edge between v and w. If v and w are adjacent, then $v \prec w$ or $w \prec v$. Now we can apply Observation 73: Without loss of generality let $i = 1$. If $w \prec v$, then it follows from $v \in V_1^{\prec}(B) \cap V_1^{\prec}(C)$ that $w \in V_1^{\prec}(B) \cap V_1^{\prec}(C) \subseteq B \cap C$. If $v \prec w$, then $w \in V_1^{\prec}(N)$ implies that $v \in V_1^{\prec}(N) \subseteq N$, a contradiction. Therefore, $w \in B \cap C$. Consequently, $w \in (B \cap C) \setminus S_{3-i}^{\prec}(e)$.

By Observation 100 we have $v \notin S_{3-i}^{\prec}(e)$. As $v \in B \cap C$ but $v \notin M \cap N$, we have $(M \cap N) \setminus S_{3-i}^{\prec}(e) \subset (B \cap C) \setminus S_{3-i}^{\prec}(e)$. □

6.3.6. Sides of Inner Ends

For this section let $e \in F_{\text{in}}$. Since inner end e is contained in at least two max cliques by Corollary 91 and these two max cliques are in $\mathcal{M}_e$, we have $e \in S_1^{\prec}(e)$ or $e \in S_2^{\prec}(e)$ for $e \in F_{\text{in}}$ according to Observation 98. Further, $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$ are disjoint sets of vertices by Observation 100. Thus, inner end e is in exactly one of the two sets $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$. Let $i \in [2]$ such that $e \in S_i^{\prec}(e)$. In the following the goal is to define the sets

$$S_e^{\prec} := S_i^{\prec}(e) \text{ and } \hat{S}_e^{\prec} := S_{3-i}^{\prec}(e). \tag{6.3}$$

We call $S_e^{\prec}$ the *$\prec$-side* and $\hat{S}_e^{\prec}$ the *$\prec$-counterside* of $e \in F_{\text{in}}$. We show that for $e \in F_{\text{in}}$ the sets $S_e^{\prec}$ and $\hat{S}_e^{\prec}$ are the same for every strict partial order $\prec$ for G, and that they can be defined in FP using a simultaneous inflationary fixed-point operator.

In order to do that, we present a sequence of vertex sets $(X^k_{S_e^{\prec}}, X^k_{\hat{S}_e^{\prec}})_{k \in \mathbb{N}}$. This sequence of sets occurs within the recursion of the simultaneous IFP-formula, which defines the sets $S_e^{\prec}$ and $\hat{S}_e^{\prec}$ and which we will present afterwards. For large enough $l \in \mathbb{N}$ we have $(X^l_{S_e^{\prec}}, X^l_{\hat{S}_e^{\prec}}) = (S_e^{\prec}, \hat{S}_e^{\prec})$. Let $e \in F_{\text{in}}$. Further, let $\mathcal{S} := \{S_e^{\prec}, \hat{S}_e^{\prec}\}$. Our goal is to define the sets X^k_{S} for all $\mathsf{S} \in \mathcal{S}$ and all $k \in \mathbb{N}$.

We start by defining the sets $X^1_{S_e^{\prec}}$ and $X^1_{\hat{S}_e^{\prec}}$. We want them to contain vertices of which we definitely know that they are in $S_e^{\prec}$ and $\hat{S}_e^{\prec}$, respectively. Thus, we let $X^1_{S_e^{\prec}}$ contain the inner end e, and we let $X^1_{\hat{S}_e^{\prec}}$ be the set of all vertices in U_e that are also contained in a max clique $D \notin \mathcal{M}_e$. The following lemma shows that all vertices in $X^1_{\hat{S}_e^{\prec}}$ belong to $\hat{S}_e^{\prec}$.

Lemma 105. *Let $e \in F_{\text{in}}$, and let $\prec$ be a strict partial order for G. Let $A \in \mathcal{M}_e$ and $D \notin \mathcal{M}_e$ be max cliques such that $A \cap D \neq \emptyset$. Then $A \cap D \subseteq \hat{S}_e^{\prec}$.*

Proof. Let $e \in F_{\text{in}}$, and let $\prec$ be a strict partial order for G. Further, let $A \in \mathcal{M}_e$ and $D \in \mathcal{M} \setminus \mathcal{M}_e$ be such that $A \cap D \neq \emptyset$. As $D \notin \mathcal{M}_e$, we have $e \notin A \cap D$. Thus, Corollary 70 implies that the intersection of A and D is not of form (c). Consequently, there exists an

$i \in [2]$ such that $A \cap D \subseteq V_i^{\prec}(A, D) \subseteq V_i^{\prec}(A) \subseteq S_i^{\prec}(e)$. Note that $A \cap D$ contains the second $\prec$-end $e' \neq e$ of A. Since we already know that $e \in S_e^{\prec}$, and Lemma 102 implies that $S_e^{\prec}$ cannot contain both $\prec$-ends of A, it follows that $A \cap D \subseteq \hat{S}_e^{\prec}$. □

To sum up, we let

$$X^1_{S_e^{\prec}} := \{e\} \quad \text{and} \quad X^1_{\hat{S}_e^{\prec}} := \bigcup \{A \cap D \mid A \in \mathcal{M}_e, D \notin \mathcal{M}_e\}. \tag{6.4}$$

Now we can recursively define the sets $X^k_{\mathtt{S}}$ for $\mathtt{S} \in \mathcal{S}$ and $k > 2$. For $\mathtt{S} = S_e^{\prec}$ we let $\hat{\mathtt{S}} := \hat{S}_e^{\prec}$, and for $\mathtt{S} = \hat{S}_e^{\prec}$ we let $\hat{\mathtt{S}} := S_e^{\prec}$. Then $\mathcal{D}^k_{\mathtt{S}} := \{(M, N) \in \mathcal{M}_e^2 \mid \exists v \in X^k_{\hat{\mathtt{S}}} \colon v \in M \triangle N\}$, and we define

$$X^{k+1}_{\mathtt{S}} := X^k_{\mathtt{S}} \cup \bigcup_{(M,N) \in \mathcal{D}^k_{\mathtt{S}}} (M \cap N) \setminus X^k_{\hat{\mathtt{S}}}. \tag{6.5}$$

First, we show that $X^k_{\mathtt{S}}$ is a subset of $\mathtt{S} \in \mathcal{S}$ for all $k \geq 1$.

Lemma 106. *Let $e \in F_{\text{in}}$ and let $\prec$ be an arbitrary strict partial order for G. Then $X^k_{\mathtt{S}} \subseteq \mathtt{S}$ for all $\mathtt{S} \in \mathcal{S}$ and all $k \geq 1$.*

Proof. We show Lemma 106 by induction. $X^1_{S_e^{\prec}} \subseteq S_e^{\prec}$ since $X^1_{S_e^{\prec}} = \{e\}$, and $X^1_{\hat{S}_e^{\prec}} \subseteq \hat{S}_e^{\prec}$ follows from Lemma 105. Thus, $X^1_{\mathtt{S}} \subseteq \mathtt{S}$ for $\mathtt{S} \in \mathcal{S}$.

Let $k \geq 1$. The set $X^k_{\mathtt{S}}$ is a subset of $\mathtt{S} \in \mathcal{S}$ by inductive assumption. In order to show that $X^{k+1}_{\mathtt{S}} \subseteq \mathtt{S}$ for each $\mathtt{S} \in \mathcal{S}$, let us consider pairs $(M, N) \in \mathcal{D}^k_{\mathtt{S}}$, that is, max cliques $M, N \in \mathcal{M}_e$ such that there exists a vertex $v \in X^k_{\hat{\mathtt{S}}}$ with $v \in M \triangle N$. We need to show that $(M \cap N) \setminus X^k_{\hat{\mathtt{S}}}$ is a subset of $\mathtt{S}$. Without loss of generality, let $v \in M \setminus N$. For a contradiction, let us assume there exists an $a \in (M \cap N) \setminus X^k_{\hat{\mathtt{S}}}$ that is not in $\mathtt{S}$. Thus, $a \in (M \cap N) \setminus \mathtt{S}$. Now, let us consider the vertex $v \in X^k_{\hat{\mathtt{S}}}$. Let $m \leq k$ be minimal with $v \in X^m_{\hat{\mathtt{S}}}$.

If $m = 1$ and $\mathtt{S} = S_e^{\prec}$, then $v \in D \cap A'$ for max cliques $A' \in \mathcal{M}_e$ and $D \notin \mathcal{M}_e$. Thus, $v \in (D \cap A' \cap M) \setminus N$. As $v \in \hat{S}_e^{\prec}$ by inductive assumption, we can apply Lemma 104 and obtain $(M \cap N) \setminus S_e^{\prec} \subset (D \cap A') \setminus S_e^{\prec}$. Consequently, $a \in (D \cap A') \setminus S_e^{\prec}$. Since $(D \cap A') \subseteq X^1_{\hat{S}_e^{\prec}} \subseteq X^k_{\hat{S}_e^{\prec}}$, we have $a \in X^k_{\hat{S}_e^{\prec}}$, which is a contradiction to $a \in (M \cap N) \setminus X^k_{\hat{S}_e^{\prec}}$.

If $m = 1$ and $\mathtt{S} = \hat{S}_e^{\prec}$, then $v = e$. Since $N \in \mathcal{M}_e$, we have $v \in N$, a contradiction.

Now let $m > 1$. Then there exist max cliques $B, C \in \mathcal{D}^{m-1}_{\hat{\mathtt{S}}}$ such that $(B \cap C) \setminus X^{m-1}_{\mathtt{S}} \subseteq X^m_{\hat{\mathtt{S}}}$ and $v \in (B \cap C) \setminus X^{m-1}_{\mathtt{S}}$. Since $X^{m-1}_{\mathtt{S}} \subseteq \mathtt{S}$ by inductive assumption and $X^m_{\hat{\mathtt{S}}} \subseteq X^k_{\hat{\mathtt{S}}}$, we have $(B \cap C) \setminus \mathtt{S} \subseteq X^k_{\hat{\mathtt{S}}}$. Further, we have $B, C \in \mathcal{M}_e$, $v \in (B \cap C \cap M) \setminus N$, and $v \in \hat{\mathtt{S}}$ by inductive assumption. Therefore, we can apply Lemma 104, and we obtain $(M \cap N) \setminus \mathtt{S} \subset (B \cap C) \setminus \mathtt{S}$. As $a \in (M \cap N) \setminus \mathtt{S}$ and $(B \cap C) \setminus \mathtt{S} \subseteq X^k_{\hat{\mathtt{S}}}$, vertex a must be in $X^k_{\hat{\mathtt{S}}}$, a contradiction. □

As $X^k_{\mathtt{S}} \subseteq X^{k+1}_{\mathtt{S}}$ for all $\mathtt{S} \in \mathcal{S}$ and $k \geq 1$, there exists an l such that $(X^l_{\mathtt{S}}, X^l_{\mathtt{S}}) = (X^{l+1}_{\mathtt{S}}, X^{l+1}_{\mathtt{S}})$. Then $(X^l_{\mathtt{S}}, X^l_{\mathtt{S}}) = (X^{l'}_{\mathtt{S}}, X^{l'}_{\mathtt{S}})$ for all $l' > l$. Let l be minimal with that property.

Proposition 107. *Let $e \in F_{\text{in}}$. For all strict partial orders $\prec$ for G, we have $X^l_{\mathtt{S}} = \mathtt{S}$ for each $\mathtt{S} \in \{S^{\prec}_e, \hat{S}^{\prec}_e\}$.*

In order to show Proposition 107 we need the following claim.

Claim 108. *Let $e \in F_{\text{in}}$. Let $A, B \in \mathcal{M}_e$ be max cliques, and let $x \in A \cap B$. If there exists an $a \in A \triangle B$ such that $a \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$, then $x \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$.*

Proof. Let $e \in F_{\text{in}}$. Let $A, B \in \mathcal{M}_e$ and $x \in A \cap B$. Let $a \in A \triangle B$ and let there be an $\mathtt{S} \in \mathcal{S}$ such that $a \in X^l_{\hat{\mathtt{S}}}$. Then $(A, B) \in \mathcal{D}^l_{\mathtt{S}}$, and $(A \cap B) \setminus X^l_{\hat{\mathtt{S}}} \subseteq X^{l+1}_{\mathtt{S}}$. Due to the choice of l, we have $(A \cap B) \setminus X^l_{\hat{\mathtt{S}}} \subseteq X^l_{\mathtt{S}}$. Thus, if $x \in A \cap B$ is not in $X^l_{\hat{\mathtt{S}}}$, then it must be in $X^l_{\mathtt{S}}$. Consequently, $x \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$. ☐

Proof of Proposition 107. Let $e \in F_{\text{in}}$, and let $\prec$ be a strict partial order for G. As $X^l_{\mathtt{S}} \subseteq \mathtt{S}$ for $\mathtt{S} \in \mathcal{S}$ according to Lemma 106 and $S^{\prec}_e, \hat{S}^{\prec}_e$ are disjoint sets by Observation 100, it suffices to show that $S^{\prec}_e \cup \hat{S}^{\prec}_e \subseteq X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$. Thus, let $x \in S^{\prec}_e \cup \hat{S}^{\prec}_e$. We need to show that $x \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$.

For $x \in S^{\prec}_e \cup \hat{S}^{\prec}_e$ there exists a max clique $A \in \mathcal{M}_e$ and a max clique $B \in \mathcal{M} \setminus \{A\}$ such that $x \in A \cap B$ by Observation 98. If $B \notin \mathcal{M}_e$, then $x \in X^1_{\hat{S}^{\prec}_e}$, and therefore, $x \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$. In the following let $B \in \mathcal{M}_e$.

If there is an $a \in A \triangle B$ such that $a \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$, then $x \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$ according to Claim 108. Thus, we only need to consider the case where there does not exist an $a \in A \triangle B$ with $a \in X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}$, that is, the case where $(A \triangle B) \cap (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}) = \emptyset$. In the following we show that this case leads to a contradiction.

We inductively construct an infinite sequence $(\mathcal{M}_i)_{i \in \mathbb{N}}$ of sets of max cliques with $\mathcal{M}_i \subseteq \mathcal{M}_e$ for all $i \geq 0$ such that $U_i := \{v \in A \mid A \in \mathcal{M}_i\}$ and $K_i := U_i \setminus (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e})$ and the following properties are satisfied for each $i \geq 0$:

1. For all $M, N \in \mathcal{M}_i$ we have $M \cap (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}) = N \cap (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e})$, and
2. $|K_{i+1}| > |K_i|$ and $|K_0| \geq 2$.

Obviously, it holds that $U_i \subseteq U_e$ and $K_i \subseteq K := U_e \setminus (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e})$ for all $i \geq 0$. Thus, such a sequence induces a contradiction, since $|K_{|K|}| > |K|$ but $K_{|K|} \subseteq K$.

Let $\mathcal{M}_0 := \{A, B\}$. First we show that the set $\mathcal{M}_0$ satisfies the two properties: Since $(A \triangle B) \cap (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}) = \emptyset$, the set $\mathcal{M}_0 = \{A, B\}$ satisfies Property 1. Further, we have $|A \triangle B| \geq 2$ as A and B are distinct max cliques. Therefore, it follows from $(A \triangle B) \cap (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e}) = \emptyset$ that $|K_0| = |(A \cup B) \setminus (X^l_{S^{\prec}_e} \cup X^l_{\hat{S}^{\prec}_e})| \geq 2$. Thus, Property 2 is satisfied for $\mathcal{M}_0$.

Now let $i \geq 0$ and let $\mathcal{M}_0, \ldots, \mathcal{M}_i$ be a sequence of subsets of $\mathcal{M}_e$ such that for all $j \in [0, i]$ the above properties are satisfied. In the following we construct a set $\mathcal{M}_{i+1} \subseteq \mathcal{M}_e$ so that for $j = i + 1$ the two properties are satisfied as well.

Since $e \in X^l_{S^{\prec}_e}$, we have $e \notin K_i$. Thus, $|V| > |K_i| \geq |K_0| \geq 2$. As K_i cannot be a non-trivial module, there must be a vertex $c \in V \setminus K_i$ and vertices $p, q \in K_i$ with $\{p, c\} \in E$ and

$\{q,c\} \notin E$. Vertex c cannot be contained in $U_i \setminus K_i$: Assume $c \in U_i \setminus K_i = U_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$. Then, since $c, q \in U_i$, there exist two max cliques, M_c and M_q, in $\mathcal{M}_i$ that contain c and q, respectively. Now, $M_c \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}) = M_q \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$ according to Property 1, and therefore, $c \in M_c \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$ is an element in M_q, which implies $c = q$ or $\{c,q\} \in E$, a contradiction. Hence, $c \notin U_i \setminus K_i$, that is, $c \in V \setminus U_i$.

Let $A_i \in \mathcal{M}$ be a max clique containing p and c. As $p \in K_i \subseteq U_i$ there is a max clique $B_i \in \mathcal{M}_i \subseteq \mathcal{M}_e$ such that $p \in B_i$. Obviously, $A_i \notin \mathcal{M}_i$, $c \notin B_i$ and $B_i \neq A_i$. If $A_i \notin \mathcal{M}_e$, then $p \in X^1_{\hat{S}_e^\prec} \subseteq X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}$ by the definition of $X^1_{\hat{S}_e^\prec}$, which cannot be as $p \in K_i$. Thus, $A_i \in \mathcal{M}_e$.

We define $\mathcal{M}_{i+1} := \mathcal{M}_i \cup \{A_i\}$. Then the two properties are satisfied for $j = i+1$: To show Property 1 let us assume $A_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}) \neq B_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$. Then there exists a vertex a such that $a \in A_i \triangle B_i$ and $a \in X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}$ By Claim 108 we have $p \in X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}$, which is a contradiction to $p \in K_i$. As a consequence, $A_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}) = B_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$, and by inductive assumption Property 1 implies $A_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}) = M \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$ for all $M \in \mathcal{M}_i$. To show Property 2, note that $c \notin K_i$ and that $c \in A_i \setminus B_i$. Thus, vertex c cannot be in $X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}$ as we have shown $A_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec}) = B_i \cap (X^l_{S_e^\prec} \cup X^l_{\hat{S}_e^\prec})$. Consequently, $c \in K_{i+1} \setminus K_i$ and $|K_{i+1}| > |K_i|$. □

Corollary 109. *There exist sets $S_e, \hat{S}_e \subseteq V$ of vertices of G such that $S_e = S_e^\prec$ and $\hat{S}_e = \hat{S}_e^\prec$ for all strict partial orders $\prec$ for G.*

To emphasize that the $\prec$-side $S_e^\prec$ and the $\prec$-counterside $\hat{S}_e^\prec$ do not depend on the strict partial order $\prec$ for G, we from now on denote $S_e^\prec$ and $\hat{S}_e^\prec$ by S_e and $\hat{S}_e$, respectively. We call S_e the *side* and $\hat{S}_e$ the *counterside* of $e \in F_{\text{in}}$. Notice that then $O_e = U_e \setminus (S_e \cup \hat{S}_e)$. We use the inductive definition of the two sets S_e and $\hat{S}_e$ via the sets $X^k_{S_e^\prec}$ and $X^k_{\hat{S}_e^\prec}$ for $k \geq 0$ to define S_e and $\hat{S}_e$ in FP for $e \in F_{\text{in}}$. We let

$$\varphi_S^{\text{in}}(x^*, x') := \varphi_{F_{\text{in}}}(x^*) \wedge \text{ifp}\begin{pmatrix} X_1(x_1) \leftarrow \varphi_1 \vee \varphi_{\text{rec}}(x^*, x_1, X_1, X_2) \\ X_2(x_2) \leftarrow \varphi_2 \vee \varphi_{\text{rec}}(x^*, x_2, X_2, X_1)\end{pmatrix}(x')$$

where

$$\begin{aligned}\varphi_1 &:= x_1 = x^*, \\ \varphi_2 &:= \exists z_1, z_2, z_1', z_2' \; \big(\varphi_{\mathcal{M}}(z_1, z_2, x_2) \wedge \varphi_{\mathcal{M}}(z_1', z_2', x_2) \wedge \\ &\qquad\qquad \varphi_{\mathcal{M}}(z_1, z_2, x^*) \wedge \neg\varphi_{\mathcal{M}}(z_1', z_2', x^*)\big), \quad \text{and}\end{aligned}$$

$$\begin{aligned}\varphi_{\text{rec}}(x^*, x, X_{\mathtt{S}}, X_{\hat{\mathtt{S}}}) := \exists y_1, y_2, y_1', y_2', y \; \big(&\varphi_{\mathcal{M}}(y_1, y_2, x^*) \wedge \varphi_{\mathcal{M}}(y_1', y_2', x^*) \wedge \\ &X_{\hat{\mathtt{S}}}(y) \wedge \varphi_{\triangle}(y_1, y_2, y_1', y_2', y) \wedge \\ &\varphi_{\mathcal{M}}(y_1, y_2, x) \wedge \varphi_{\mathcal{M}}(y_1', y_2', x) \wedge \neg X_{\hat{\mathtt{S}}}(x)\big).\end{aligned}$$

Simultaneous IFP-formula $\varphi_S^{\text{in}}(x^*, x')$ first uses the formula $\varphi_{F_{\text{in}}}$ from Corollary 96 to check whether the vertex for x^* is an inner end e. Then it recursively determines whether the vertex for x' belongs to S_e. It is not hard to see that the sets X_1^k and X_2^k for $k \geq 1$

occurring in the recursion when interpreting the simultaneous fixed-point operator of $\varphi_S^{\text{in}}(x^*, x')$ match the previously defined sets $X^k_{S_e^\prec}$ and $X^k_{\hat{S}_e^\prec}$:

Formulas φ_1 and φ_2 ensure that $X_1^1 = X^1_{S_e^\prec}$ and $X_2^1 = X^1_{\hat{S}_e^\prec}$, respectively. Formula φ_1 guaranties that X_1^1 (only) contains the inner end e. Formula φ_2 makes sure that the set X_2^1 contains all vertices that are both in a max clique $A \in \mathcal{M}_e$ and in a max clique $A' \notin \mathcal{M}_e$. Remember that the formula $\varphi_{\mathcal{M}}(z_1, z_2, x)$ (see (2.3) in Section 2.8.2) is satisfied for $(v_1, v_2, w) \in V^3$ precisely if v_1, v_2 is a spanning pair of a max clique A and $w \in A$.

Due to formula φ_{rec} all necessary vertices are added in each round of the recursion: If $e \in F_{\text{in}}$, $w \in V$ and the sets X^k_{S} and $X^k_{\hat{\mathsf{S}}}$ for $\mathsf{S} \in \{S_e^\prec, \hat{S}_e^\prec\}$ and $k \geq 0$ are the sets defined recursively in (6.4) and (6.5), then $(e, w, X^k_{\mathsf{S}}, X^k_{\hat{\mathsf{S}}})$ satisfies formula $\varphi_{\text{rec}}(x^*, x, X_{\mathsf{S}}, X_{\hat{\mathsf{S}}})$ if and only if $w \in \bigcup_{(M,N) \in \mathcal{D}^k_{\mathsf{S}}} (M \cap N) \setminus X^k_{\hat{\mathsf{S}}}$. The variables y_1, y_2 and y_1', y_2' can only be interpreted by spanning pairs of two max cliques N and M in $\mathcal{M}_e$, respectively. Then each possible value for variable y is a vertex $v \in X^k_{\hat{\mathsf{S}}}$ such that $v \in M \bigtriangleup N$. Subformula $\varphi_\bigtriangleup$ expresses the symmetric difference. Thus, $\varphi_\bigtriangleup(y_1, y_2, y_1', y_2', y)$ is satisfied for $(a_1, a_2, a_1', a_2', a) \in V^5$ if, and only if, a_1 and a_2 span a max clique A, a_1' and a_2' span a max clique A' and $a \in A \bigtriangleup A'$. Clearly, formula $\varphi_\bigtriangleup$ is definable in FO. Hence, the variables y_1, y_2 and y_1', y_2' are to be interpreted by spanning pairs for two max cliques N and M with $(N, M) \in \mathcal{D}^k_{\mathsf{S}}$. Then, it is not hard to see that $\varphi_{\text{rec}}(x^*, x, X_{\mathsf{S}}, X_{\hat{\mathsf{S}}})$ is satisfied for a tuple $(e, w, X^k_{\mathsf{S}}, X^k_{\hat{\mathsf{S}}})$ as defined above if and only if $w \in \bigcup_{(M,N) \in \mathcal{D}^k_{\mathsf{S}}} (M \cap N) \setminus X^k_{\hat{\mathsf{S}}}$.

Symmetrically, we can define $\hat{S}_e$ for inner ends $e \in F_{\text{in}}$:

$$\varphi_{\hat{S}}^{\text{in}}(x^*, x') := \varphi_{F_{\text{in}}}(x^*) \wedge \text{ifp}\begin{pmatrix} X_2(x_2) \leftarrow \varphi_2 \wedge \varphi_{\text{rec}}(x^*, x_2, X_2, X_1) \\ X_1(x_1) \leftarrow \varphi_1 \wedge \varphi_{\text{rec}}(x^*, x_1, X_1, X_2) \end{pmatrix}(x').$$

As a consequence we obtain the following result:

Corollary 110. *There exists an* FP*-formula* $\varphi_S^{\text{in}}(x^*, x')$ *(or* $\varphi_{\hat{S}}^{\text{in}}(x^*, x')$*) that is satisfied for vertices* $e, a \in V$ *in a graph* G *if, and only if,* e *is an inner end of* G *and* $a \in S_e$ *(or* $a \in \hat{S}_e$*).*

6.3.7. Outer Ends

Now we can use Corollary 109 to show that the outer $\prec$-ends do not depend on the strict partial order $\prec$ for G and that we can define them in FP.

Let $\prec$ be an arbitrary strict partial order for G. By Lemma 82 we know that each outer $\prec$-end is a $\prec$-end of an outer max clique. This is why we consider outer max cliques in this section. Each outer max clique has an outer and an inner $\prec$-end according to Corollary 88. The inner $\prec$-end does not depend on the strict partial order $\prec$ of G (Corollary 95).

Let A be an outer max clique, and let e be the inner end of A. Then $A \in \mathcal{M}_e$, and the set U_e is the disjoint union of S_e, O_e and $\hat{S}_e$. We let $V_e := S_e \cup O_e$ and $\hat{V}_e := \hat{S}_e \cup O_e$. Since S_e, $\hat{S}_e$ and O_e are FP-definable for inner ends e (shown in the previous two sections), V_e and $\hat{V}_e$ are definable in FP as well. We use the set $\hat{V}_e$ to show that the outer $\prec$-end of A does not depend on the strict partial order $\prec$ for G. We let

$$E'(A) := \bigcap \{D \cap \hat{V}_e \mid D \in \mathcal{M} \text{ and } A \cap D \cap \hat{V}_e \neq \emptyset\}. \tag{6.6}$$

Proposition 111. *Let $A \in \mathcal{M}$ be an outer max clique of G. Then $|E'(A)| = 1$. More precisely, for every strict partial order $\prec$ for G, which induces the order $a_1, \dots, a_m$ on the elements of A, we have either $E'(A) = \{a_1\}$ or $E'(A) = \{a_m\}$, where $E'(A)$ contains the outer $\prec$-end of A.*

Proof. The proof is similar to the proof of Proposition 93. Let $\prec$ be an arbitrary strict partial order for G. Let $A = (a_1, \dots, a_m)_\prec$ be an outer max clique of G. Without loss of generality, let a_1 be the inner end e of A. We show that the outer end a_m of A is in $E'(A)$. Vertex a_m must be in $\hat{V}_e$ as $a_m \notin S_1^\prec(e) = S_e$ (Lemma 102). Let $\mathcal{D} := \{D \in \mathcal{M} \mid A \cap D \cap \hat{V}_e \neq \emptyset\}$. Then $A \in \mathcal{D}$ and $\mathcal{D} \neq \emptyset$. Let us assume there exists a max clique $D \in \mathcal{D}$ with $a_m \notin D$. Then $D \neq A$. Let d be an element of the non-empty set $A \cap D \cap \hat{V}_e$. As $a_m \notin D$, Corollary 70 implies that the intersection of A and D is of form (a). Consequently, $A \cap D = V_1^\prec(A, D)$. It follows that $d \in A \cap D = V_1^\prec(A, D) \subseteq V_1^\prec(A) \subseteq S_1^\prec(e) = S_e$, which is a contradiction since $d \in \hat{V}_e$. We obtain that a_m must be contained in all max cliques $D \in \mathcal{D}$. Hence, $a_m \in E'(A)$. As a result, $|E'(A)| \geq 1$. Since $e \in S_e$, we have $e \notin \hat{V}_e$, and therefore $e \notin E'(A)$. Thus, $|E'(A)| < |V|$. We can show that $E'(A)$ is a module in the same way as in the proof of Proposition 93. Since there are no non-trivial modules, it follows that $|E'(A)| = 1$. □

Corollary 112. *Let A be an outer max clique of a graph G. Then there exists a vertex $f \in A$ such that f is the outer $\prec$-end of A for all strict partial orders $\prec$ for G.*

Let $\prec$ be an arbitrary strict partial order for G. Since each outer $\prec$-end is a $\prec$-end of an outer max clique by Lemma 82, we can conclude the following:

Corollary 113. *There exists a set $F_{\text{out}} \subseteq V$ of vertices of G such that $F_{\text{out}} = F_{\text{out}}^\prec$ for all strict partial orders $\prec$ for G.*

We summarize the results of Corollary 95 and 113 in the following corollary.

Corollary 114. *There exists a set $F \subseteq V$ of vertices of G such that $F = F^\prec$ for all strict partial orders $\prec$ for G.*

As a consequence, we can omit the "$\prec$" in the notation of the outer $\prec$-ends and $\prec$-ends in general. From now on, we simply write *outer ends* and denote $F_{\text{out}}^\prec$ by F_{out}. Further, we use F to denote $F^\prec$, the set $F_{\text{in}} \cup F_{\text{out}}$ of all *ends*.

From the description of the construction of formula $\varphi_{F_{\text{in}}}(x^*)$ in Section 6.3.4, it should be clear how to use the definition of $E'(A)$ in (6.6) for the construction of an FP-formula $\varphi_{F_{\text{out}}}(x^*)$ that is satisfied for prime chordal comparability graphs G and vertices $e \in V$ if and only if e is an outer end of G. In combination with Corollary 96, we obtain the following result:

Corollary 115. *There exist FP-formulas $\varphi_{F_{\text{in}}}(x^*)$, $\varphi_{F_{\text{out}}}(x^*)$ and $\varphi_F(x^*)$ that are satisfied for a vertex $e \in V$ in a prime chordal comparability graph G if, and only if, e is an inner end, an outer end and an end of G, respectively.*

Since the $\prec$-ends do not depend on the strict partial order $\prec$ for G, we obtain that the $\prec$-bundle tree $G[F^\prec]$ does not depend on the strict partial order $\prec$ for prime chordal comparability graphs G according to Corollary 114. We call $G[F]$ the *bundle tree* of prime graph G. Since there is an FP-formula $\varphi_F(x^*)$ that defines the unique set F of ends (Corollary 115), the bundle tree $G[F]$ is FP-definable for prime chordal comparability graphs. Further, 2-colorings of connected bipartite graphs are definable in symmetric transitive closure logic.[1] Thus, a 2-coloring $\{F_1, F_2\}$ of the tree $G[F]$ is definable in FP.

Corollary 116. *There exists an FP-formula $\varphi_{F,\approx}(x^*, y^*)$ such that for all vertices e and f of a prime chordal comparability graph G we have*

$$G \models \varphi_{F,\approx}(e,f) \iff \text{e and f are in the same color class of the 2-coloring $\{F_1,F_2\}$ of $G[F]$.}$$

6.3.8. Sides of Outer Ends

For all inner ends $e \in F_{\text{in}}$ we have already defined the side S_e and counterside $\hat{S}_e$ of e. Now let us define the side and counterside of outer ends.

Let $f \in F_{\text{out}}$ be an outer end of G. Then f is a leaf of the bundle tree $G[F]$. By Corollary 86, f has a unique neighbor $e \in F$, which is an inner end, in $G[F]$. Since end e is the only end adjacent to f, all max cliques that have f as an end must also have e as an end. Thus, $\mathcal{M}_f \subseteq \mathcal{M}_e$. We use $\hat{S}_e$ and S_e to define the *side* S_f and the *counterside* $\hat{S}_f$ of outer end f. We let

$$S_f := \hat{S}_e \cap U_f \text{ and } \hat{S}_f := S_e \cap U_f. \tag{6.7}$$

Figure 6.10 shows the intersection of the set U_f with $\hat{S}_e$ and S_e.

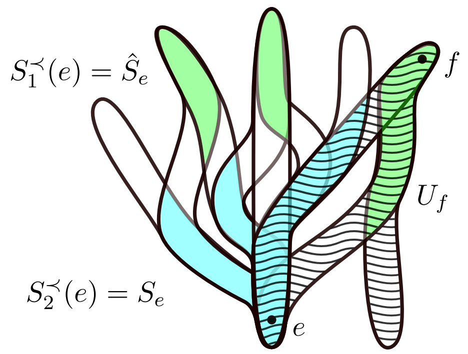


Figure 6.10.: The sets $\hat{S}_e$, S_e and U_f

The side S_f and counterside $\hat{S}_f$ of outer ends $f \in F_{\text{out}}$ are definable in FP: We can define in fixed-point logic all pairs of vertices f and e where f is an outer end and e is an

[1] We can test whether two vertices a and b have to be colored with the same color in a connected bipartite graph $H = (V, E)$ by checking whether a and b are connected in the graph $H' = (V, E')$ where $E' := \big\{\{v, w\} \in \binom{V}{2} \big| \exists z \in V : \{v, z\}, \{z, w\} \in E\big\}$.

inner end that is adjacent to f (Corollary 115). Further, U_f is FO-definable, and the side S_e and counterside $\hat{S}_e$ of each inner end $e \in F_{\text{in}}$ are definable in fixed-point logic (Corollary 110). Thus, there exists an FP-formula that defines S_f and $\hat{S}_f$ for outer ends $f \in F_{\text{out}}$. This and Corollary 110 yield the following corollary.

Corollary 117. *There exists an* FP*-formula* $\varphi_S(x^*, x)$ *(or* $\varphi_{\hat{S}}(x^*, x)$*) that is satisfied for vertices* $e, a \in V$ *in* G *if, and only if,* $e \in F$ *and* $a \in S_e$ *(or* $a \in \hat{S}_e$*).*

In the following we present the connection between S_f and $\hat{S}_f$, and $S_1^{\prec}(f)$ and $S_2^{\prec}(f)$ for $f \in F_{\text{out}}$ where $\prec$ is the strict partial order for G.

Proposition 118. *Let* $\prec$ *be a strict partial order for* G*, and let* $f \in F_{\text{out}}$*. There exists exactly one* $i \in [2]$ *such that* $f \in S_i^{\prec}(f)$ *or* $S_i^{\prec}(f) = \emptyset$*, and for this* $i \in [2]$ *we have*

$$S_f = S_i^{\prec}(f) \text{ and } \hat{S}_f = S_{3-i}^{\prec}(f).$$

To prove Proposition 118 we need the three subsequent lemmas.

Lemma 119. *Let* $\prec$ *be a strict partial order for* G*, and let* $f \in F_{\text{out}}$*. Let* $i \in [2]$ *be such that* $f \in S_i^{\prec}(f)$ *or* $S_i^{\prec}(f) = \emptyset$*. Then* $S_f = S_i^{\prec}(f)$ *and* $\hat{S}_f = S_{3-i}^{\prec}(f)$*.*

Proof. Let $\prec$ be a strict partial order for G, and let $f \in F_{\text{out}}$. Let $e \in F_{\text{in}}$ be the unique neighbor of end f in the bundle tree $G[F]$ of graph G. We first prove that $S_j^{\prec}(f) = S_j^{\prec}(e) \cap U_f$ for all $j \in [2]$: Since $\mathcal{M}_f \subseteq \mathcal{M}_e$ and $V_j^{\prec}(A) \subseteq A$ for all $A \in \mathcal{M}$, we obtain $\bigcup_{A \in \mathcal{M}_f} V_j^{\prec}(A) \subseteq \bigcup_{A \in \mathcal{M}_e} V_j^{\prec}(A)$ and $\bigcup_{A \in \mathcal{M}_f} V_j^{\prec}(A) \subseteq U_f$, respectively. Consequently, $S_j^{\prec}(f) = \bigcup_{A \in \mathcal{M}_f} V_j^{\prec}(A)$ is a subset of $\bigcup_{A \in \mathcal{M}_e} V_j^{\prec}(A) \cap U_f = S_j^{\prec}(e) \cap U_f$. Next let $v \in S_j^{\prec}(e) \cap U_f$. Then there exists a max clique $A \in \mathcal{M}_f$ such that $v \in A$. As $v \in S_j^{\prec}(e)$, we have $v \in V_j^{\prec}(A)$ according to Observation 103. Thus, $v \in \bigcup_{A \in \mathcal{M}_f} V_j^{\prec}(A) = S_j^{\prec}(f)$, and $S_j^{\prec}(e) \cap U_f \subseteq S_j^{\prec}(f)$.

Now let $S_i^{\prec}(f) = \emptyset$. Then $S_i^{\prec}(e) \cap U_f = \emptyset$, and inner end e cannot be in $S_i^{\prec}(e)$ because $e \in U_f$. Thus, $S_i^{\prec}(e) = \hat{S}_e$ and $S_{3-i}^{\prec}(e) = S_e$. Hence, $S_i^{\prec}(f) = \hat{S}_e \cap U_f$ and $S_{3-i}^{\prec}(f) = S_e \cap U_f$. If $f \in S_i^{\prec}(f)$, then $f \in S_i^{\prec}(e)$, and Lemma 102 yields $e \notin S_i^{\prec}(e)$. Again, we have $S_i^{\prec}(e) = \hat{S}_e$ and $S_{3-i}^{\prec}(e) = S_e$, which results in $S_i^{\prec}(f) = S_f$ and $S_{3-i}^{\prec}(f) = \hat{S}_f$. □

The following two lemmas help us to gain a better understanding of the sets $S_1^{\prec}(f)$ and $S_2^{\prec}(f)$ for outer ends f. Further, they show that there must exist an $i \in [2]$ such that $f \in S_i^{\prec}(f)$ or $S_i^{\prec}(f) = \emptyset$ for $f \in F_{\text{out}}$.

Lemma 120. *Let* $f \in F_{\text{out}}$*, and* $\prec$ *be a strict partial order for* G*. Then* $f \in S_i^{\prec}(f)$ *for an* $i \in [2]$ *if and only if* $|\mathcal{M}_f| > 1$*.*

Proof. Let $f \in F_{\text{out}}$, and $\prec$ be a strict partial order for G. Let $f \in S_i^{\prec}(f)$ for $i \in [2]$. Then there exist a max clique $A \in \mathcal{M}_f$ such that $f \in V_i^{\prec}(A)$, and thus, there also exists a max clique $B \neq A$ such that $f \in V_i^{\prec}(A, B) \subseteq A \cap B$. Hence, $|\mathcal{M}_f| > 1$.

If $|\mathcal{M}_f| > 1$, there exist max cliques $A, B \in \mathcal{M}_f$ with $A \neq B$ and $f \in A \cap B$. It follows from Observation 98 that there is an $i \in [2]$ such that $f \in S_i^{\prec}(f)$. □

Lemma 121. *Let $f \in F_{\text{out}}$ be an outer end, and let $\prec$ be a strict partial order for G. Then there exists an $i \in [2]$ such that $S_i^{\prec}(f) = \emptyset$ if and only if $|\mathcal{M}_f| = 1$.*

Proof. Let $f \in F_{\text{out}}$ be an outer end, and let $\prec$ be a strict partial order for G. To show the first direction, let $S_i^{\prec}(f) = \emptyset$ for an $i \in [2]$. Let us suppose $|\mathcal{M}_f| > 1$, and let e be the inner end adjacent to f in $G[F]$. Then ends e and f are contained in at least two max cliques from $\mathcal{M}_f$, and by Observation 98 each end is in $S_1^{\prec}(f)$ or $S_2^{\prec}(f)$. Since $S_i^{\prec}(f) = \emptyset$, we have $e, f \in S_{3-i}^{\prec}(f)$, which is a contradiction to Lemma 102.

For the other direction, let $|\mathcal{M}_f| = 1$, and let $A = (a_1, \ldots, a_m)_{\prec}$ be the only max clique in $\mathcal{M}_f$. Then $S_i^{\prec}(f) = V_i^{\prec}(A)$ for each $i \in [2]$. Further, let us assume $S_i^{\prec}(f) \neq \emptyset$ for both $i \in [2]$. Then, we must have $a_1 \in V_1^{\prec}(A)$ and $a_m \in V_2^{\prec}(A)$ (Observation 73). As $f = a_1$ or $f = a_m$, there exists a $j \in [2]$ such that $f \in V_j^{\prec}(A) \subseteq S_j^{\prec}(f)$, which according to Lemma 120 is a contradiction. □

Let $f \in F_{\text{out}}$. Lemma 119 states that for all $i \in [2]$ with $f \in S_i^{\prec}(f)$ or $S_i^{\prec}(f) = \emptyset$, we have $S_f = S_i^{\prec}(f)$ and $\hat{S}_f = S_{3-i}^{\prec}(f)$. Further, Lemma 120 and 121 show that there exists an $i \in [2]$ such that $f \in S_i^{\prec}(f)$ or $S_i^{\prec}(f) = \emptyset$. In order to show Proposition 118, it remains to prove that there exists only one $i \in [2]$ like that:

Proof of Proposition 118. Let us assume we have $f \in S_i^{\prec}(f)$ or $S_i^{\prec}(f) = \emptyset$ for both $i \in [2]$. According to Lemma 119 we have $S_f = S_1^{\prec}(f) = S_2^{\prec}(f)$ and $\hat{S}_f = S_2^{\prec}(f) = S_1^{\prec}(f)$. Since $S_1^{\prec}(f)$ and $S_2^{\prec}(f)$ are disjoint sets of vertices by Observation 100, it follows that $S_1^{\prec}(f) = S_2^{\prec}(f) = \emptyset$. Let e be the inner end adjacent to f in $G[F]$. Then e is contained in a max clique A in $\mathcal{M}_f \subseteq \mathcal{M}_e$, which is an outer max clique by Lemma 82. As e is an inner end, it is contained in a further (inner) max clique (Lemma 89). According to Observation 98, $e \in S_{j'}^{\prec}(f)$ for a $j' \in [2]$, a contradiction. □

6.3.9. Sides and the Middle of a Graph

For each end $e \in F$ we call the set S_e a *side* of G. Further, we let $O := V \setminus \bigcup\{S_e \mid e \in F\}$ be the *middle* of G. Remember that $S_1^{\prec}(e)$ and $S_2^{\prec}(e)$ are subsets of U_e and that O_e is the set $U_e \setminus (S_1^{\prec}(e) \cup S_2^{\prec}(e))$ where $\prec$ is an arbitrary strict partial order for G (see Section 6.3.5). By Proposition 118 we have $S_1^{\prec}(e) \cup S_2^{\prec}(e) = S_e \cup \hat{S}_e$ not only for inner ends $e \in F_{\text{in}}$ but also for outer ends $e \in F_{\text{out}}$. Thus, $S_e \cup \hat{S}_e \subseteq U_e$ and $O_e = U_e \setminus (S_e \cup \hat{S}_e)$ for all $e \in F$. We let $V_e := S_e \,\dot{\cup}\, O_e$.

According to Corollary 117 we can define S_e and $\hat{S}_e$ in fixed-point logic. Further, U_e is FO-definable. As a consequence we can construct FP-formulas $\varphi_O(x^*, x)$ and $\varphi_V(x^*, x)$ such that for each prime chordal comparability graph $G = (V, E)$ with $|V| > 2$ and all $e, v \in V$ we have

$$\begin{aligned} G \models \varphi_O[e, v] \quad &\iff \quad e \in F \text{ and } v \in O_e, \quad \text{and} \\ G \models \varphi_V[e, v] \quad &\iff \quad e \in F \text{ and } v \in V_e. \end{aligned} \tag{6.8}$$

In the following we present results regarding the sides and the middle of G.

Observation 122. *Let A and B be distinct max cliques containing a vertex v. Then there exists a common end $e \in F$ of A and B such that $v \in S_e$.*

Proof. Let $A, B \in \mathcal{M}$ with $A \neq B$ and let $v \in A \cap B$. Let $\prec$ be a strict partial order for G and let $i \in [2]$ be such that $v \in V_i^{\prec}(A, B)$ (Observation 71). Further, let e be the common end of A and B that satisfies $e \in V_i^{\prec}(A, B)$. Then $V_i^{\prec}(A, B) \subseteq V_i^{\prec}(A) \subseteq S_i^{\prec}(e)$, and $S_i^{\prec}(e) = S_e$ since $e \in S_i^{\prec}(e)$. □

Lemma 123. *Let $v \in V$. Then $v \in O$ if and only if v is contained in only one max clique.*

Proof. Let $v \in V$. In order to show the first direction, let us assume v is contained in max cliques A and B with $A \neq B$. Then $v \in S_e$ for a common end e of A and B according to Observation 122. Thus, $v \notin O$. For the other direction, suppose $v \notin O$. Then $v \in S_e$ for an end $e \in F$. Thus, $v \notin O_e$ and we can apply Observation 101. As a consequence $v \in U_e$ is contained in more than one max clique. □

Observation 124. *Let $\{F_1, F_2\}$ be the 2-coloring of $G[F]$. Then $O = \bigcup_{e \in F_i} O_e$ for each $i \in [2]$.*

Proof. Let $i \in [2]$. By Observation 101 and 123, we have $O_e \subseteq O$ for all $e \in F_i$. Hence, $\bigcup_{e \in F_i} O_e \subseteq O$. It remains to show that $O \subseteq \bigcup_{e \in F_i} O_e$. Let $v \in O$. According to Lemma 123, vertex v is in only one max clique A. Let e be the end of this max clique with $e \in F_i$. Then $v \in U_e$. Now we can apply Observation 101, and obtain that $v \in O_e$. □

Corollary 125. *There is an FP-formula $\varphi_O(x)$ that is satisfied for a vertex $v \in V$ in a prime chordal comparability graph G if, and only if, $v \in O$.*

Proof. We let $\varphi_O(z) := \exists x^* \varphi_O(x^*, z)$, where $\varphi_O(x^*, z)$ is the formula defined in (6.8). □

Observation 126. *Let $f \in F_{\text{out}}$ be an outer end. Then $f \in O$ if and only if $S_f = \emptyset$.*

Proof. Let $f \in F_{\text{out}}$ be an outer end, and let $\prec$ be a strict partial order for G. Let $f \in O$. Then f is contained in only one max clique by Lemma 123. According to Lemma 121 there is an $i \in [2]$ such that $S_i^{\prec}(f) = \emptyset$. It follows from Proposition 118 that $S_f = \emptyset$. Now let $f \notin O$. By Lemma 123, f is contained in more than one max clique. Lemma 120 implies that $f \in S_i^{\prec}(f)$ for an $i \in [2]$, and Proposition 118 yields that $f \in S_f$. □

Observation 127. *Let $\prec$ be a strict partial order for G and $e \in F$. If e is $\prec$-minimal, then $S_e = S_1^{\prec}(e)$ and $\hat{S}_e = S_2^{\prec}(e)$. If e is $\prec$-maximal, then $S_e = S_2^{\prec}(e)$ and $\hat{S}_e = S_1^{\prec}(e)$.*

Proof. Let $\prec$ be a strict partial order for G and $e \in F$. Without loss of generality, let e be $\prec$-minimal. Let us assume there exist two max cliques $A, B \in \mathcal{M}$ that contain e. Then $e \in V_1^{\prec}(A, B) \subseteq S_1^{\prec}(e)$. By the definition of S_e and $\hat{S}_e$ for inner ends (see (6.3)) and Proposition 118, we have $S_e = S_1^{\prec}(e)$ and $\hat{S}_e = S_2^{\prec}(e)$. Now let us suppose there exists only one max clique $A \in \mathcal{M}$ that contains e. Then e cannot be an inner end by Corollary 91. Thus, e is an outer end. Let us assume there exists a vertex $v \in V$ such that $v \in S_1^{\prec}(e)$. Since $v \in S_1^{\prec}(e)$, vertex v is contained in a max clique $M \in \mathcal{M}_e$. Thus, e and v are adjacent, and as e is $\prec$-minimal, we have $e \prec v$. It follows from Observation 97 that $e \in S_1^{\prec}(e)$. Then, Lemma 120 implies that $|\mathcal{M}_e| > 1$, which is a contradiction. Thus, $S_1^{\prec}(e)$ must be empty. By Proposition 118, we have $S_e = S_1^{\prec}(e)$ and $\hat{S}_e = S_2^{\prec}(e)$. $\square$

Lemma 128. *For all $e, e' \in F$ with $e \neq e'$ we have $S_e \cap S_{e'} = \emptyset$.*

Proof. Let $e, e' \in F$ with $e \neq e'$. First, let e and e' be adjacent vertices. Let $\prec$ be a strict partial order for G. Without loss of generality, let $e \prec e'$. Then Observation 76 yields that e is $\prec$-minimal and e' is $\prec$-maximal. It follows that $S_e = S_1^{\prec}(e)$ and $S_{e'} = S_2^{\prec}(e')$ by Observation 127. Let us assume there exists a vertex $v \in S_e \cap S_{e'}$. Vertex v is contained in a max clique A. By Observation 103 we have $v \in V_1^{\prec}(A)$, and $v \in V_2^{\prec}(A)$, a contradiction to Observation 72.

It remains to consider ends e and e' that are non-adjacent vertices. Let us suppose there exists a vertex $v \in S_e \cap S_{e'} \subseteq U_e \cap U_{e'}$. Let $A \in \mathcal{M}_e$ and $A' \in \mathcal{M}_{e'}$ be max cliques containing v. As e and e' are not adjacent, there does not exist a max clique B with $e, e' \in B$. Thus, we have $A \neq A'$. By Observation 122 we have $v \in S_f$ where f is a common end of A and A'. Since there does not exist a max clique that contains e and e', we have $f \neq e$ and $f \neq e'$. Then ends e and f are adjacent and $v \in S_e \cap S_f$, which is a contradiction as shown in the previous case. $\square$

Lemma 128 shows that for all $v \in V \setminus O$, there exists a unique end $e \in F$ such that $v \in S_e$.

Lemma 129. *If a vertex $v \in V \setminus O$ belongs to a max clique with ends e and e', then either $v \in S_e$ or $v \in S_{e'}$.*

Proof. Let $v \in V \setminus O$ be contained in a max clique A with ends e and e'. Since $v \notin O$, there must exist a further max cliques $B \neq A$ with $v \in A \cap B$ (Lemma 123). According to Observation 122 we have $v \in S_f$ where f is a common end of A and B. It follows that $f \in \{e, e'\}$. $\square$

Lemma 129 directly implies the following corollary.

Corollary 130. *Let A be a max clique with ends $e, f \in F$. Then $A \subseteq S_e \cup O \cup S_f$.*

Corollary 131. *Let $e \in F$. If $v \in V_e$, then every max clique A with $v \in A$ has e as an end.*

Proof. Let $e \in F$ and $v \in S_e \cup O_e$. Let us assume there exists a max clique A with $v \in A$ that does not have e as an end. Then A must have two ends f and f' different from e. By Corollary 130 $v \in A \subseteq S_f \cup O \cup S_{f'}$. If $v \in S_e$, then v is not in O, and v is also not contained in S_f or $S_{f'}$ by Lemma 128. Thus, we must have $v \in O_e$. As a consequence, there is only one max clique, namely A, that contains v (Observation 101). Since $v \in O_e \subseteq U_e$, this max clique is in $\mathcal{M}_e$. Hence, max clique A must have e as an end, a contradiction. $\square$

Lemma 132. *Let $e, f \in F$ with $e \neq f$. If there is no edge between e and f (in $G[F]$), then $V_e \cap V_f = \emptyset$.*

Proof. Let us assume there exists a vertex $v \in V_e \cap V_f$. Then $v \in O_e$ or $v \in O_f$ according to Lemma 128. By Observation 101 vertex v is contained in only one max clique $A \in \mathcal{M}$. Now we can apply Corollary 131, and obtain that e and f are the ends of A, a contradiction to e and f being non-adjacent. □

6.3.10. Side Depth and Side Trees

Throughout this section let G be a prime chordal comparability graphs with more than two vertices. We introduce the side depth and side trees in this section. For each end $e \in F$, we define a directed tree T_e, the side tree of e, which will be FP+C-definable. The vertex set of T_e is V_e and the symmetric closure of the transitive closure of the edge relation of T_e is the edge relation of the induced subgraph $G[V_e]$. We begin this section with introducing the side depth of a vertex $v \in V_e$ for an end $e \in F$. The side depth of $v \in V_e$ corresponds to the depth of v in the side tree T_e of e.

For $e \in F$ and $v \in V_e$ we define the *side depth* $\mathrm{sd}_e(v)$ of v regarding e as follows: We let

$$\mathrm{sd}_e(v) := |A_e^v \cup \{v\}|$$

where $A_e^v := \{w \in V_e \mid \{v, w\} \in E \text{ and } \exists M \in \mathcal{M}\colon w \in M, v \notin M\}$. Notice that the set A_e^v does not contain vertex v. Thus, $\mathrm{sd}_e(v) = |A_e^v| + 1$. Further, for each end $e \in F$, it follows from Corollary 131 that $A_e^e = \emptyset$. Therefore, we have $\mathrm{sd}_e(e) = 1$ for all $e \in F$.

There exists an FP+C-formula for the side depth. We let

$$\varphi_{\mathrm{sd}}(x^*, x, p) := \varphi_V(x^*, x) \wedge p = \#y \Big(\Big(\varphi_V(x^*, y) \wedge E(x, y) \wedge \exists z_1, z_2 \big(\varphi_{\mathcal{M}}(z_1, z_2, y) \wedge \neg\varphi_{\mathcal{M}}(z_1, z_2, x)\big)\Big) \vee y = x\Big), \tag{6.9}$$

where φ_V and $\varphi_{\mathcal{M}}$ are the formulas defining the set V_e for $e \in F$ and max cliques $M \in \mathcal{M}$, respectively, from (6.8) and (2.3). Then for each prime chordal comparability graph $G = (V, E)$ with $|V| > 2$, all $e, v \in V$ and all $l \in N(G)$ we have

$$G \models \varphi_{\mathrm{sd}}[e, v, l] \iff e \in F,\ v \in V_e \text{ and } \mathrm{sd}_e(v) = l \text{ in } G.$$

Lemma 133. *Let $\prec$ be a strict partial order for G. Let $e \in F$ and $v \in V_e$. If e is $\prec$-minimal (or $\prec$-maximal), then $A_e^v = \{z \in V \mid z \prec v\}$ (or $A_e^v = \{z \in V \mid v \prec z\}$).*

Proof. Let $e \in F$, $v \in V_e$ and $\prec$ be a strict partial order for G. Without loss of generality, let e be $\prec$-minimal. Then $S_e = S_1^{\prec}(e)$ and $S_2^{\prec}(e) = \hat{S}_e$ by Observation 127.

First we show that $A_e^v \subseteq \{z \in V \mid z \prec v\}$. Thus, let $w \in A_e^v$. Then $w \in V_e$, $\{v, w\} \in E$ and there exists a max clique $M \in \mathcal{M}$ such that $w \in M$ and $v \notin M$. Since $\{v, w\} \in E$, there must exist a max clique $A \in \mathcal{M}$ with $v, w \in A$. As a consequence of Observation 101, we have $w \notin O_e$. Hence, $w \in S_e = S_1^{\prec}(e)$ and by Observation 103 we have $w \in V_1^{\prec}(M)$. As $\{v, w\} \in E$, we have either $v \prec w$ or $w \prec v$. Now, $v \prec w$ implies that $v \in V_1^{\prec}(M) \subseteq M$ (Observation 73), a contradiction to $v \notin M$. Therefore, we must have $w \prec v$. Hence, w is in $\{z \in V \mid z \prec v\}$.

To prove that $A_e^v \supseteq \{z \in V \mid z \prec v\}$, let $w \in \{z \in V \mid z \prec v\}$. Thus, $w \prec v$. Then $\{v, w\} \in E$, and there exists a max clique A that contains v, w. Let us assume that there does not exist a max clique $M \in \mathcal{M}$ such that $w \in M$ and $v \notin M$, that is, $w \notin M$ or $v \in M$ for all max cliques $M \in \mathcal{M}$. Then $\{v, w\}$ is a module: First let us suppose that there exists a $z \notin \{v, w\}$ such that $\{z, v\} \notin E$ and $\{z, w\} \in E$. Then there exists a max clique $B \in \mathcal{M}$ with $z, w \in B$. Since B is a max clique with $w \in B$, we must have $v \in B$ according to our assumption. Consequently, there is an edge between z and v, a contradiction. Now let us assume that there is a $z \notin \{v, w\}$ such that $\{z, v\} \in E$ and $\{z, w\} \notin E$. Then there exists a max clique $B \in \mathcal{M}$ with $z, v \in B$. As z and w are not adjacent, vertex w cannot be in B. Thus, $B \neq A$ and $v \notin O_e$ by Observation 101. Therefore, we must have $v \in S_e = S_1^{\prec}(e)$. Then Observation 103 yields that $v \in V_1^{\prec}(B)$. Now $w \prec v$ implies that $w \in V_1^{\prec}(B) \subseteq B$ (Observation 73), a contradiction. Thus, $\{v, w\}$ is a module in G. Moreover, $\{v, w\}$ is a non-trivial module. As G is prime, our assumption was wrong and there exists a max clique $M \in \mathcal{M}$ such that $w \in M$ and $v \notin M$. It remains to show that $w \in V_e$. Since $v \in V_e$, max clique A has e as an end according to Corollary 131. Therefore, $w \in U_e$. Let us suppose that $w \in \hat{S}_e = S_2^{\prec}(e)$. Then $w \in V_2^{\prec}(A)$ (Observation 103), and from $w \prec v$ it follows that $v \in V_2^{\prec}(A)$ as well. Hence, v also is in $S_2^{\prec}(e) = \hat{S}_e$, a contradiction. □

Lemma 133 directly implies the following two corollaries:

Corollary 134. *Let $\prec$ be a strict partial order for G. Let $e \in F$ and $v \in V_e$. If e is $\prec$-minimal (or $\prec$-maximal), then $\mathrm{sd}_e(v)$ is the number of vertices $z \in V$ with $z \preceq v$ (or $v \preceq z$).*

Corollary 135. *Let $\prec$ be a strict partial order for G. If $e \in F$ is $\prec$-minimal (or $\prec$-maximal), then for all $v, v' \in V_e$ with $\{v, v'\} \in E$ we have $\mathrm{sd}_e(v) < \mathrm{sd}_e(v')$ if and only if $v \prec v'$ (or $v' \prec v$).*

For $e \in F$ we define the directed graph $T_e = (V_e, E_e)$ as follows: We let $(v, w) \in V_e^2$ be an edge of T_e if $\mathrm{sd}_e(v) = \mathrm{sd}_e(w) - 1$ and $\{v, w\} \in E$. In the following we show that T_e is a directed tree. We call T_e the *side tree* of e.

Lemma 136. *T_e is a directed tree for all $e \in F$, and the root of T_e is end e.*

Proof. Let $\tilde{T}_e$ be the undirected version of T_e, and suppose $C = v_1, \ldots, v_k, v_1$ is a cycle of minimal length in $\tilde{T}_e$. Let us choose $j \in [k]$ such that $l := \mathrm{sd}_e(v_j)$ is maximal. Without loss of generality, let $j = 2$. Then there are edges between v_1 and v_2 and between v_2 and v_3 in G, and $\mathrm{sd}(v_1) = l - 1$ and $\mathrm{sd}(v_3) = l - 1$. Let $\prec$ be a strict partial order for G, and without loss of generality let e be $\prec$-minimal. By Corollary 135 we have $v_1 \prec v_2$ and $v_3 \prec v_2$. First, let us suppose that $v_3 \prec v_1$. Then v_3 and v_1 are adjacent, and Corollary 135 implies that $\mathrm{sd}_e(v_3) < \mathrm{sd}_e(v_1)$, a contradiction. Now, let us assume $v_3 \not\prec v_1$. Then we can apply Lemma 133. We obtain that $v_3 \notin A_e^{v_1}$ and that $A_e^{v_1} \,\dot{\cup}\, \{v_1, v_3\} \subseteq A_e^{v_2}$. An illustration can be found in Figure 6.11a. As a consequence, $\mathrm{sd}_e(v_2) = |A_e^{v_2}| + 1 \geq |A_e^{v_1}| + 3 = \mathrm{sd}_e(v_1) + 2$, again a contradiction. It follows that there does not exist a cycle in $\tilde{T}_e$.

In the following, we show that there is a directed path from e to each vertex $v \in V_e$. (This part of the proof is illustrated in Figure 6.11b.) Let W_e be the set of vertices $w' \in V_e$ that e is connected to by a directed path. Clearly, $e \in W_e$. Further, let $W := V_e \setminus W_e$. Let us assume that $W \neq \emptyset$. Let $\prec$ be a strict partial order for G, and without loss of generality let e be $\prec$-minimal. Further, let w be a vertex in W such that $\mathrm{sd}_e(w)$ is

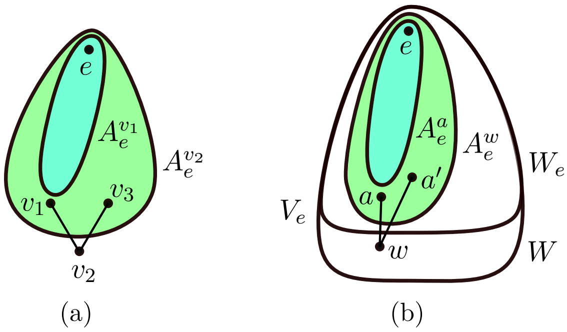


Figure 6.11.: Illustrations for the proof of Lemma 136

minimal. Then $A_e^w \cap W = \emptyset$ according to Lemma 133 and Corollary 135. Hence, $A_e^w \subseteq W_e$. As $w \in U_e$, we have $e \prec w$. Thus, the set A_e^w is not empty. Let $a \in A_e^w$ be such that $\mathrm{sd}_e(a)$ is maximal. Notice that $a \in A_e^w$ implies that $a \prec w$ and $\{a, w\} \in E$. According to Corollary 135, it holds that $\mathrm{sd}_e(a) < \mathrm{sd}_e(w)$, and as there is no directed edge from a to w in T_e, we must have $\mathrm{sd}_e(a) < \mathrm{sd}_e(w) - 1$. Hence, $|A_e^a| < |A_e^w| - 1$. Since $a \prec w$, we have $A_e^a \subseteq A_e^w$ (Lemma 133). Let a' be a vertex in A_e^w such that $a' \notin A_e^a \cup \{a\}$. Then $a' \prec w$ by Lemma 133. Let $M \in \mathcal{M}_e$ be a max clique with $w \in M$. We show that both a and a' are also in max clique M: If $w \in S_e$, then $w \in S_1^{\prec}(e)$ (Observation 127) and $w \in V_1^{\prec}(M)$ (Observation 103), and it follows from Observation 73 that $a, a' \in V_1^{\prec}(M) \subseteq M$. If $w \in O_e$, then M is the only max clique containing w. Since w is adjacent to a and a', the vertices a, a' must be in M. Consequently, $a, a' \in M$, and there must be an edge between them. Thus, $a \prec a'$ or $a' \prec a$. As $a' \notin A_e^a$, we cannot have $a' \prec a$ (Lemma 133); and we cannot have $a \prec a'$ either because it implies that $\mathrm{sd}_e(a) < \mathrm{sd}_e(a')$ (Corollary 135), and $a \in A_e^w$ was chosen such that $\mathrm{sd}_e(a)$ is maximal, a contradiction. □

Given the FP+C-formula in (6.9) for the side depth of a vertex $v \in V_e$ regarding an end $e \in F$, it is easy to see that we can define the edge relation of the side tree T_e of $e \in F$ in fixed-point logic with counting.

Observation 137. *There exists an* FP+C*-formula* $\varphi_T(x^*, x_1, x_2)$ *that is satisfied for a graph* G *and vertices* $e, v, w \in V$ *if and only if* $e \in F$ *and* (v, w) *is an edge of the directed tree* T_e*.*

For every vertex v of side tree T_e, the depth of v in T_e is $\mathrm{sd}_e(v) - 1$. We let $\mathrm{anc}_e(v)$ be the set of all ancestors and $\mathrm{dec}_e(v)$ be the set of all descendants of $v \in V_e$ in T_e. In the following we show more properties of T_e.

Lemma 138. *Let* $\prec$ *be a strict partial order for* G*, and let* $e \in F$ *be an end that is* $\prec$*-minimal (or* $\prec$*-maximal). Then for all* $v, w \in V_e$*, we have* $v \prec w$ *(or* $w \prec v$*) if and only if* v *is a proper ancestor of* w *in* T_e*.*

Proof. Let $\prec$ be a strict partial order for G. Let $e \in F$ and $v, w \in V_e$. Without loss of generality, let us assume e is $\prec$-minimal.

Let v be a proper ancestor of w, and let $v = v_1, \dots, v_k = w$ be the directed path from v to w in the directed tree T_e. Then $\mathrm{sd}_e(v_i) = \mathrm{sd}_e(v_{i+1}) - 1$ and $\{v_i, v_{i+1}\} \in E$ for all

$i \in [k-1]$. By Corollary 135 we have $v_i \prec v_{i+1}$ for all $i \in [k-1]$, and the transitivity of $\prec$ implies that $v \prec w$.

Now let us assume there exist $v, w \in V_e$ with $v \prec w$ where v is not a proper ancestor of w in T_e. Let w be of minimal depth in T_e with this property. As $v \prec w$, we have $w \neq e$. Let w' be the parent of w in T_e. Clearly, $w' \neq v$. According to Corollary 135 we have $w' \prec w$. If $v \prec w'$, then due to our choice of w, vertex v is a proper ancestor of w', and therefore, v is a proper ancestor of w, a contradiction. Consequently, we cannot have $v \prec w'$. By Lemma 133 we obtain $v \notin A_e^{w'}$. However, $v \in A_e^w$, and from $w' \prec w$ it follows that $w' \in A_e^w$ and $A_e^{w'} \subseteq A_e^w$. Therefore, $A_e^{w'} \dot\cup \{w', v\} \subseteq A_e^w$. Then $|A_e^w| \geq |A_e^{w'}| + 2$, but $\mathrm{sd}_e(w) = \mathrm{sd}_e(w') + 1$, a contradiction. □

Corollary 139. *The edge relation of the induced subgraph $G[V_e]$ corresponds to the symmetric closure of the transitive closure of the edge relation of T_e for all $e \in F$.*

Lemma 140. *Let $v, w \in V_e$ for $e \in F$. Then $w \in \mathrm{anc}_e(v)$ if, and only if, $v \in A$ implies $w \in A$ for all max cliques $A \in \mathcal{M}$.*

Proof. Let $e \in F$ and $v, w \in V_e$. First we show that if $w \in \mathrm{anc}_e(v)$, then $v \in A$ implies $w \in A$ for all max cliques $A \in \mathcal{M}$. Let $v \in O_e$. Then there exists only one max clique $A \in \mathcal{M}$ with $v \in A$. If $w \in \mathrm{anc}_e(v)$, then either $w = v$ or there is an edge between v and w in G according to Corollary 139. Hence, $w \in A$. Now let $v \in S_e$. Without loss of generality, let e be $\prec$-minimal. Thus, $S_e = S_1^{\prec}(e)$ (Observation 127). Let $w \in \mathrm{anc}_e(v)$. By Lemma 138 we know $w \preceq v$. Let $A \in \mathcal{M}$ be such that $v \in A$. Then we obtain $w \in V_1^{\prec}(A) \subseteq A$ as a direct consequence of Observation 103 and Observation 73.

Now let us prove that $w \in \mathrm{anc}_e(v)$ if $v \in A$ implies $w \in A$ for all max cliques $A \in \mathcal{M}$. Let $w \in A$ for all max cliques $A \in \mathcal{M}$ with $v \in A$. Clearly, $w \in \mathrm{anc}_e(v)$ if $v = w$. Thus, let $v \neq w$. Let A be a max clique with $v \in A$. Then $w \in A$. Therefore, there is an edge between v and w. It follows that $v \prec w$ or $w \prec v$. By Lemma 138 we obtain that w is a proper ancestor of v or that v is a proper ancestor of w in T_e. In the first case we are done. Thus, let us assume v is a proper ancestor of w in T_e. Then it follows from the above that $w \in A$ implies $v \in A$ for all max cliques $A \in \mathcal{M}$. Hence, for all max cliques A, we have $v \in A$ if and only if $w \in A$. It is not hard to see, that this yields that $\{v, w\}$ is a non-trivial module, which is a contradiction because G is prime. □

Observation 141. *Let $e \in F$. The set O_e is a subset of the set of leaves of T_e.*

Proof. Let $e \in F$. Let us assume there exists a vertex w with $w \in O_e$ but w is not a leaf of T_e. Let v be a child of w in T_e. Then $w \in \mathrm{anc}_e(v)$, and by Lemma 140, $v \in A$ implies $w \in A$ for all max cliques $A \in \mathcal{M}$. As $w \in O_e$ is contained in just one max clique A, max clique A is the only max clique containing vertex v. Consequently, v and w are each solely adjacent to the vertices in A except for itself. We obtain that $\{v, w\}$ is a non-trivial module, a contradiction. □

Let $\{F_1, F_2\}$ be the 2-coloring of the bundle tree $G[F]$ of G. For each $i \in [2]$ we define a relation $\prec_i$ on V. We let $a \prec_i b$ if, and only if, one of the following holds:

- there exists an $e \in F_i$ such that $a, b \in V_e$ and $a \in \mathrm{anc}_e(b) \setminus \{b\}$,
- there exists an $f \in F_{3-i}$ such that $a, b \in V_f$ and $a \in \mathrm{dec}_e(b) \setminus \{b\}$,
- there exist $e \in F_i$, $f \in F_{3-i}$ and a max clique $A \in \mathcal{M}$ such that $a \in S_e$, $b \in S_f$ and $a, b \in M$.

Observation 142. *Let $\prec$ be a strict partial order for G, and let $\{F_1, F_2\}$ be the 2-coloring of the bundle tree $G[F]$. Then $\prec = \prec_1$ if $F_1 = F_1^{\prec}$ and $F_2 = F_2^{\prec}$, and $\prec = \prec_2$, if $F_1 = F_2^{\prec}$ and $F_2 = F_1^{\prec}$.*

Proof. Let $\prec$ be a strict partial order for G, and let $\{F_1, F_2\}$ be the 2-coloring of $G[F]$. Without loss of generality, let $F_1 = F_1^{\prec}$ and $F_2 = F_2^{\prec}$. Then all ends in F_1 are minimal and all ends in F_2 maximal with respect to $\prec$.

Let $a \prec b$. We have to show that $a \prec_1 b$. If there exists an $e \in F$ such that $a, b \in V_e$, then $a \prec_1 b$ follows directly from Lemma 138. Thus, let there be ends $e, f \in F$ with $e \neq f$ such that $a \in V_e$ and $b \in V_f$. Since $\{a, b\} \in E$ there exists a max clique A such that $a, b \in A$. By Corollary 131 vertices e and f are the ends of A. Let $a \in O_e$ or $b \in O_f$. Then it follows from Observation 101 that $a \in O_f$ or $b \in O_e$. Hence, $a, b \in V_f$ or $a, b \in V_e$ in this case, and we have $a \prec_1 b$ as shown above. In the following let $a \in S_e$ and $b \in S_f$. If e is $\prec$-maximal, then $a \in S_e$ implies that $b \in S_e$ according to Observation 127 and Observation 97, which is a contradiction to Lemma 128. Hence, e must be $\prec$-minimal, that is, $e \in F_1$. As e and f are the ends of A, ends e and f are adjacent in $G[F]$. Thus, $e \in F_1$ yields $f \in F_2$. Therefore, $a \prec_1 b$.

Next let us prove that $a \prec b$ if $a \prec_1 b$. If $a \prec_1 b$ holds because of one of the first two cases, then $a \prec b$ follows directly from Lemma 138. Thus, let us consider the third case: There exist $e \in F_1$, $f \in F_2$ and a max clique $A \in \mathcal{M}$ such that $a \in S_e$, $b \in S_f$ and $a, b \in M$. Since a and b are adjacent, we have either $a \prec b$ or $b \prec a$. If $b \prec a$, then we obtain a contradiction to Lemma 128, as $a \in S_e$ yields $b \in S_e$ by Observation 127 and Observation 97. Consequently, we have $a \prec b$. ☐

It is not hard to see that $\prec_{3-i}$ is exactly the reverse relation $\prec_{iR}$ of strict partial order $\prec_i$.

Corollary 143. *For each $i \in [2]$ the relation $\prec_i$ is a strict partial order for G.*

6.4. The Bundle Extension and Extended Valid Subgraphs

Let $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$ be an LO-colored graph, where the underlying graph $G = (V, E)$ is a prime chordal comparability graph with $|V| \geq 4$. In this section we transform the LO-colored graph $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$ into a structure $\mathtt{H}^*$, a bundle extension. We do this in two steps. First we define the O-extension $G^{*\prime}$ of LO-colored graph G^*, and afterwards, the bundle extension $\mathtt{H}^*$ of $G^{*\prime}$. The bundle extension allows us to retrieve G^* but also contains additional information about the structure of G. Further, we define valid subgraphs and extended valid subgraphs in this section. Valid subgraphs are induced subgraphs of the underlying graph of bundle extension $\mathtt{H}^*$. As we need knowlegde about the structure of the valid subgraph, we equip the valid subgraph with additional relations, which contain information about the structure. We call the resulting structure an extended valid subgraph. Extended valid subgraphs are used in the following section to construct a decomposition tree that can be used for canonization.

6.4.1. The O-Extension

Let $G^* = (U, V, E, M, \trianglelefteq, L) \in \mathcal{C}^*_{\mathrm{ChCo}}$ be an LO-colored graph with $|V| \geq 4$. In this section we transform G^* into an O-extension $G^{*\prime}$. We add vertices to G^* such that subsequently in the underlying graph every max clique has a vertex that is contained in only one max clique, that is, a vertex that belongs to the set O. Then for every end e the set of leaves of the tree T_e coincides with the set O_e. We mark the newly added vertices by adding a unary relation that contains these vertices. So that we can identify the original LO-colored graph G^* given the O-extension $G^{*\prime}$.

First, let us extend the underlying graph $G = (V, E)$ of G^* to a graph $G' = (V', E')$. Let $\mathcal{M}$ be the set of max cliques of G, and let O be the middle of G. For every max clique $A \in \mathcal{M}$ of G with $A \cap O = \emptyset$, let $v_A \notin V$ be a new vertex. Let $V_{\mathcal{M}} := \{v_A \mid A \in \mathcal{M} \colon A \cap O = \emptyset\}$. We let

$$
\begin{aligned}
V' &:= V \cup V_{\mathcal{M}} \quad \text{and} \\
E' &:= E \cup \big\{\{v_A, v\} \,\big|\, v_A \in V_{\mathcal{M}}, v \in A\big\}.
\end{aligned}
$$

Note that G is the subgraph of G' induced by V.

In the following we show that G' is a prime chordal comparability graph.

Lemma 144. *G' is prime.*

Proof. Let us assume that there exists a non-trivial module M of G'. As M is a module of G', the set $M \cap V$ is a module of G. Since $G = (V, E)$ is a prime graph, which means there are no non-trivial modules in G, we must have $|M \cap V| \leq 1$ or $M \cap V = V$. In the following we show that each of the two conditions lead to a contradiction. Note that $G = (V, E)$ is a prime graph with $|V| \geq 4$. Thus, G has at least two max cliques and each max clique consists of at least two vertices.

First let us suppose that $|M \cap V| = 0$. Then there exist vertices $v_A, v_B \in M$ with $v_A, v_B \in V_{\mathcal{M}}$ and $v_A \neq v_B$. Thus, for max cliques A and B of G we have $A \neq B$. As A and B are max cliques of G, there exists a vertex $a \in A \setminus B$. Vertex a is not in M because $|M \cap V| = 0$. Now, v_A and a are adjacent but v_B and a are not adjacent. Since M is a module of G', we obtain a contradiction.

Next let us suppose that $|M \cap V| = 1$. Let v be the vertex in $M \cap V$, and let $v_A \in V_{\mathcal{M}}$ be a further vertex in M. Let us consider the case where $v \notin A$. Then $A \cap M = \emptyset$ and as v_A is adjacent to all $a \in A$, v must be adjacent to all $a \in A$ as well. Hence, $A \dot{\cup} \{v\}$ is a clique, which is a contradiction to A being a max clique. It remains to consider the case where $v \in A$. As $v_A \in V_{\mathcal{M}}$, we have $A \cap O = \emptyset$. Thus, v is contained in at least two max cliques of G by Lemma 123. Let $B \in \mathcal{M}$ be a max clique of G with $A \neq B$ and $v \in B$. Since A and B are distinct, there exists a vertex $b \in B \setminus A$. Clearly, $b \neq v$, $b \neq v_A$ and $b \notin M$. We obtain that v and b are adjacent but v_A and b are not adjacent. As M is a module of G', this is a contradiction.

Finally, we assume that $M \cap V = V$. Since M is a non-trivial module of G' we have $V' \setminus M \neq \emptyset$, and it follows from $M \cap V = V$ that $V' \setminus M \subseteq V_{\mathcal{M}}$. Let $v_A \in V_{\mathcal{M}}$ be a vertex of G' that is not in M. Let B be a max clique of G with $B \neq A$. Max cliques A and B

are subsets of M. Let $a \in A \setminus B$ and $b \in B \setminus A$. Then a and v_A are adjacent but b and v_A are not adjacent. Again, we obtain a contradiction because M is a module of G'. □

Observation 145. *G' is a chordal graph.*

Proof. Let us assume there exists an induced cycle $C = c_1, \dots, c_k, c_1$ of length at least 4 in G', and let us suppose C contains a vertex $v_A \in V_{\mathcal{M}}$. Without loss of generality, let $c_2 = v_A$. As v_A is only adjacent to the vertices in max clique A of G, the vertices c_1 and c_3, which are adjacent to c_2, must be contained in max clique A. Thus, there is an edge between c_1 and c_3, and C is not an induced cycle, a contradiction. Hence, C contains only vertices in V, which is a contradiction as G is a chordal graph. □

In the following we prove that G' is a comparability graph. Let $\prec$ be a strict partial order for G. We define a relation $\prec'$ on V' that extends the strict partial order $\prec$ for G, and use the subsequent lemmas to show that $\prec'$ is a strict partial order for G'. We let

$$v \prec' w \text{ for all } v, w \in V \text{ with } v \prec w,$$

and for each vertex $v_A \in V_{\mathcal{M}}$ we let

$$a \prec' v_A \text{ for all } a \in V_1^{\prec}(A) \text{ and}$$
$$v_A \prec' a \text{ for all } a \in V_2^{\prec}(A).$$

Lemma 146. *The relation $\prec'$ is a strict partial order.*

Proof. Since $\prec$ is irreflexive, it is not hard to see that $\prec'$ is irreflexive as well. Let us show that $\prec'$ is transitive. Let $u, v, w \in V'$ be such that $u \prec' v$ and $v \prec' w$. Clearly, we have $u \prec' w$ if $u, v, w \in V$.

Let us consider the case where $v \in V_{\mathcal{M}}$. Then $v = v_A$ for a max clique A of G. As $u \prec' v$ and $v \prec' w$, we have $u \in V_1^{\prec}(A)$ and $w \in V_2^{\prec}(A)$. If follows from Observation 72 that $u \prec w$, and therefore, $u \prec' w$.

Now, let us consider the case where $u \in V_{\mathcal{M}}$. Then $u = v_A$ for a max clique A of G. As $u \prec' v$, we have $v \in V_2^{\prec}(A)$. If $w \in V$, then $v \prec w$ and $w \in V_2^{\prec}(A)$ according to Observation 73. Thus, if $w \in V$, we have $u \prec' w$. Let $w \in V_{\mathcal{M}}$. Then $w = v_B$ for a max clique B of G, and $v \in V_1^{\prec}(B)$. Thus, $v \in V_1^{\prec}(B) \subseteq B$ and $v \in V_2^{\prec}(A) \subseteq A$. If $A = B$, we obtain a contradiction according to Observation 72. Therefore, let $A \neq B$. Since $v \in A \cap B$, there exists an $i \in [2]$ such that $v \in V_i^{\prec}(A, B)$ by Observation 71. Without loss of generality, let $i = 1$. As $v \in V_1^{\prec}(A, B) \subseteq V_1^{\prec}(A)$, we obtain a contradiction to Observation 72. The case where $w \in V_{\mathcal{M}}$ can be handled analogously. □

Lemma 147. *Let $\prec$ be a strict partial order for G, and let A be a max clique of G. If $A \cap O = \emptyset$, then $V_1^{\prec}(A) \cup V_2^{\prec}(A) = A$.*

Proof. Let $\prec$ be a strict partial order for G. Let A be a max clique of G where $A \cap O = \emptyset$. Clearly, $V_1^{\prec}(A) \cup V_2^{\prec}(A) \subseteq A$. Thus, we only need to prove that each $a \in A$ is also contained in $V_1^{\prec}(A) \cup V_2^{\prec}(A)$. As $A \cap O = \emptyset$, every vertex $a \in A$ is contained in at least two max cliques according to Lemma 123. By Observation 122 there exists an end e

such that $a \in S_e$, where S_e is the side of e of G. Without loss of generality let e be $\prec$-minimal. Then $S_e = S_1^{\prec}(e)$ by Observation 127. From Observation 103 it follows that $a \in V_1^{\prec}(A)$. $\square$

Lemma 148. *The relation $\prec'$ is a strict partial order for G'.*

Proof. According to Lemma 146, $\prec'$ is a strict partial order. In order to show that $\prec'$ is a strict partial order for G', it remains to prove that for all vertices $u, v \in V'$ we have $\{u, v\} \in E'$ if and only if u and v are comparable with respect to $\prec'$. For $u, v \in V$ this follows from $\prec$ being a strict partial order for G. Every vertex $v_A \in V_{\mathcal{M}}$ is only comparable to all $a \in V_1^{\prec}(A) \cup V_2^{\prec}(A)$ with respect to $\prec'$, and each vertex $v_A \in V_{\mathcal{M}}$ is only adjacent to all $a \in A$ in G'. Therefore, we need to show that $V_1^{\prec}(A) \cup V_2^{\prec}(A) = A$ for all $v_A \in V_{\mathcal{M}}$, which follows directly from Lemma 147 because for each $v_A \in V_{\mathcal{M}}$ we have $A \cap O = \emptyset$. $\square$

Corollary 149. *G' is a comparability graph.*

Now we consider the max cliques of G' and show that all added vertices $v_A \in V_{\mathcal{M}}$ are contained in the middle O' of G'. We will see in Corollary 152 that the middle O' of G' is exactly the extension of O by the vertices from $V_{\mathcal{M}}$.

For a max clique $A \in \mathcal{M}$ of G, let the set $g'(A)$ of vertices of G' be defined as follows:

$$g'(A) := \begin{cases} A \cup \{v_A\} & \text{if } A \cap O = \emptyset \\ A & \text{otherwise.} \end{cases}$$

In the following we show that g' is a bijection between the set $\mathcal{M}$ of max cliques of G and the set $\mathcal{M}'$ of max cliques of G'.

Observation 150. *Let A be a max clique of G. Then $g'(A)$ is a max clique of G'.*

Proof. Let A be a max clique of G. The set $g'(A)$ is a clique because A is a clique and vertex v_A is adjacent to all vertices in A if $A \cap O = \emptyset$. Let us assume $g'(A)$ is not maximal. Then there is a vertex $w \notin g'(A)$ such that w is adjacent to all vertices in $g'(A)$. If $w \in V$, then A is not a max clique of G, and we have a contradiction. Let $w \notin V$. Then $w \in V_{\mathcal{M}}$, and $w = v_B$ for a max clique $B \neq A$. Since A and B are distinct max cliques of G, there exists a vertex $v \in A \setminus B$. By definition of E', vertex w is not adjacent to v, a contradiction. $\square$

Observation 151. *Let A' be a max clique of G'. Then there is a max clique A of G such that $A' = g'(A)$.*

Proof. Let A' be a max clique of G'. Let $C := A' \cap V$. As G is the subgraph of G' induced by V, the set C must be a clique of G. Let us consider $A' \cap V_{\mathcal{M}}$. Since there are no edges between vertices in $V_{\mathcal{M}}$, there exists at most one vertex in $A' \cap V_{\mathcal{M}}$. If there does not exist a vertex in $A' \cap V_{\mathcal{M}}$, we let A be an arbitrary max clique of G that contains C. If there exists a vertex $v_B \in A' \cap V_{\mathcal{M}}$, then clique C must be a subset of B because v_B is only adjacent to $v \in V$ if v is contained in max clique B of G, and we let A be the max clique B. Now it is not hard to see, that A' is a subset of $g'(A)$. As A' is a max clique of G', Observation 150 yields that $A' = g'(A)$. $\square$

It follows from Observations 150 and 151 that g' is a surjective mapping between the set of max cliques of G and the set of max cliques of G'. It is not hard to see that g' must be injective as well. Hence, g' is bijective. We can observe that the max cliques of G and G' essentially are the same, except that some max cliques of G' contain an additional new vertex $v_A \in V_{\mathcal{M}}$. Thus, for all vertices $v \in V$, vertex v is only contained in one max clique of G if and only if v is only contained in one max clique of G'. Further, each vertex $v_A \in V_{\mathcal{M}}$ is only contained in the max clique $A \cup \{v_A\}$ of G'. With Lemma 123 we obtain the following corollary.

Corollary 152. *The middle O' of G' is the disjoint union of $V_{\mathcal{M}}$ and the middle O of G.*

Now for every max clique A of G it follows that $g'(A) \cap O' \neq \emptyset$: If $g'(A) = A \cup \{v_A\}$, then $v_A \in g'(A) \cap O'$; and if $g'(A) = A$, then $A \cap O \neq \emptyset$, and $g'(A) \cap O' \neq \emptyset$ follows from $A \cap O \subseteq g'(A) \cap O'$. We conclude the following:

Corollary 153. *Let O' be the middle of G'. Then for every max clique A' of G' we have $A' \cap O' \neq \emptyset$.*

The subsequent corollary follows directly from Lemma 123 and Corollary 153.

Corollary 154. *For every max clique A' of G' there is a vertex $v \in O'$ such that A' is spanned by vertex v.*

Lemma 155. *Let $e \in F'$ be an end of G', and let O'_e be the middle and T'_e be the side tree of e of G'. Then O'_e is the set of leaves of T'_e.*

Proof. Let $e \in F'$ be an end of G'. Let O'_e be the middle and T'_e be the side tree of e of G'. By Observation 141 the set O'_e is a subset of the set of leaves of T'_e. Let us suppose there is a leaf w of T'_e that is not in O'_e. Then w is contained in the side S'_e of e. It follows from Lemma 123 that w is contained in at least two max cliques A' and B' of G'. Let us consider max clique A'. Max clique A' has e as an end by Corollary 131. According to Corollary 153 there exists a vertex $v \in A' \cap O'$. As $v \in O'$, vertex v is only contained in max clique A' (Lemma 123). Since $v \in U_e$, Observation 101 implies that $v \in O_e \subseteq V_e$. By Lemma 140 it follows that w is an ancestor of v in T_e. As $v \neq w$, we obtain that w is not a leaf, a contradiction. ☐

Now let us define the O-extension of the LO-colored graph $G^* = (U, V, E, M, \trianglelefteq, L) \in \mathcal{C}^*_{\text{ChCo}}$. Let $U' := V' \cup M$, and $Z := V_{\mathcal{M}}$. Then $G^{*\prime} := (U', V', E', M, \trianglelefteq, L, Z)$ is the *O-extension of* G^*. Generally, a $\{V', E', M', \trianglelefteq', L', Z'\}$-structure is an *$O$-extension* if it is the O-extension of an LO-colored graph $G^* \in \mathcal{C}^*_{\text{ChCo}}$. As the graph $G' = (V', E')$ is a prime chordal comparability graph, we can interpret the O-extension $G^{*\prime} = (U', V', E', M', \trianglelefteq', L', Z')$ of $G^* \in \mathcal{C}^*_{\text{ChCo}}$ as an LO-colored graph $(U', V', E', M', \trianglelefteq', L')$ from $\mathcal{C}^*_{\text{ChCo}}$ with an additional unary relation Z', which contains all vertices from V' that are not in V. We transfer all names from LO-colored graphs to O-extensions. Thus, $G' = (V', E')$ is the *underlying graph*, M' the set of *basic color elements* and L' the *color relation* of $G^{*\prime}$.

Given an O-extension $G^{*\prime} = (U', V', E', M', \trianglelefteq', L', Z')$ we can easily determine the unique LO-colored graph $G^* \in \mathcal{C}^*_{\text{ChCo}}$ of which $G^{*\prime}$ is the O-extension: We obtain G^* from $G^{*\prime}$

by dropping relation Z' and by using the subgraph induced by $V' \setminus Z'$ of the underlying graph $G' = (V', E')$ of $G^{*\prime}$ as underlying graph $G = (V, E)$ of G^*.

The following transduction Θ' maps each LO-colored graph $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$ to an isomorphic copy of its O-extension $G^{*\prime}$.

$$\begin{aligned}\Theta' = \big(&\theta_{U'}(x,y,p), \theta_{\approx'}(x,y,p,x',y',p'), \theta_{V'}(x,y,p), \theta_{E'}(x,y,p,x',y',p'),\\ &\theta_{M'}(x,y,p), \theta_{\trianglelefteq'}(x,y,p,x',y',p'), \theta_{L'}(x,y,p,x',y',p',x'',y'',p''), \theta_{Z'}(x,y,p)\big)\end{aligned}$$

where

$$\begin{aligned}
\theta_{U'}(x,y,p) :=\ & p = 0\\
& \vee\ \big(p = 1 \wedge \varphi_{\mathrm{span}}(x,y) \wedge \neg\exists z\, (\varphi_{\mathcal{M}}(x,y,z) \wedge \varphi_O(z))\big),\\
\theta_{\approx'}(x,y,p,x',y',p') :=\ & (p = 0 \wedge p' = 0 \wedge x = x')\\
& \vee\ \big(p = 1 \wedge p' = 1 \wedge \varphi_{\mathrm{span},\approx}(x,y,x',y')\big),\\
\theta_{V'}(x,y,p) :=\ & (p = 0 \wedge V(x)) \vee p = 1,\\
\theta_{E'}(x,y,p,x',y',p') :=\ & (p = 0 \wedge p' = 0 \wedge E(x,x'))\\
& \vee\ (p = 0 \wedge p' = 1 \wedge \varphi_{\mathcal{M}}(x',y',x))\\
& \vee\ \big(p = 1 \wedge p' = 0 \wedge \varphi_{\mathcal{M}}(x,y,x')\big),\\
\theta_{M'}(x,y,p) :=\ & p = 0 \wedge M(x),\\
\theta_{\trianglelefteq'}(x,y,p,x',y',p') :=\ & p = 0 \wedge p' = 0 \wedge \trianglelefteq(x,x'),\\
\theta_{L'}(x,y,p,x',y',p',x'',y'',p'') :=\ & p = 0 \wedge p' = 0 \wedge p'' = 0 \wedge L(x,x',x''),\\
\theta_{Z'}(x,y,p) :=\ & p = 1.
\end{aligned}$$

Within the above definition of transduction Θ' the formulas φ_{span}, $\varphi_{\mathrm{span},\approx}$ and $\varphi_{\mathcal{M}}$ are not the $\{E\}$-formulas for graphs from Section 2.8.2 and φ_O not the $\{E\}$-formula for graphs $G \in \mathcal{C}_{\mathrm{ChCo}}$ from Corollary 125 but matching $\{V, E, M, \trianglelefteq, L\}$-formulas referring to the underlying graphs of LO-colored graphs $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$. We obtain them easily by an application of the Transduction Lemma. The transduction used is the $(\{V, E, M, \trianglelefteq, L\}, \{E\})$-transduction $\Theta = (V(x), E(x, x'))$.

Lemma 156. *For every $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$, structure $\Theta'[G^*]$ is isomorphic to $G^{*\prime}$.*

Proof. Let $G^* = (U, V, E, M, \trianglelefteq, L) \in \mathcal{C}^*_{\mathrm{ChCo}}$ and let $G^{*\prime} = (U', V', E', M', \trianglelefteq', L', Z')$ be the O-extension of G^*. Further, let $G = (V, E)$ and $G' = (V', E')$ be the underlying graphs of G^* and $G^{*\prime}$. The set $U'' := \theta_{U'}[G^*; x, y, p]$ consists of all triples $(v, w, j) \in U(G^*)^2 \times N(G^*)$ where $j \in \{0, 1\}$ and if $j = 1$, vertices $v, w \in V$ are spanning vertices of a max clique of G that is disjoint to the middle O of G. Thus, U'' is the disjoint union of $U''_0 := U(G^*)^2 \times \{0\}$ and $U''_1 := \{(v, w) \in V(G^*) \mid \exists A \in \mathcal{M}\colon\ v, w \text{ span } A,\ A \cap O = \emptyset\} \times \{1\}$. Let $\approx'$ be the equivalence relation generated by $\theta_{\approx'}[G^*; x, y, p, x', y', p']$. Triples $(v, w, 0)$ and $(v', w', 0)$ from U''_0 are equivalent regarding $\approx'$ if, and only if, $v = v'$. Thus, the mapping $h\colon (v, w, 0)/_{\approx'} \mapsto v$ is a bijection between $U''_0/_{\approx'}$ and U. Notice that there are no triples from U''_0 that are equivalent to triples from U''_1. Triples $(v, w, 1)$ and $(v', w', 1)$ from U''_1 are equivalent if, and only if, v, w and v', w' span the same max clique A of G (where $A \cap O = \emptyset$). We obtain that $U''_1/_{\approx'}$ consists of exactly one vertex for each max clique A of G with $A \cap O = \emptyset$. We let $h((v, w, 1)/_{\approx'}) := v_A$ if v, w span max clique A. Then h bijectively maps $U''_1/_{\approx'}$ to $V_{\mathcal{M}}$. Thus, h is a bijection between $U''/_{\approx'}$ and U'.

Now let us show that formulas $\theta_{V'}$, $\theta_{\approx'}$ and $\theta_{E'}$ define an isomorphic copy of the graph G'. It is not hard to see that $V'' := \theta_{V'}[G^*; x, y, p] \cap U''$ is the union of the sets $V''_0 := \{(v, w, 0) \in U''_0 \mid v \in V\}$ and $V''_1 := U''_1$. Since $h(V''_0/_{\approx'}) = V$ and $h(V''_1/_{\approx'}) = V_{\mathcal{M}}$, it follows that $h(V''/_{\approx'}) = V'$. Let E'' be the relation $\theta_{E'}[G^*; x, y, p, x', y', p'] \cap U''^2$, and let $E''_{0,0}$, $E''_{0,1}$ and $E''_{1,0}$ be the set of all $(v, w, j, v', w', j') \in E''$ where $j = 0$ and $j' = 0$, $j = 0$ and $j' = 1$, and $j = 1$ and $j' = 0$, respectively. Then $E'' = E''_{0,0} \cup E''_{0,1} \cup E''_{1,0}$. Let $h' \colon (U''/_{\approx'})^2 \mapsto (U')^2$ be the mapping where

$$h'\big((v, w, j)/_{\approx'}, (v', w', j')/_{\approx'}\big) := \big(h((v, w, j)/_{\approx'}), h((v', w', j')/_{\approx'})\big).$$

Clearly, $h'(E''_{0,0}/_{\approx'}) = E$. We have $(v, w, 0, v', w', 1) \in E''_{0,1}$ if, and only if, $(v, w, 0) \in U''_0$, $(v', w', 1) \in U''_1$, and v' and w' are vertices of G that span a max clique A of G with $v \in A$. Hence, $h'(E''_{0,1}/_{\approx'}) = \{(v, v_A) \in U \times V_{\mathcal{M}} \mid v \in A\}$. Analogously, we can show that $h'(E''_{1,0}/_{\approx'}) = \{(v_A, v) \in V_{\mathcal{M}} \times U \mid v \in A\}$. Then $h'(E''_{0,1}/_{\approx'} \cup E''_{1,0}/_{\approx'})$ is the set $\{\{v_A, v\} \mid v_A \in V_{\mathcal{M}}, v \in A\}$. Thus, $h'(E''/_{\approx'}) = E'$. It follows that formulas $\theta_{V'}$, $\theta_{\approx'}$ and $\theta_{E'}$ define an isomorphic copy of the graph G'.

It is not hard to see that formulas $\theta_{M'}$, $\theta_{\trianglelefteq'}$ and $\theta_{L'}$ define the relations corresponding to M', $\trianglelefteq'$ and L', and that $\theta_{Z'}$ and $\theta_{\trianglelefteq'}$ define the relation corresponding to Z' on the universe $U''/_{\approx'}$. □

6.4.2. The Bundle Extension

We consider O-extensions in this section, and extend them further into what we call bundle extensions. O-extensions are a sort of LO-colored graphs from $\mathcal{C}^*_{\text{ChCo}}$ with an additional unary relation Z of vertices. We can consider O-extensions as LO-colored graphs from $\mathcal{C}^*_{\text{ChCo}}$ where certain vertices are marked. That is why we denote the O-extensions in this section by G^*. In order to obtain the bundle extension $\mathtt{H}^*$ of G^*, we add two new vertices, f_1 and f_2, to the underlying graph G of O-extension G^*. We let each of these two vertices be adjacent to all other vertices. Further, we construct two side trees for the bundle extension. The vertices f_1 and f_2 become the roots of these two side trees, and the new side trees contain all side trees of the underlying graph G as subtrees.

Defining the Bundle Extension

Let $G^* = (U, V, E, M, \trianglelefteq, L, Z)$ be an O-extension with underlying graph $G = (V, E)$. We extend the underlying graph $G = (V, E)$ of our O-extension G^* to a graph $\mathtt{H} = (\mathtt{V}, \mathtt{E})$. We let the vertex set $\mathtt{V}$ consist of all vertices in V and two distinct vertices f_1 and f_2 for the color classes F_1 and F_2, respectively, of the bundle tree $G[F]$ of G. The edge relation $\mathtt{E}$ of $\mathtt{H}$ is the set $E \cup \{\{v, w\} \in \binom{\mathtt{V}}{2} \mid v \in \{f_1, f_2\}\}$.

Note that G is the subgraph of $\mathtt{H}$ induced by V. Further, each of the two additional vertices f_1 and f_2 is completely connected to the rest of the graph. Thus, $\{f_1, f_2\}$ is a non-trivial module of $\mathtt{H}$, and therefore, the graph $\mathtt{H}$ is not prime. However, $\mathtt{H}$ remains chordal because an induced cycle of length ≥ 4 cannot contain f_1 or f_2 and must therefore be an induced cycle in G. Further, $\mathtt{H}$ is a comparability graph as we can extend a strict partial order $\prec$ for G to a strict partial order for $\mathtt{H}$ by letting $f_1 \prec v$ for all $v \in \mathtt{V} \setminus \{f_1\}$ and $v \prec f_2$ for all $v \in \mathtt{V} \setminus \{f_2\}$. Notice that we can extend a strict partial order $\prec$ for G also in a different way to a strict partial order for $\mathtt{H}$. It follows that the set of $\prec$-ends of $\mathtt{H}$ depends on the strict partial order $\prec$ for $\mathtt{H}$, and that we cannot define a unique set of ends of $\mathtt{H}$.

Next we define the *side trees*, $\mathtt{T}_{f_1}$ and $\mathtt{T}_{f_2}$, of bundle extension $\mathtt{H}$. The side tree $\mathtt{T}_{f_1}$ is a directed tree that has f_1 as root vertex and all side trees T_e of $e \in F_1$ of G as subtrees. We integrate the trees T_e into $\mathtt{T}_{f_1}$ by letting the root vertices $e \in F_1$ be the children of f_1. Notice, that according to Lemma 132 the vertex sets V_e of the directed trees T_e with $e \in F_1$ are disjoint. Equivalently, we construct $\mathtt{T}_{f_2}$, where we use the set F_2 of end vertices of G.

Let $\mathtt{U} := \mathtt{V} \cup M$, and $\mathtt{Z} := Z \cup \{f_1, f_2\}$. Further, let $\mathtt{T}$ be the set of all pairs $(v, w) \in \mathtt{V}^2$ where $(v, w) \in E(\mathtt{T}_f)$ for $f \in \{f_1, f_2\}$. We call $\mathtt{H}^* := (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, L, \mathtt{T}, \mathtt{Z})$ the *bundle extension of* G^*. Generally, a $\{\mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z}\}$-structure is a *bundle extension* if it is the bundle extension of an O-extension. We let $\mathtt{C}^*_{\mathrm{ChCo}}$ be the class of all bundle extensions. We transfer names from O-extensions, that is, LO-colored graphs, to the bundle extensions. Thus, $\mathtt{H} = (\mathtt{V}, \mathtt{E})$ is the *underlying graph*, $\mathtt{M}$ the set of *basic color elements* and $\mathtt{L}$ defines the *color relation* of bundle extension $\mathtt{H}^*$.

Having a bundle extension $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z}) \in \mathtt{C}^*_{\mathrm{ChCo}}$ it is not hard to identify the unique O-extension $G^{*\prime}$ of which $\mathtt{H}^*$ is a bundle extension: We can determine the two vertices of the directed graph $(\mathtt{V}, \mathtt{T})$ where the in-degree is 0. These vertices are f_1 and f_2. Then we obtain $G^{*\prime}$ by dropping the relation $\mathtt{T}$ from $\mathtt{H}^*$, removing f_1 and f_2 from $\mathtt{Z}$, and by using the induced subgraph $\mathtt{H}[\mathtt{V} \setminus \{f_1, f_2\}]$, where $\mathtt{H} = (\mathtt{V}, \mathtt{E})$, as underlying graph.

The following transduction Θ maps each O-extension G^* to an isomorphic copy of its bundle extension $\mathtt{H}^*$.

$$\begin{aligned}\Theta = (&\theta_{\mathtt{U}}(x,y,p), \theta_{\approx}(x,y,p,x',y',p'), \theta_{\mathtt{V}}(x,y,p), \theta_{\mathtt{E}}(x,y,p,x',y',p'), \theta_{\mathtt{M}}(x,y,p),\\ &\theta_{\trianglelefteq}(x,y,p,x',y',p'), \theta_{\mathtt{L}}(x,y,p,x',y',p',x'',y'',p''), \theta_{\mathtt{T}}(x,y,p,x',y',p'), \theta_{\mathtt{Z}}(x,y,p))\end{aligned}$$

where

$$\begin{aligned}\theta_{\mathtt{U}}(x,y,p) &:= \big(p=0 \lor p=1\big) \land \varphi_F(y),\\ \theta_{\approx}(x,y,p,x',y',p') &:= \big(p=0 \land p'=0 \land x=x'\big)\\ &\quad\lor\ \big(p=1 \land p'=1 \land \varphi_{F,\approx}(y,y')\big),\\ \theta_{\mathtt{V}}(x,y,p) &:= \big(p=0 \land V(x)\big) \lor p=1,\\ \theta_{\mathtt{E}}(x,y,p,x',y',p') &:= \big(p=0 \land p'=0 \land E(x,x')\big)\\ &\quad\lor\ \big(p=1 \land p'=1 \land \neg\varphi_{F,\approx}(y,y')\big)\\ &\quad\lor\ \big(p=0 \land p'=1\big) \lor \big(p=1 \land p'=0\big),\end{aligned}$$

$$\begin{aligned}
\theta_{\mathtt{M}}(x,y,p) &:= \; p=0 \wedge M(x),\\
\theta_{\trianglelefteq}(x,y,p,x',y',p') &:= \; p=0 \wedge p'=0 \wedge \trianglelefteq(x,x'),\\
\theta_{\mathtt{L}}(x,y,p,x',y',p',x'',y'',p'') &:= \; p=0 \wedge p'=0 \wedge p''=0 \wedge L(x,x',x''),\\
\theta_{\mathtt{T}}(x,y,p,x',y',p') &:= \; (p=1 \wedge p'=0 \wedge y=x')\\
&\quad\;\vee\; (p=0 \wedge p'=0 \wedge \exists x^*\, \varphi_T(x^*,x,x')),\\
\theta_{\mathtt{Z}}(x,y,p) &:= \; (p=0 \wedge Z(x)) \vee p=1.
\end{aligned}$$

Within the above definition we assume that formulas φ_F, $\varphi_{F,\approx}$ and φ_T are not the $\{E\}$-formulas for graphs $G \in \mathcal{C}_{\mathrm{ChCo}}$ from Section 6.3 (Corollaries 115 and 116, Observation 137) but matching $\{V,E,M,\trianglelefteq,L,Z\}$-formulas referring to the underlying graphs of O-extensions.

Lemma 157. *For every O-extension G^*, structure $\Theta[G^*]$ is isomorphic to $\mathtt{H}^*$.*

Proof. Let $G^* = (U,V,E,M,\trianglelefteq,L,Z)$ be an O-extension, and let $\mathtt{H}^* = (\mathtt{U},\mathtt{V},\mathtt{E},\mathtt{M},\trianglelefteq,\mathtt{L},\mathtt{T},\mathtt{Z})$ be the bundle extension of G^*. Further, let $G=(V,E)$ and $\mathtt{H}=(\mathtt{V},\mathtt{E})$ be the underlying graphs of G^* and $\mathtt{H}^*$, respectively. The set $\mathtt{U}' := \theta_{\mathtt{U}}[G^*;x,y,p]$ consists of all triples $(v,e,j) \in U(G^*)^2 \times N(G^*)$ where $j \in \{0,1\}$ and e is in the set F of ends of G. Thus, $\mathtt{U}' = U(G^*) \times F \times \{0,1\}$. Let $\approx$ be the equivalence relation generated by $\theta_{\approx}[G^*;x,y,p,x',y',p']$. Triples $(v,e,0)$ and $(v',e',0)$ from $U(G^*) \times F \times \{0\}$ are equivalent if, and only if, $v=v'$. Therefore, the mapping $h\colon (v,e,0)/_{\approx} \mapsto v$ is a bijection between $(U(G^*) \times F \times \{0\})/_{\approx}$ and $U(G^*)$. Note that there are no triples from $U(G^*) \times F \times \{0\}$ that are equivalent to triples from $U(G^*) \times F \times \{1\}$. Triples $(v,e,1)$ and $(v',e',1)$ from $U(G^*) \times F \times \{1\}$ are equivalent if and only if e and e' are in the same color class of bipartition $\{F_1,F_2\}$ of $G[F]$. We obtain that $(U(G^*) \times F \times \{1\})/_{\approx}$ consists of exactly two equivalence classes, which correspond to the vertices f_1 and f_2 of $\mathtt{H}$; and we let $h((v,e,1)/_{\approx}) := f_i$ if $e \in F_i$ for $i \in [2]$. Then h is a bijection between the universe of $\Theta[G^*]$ and the universe $\mathtt{U}$ of the bundle extension $\mathtt{H}^*$.

Now, it is not hard to see that formulas $\theta_{\mathtt{V}}$, $\theta_{\approx}$ and $\theta_{\mathtt{E}}$ define an isomorphic copy of the graph $\mathtt{H}$, that formulas $\theta_{\mathtt{M}}$, $\theta_{\approx}$, $\theta_{\trianglelefteq}$ and $\theta_{\mathtt{L}}$ define the relations corresponding to $\mathtt{M}$, $\trianglelefteq$ and $\mathtt{L}$, and $\theta_{\mathtt{Z}}$ and $\theta_{\approx}$ the relation corresponding to $\mathtt{Z}$ on the universe $(U(G^*) \times F \times \{0,1\})/_{\approx}$.

Let us examine the relation $\mathtt{T}' := \theta_{\mathtt{T}}[G^*;x,y,p,x',y',p'] \cap \mathtt{U}'^2$. For all $(v,e,j) \in \mathtt{U}'$ let $\mathtt{T}'(v,e,j) := \{(v',e',j') \mid (v,e,j,v',e',j') \in \mathtt{T}'\}$. First let us consider all triples $(v,e,j) \in \mathtt{U}'$ with $j=1$. Let $v \in U(G^*)$ and $e \in F$. Then $h((v,e,1)/_{\approx})$ is either f_1 or f_2. For each $i \in [2]$, the equivalence class $h^{-1}(f_i)$ is the set $U(G^*) \times F_i \times \{1\}$. For $(v,e,1) \in h^{-1}(f_i)$ we have $\mathtt{T}'(v,e,1) = \{(e,e',0) \mid e' \in F\}$, and thus, $\mathtt{T}'(h^{-1}(f_i)) = \{(e,e',0) \mid e \in F_i,\, e' \in F\}$. Notice that $(e,e',0)/_{\approx} = (e,e,0)/_{\approx}$ for all vertices $e,e' \in F$. Therefore, we obtain $\mathtt{T}'(h^{-1}(f_i))/_{\approx} = \{(e,e,0)/_{\approx} \mid e \in F_i\}$. Let $\mathtt{T}'_{i'}$ be the set of all tuples $(v,e,j,v',e',j') \in \mathtt{T}'$ where $j=i'$, and let $h'\colon (U''/_{\approx'})^2 \mapsto (U')^2$ be the mapping where

$$h'\big((v,e,j)/_{\approx'},(v',e',j')/_{\approx'}\big) := \big(h((v,e,j)/_{\approx'}),h((v',e',j')/_{\approx'})\big).$$

Then $h(\mathtt{T}'_1/_{\approx}) = \{(f_i,e) \mid e \in F_i\}$. Next let us consider all $(v,e,j) \in \mathtt{U}'$ with $j=0$. Let $v \in U(G^*)$ and $e \in F$. The set $\mathtt{T}'(v,e,0)$ is the set of all triples $(v',e',0)$ where $e' \in F$ and there exists an $e'' \in F$ such that (v,v') is an edge of the directed tree $T_{e''}$. Note that $h((v,e,0)/_{\approx}) = v$ and for each triple $(v',e',0)$ we have $h((v',e',0)/_{\approx}) = v'$. Thus,

$h(\mathtt{T}'_0/_\approx) = \{(v,v') \in E(T_{e''}) \mid e'' \in F\}$. It follows that h' maps $\mathtt{T}'/_\approx$, that is, $\mathtt{T}'_1/_\approx \dot\cup \mathtt{T}'_0/_\approx$, to $\mathtt{T}$. Hence, h is an isomorphism between G^* and $\mathtt{H}^*$. □

Having a bundle extension $\mathtt{H}^* = (\mathtt{U},\mathtt{V},\mathtt{E},\mathtt{M},\trianglelefteq,\mathtt{L},\mathtt{T},\mathtt{Z}) \in \mathtt{C}^*_{\mathrm{ChCo}}$ we cannot only determine the unique O-extension $G^{*\prime}$ of which $\mathtt{H}^*$ is a bundle extension, but also the unique LO-colored graph $G^* \in \mathcal{C}^*_{\mathrm{ChCo}}$ of whose O-extension $G^{*\prime}$ the structure $\mathtt{H}^*$ is a bundle extension. We obtain G^* by dropping the relations $\mathtt{T}$ and $\mathtt{Z}$ from $\mathtt{H}^*$, and by using the induced subgraph $\mathtt{H}[\mathtt{V}\setminus\mathtt{Z}]$ as underlying graph, where $\mathtt{H} = (\mathtt{V},\mathtt{E})$ is the underlying graph of $\mathtt{H}^*$.

Definitions and Properties

Now we consider a bundle extension $\mathtt{H}^* = (\mathtt{U},\mathtt{V},\mathtt{E},\mathtt{M},\trianglelefteq,\mathtt{L},\mathtt{T},\mathtt{Z})$. Let $\mathtt{H} = (\mathtt{V},\mathtt{E})$ be the underlying graph of $\mathtt{H}^*$. We let f_1 and f_2 be the two vertices in $\mathtt{V}$ that have in-degree 0 in the directed graph $(\mathtt{V},\mathtt{T})$. We call f_1 and f_2 the *termini* of the underlying graph $\mathtt{H}$ and we let $\mathtt{F} := \{f_1, f_2\}$. For each $i \in [2]$ we let $\mathtt{V}_{f_i}$ be the set of all vertices $v \in \mathtt{V}$ that are reachable from f_i in $(\mathtt{V},\mathtt{T})$. Obviously, $f_i \in V_{f_i}$ for $i \in [2]$ and $\mathtt{V}_{f_1} \cup \mathtt{V}_{f_2} = \mathtt{V}$. Then the side tree $\mathtt{T}_{f_i}$ is $(\mathtt{V}_{f_i},\mathtt{E}_{f_i})$ where $\mathtt{E}_{f_i} := \mathtt{T} \cap (\mathtt{V}_{f_i})^2$. Further, we let $\mathtt{O}_{f_i}$ be the set of all *leaves* of $\mathtt{T}_{f_i}$. Since $\mathtt{O}_{f_1} = \mathtt{O}_{f_2}$ (Observation 124, Lemma 155), we simply denote the set of all leaves by $\mathtt{O}$. Clearly, $f_1, f_2 \notin \mathtt{O}$. We call $\mathtt{O}$ the *middle* of $\mathtt{H}$. We define $\mathtt{S}_{f_i} := \mathtt{V}_{f_i} \setminus \mathtt{O}$ as the *side* of $f_i \in \mathtt{F}$. As we have $\mathtt{O}_{f_1} = \mathtt{O}_{f_2}$, Lemma 128 implies the following observation.

Observation 158. *We have* $\mathtt{V}_{f_1} \cap \mathtt{V}_{f_2} = \mathtt{O}$ *and* $\mathtt{S}_{f_1} \cap \mathtt{S}_{f_2} = \emptyset$.

We let $\varphi_{\mathtt{F}}(x)$, $\varphi_{\mathtt{V}}(x^*,x)$, $\varphi_{\mathtt{E}}(x^*,x,x')$, $\varphi_{\mathtt{O}}(x)$ and $\varphi_{\mathtt{S}}(x^*,x)$ be TC-formulas for $\mathtt{F}$, $\mathtt{V}_{f_i}$, $\mathtt{E}_{f_i}$, $\mathtt{O}$ and $\mathtt{S}_{f_i}$ respectively. Hence,

$$\begin{aligned}
\mathtt{H}^* &\models \varphi_{\mathtt{F}}[f] &&\iff f \in \mathtt{F}, \\
\mathtt{H}^* &\models \varphi_{\mathtt{V}}[f,v] &&\iff f \in \mathtt{F} \text{ and } v \in \mathtt{V}_f, \\
\mathtt{H}^* &\models \varphi_{\mathtt{E}}[f,v,v'] &&\iff f \in \mathtt{F} \text{ and } (v,v') \in \mathtt{E}_f, \\
\mathtt{H}^* &\models \varphi_{\mathtt{O}}[v] &&\iff v \in \mathtt{O}, \quad \text{and} \\
\mathtt{H}^* &\models \varphi_{\mathtt{S}}[f,v] &&\iff f \in \mathtt{F} \text{ and } v \in \mathtt{S}_f.
\end{aligned} \tag{6.10}$$

We let $\mathcal{M}_{\mathtt{H}}$ be the set of max cliques of $\mathtt{H}$. Let $\mathtt{H}^*$ be the bundle extension of O-extension G^*. Let G be the underlying graph of G^*. It holds that A is a max clique of G if, and only if, $A \cup \{f_1, f_2\}$ is a max clique of $\mathtt{H}$. Therefore, the property that every max clique of G is spanned by one vertex (cf. Corollary 154) is also satisfied for $\mathtt{H}$.

Corollary 159. *For every max clique* $A \in \mathcal{M}_{\mathtt{H}}$ *of* $\mathtt{H}$ *there is a vertex* $v \in A$ *such that* A *is spanned by vertex* v.

Hence, there are FO-formulas (see Section 2.8.2) that define the max cliques of $\mathtt{H}$ for bundle extensions $\mathtt{H}^*$. In the following, we show that vertex v is in the middle $\mathtt{O}$ of $\mathtt{H}$ if, and only if, v is contained in only one max clique of $\mathtt{H}$. We use the following observation to do this.

Observation 160. *Let* $v \in \mathtt{V}$. *Vertex* v *is in the middle* $\mathtt{O}$ *of* $\mathtt{H}$ *if, and only if,* v *is in the middle* O *of* G.

Proof. Let $i \in [2]$ and $v \in \mathtt{V}$. Vertex v is in $\mathtt{O}$ exactly if v is a leaf of $\mathtt{T}_{f_i}$. By construction of the side tree $\mathtt{T}_{f_i}$, v is a leaf of $\mathtt{T}_{f_i}$ if, and only if, there is an end $e \in F_i$ such that v is a leaf of the side tree T_e of G. According to Lemma 155, the set of leaves of T_e is the middle O_e of e in the underlying graph G of O-extension G^*. Further, as G is a prime chordal comparability graph, the union of all sets O_e with $e \in F_i$ is the middle O of G (Observation 124). Thus, we obtain that $v \in \mathtt{O}$ if and only if $v \in O$. □

Lemma 161. *Let $v \in \mathtt{V}$. Vertex v is in the middle $\mathtt{O}$ of $\mathtt{H}$ if and only if v is contained in only one max clique of $\mathtt{H}$.*

Proof. First of all, Lemma 161 is true for f_1 and f_2 since $f_1, f_2 \notin \mathtt{O}$ and f_1 and f_2 are contained in more than one max clique. (The prime chordal comparability graph G has at least two max cliques. Therefore, $\mathtt{H}$ has at least two max cliques.) Now let $v \in V = \mathtt{V} \setminus \mathtt{F}$. According to Observation 160, v is in the middle $\mathtt{O}$ of $\mathtt{H}$ precisely if v is in the middle O of G. Further, $v \in O$ if and only if v is contained in only one max clique of G by Lemma 123. Since A is a max clique of G exactly if $A \cup \mathtt{F}$ is a max clique of $\mathtt{H}$, it follows that v is contained in only one max clique of G if and only if v is contained in only one max clique of $\mathtt{H}$. □

Corollary 159 and Lemma 161 yield the following corollary.

Corollary 162. *We have $A \cap \mathtt{O} \neq \emptyset$ for every max clique $A \in \mathcal{M}_{\mathtt{H}}$ of $\mathtt{H}$.*

We let $\mathtt{anc}_f(v)$ be the set of ancestors and $\mathtt{dec}_f(v)$ be the set of descendants of $v \in \mathtt{V}_f$ in $\mathtt{T}_f$ for $f \in \mathtt{F}$. Clearly, the ancestors $\mathtt{anc}_f(v)$ and descendants $\mathtt{dec}_f(v)$ are definable in transitive closure logic for bundle extensions $\mathtt{H}^*$. Thus, there exist TC-formulas $\varphi_{\mathtt{anc}}(x^*, x, y)$ and $\varphi_{\mathtt{dec}}(x^*, x, y)$ such that for all elements $f, v, w \in \mathtt{U}$ of a bundle extension $\mathtt{H}^*$ we have

$$\begin{aligned} \mathtt{H}^* \models \varphi_{\mathtt{anc}}[f, v, w] \quad &\Longleftrightarrow \quad f \in \mathtt{F},\ v, w \in \mathtt{V}_f \text{ and } w \in \mathtt{anc}_f(v), \\ \mathtt{H}^* \models \varphi_{\mathtt{dec}}[f, v, w] \quad &\Longleftrightarrow \quad f \in \mathtt{F},\ v, w \in \mathtt{V}_f \text{ and } w \in \mathtt{dec}_f(v). \end{aligned} \tag{6.11}$$

For $i \in [2]$ bundle extension $\mathtt{H}^*$ allows us to define a relation $\prec_{f_i}$ for $\mathtt{H}$. For $v, w \in \mathtt{V}$, we let $v \prec_{f_i} w$ if and only if one of the following holds:

- $v, w \in \mathtt{V}_{f_i}$ and $v \in \mathtt{anc}_{f_i}(w) \setminus \{w\}$,
- $v, w \in \mathtt{V}_{f_{3-i}}$ and $v \in \mathtt{dec}_{f_i}(w) \setminus \{w\}$,
- $v \in \mathtt{S}_{f_i}$ and $w \in \mathtt{S}_{f_{3-i}}$, and there is a max clique $M \in \mathcal{M}_{\mathtt{H}}$ with $v, w \in M$.

It is not hard to see that $\prec_{f_{3-i}}$ is exactly the reverse relation $(\prec_{f_i})_R$ of relation $\prec_{f_i}$. If we restrict $\prec_{f_i}$ to the vertex set $\mathtt{V}$, we obtain the strict partial order $\prec_i$ for G from the end of Section 6.3.10. Further, $f_i \prec_{f_i} v$ for all $v \in \mathtt{V} \setminus \{f_i\}$, and $v \prec_{f_i} f_{3-i}$ for all $v \in \mathtt{V} \setminus \{f_{3-i}\}$. Therefore, we obtain the following corollary.

Corollary 163. *Ordering $\prec_f$ is a strict partial order for $\mathtt{H}$ for all $f \in \mathtt{F}$.*

Observation 164. *Let $f \in \mathtt{F}$ and $v, w \in \mathtt{V}$. If $w \in \mathtt{V}_f$ and $v \prec_f w$, then $v \in \mathtt{V}_f$ and v is a proper ancestor of w in $\mathtt{T}_f$.*

Proof. Let $i \in [2]$ and $v, w \in \mathtt{V}$. Let $w \in \mathtt{V}_{f_i}$ and $v \prec_{f_i} w$. Since $w \notin \mathtt{V}_{f_{3-i}} \setminus \mathtt{O} = \mathtt{S}_{f_{3-i}}$ (Observation 158), the definition of $\prec_{f_i}$ implies that $v, w \in \mathtt{V}_{f_i}$ and $v \in \mathtt{anc}_{f_i}(w) \setminus \{w\}$, or $v, w \in \mathtt{V}_{f_{3-i}}$ and $v \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$. Suppose $v, w \in \mathtt{V}_{f_{3-i}}$ and $v \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$. Then $w \in \mathtt{O}$ by Observation 158. Thus, w is a leaf of $\mathtt{T}_{f_{3-i}}$, and we obtain a contradiction as v cannot be a proper descendant of w in $\mathtt{T}_{f_{3-i}}$. It follows that $v \in \mathtt{V}_{f_i}$ and $v \in \mathtt{anc}_{f_i}(w) \setminus \{w\}$. □

Observation 165. *Let $i \in [2]$ and $v, w \in \mathtt{V}$. Then $v \prec_{f_i} w$ if, and only if, one of the following holds:*

- *$v, w \in \mathtt{V}_{f_i}$ and $v \in \mathtt{anc}_{f_i}(w) \setminus \{w\}$,*
- *$v, w \in \mathtt{V}_{f_{3-i}}$ and $v \in \mathtt{dec}_{f_i}(w) \setminus \{w\}$,*
- *$v \in \mathtt{S}_{f_i}$ and $w \in \mathtt{S}_{f_{3-i}}$, and there is a vertex $o \in \mathtt{O}$ such that $v \in \mathtt{anc}_{f_i}(o) \setminus \{o\}$ and $o \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$.*

Proof. Let $i \in [2]$ and $v, w \in \mathtt{V}$. Let $v \in \mathtt{S}_{f_i}$ and $w \in \mathtt{S}_{f_{3-i}}$. We only need to show that there is a vertex $o \in \mathtt{O}$ such that $v \in \mathtt{anc}_{f_i}(o) \setminus \{o\}$ and $o \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$ if, and only if, there is a max clique $M \in \mathcal{M}_{\mathtt{H}}$ with $v, w \in M$.

First, let there be a max clique $M \in \mathcal{M}_{\mathtt{H}}$ with $v, w \in M$. We show that there is a vertex $o \in \mathtt{O}$ such that $v \in \mathtt{anc}_{f_i}(o) \setminus \{o\}$ and $o \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$. By Corollary 162 we have $M \cap \mathtt{O} \neq \emptyset$. Let $o \in M \cap \mathtt{O}$. Then v and o are adjacent. Thus, $v \prec_{f_i} o$ or $o \prec_{f_i} v$. Since v and o are in $\mathtt{V}_{f_i}$, vertex v is a proper ancestor of o or vertex o a proper ancestor of v in $\mathtt{T}_{f_i}$ (Observation 164). As $o \in \mathtt{O}$ is a leaf of $\mathtt{T}_{f_i}$, we have $v \in \mathtt{anc}_{f_i}(o) \setminus \{o\}$. Analogously, we can show that $o \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$.

Now let there be a vertex $o \in \mathtt{O}$ such that $v \in \mathtt{anc}_{f_i}(o) \setminus \{o\}$ and $o \in \mathtt{dec}_{f_{3-i}}(w) \setminus \{w\}$. We prove that there is a max clique $M \in \mathcal{M}_{\mathtt{H}}$ with $v, w \in M$. According to Lemma 161 there is only one max clique that contains vertex o. Let M be this max clique. Thus, each vertex that is adjacent to o is contained in max clique M. As v and o are in $\mathtt{V}_{f_i}$ and $v \in \mathtt{anc}_{f_i}(o) \setminus \{o\}$, we have $v \prec_{f_i} o$. Thus, v and o are adjacent, and $v \in M$. Similarly, we can show that $w \in M$. □

As there are formulas for $\mathtt{F}$, $\mathtt{V}_f$, $\mathtt{S}_f$, $\mathtt{O}$, $\mathtt{anc}_f(v)$ and $\mathtt{dec}_f(v)$ (cf. (6.10) and (6.11)), it is easy to see that there exists a TC-formula $\varphi_{\prec}(x^*, x_1, x_2)$ such that for elements $f, v_1, v_2 \in \mathtt{U}$ of bundle extension $\mathtt{H}^*$ we have

$$\mathtt{H}^* \models \varphi_{\prec}[f, v_1, v_2] \iff f \in \mathtt{F},\ v_1, v_2 \in \mathtt{V} \text{ and } v_1 \prec_f v_2. \tag{6.12}$$

The following corollary follows immediately from Corollary 139 and the structure of $\mathtt{H}^*$.

Corollary 166. *The edge relation of the induced subgraph $\mathtt{H}[\mathtt{V}_f]$ is the symmetric closure of the transitive closure of the edge relation of $\mathtt{T}_f$ for all $f \in \mathtt{F}$.*

Corollary 167. *The set $\mathtt{O}$ is an independent set.*

Lemma 168. *Let $f \in \mathtt{F}$ and $v, w \in \mathtt{V}_f$. Then $w \in \mathtt{anc}_f(v)$ if, and only if, $v \in A$ implies $w \in A$ for all max cliques $A \in \mathcal{M}_{\mathtt{H}}$.*

Proof. Let $f \in \mathtt{F}$ and $v, w \in \mathtt{V}_f$. First, let us consider $w = f$. Then $w \in \mathtt{anc}_f(v)$ and as w is contained in all max cliques $A \in \mathcal{M}_{\mathtt{H}}$, $v \in A$ implies $w \in A$ for all $A \in \mathcal{M}_{\mathtt{H}}$. Next, let us consider $v = f$. Clearly, $\mathtt{anc}_f(f) = \{f\}$ and by Observation 158 and Lemma 161 vertex f is the only vertex in $\mathtt{V}_f$ that is contained in all max cliques $A \in \mathcal{M}_{\mathtt{H}}$. Now, let us consider $v \neq f$ and $w \neq f$. Let $f = f_i$ with $i \in [2]$. If there is an $e \in F_i$ such that $v, w \in V_e$, then Lemma 168 follows from Lemma 140, and the fact that A is a max clique of G if and only if $A \cup \mathtt{F}$ is a max clique of $\mathtt{H}$. If there is no $e \in F_i$ such that $v, w \in V_e$, then there exist $e_1, e_2 \in F_i$ with $e_1 \neq e_2$ such that $v \in V_{e_1}$ and $w \in V_{e_2}$. Then $w \notin \mathtt{anc}_{f_i}(v)$ by Lemma 132. Let us assume $v \in A$ implies $w \in A$ for all max cliques $A \in \mathcal{M}_{\mathtt{H}}$. Clearly, there exists a max clique $A \in \mathcal{M}_{\mathtt{H}}$ with $v \in A$. Thus, $w \in A$. According to Corollary 131 the max clique $A \setminus \mathtt{F}$ of G has e_1 and e_2 as ends. It follows that e_1 and e_2 are adjacent in $G[F]$, a contradiction to $e_1, e_2 \in F_i$. □

Lemma 161 and 168 imply the following corollary.

Corollary 169. *Let $v \in \mathtt{O}$. Then v spans the max clique $\mathtt{anc}_{f_1}(v) \cup \mathtt{anc}_{f_2}(v)$.*

6.4.3. Subbundle Pairs

We now introduce subbundle pairs. Subbundle pairs play a major role within the construction of the decomposition tree in Section 6.5.

Let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be a bundle extension, and $\mathtt{H} = (\mathtt{V}, \mathtt{E})$ be its underlying graph. Let $a, b \in \mathtt{V}$. We call the binary multiset $[a, b]$ a *subbundle pair* of $\mathtt{H}$ if $a, b \in \mathtt{V}$ and either $a = b$ or there is an edge between a and b. Of course, if $[a, b]$ is a subbundle pair of $\mathtt{H}$, then so is $[b, a]$. We let $\mathtt{P}_{\text{sub}}$ be the set of all subbundle pairs of $\mathtt{H}$. Clearly, there exists an FO-formula $\varphi_{\text{sub}}(x_1, x_2)$ that decides whether $[a_1, a_2]$ is a subbundle pair of $\mathtt{H}$ for $a_1, a_2 \in \mathtt{V}$ and $\mathtt{H}^* \in \mathtt{C}^*_{\text{ChCo}}$. We call a subbundle pair $[a, b]$ of $\mathtt{H}$ *trivial* if there is a terminus $f \in \mathtt{F}$ such that $a, b \in \mathtt{V}_f$, and *non-trivial* otherwise. Then, a subbundle pair $[a, b]$ of $\mathtt{H}$ is non-trivial if there is an $i \in [2]$ such that $a \in \mathtt{S}_{f_i}$ and $b \in \mathtt{S}_{f_{3-i}}$.

Let $[a, b]$ be a subbundle pair of $\mathtt{H}$. If there is an edge between a and b in $\mathtt{H}$, then for all $f \in \mathtt{F}$ either $a \prec_f b$ or $b \prec_f a$ as order $\prec_f$ is a strict partial order for $\mathtt{H}$. Therefore, we have $a \preceq_f b$ or $b \preceq_f a$ for all subbundle pairs $[a, b]$ of $\mathtt{H}$ and all $f \in \mathtt{F}$. Let us fix a terminus $f \in \mathtt{F}$. We let $\mathtt{V}([a, b])$ be the set of vertices $v \in \mathtt{V}$ where $a \preceq_f v \preceq_f b$ or $b \preceq_f v \preceq_f a$. Since order $\prec_{f_{3-i}}$ is the reverse of $\prec_{f_i}$ for $i \in [2]$, the definition of $\mathtt{V}([a, b])$ does not depend on the terminus $f \in \mathtt{F}$. Obviously, if $a = b$, then $\mathtt{V}([a, b]) = \{a\}$. Further, if $a \prec_f b$, then there do not exist vertices v with $b \preceq_f v \preceq_f a$. Thus, if $a \preceq_f b$ for subbundle pair $[a, b]$, then $\mathtt{V}([a, b]) = \{v \in \mathtt{V} \mid a \preceq_f v \preceq_f b\}$. We also denote $\mathtt{V}([a, b])$ by $\mathtt{V}(a, b)$. Since $\prec_f$ is definable in transitive closure logic, we also have a TC-formula $\varphi_{\mathtt{V}(\cdot,\cdot)}(x_1, x_2, y)$ that defines the set $\mathtt{V}(a, b)$. Hence, for a bundle extension $\mathtt{H}^*$ and elements $a, b, v \in \mathtt{U}$ of $\mathtt{H}^*$, it holds that

$$\mathtt{H}^* \models \varphi_{\mathtt{V}(\cdot,\cdot)}[a, b, v] \iff [a, b] \text{ is a subbundle pair of } \mathtt{H} \text{ and } v \in \mathtt{V}(a, b). \tag{6.13}$$

Observation 170. *Let $[a, b]$ be a trivial subbundle pair of $\mathtt{H}$ where $a, b \in \mathtt{V}_f$ with $f \in \mathtt{F}$. Then $\mathtt{V}(a, b) \subseteq \mathtt{V}_f$ and $\mathtt{V}(a, b)$ induces a directed path from a to b or from b to a in $\mathtt{T}_f$.*

Proof. Let $[a, b]$ be a trivial subbundle pair of $\mathtt{H}$ where $a, b \in \mathtt{V}_f$ with $f \in \mathtt{F}$. If $a = b$, then $\mathtt{V}(a, b) = \{a\}$. Thus, $\mathtt{V}(a, b) \subseteq \mathtt{V}_f$ and $\mathtt{V}(a, b)$ induces a directed path from a to b in $\mathtt{T}_f$ in this case. Let $a \neq b$. Then a and b are adjacent, and either $a \prec_f b$ or $b \prec_f a$. Without loss of generality, let $a \prec_f b$. Then Observation 164 and the definition of $\prec_f$ imply that for all $v \in \mathtt{V}$ we have $a \preceq_f v \preceq_f b$ if and only if vertex v is a descendant of a and an ancestor of b in the directed tree $\mathtt{T}_f$. Since $\mathtt{V}(a, b) = \{v \in \mathtt{V} \mid a \preceq_f v \preceq_f b\}$, the set $\mathtt{V}(a, b)$ is a subset of $\mathtt{V}_f$ and $\mathtt{V}(a, b)$ induces a directed path from a to b in $\mathtt{T}_f$. ☐

Observation 171. *Let $i \in [2]$. Let $[a, b]$ be a subbundle pair of $\mathtt{H}$ with $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{3-i}}$. Then $a \preceq_{f_i} b$.*

Proof. Let $i \in [2]$ and let $[a, b]$ be a subbundle pair of $\mathtt{H}$ with $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{3-i}}$. Clearly, we have $a \preceq_{f_i} b$ if $a = b$. Thus, let a and b be adjacent. Then $a \prec_{f_i} b$ or $b \prec_{f_i} a$. Let us suppose we have $b \prec_{f_i} a$. Then $b \in \mathtt{V}_{f_i}$ and b is a proper ancestor of a in $\mathtt{T}_{f_i}$ by Observation 164. As $b \in \mathtt{V}_{f_i} \cap \mathtt{V}_{f_{3-i}} = \mathtt{O}$ (Observation 158), b is a leaf of $\mathtt{T}_{f_i}$ and we obtain a contradiction. Hence, $a \prec_{f_i} b$. ☐

Let $f \in \mathtt{F}$. For $a \in \mathtt{V}_f$ we let $\mathtt{O}_a$ be the set of all leaves $v \in \mathtt{O}$ such that there is a path from a to v in $\mathtt{T}_f$. Let $[a, b]$ be a non-trivial subbundle pair of $\mathtt{H}$. Then $a \neq b$ and there is an $i \in [2]$ such that $a \in \mathtt{S}_{f_i}$ and $b \in \mathtt{S}_{f_{3-i}}$. For $f \in \mathtt{F}$ let $r_f(a, b) := a$ if $a \in \mathtt{S}_f$ and $r_f(a, b) := b$ if $b \in \mathtt{S}_f$. We let $\mathtt{T}_f(a, b)$ be the unique subtree of $\mathtt{T}_f$ that has $r_f(a, b)$ as root and $O_a \cap O_b$ as set of leaves. We denote the vertex set of $\mathtt{T}_f(a, b)$ by $\mathtt{V}_f(a, b)$. Thus, if $a \in \mathtt{S}_{f_i}$ for $i \in [2]$, then $\mathtt{V}_{f_i}(a, b)$ consists of all vertices that lie on a path from a to a vertex $o \in O_a \cap O_b$. Clearly, $\mathtt{V}_f(a, b)$ is a subset of $\mathtt{V}_f$. Further, the set of leaves of $\mathtt{T}_f(a, b)$ is $\mathtt{O}_a \cup \mathtt{O}_b$ for each $f \in \mathtt{F}$, and no vertex from $\mathtt{O} \setminus (\mathtt{O}_a \cup \mathtt{O}_b)$ is contained in $\mathtt{V}_f(a, b)$. Then Observation 158 implies the following observation.

Observation 172. *Let $[a, b] \in \mathtt{P}_{\mathrm{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$. Then $\mathtt{V}_{f_1}(a, b) \cap \mathtt{V}_{f_2}(a, b) = \mathtt{O}_a \cap \mathtt{O}_b$.*

Lemma 173. *Let $[a, b] \in \mathtt{P}_{\mathrm{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$. Then $\mathtt{V}(a, b) = \mathtt{V}_{f_1}(a, b) \cup \mathtt{V}_{f_2}(a, b)$.*

Proof. Let $[a, b]$ be a non-trivial subbundle pair of $\mathtt{H}$. Without loss of generality, let $a \in \mathtt{S}_{f_1}$ and $b \in \mathtt{S}_{f_2}$. Then $a \prec_{f_1} b$ by Observation 171.

First, let us prove that $\mathtt{V}(a, b) \subseteq \mathtt{V}_{f_1}(a, b) \cup \mathtt{V}_{f_2}(a, b)$. Let $v \in \mathtt{V}(a, b)$. Then $a \preceq_{f_1} v \preceq_{f_1} b$. Let us consider the case where $v \in \mathtt{O}$. Then v is in $\mathtt{V}_{f_1}$, and $a \preceq_{f_1} v$ implies that a is an ancestor of v in $\mathtt{T}_{f_1}$ (Observation 164). Further, v is contained in $\mathtt{V}_{f_2}$, and $v \preceq_{f_1} b$ yields that v is an descendant of b in $\mathtt{T}_{f_2}$. Hence, $v \in \mathtt{O}_a \cap \mathtt{O}_b$. Clearly, v lies on the path from a to v in $\mathtt{T}_{f_1}$. Thus, $v \in \mathtt{V}_{f_1}(a, b)$. Now let us consider the case where $v \in \mathtt{S}_{f_1}$. Vertex a is an ancestor of v in $\mathtt{T}_{f_1}$, because $a \preceq_{f_1} v$. Since $v \in \mathtt{S}_{f_1}$, $b \in \mathtt{S}_{f_2}$, we have $v \neq b$ (Observation 158). Thus, $v \prec_{f_1} b$, and Observation 165 implies that there exists an $o \in O$ such that v is an ancestor of o in $\mathtt{T}_{f_1}$ and b is an ancestor of o in $\mathtt{T}_{f_2}$. Then a is an ancestor of o in $\mathtt{T}_{f_1}$. Thus, $o \in \mathtt{O}_a \cap \mathtt{O}_b$, and vertex v lies on the path from a to o in $\mathtt{T}_{f_1}$. Hence, $v \in \mathtt{V}_{f_1}(a, b)$. Analogously, we can show that $v \in \mathtt{S}_{f_2}$ implies $v \in \mathtt{V}_{f_2}(a, b)$. Consequently, $\mathtt{V}(a, b)$ is a subset of $\mathtt{V}_{f_1}(a, b) \cup \mathtt{V}_{f_2}(a, b)$.

Next, we prove that $\mathtt{V}_{f_1}(a, b) \subseteq \mathtt{V}(a, b)$. Let $v \in \mathtt{V}_{f_1}(a, b)$. Then there is a vertex $o \in \mathtt{O}_a \cap \mathtt{O}_b$ such that v lies on the path from a to o in $\mathtt{T}_{f_1}$. Hence, a is an ancestor of v and v is an ancestor of o in $\mathtt{T}_{f_1}$. Moreover, since $o \in \mathtt{O}_b$, vertex o is a descendant of b in $\mathtt{T}_{f_2}$. We obtain that $a \preceq_{f_1} v \preceq_{f_1} o \preceq_{f_1} b$. As $\preceq_{f_1}$ is transitive, we have $a \preceq_{f_1} v \preceq_{f_1} b$. It follows that $v \in \mathtt{V}(a,b)$. Analogously, it can be shown that $\mathtt{V}_{f_2}(a,b) \subseteq \mathtt{V}(a,b)$. □

Corollary 174. *Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$. Then for all $f \in \mathtt{F}$ we have $\mathtt{V}(a,b) \cap \mathtt{V}_f = \mathtt{V}_f(a,b)$.*

Proof. Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of $\mathtt{H}$. As $\mathtt{V}_f(a,b) \subseteq \mathtt{V}_f$ for each $f \in \mathtt{F}$ and $\mathtt{V}_f(a,b) \subseteq \mathtt{V}(a,b)$ by Lemma 173 we have $\mathtt{V}_f(a,b) \subseteq \mathtt{V}(a,b) \cap \mathtt{V}_f$. Lemma 173 also implies that $\mathtt{V}(a,b) \cap \mathtt{V}_f \subseteq \mathtt{V}_{f_1}(a,b) \cup \mathtt{V}_{f_2}(a,b)$. Without loss of generality, let $f = f_1$ Let us assume there is a vertex $v \in \mathtt{V}(a,b) \cap \mathtt{V}_{f_1}$ that is contained in $\mathtt{V}_{f_2}(a,b) \setminus \mathtt{V}_{f_1}(a,b)$. Then $v \in \mathtt{V}_{f_1}$ and $v \in \mathtt{V}_{f_2}(a,b) \subseteq \mathtt{V}_{f_2}$. Hence, $v \in \mathtt{O}$ by Observation 158. Since $\mathtt{V}_{f_2}(a,b)$ does not contain any vertices from $\mathtt{O} \setminus (\mathtt{O}_a \cup \mathtt{O}_b)$, vertex v is in $\mathtt{O}_a \cup \mathtt{O}_b$ and therefore also in $\mathtt{V}_{f_1}(a,b)$, a contradiction. It follows that $\mathtt{V}(a,b) \cap \mathtt{V}_{f_1} \subseteq \mathtt{V}_{f_1}(a,b)$. □

Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of $\mathtt{H}$. We let $\mathtt{V}^{-}([a,b])$ (or short $\mathtt{V}^{-}(a,b)$) be the set $\mathtt{V}(a,b) \setminus \{a,b\}$. Then, $\mathtt{V}^{-}(a,b) = \{v \in \mathtt{V} \mid \exists f \in \mathtt{F}\colon a \prec_f v \prec_f b\}$. Let $f \in \mathtt{F}$. If $a \preceq_f b$, it follows that $\mathtt{V}^{-}(a,b) = \{v \in \mathtt{V} \mid a \prec_f v \prec_f b\}$. Further, we let $\mathtt{V}^{-}_f(a,b)$ be the set $\mathtt{V}_f(a,b) \setminus \{r_f(a,b)\}$, and we let $\mathtt{T}^{-}_f(a,b)$ be the subgraph of $\mathtt{T}_f(a,b)$ induced by $\mathtt{V}^{-}_f(a,b)$. Then $\mathtt{T}^{-}_f(a,b)$ is a directed forest. Let $\mathtt{C}_f(a,b)$ be the set of children of $r_f(a,b)$ in $\mathtt{T}_f(a,b)$. The set $\mathtt{C}_f(a,b)$ is the set of roots of the directed forest $\mathtt{T}^{-}_f(a,b)$.

For each vertex $e \in \mathtt{C}_f(a,b)$ let $\mathtt{V}^e_f(a,b)$ be the connected component of $\mathtt{T}^{-}_f(a,b)$ with $e \in \mathtt{V}^e_f(a,b)$. Let $\mathtt{T}^e_f(a,b)$ be the subtree induced by $\mathtt{V}^e_f(a,b)$ in $\mathtt{T}_f$, and let $\mathtt{O}^e_f(a,b)$ be the set of leaves of $\mathtt{T}^e_f(a,b)$. Clearly, e is the root of $\mathtt{T}^e_f(a,b)$, and the set of leaves $\mathtt{O}^e_f(a,b)$ is a subset of $\mathtt{O}_a \cap \mathtt{O}_b = \mathtt{V}^{-}(a,b) \cap \mathtt{O}$. Further, $\mathtt{V}^{-}_f(a,b)$ is the union of all sets $\mathtt{V}^e_f(a,b)$ where $e \in \mathtt{C}_f(a,b)$. We can easily observe the following.

Observation 175. *Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$, and let $f \in \mathtt{F}$. For all $e_1, e_2 \in \mathtt{C}_f(a,b)$ with $e_1 \neq e_2$, the sets $\mathtt{V}^{e_1}_f(a,b)$ and $\mathtt{V}^{e_2}_f(a,b)$ are disjoint.*

Observation 176. *Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$. Let $i \in [2]$. Then*

$$\bigcup_{e \in \mathtt{C}_{f_i}(a,b)} \mathtt{O}^e_{f_i}(a,b) \;=\; \mathtt{O}_a \cap \mathtt{O}_b.$$

Observation 172 and Corollary 174 imply the following two corollaries.

Corollary 177. *Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$. For all $e_1 \in \mathtt{C}_{f_1}(a,b)$ and all $e_2 \in \mathtt{C}_{f_2}(a,b)$, we have $\mathtt{V}^{e_1}_{f_1}(a,b) \cap \mathtt{V}^{e_2}_{f_2}(a,b) \subseteq \mathtt{O}_a \cap \mathtt{O}_b$.*

Corollary 178. *Let $[a,b] \in \mathtt{P}_{\text{sub}}$ be a non-trivial subbundle pair of the graph $\mathtt{H}$. Then for all $f \in \mathtt{F}$ we have*

$$\mathtt{V}^{-}(a,b) \cap \mathtt{V}_f \;=\; \bigcup_{e \in \mathtt{C}_f(a,b)} \mathtt{V}^e_f(a,b).$$

6.4.4. Valid Triples and (Extended) Valid Subgraphs

For each subbundle pair $[a, b]$, let $\mathtt{H}^{-}(a, b)$ be the graph $\mathtt{H}[\mathtt{V}^{-}(a, b)]$. The graph $\mathtt{H}^{-}(a, b)$ is not necessarily connected. We call $([a, b], c)$ a *valid triple* of $\mathtt{H}$ if $[a, b]$ is a non-trivial subbundle pair of $\mathtt{H}$ and $c \in \mathtt{V}^{-}(a, b)$. It is not hard to see that formula

$$\begin{aligned}\theta_{\mathrm{dom}}(x_1, x_2, x_3) := \varphi_{\mathtt{V}(\cdot,\cdot)}(x_1, x_2, x_3) \wedge \neg x_1 = x_3 \wedge \neg x_2 = x_3 \\ \wedge \neg \exists x^* \big(\varphi_{\mathtt{V}}(x^*, x_1) \wedge \varphi_{\mathtt{V}}(x^*, x_2)\big)\end{aligned}$$

defines all valid triples, where $\varphi_{\mathtt{V}}$ and $\varphi_{\mathtt{V}(\cdot,\cdot)}$ are the formulas from (6.10) and (6.13).

For every valid triple $([a, b], c)$ we let $V_{([a,b],c)}$ be the connected component of $\mathtt{H}^{-}(a, b)$ that contains vertex c. The induced subgraph $H_{([a,b],c)} := \mathtt{H}[V_{([a,b],c)}]$ is a chordal comparability graph as the class of chordal comparability graphs is closed under induced subgraphs. Note that $H_{([a,b],c)}$ is not necessarily prime. We call $H_{([a,b],c)}$ the *valid subgraph* of $\mathtt{H}$ defined by the valid triple $([a, b], c)$. Let $E_{([a,b],c)}$ be the edge relation of $H_{([a,b],c)}$. Since we already have a TC-formula $\varphi_{\mathtt{V}(\cdot,\cdot)}$ that defines $\mathtt{V}(a, b)$ and connectivity is expressible in STC, the graph $H_{([a,b],c)}$ is definable in transitive closure logic. Thus, there exist TC-formulas $\theta_V(x_1, x_2, x_3, y_1)$ and $\theta_E(x_1, x_2, x_3, y_1, y_2)$ such that for all elements $v_1, v_2 \in \mathtt{U}$ of a bundle extensions $\mathtt{H}^* \in \mathtt{C}^*_{\mathrm{ChCo}}$ and for all valid triples $([a, b], c)$ of the underlying graph $\mathtt{H}$ of $\mathtt{H}^*$ we have

$$\begin{aligned}\mathtt{H}^* \models \theta_V[a, b, c, v_1] &\iff v_1 \in V_{([a,b],c)}, \quad \text{and}\\ \mathtt{H}^* \models \theta_E[a, b, c, v_1, v_2] &\iff \{v_1, v_2\} \in E_{([a,b],c)}.\end{aligned}$$

Thus, $\Theta_{\mathrm{val}} = (\theta_{\mathrm{dom}}, \theta_V, \theta_E)$ is a parameterized TC-transduction, that assigns each bundle extension $\mathtt{H}^*$ and valid triple $([a, b], c)$ of the underlying graph $\mathtt{H}$ of $\mathtt{H}^*$ to the valid subgraph $H_{([a,b],c)}$ of $\mathtt{H}$ defined by $([a, b], c)$.

We need valid subgraphs to create a decomposition tree, which we use for canonization. To create the decomposition tree, we need more of the information we already have about the structure of these valid subgraphs. Hence, we define extended valid subgraphs and show different properties of extended valid subgraphs. Extended valid subgraphs additionally include a strict partial order (and its reverse) for the valid subgraph, and side trees.

Let $([a, b], c)$ be a valid triple. Thus, $[a, b]$ is a non-trivial subbundle pair.

Since for all $f \in \mathtt{F}$ and all $e \in \mathtt{C}_f(a, b)$ the tree $\mathtt{T}^e_f(a, b)$ is a subtree of $\mathtt{T}_f$ and $H_{([a,b],c)}$ is an induced subgraph of $\mathtt{H}$ with $\mathtt{V}^e_f(a, b) \subseteq V_{([a,b],c)}$, Corollary 166 implies the following corollary.

Corollary 179. *The edge relation of the induced subgraph $H_{([a,b],c)}[\mathtt{V}^e_f(a, b)]$ is the symmetric closure of the transitive closure of the edge relation of $\mathtt{T}^e_f(a, b)$ for all $f \in \mathtt{F}$ and all $e \in \mathtt{C}_f(a, b)$.*

Let $f \in \mathtt{F}$. By Observation 175 and Corollary 178 we know that $\mathtt{V}^{-}(a, b) \cap \mathtt{V}_f$ is the disjoint union of all sets $\mathtt{V}^e_f(a, b)$ with $e \in \mathtt{C}_f(a, b)$. By Corollary 179 the set $\mathtt{V}^e_f(a, b)$ of vertices of the subtree $\mathtt{T}^e_f(a, b)$ of $\mathtt{T}_f$ is connected in $H_{([a,b],c)}$ for every $e \in \mathtt{C}_f(a, b)$. Hence, the connected component $V_{([a,b],c)}$ of $\mathtt{H}^{-}(a, b)$ that contains vertex c is the union of several connected sets $\mathtt{V}^e_f(a, b)$ with $e \in \mathtt{C}_f(a, b)$ and $f \in \mathtt{F}$. We obtain the following corollary.

Corollary 180. *Let $([a,b],c)$ be a valid triple. For each $f \in \mathtt{F}$ there is a unique subset $\mathtt{C}_f \subseteq \mathtt{C}_f(a,b)$ such that*

$$V_{([a,b],c)} \cap \mathtt{V}_f = \dot{\bigcup_{e \in \mathtt{C}_f}} \mathtt{V}^e_f(a,b).$$

Corollary 181. *Let $([a,b],c)$ be a valid triple. Let $f \in \mathtt{F}$. Then $\mathtt{C}_f \neq \emptyset$.*

Proof. Let $([a,b],c)$ be a valid triple, and let $f \in \mathtt{F}$. As $[a,b]$ is a non-trivial subbundle pair, we have $a \in \mathtt{V}_f$ or $b \in \mathtt{V}_f$. Thus, $V_{([a,b],c)} \cap \mathtt{V}_f \neq \emptyset$, and it follows from Corollary 180 that $\mathtt{C}_f \neq \emptyset$. □

For every $f \in \mathtt{F}$ let $\mathtt{C}_f$ be the unique subset mentioned in Corollary 180. We call each vertex $e \in \mathtt{C}_f$ with $f \in \mathtt{F}$ an *(inherited) terminus* of $H_{([a,b],c)}$. We let $F_{([a,b],c)} = \mathtt{C}_{f_1} \cup \mathtt{C}_{f_2}$ be the set of all inherited termini of $H_{([a,b],c)}$. We call the directed tree $\mathtt{T}^e_f(a,b)$ for $e \in F_{([a,b],c)}$ the *(inherited) side tree of* e of $H_{([a,b],c)}$, and the set $\mathtt{O}^e_f(a,b)$ the *(inherited) middle of* e of $H_{([a,b],c)}$. We let the *(inherited) middle* $O_{([a,b],c)}$ of $H_{([a,b],c)}$ be the union of all sets $\mathtt{O}^e_f(a,b)$ with $f \in \mathtt{F}$ and $e \in \mathtt{C}_f$. Clearly, we can observe the following.

Observation 182. *Let $([a,b],c)$ be a valid triple. Then $O_{([a,b],c)} = \mathtt{O} \cap V_{([a,b],c)}$.*

Lemma 183. *Let $([a,b],c)$ be a valid triple. Let $f \in \mathtt{F}$. Then*

$$O_{([a,b],c)} = \bigcup_{e \in \mathtt{C}_f} \mathtt{O}^e_f(a,b).$$

Proof. Let $([a,b],c)$ be a valid triple. Let $f \in \mathtt{F}$. Without loss of generality, let $f = f_2$. Clearly, $\bigcup_{e \in \mathtt{C}_{f_2}} \mathtt{O}^e_{f_2}(a,b) \subseteq O_{([a,b],c)}$. Let us assume $O_{([a,b],c)} \not\subseteq \bigcup_{e \in \mathtt{C}_{f_2}} \mathtt{O}^e_{f_2}(a,b)$. Then there is a root $e_1 \in \mathtt{C}_{f_1}$ and a vertex $v \in \mathtt{O}^{e_1}_{f_1}(a,b)$ such that for all $e \in \mathtt{C}_{f_2}$ we have $v \notin \mathtt{O}^e_{f_2}(a,b)$. According to Observation 176 there is a root $e_2 \in \mathtt{C}_{f_2}(a,b)$ such that $v \in \mathtt{O}^{e_2}_{f_2}(a,b)$. As $v \in \mathtt{O}^{e_1}_{f_1}(a,b) \subseteq \mathtt{V}^{e_1}_{f_1}(a,b)$ and $e_1 \in \mathtt{C}_{f_1}$, it follows from Corollary 180 that $v \in V_{([a,b],c)}$. Since $v \in \mathtt{O}^{e_2}_{f_2}(a,b) \subseteq \mathtt{V}^{e_2}_{f_2}(a,b)$ and $\mathtt{V}^{e_2}_{f_2}(a,b)$ is connected in $\mathtt{H}^-(a,b)$, the set $\mathtt{V}^{e_2}_{f_2}(a,b)$ is a subset of the connected component $V_{([a,b],c)}$. Further, $\mathtt{V}^{e_2}_{f_2}(a,b) \subseteq \mathtt{V}_{f_2}$ (Corollary 178). Hence, Observation 175 and Corollary 180 imply that $e_2 \in \mathtt{C}_{f_2}$, a contradiction. □

A strict partial order for the underlying graph $\mathtt{H}$ of a bundle extension yields a strict partial order for every induced subgraph of $\mathtt{H}$. For the strict partial order $\prec_{f_i}$ for $\mathtt{H}$, the restriction $\prec_{f_i,([a,b],c)}$ of $\prec_{f_i}$ to the vertex set of the valid subgraph $H_{([a,b],c)}$ is a strict partial order for $H_{([a,b],c)}$. We call $\prec_{f_1,([a,b],c)}$ and $\prec_{f_1,([a,b],c)}$ the *inherited strict partial orders for* $H_{([a,b],c)}$.

Analogously to subbundle pairs of $\mathtt{H}$, we define subbundle pairs of the induced subgraph $H_{([a,b],c)}$. Thus, $[x,y]$ is a *subbundle pair* of $H_{([a,b],c)}$ if and only if $x = y$ or x and y are adjacent in $H_{([a,b],c)}$. As $H_{([a,b],c)}$ is an induced subgraph of $\mathtt{H}$, the multiset $[x,y]$ is a subbundle pair of $H_{([a,b],c)}$ precisely if $[x,y]$ is a subbundle pair of $\mathtt{H}$, for all $x,y \in V_{([a,b],c)}$. We let $V_{([a,b],c)}(x,y)$ be the set of all $v \in V_{([a,b],c)}$ for which there is an $i \in [2]$ such that $x \preceq_{f_i,([a,b],c)} v \preceq_{f_i,([a,b],c)} y$.

Lemma 184. *Let $([a,b],c)$ be a valid triple and $[x,y]$ be a subbundle pair of $H_{([a,b],c)}$. Then $V_{([a,b],c)}(x,y) = \mathtt{V}(x,y)$.*

Proof. Let $([a,b],c)$ be a valid triple and $[x,y]$ be a subbundle pair of $H_{([a,b],c)}$. First let us show that $V_{([a,b],c)}(x,y) \subseteq \mathtt{V}(x,y)$. Let $v \in V_{([a,b],c)}(x,y)$. Then $v \in V_{([a,b],c)}$ and there is an $i \in [2]$ such that $x \preceq_{f_i,([a,b],c)} v \preceq_{f_i,([a,b],c)} y$. Since $V_{([a,b],c)} \subseteq \mathtt{V}$ and $\preceq_{f_i,([a,b],c)}$ is the restriction of $\preceq_{f_i}$ to the vertex set $V_{([a,b],c)}$, it follows that $v \in \mathtt{V}$ and there is an $i \in [2]$ such that $x \preceq_{f_i} v \preceq_{f_i} y$. Hence, $v \in \mathtt{V}(x,y)$.

Next let us prove that $\mathtt{V}(x,y) \subseteq V_{([a,b],c)}(x,y)$. Let $v \in \mathtt{V}(x,y)$. Then $v \in \mathtt{V}$ and there is an $i \in [2]$ such that $x \preceq_{f_i} v \preceq_{f_i} y$. Without loss of generality, let $i = 1$. As $x, y \in V_{([a,b],c)}$, we have $x, y \in \mathtt{V}^{-}(a,b)$. Thus, $a \prec_{f_1} x, y \prec_{f_1} b$ or $a \prec_{f_2} x, y \prec_{f_2} b$. Without loss of generality, let $a \prec_{f_1} x, y \prec_{f_1} b$. Then $a \prec_{f_1} x \preceq_{f_1} v \preceq_{f_1} y \prec_{f_1} b$, and by transitivity it follows that $a \prec_{f_1} v \prec_{f_1} b$. Hence, $v \in \mathtt{V}^{-}(a,b)$. Since $x \preceq_{f_1} v$, either $x = v$ or there is an edge between x and v. Consequently, x and v are in the same connected component of $\mathtt{H}^{-}(a,b)$, and therefore, $v \in V_{([a,b],c)}$. Now, $x \preceq_{f_1} v \preceq_{f_1} y$ implies that $x \preceq_{f_1,([a,b],c)} v \preceq_{f_1,([a,b],c)} y$. Hence, $v \in V_{([a,b],c)}(x,y)$. □

We already know that there is a TC-formula $\varphi_{\prec}(x^*, x_1, x_2)$ that is satisfied for elements $f, v_1, v_2 \in \mathtt{U}$ of bundle extension $\mathtt{H}^*$ if, and only if, $f \in \mathtt{F}$, $v_1, v_2 \in \mathtt{V}$ and $v_1 \prec_f v_2$ (cf. (6.12)). We slightly modify this formula, and we let

$$\theta_{\prec}(x, x_1, x_2) := \exists x^* \big(\varphi_{\mathtt{S}}(x^*, x) \land \varphi_{\prec}(x^*, x_1, x_2)\big).$$

Then, elements $v, v_1, v_2 \in \mathtt{U}$ of bundle extension $\mathtt{H}^*$ satisfy $\theta_{\prec}(x, x_1, x_2)$ if, and only if, $v, v_1, v_2 \in \mathtt{V}$ and there is an $f \in \mathtt{F}$ such that $v \in \mathtt{S}_f$ and $v_1 \prec_f v_2$. Let $\prec_{([a,b],c)}$ be the restriction of the relation defined by formula $\theta_{\prec}$ to the vertex set of $H_{([a,b],c)}$. As $H_{([a,b],c)}$ is connected, either $|V_{([a,b],c)}| = 1$ or $V_{([a,b],c)}$ contains at least one vertex from $\mathtt{S}_{f_1} \cup \mathtt{S}_{f_2}$ (Corollary 167). Since $\prec_{f_1,([a,b],c)}$ and $\prec_{f_2,([a,b],c)}$ are the reverse of each other, $\prec_{([a,b],c)}$, and therefore formula $\theta_{\prec}$, can be used to obtain the inherited strict partial orders $\prec_{f_1,([a,b],c)}$ and $\prec_{f_2,([a,b],c)}$ for $H_{([a,b],c)}$.

We let $T_{([a,b],c)}$ be the restriction of relation $\mathtt{T}$ to $V_{([a,b],c)}$. Then

$$H'_{([a,b],c)} = (V_{([a,b],c)}, E_{([a,b],c)}, \prec_{([a,b],c)}, T_{([a,b],c)})$$

is the *extended valid subgraph* of $\mathtt{H}$ defined by $([a,b],c)$.

We let

$$\theta_T(x_1, x_2, x_3, y_1, y_2) := \mathtt{T}(y_1, y_2).$$

Then we can use formulas θ_{dom}, θ_V, θ_E, $\theta_{\prec}$ and θ_T to define a parameterized TC-transduction

$$\Theta_{\text{val}}(x_1, x_2, x_3) = (\theta_{\text{dom}}, \theta_V, \theta_E, \theta_{\prec}, \theta_T) \tag{6.14}$$

that maps each bundle extension $\mathtt{H}^*$ and every valid triple $([a,b],c)$ of the underlying graph $\mathtt{H}$ of $\mathtt{H}^*$ to the extended valid subgraph $H'_{([a,b],c)}$ of $\mathtt{H}$ defined by $([a,b],c)$.

We now consider extended valid subgraphs and the properties they inherit from the bundle extension. Let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be a bundle extension with underlying graph $\mathtt{H}$. Let $([a,b],c)$ be a valid triple, and let $H' = (V, E, \prec, T)$ be the extended valid subgraph of $\mathtt{H}$ defined by $([a,b],c)$. Let $H = (V, E)$. We call H the *underlying graph* of H'. It should be clear how to use relation $\prec$ to obtain an inherited strict partial order $\prec_1$ or $\prec_2$ for H. We can use $\prec_i$ for $i \in [2]$ to define the set $F^{\prec_i}$ of $\prec_i$-ends of H, that is the $\prec_i$-minimums and $\prec_i$-maximums, in FO. As $\prec_1$ is the reverse of $\prec_2$ and vice versa, the set of $\prec_1$-ends is equal to the set of $\prec_2$-ends. Let F be the set of inherited termini of H. Then $F = F^{\prec_i}$ for each $i \in [2]$, as shown in the following lemma.

Lemma 185. *The set F of inherited termini of H is equal to the set $F^{\prec_i}$ of $\prec_i$-ends of H for each $i \in [2]$, where $\prec_1$ and $\prec_2$ are the inherited strict partial orders for H.*

Proof. Let $\mathtt{F} = \{f_1, f_2\}$ be the set of termini of $\mathtt{H}$, $\mathtt{S}_{f_1}$ and $\mathtt{S}_{f_2}$ be the sides, and $\mathtt{T}_{f_1}$ and $\mathtt{T}_{f_2}$ be the side trees of $\mathtt{H}$. Without loss of generality, let $a \in \mathtt{S}_{f_1}$ and $b \in \mathtt{S}_{f_2}$. Then $a \prec_{f_1} b$. For $i \in [2]$ let $\prec_i$ be the inherited strict partial order for H that is obtained by restricting $\prec_{f_i}$ to the vertex set V. For each $f \in \mathtt{F}$ let $\mathtt{C}_f$ be the unique subset of $\mathtt{C}_f(a,b)$ from Corollary 180.

First we prove that every terminus $e \in F$ of H is a $\prec_1$-end of H. Clearly, this implies that e is a $\prec_2$-end of H as well. Let $e \in F$. Then there is an $f \in \mathtt{F}$ so that $e \in \mathtt{C}_f$. Thus, e is the root of the directed subtree $\mathtt{T}^e_f(a,b)$ of $\mathtt{T}_f$. The root e of $\mathtt{T}^e_f(a,b)$ is a child of $r_f(a,b)$ in $\mathtt{T}_f$. Without loss of generality, let $f = f_1$. Then e is a child of a in $\mathtt{T}_{f_1}$. Let us suppose e is not $\prec_1$-minimal. Then there is a vertex $v \in V$ such that $v \prec_1 e$. As $\prec_1$ is the restriction of $\prec_{f_1}$ to V, we have $v \prec_{f_1} e$. Since $e \in \mathtt{V}_{f_1}$, vertex $v \in V$ is a proper ancestor of e in $\mathtt{T}_{f_1}$ (Observation 164), and therefore, an ancestor of a in $\mathtt{T}_{f_1}$. Hence, $v \preceq_{f_1} a$, and vertex v is not contained in $\mathtt{V}^-(a,b) = \{v \in \mathtt{V} \mid a \prec_{f_1} v \prec_{f_1} b\}$. Thus, $v \notin V \subseteq \mathtt{V}^-(a,b)$, a contradiction. It follows that e is $\prec_1$-minimal.

Next let us show that every $\prec_i$-end e of H is a terminus of H. Without loss of generality, let e be $\prec_1$-minimal. Let us consider the case where $e \in \mathtt{V}_{f_1}$. Since $V \cap \mathtt{V}_{f_1} = \dot{\bigcup}_{\varepsilon \in \mathtt{C}_{f_1}} \mathtt{V}^\varepsilon_{f_1}(a,b)$ (Corollary 180), there is a vertex $\varepsilon \in \mathtt{C}_{f_1}$ such that $e \in \mathtt{V}^\varepsilon_{f_1}(a,b)$. As there is no vertex $v \in V$ such that $v \prec_1 e$, there is no vertex $v \in V$ with $v \prec_{f_1} e$. Thus, there is no ancestor $v \in V$ of e in $\mathtt{T}_{f_1}$, and therefore, no ancestor $v \in \mathtt{V}^\varepsilon_{f_1}(a,b)$ of e in $\mathtt{T}^\varepsilon_{f_1}(a,b)$. It follows that e is the root of $\mathtt{T}^\varepsilon_{f_1}(a,b)$. Consequently, $e = \varepsilon$, and e is an inherited terminus of H. Now let us consider the case where $e \in V \setminus \mathtt{V}_{f_1} \subseteq \mathtt{S}_{f_2}$ (Observation 158). As $V \cap \mathtt{V}_{f_2} = \dot{\bigcup}_{\varepsilon \in \mathtt{C}_{f_2}} \mathtt{V}^\varepsilon_{f_2}(a,b)$, there is a vertex $\varepsilon \in \mathtt{C}_{f_2}$ such that $e \in \mathtt{V}^\varepsilon_{f_2}(a,b)$. Let o be a leaf of $\mathtt{T}^\varepsilon_{f_2}(a,b)$ that is a descendant of e in the directed tree $\mathtt{T}^\varepsilon_{f_2}(a,b)$. All leaves of $\mathtt{T}^\varepsilon_{f_2}(a,b)$ are in $\mathtt{O}_a \cap \mathtt{O}_b \subseteq \mathtt{O}$. Thus, $o \in \mathtt{O}$. As $e \in \mathtt{S}_{f_2}$, it follows that o is a proper descendant of e in $\mathtt{T}_{f_2}$. Therefore, $o \prec_{f_1} e$. Since $\mathtt{V}^\varepsilon_{f_2}(a,b) \subseteq V$, we have $o \in V$. Hence, $o \prec_1 e$, which is a contradiction because e is $\prec_1$-minimal. □

Corollary 186. *The set F of termini of H is FO-definable for every extended valid subgraph H' with underlying graph H.*

As H is a chordal comparability graph, the induced subgraph $H[F^{\prec_i}]$ is a tree (Lemma 80), the $\prec_i$-bundle tree of H, where $\prec_1$ and $\prec_2$ are the inherited strict partial orders for H. Since $F = F^{\prec_1} = F^{\prec_2}$ by Lemma 185, we have $H[F^{\prec_i}] = H[F]$ for $i \in [2]$. We call $H[F]$ the *(inherited) bundle tree* of H. It follows from Corollary 186 that $H[F]$ is definable in FO for each extended valid subgraph H' with underlying graph H. We let the set of *inner termini* F_{in} and the set of *outer termini* F_{out} be the set of inner and outer nodes of $H[F]$, respectively. Clearly, the sets of inner and outer termini are FO-definable. As $H[F]$ is a tree, there is a unique 2-coloring $\{F_1, F_2\}$ of $H[F]$. It is not hard to see that $\{F_1, F_2\} = \{\mathtt{C}_{f_1}, \mathtt{C}_{f_2}\}$ where $\mathtt{C}_{f_1}$ and $\mathtt{C}_{f_2}$ are the sets from Corollary 180. Since $\{F_1, F_2\} = \{F_1^{\prec_i}, F_2^{\prec_i}\}$ by Lemma 79, the equivalence relation corresponding to the 2-coloring is definable in FO.

For every $e \in F$, we let V_e be the set of vertices that are reachable from e in the directed graph (V, T). We let $T_e := (V_e, T \cap V_e^2)$, and we let O_e be the set of leaves of T_e. It is not hard to see that the directed tree T_e is the inherited side tree of e of H. Thus, O_e is the middle of $e \in F$ of H, and $O := \bigcup_{e \in F}$ is the middle of H. We let the *(inherited) side* S_e of terminus e be the set $V_e \setminus O_e$. Hence, $S_e = V_e \setminus O$. It is not hard to see that V_e, $E(T_e)$, O_e and S_e are TC-definable for extended valid subgraphs H'.

We let $\psi_F(x)$, $\psi_{F_{\text{in}}}(x^*)$, $\psi_V(x^*, x)$, $\psi_E(x^*, x, x')$, $\psi_O(x)$ and $\psi_S(x^*, x)$, respectively, be TC-formulas for the set/relations F, F_{in}, V_e, $E(T_e)$, O and S_e for an extended valid subgraph H'. Hence,

$$\begin{aligned}
H' &\models \psi_F[e] &&\iff e \in F, \\
H' &\models \psi_{F_{\text{in}}}[e] &&\iff e \in F_{\text{in}}, \\
H' &\models \psi_V[e, v] &&\iff e \in F \text{ and } v \in V_e, \\
H' &\models \psi_E[e, v, v'] &&\iff e \in F \text{ and } (v, v') \in E(T_e), \\
H' &\models \psi_O[v] &&\iff e \in F \text{ and } v \in O_e, \text{ and} \\
H' &\models \psi_S[e, v] &&\iff e \in F \text{ and } v \in S_e.
\end{aligned} \tag{6.15}$$

Further, it is not hard to see there are FO-formulas $\psi_{\text{sub}}(x_1, x_2)$ and $\psi_{V(\cdot,\cdot)}(x_1, x_2, y)$ such that for $v_1, v_2, w \in V$ of an extended valid subgraph H' with underlying graph H we have

$$\begin{aligned}
H' &\models \psi_{\text{sub}}[v_1, v_2] &&\iff [v_1, v_2] \text{ is a subbundle pair of } H, \\
H' &\models \psi_{V(\cdot,\cdot)}[v_1, v_2, w] &&\iff [v_1, v_2] \text{ is a subbundle pair of } H \text{ and } w \in V(v_1, v_2).
\end{aligned} \tag{6.16}$$

The following corollary is a direct consequence of Observation 175.

Corollary 187. *Let $i \in [2]$ and $e_1, e_2 \in F_i$ with $e_1 \neq e_2$. Then $V_{e_1} \cap V_{e_2} = \emptyset$.*

Corollary 188. *Let $i \in [2]$ and $e_1 \in F_1$ and $e_2 \in F_2$. Then $V_{e_1} \cap V_{e_2} \subseteq O$.*

Proof. Let $i \in [2]$ and $e_1 \in F_1$ and $e_2 \in F_2$. Then either $e_1 \in \mathtt{C}_{f_1}$ and $e_2 \in \mathtt{C}_{f_2}$, or $e_1 \in \mathtt{C}_{f_2}$ and $e_2 \in \mathtt{C}_{f_1}$. As $\mathtt{C}_{f_i} \subseteq \mathtt{C}_{f_i}(a, b)$ for each $i \in [2]$, Corollary 177 implies that $V_{e_1} \cap V_{e_2} \subseteq \mathtt{O}_a \cap \mathtt{O}_b \subseteq \mathtt{O}$, where $\mathtt{O}$ is the middle of $\mathtt{H}$. Hence, $V_{e_1} \cap V_{e_2}$ is a subset of $V \cap \mathtt{O}$, which is equal to O by Observation 182. □

Lemma 189. *Let $e_1, e_2 \in F$ with $e_1 \neq e_2$. Then $S_{e_1} \cap S_{e_2} = \emptyset$.*

Proof. Let $e_1, e_2 \in F$ with $e_1 \neq e_2$. If e_1 and e_2 are in the same color class of $\{F_1, F_2\}$, then $S_{e_1} \cap S_{e_2} = \emptyset$ follows from Corollary 187. If e_1 and e_2 are in different color classes of the 2-coloring $\{F_1, F_2\}$, then $V_{e_1} \cap V_{e_2} \subseteq O$ by Corollary 188, and it follows that $S_{e_1} \cap S_{e_2} = \emptyset$. □

Observation 190. *Let $[x, y]$ be a subbundle pair of H. Then each $z \in V(x, y)$ with $z \neq x$ is adjacent to x.*

Proof. Let $[x, y]$ be a subbundle pair of H, and let $z \in V(x, y)$ with $z \neq x$. Let $\prec_1$ be one of the inherited strict partial orders for H. Since $z \in V(x, y)$ and $z \neq x$, we have $x \prec_1 z$ or $z \prec_1 x$. Hence, there in an edge between x and z in H. □

Lemma 191. *Let $f \in \mathtt{F}$ and $e \in \mathtt{C}_f$ be a terminus. Let $[x, y]$ be a subbundle pair of H where $x \in V_e$ and $y \in \mathtt{V}_f$. Then $V(x, y) \subseteq V_e$ and $V(x, y)$ induces a directed path from x to y or from y to x in T_e.*

Proof. Let $f \in \mathtt{F}$ and $e \in \mathtt{C}_f$. Let $[x, y]$ be a subbundle pair of H with $x \in V_e$ and $y \in \mathtt{V}_f$. As $e \in \mathtt{C}_f$ and $x \in V_e$, the multiset $[x, y]$ is a subbundle pair of $\mathtt{H}$ with $x, y \in \mathtt{V}_f$. Lemma 184 implies that $V(x, y) = \mathtt{V}(x, y)$. By Observation 170 we have $\mathtt{V}(x, y) \subseteq \mathtt{V}_f$ and $\mathtt{V}(x, y)$ induces a path from x to y or from y to x in $\mathtt{T}_f$. It follows that $\mathtt{V}(x, y) \subseteq V \cap \mathtt{V}_f$. The set $V \cap \mathtt{V}_f$ is the disjoint union of all sets $V_{e'} = \mathtt{V}_f^{e'}(a, b)$ for $e' \in \mathtt{C}_f$ by Corollary 180. As $V(x, y)$ is connected (Observation 190) and $x \in V_e$, we have $V(x, y) \subseteq V_e$. Since the side tree T_e of e of H is the subtree of $\mathtt{T}_f$ induced by V_e, the set $V(x, y)$ induces a path from x to y or from y to x in T_e. □

Corollary 192. *Let $e \in F$ be a terminus, and $[x, y]$ be a subbundle pair of H where $x, y \in V_e$. Then $V(x, y) \subseteq V_e$ and $V(x, y)$ induces a directed path from x to y or from y to x in T_e.*

Proof. Let $e \in F$ be a terminus, and $[x, y]$ be a subbundle pair of H with $x, y \in V_e$. Then there is an $f \in \mathtt{F}$ such that $e \in \mathtt{C}_f$. Clearly, $y \in \mathtt{V}_f$. Thus, Corollary 192 follows from Lemma 191. □

Corollary 193. *Let $e \in F$ be a terminus, and $[x, y]$ be a subbundle pair of H with $x, y \in S_e$. Then $V(x, y) \subseteq S_e$.*

Proof. Let $e \in F$ be a terminus, and $[x, y]$ be a subbundle pair with $x, y \in S_e$. According to Corollary 192, $V(x, y)$ is a subset of V_e and induces a directed path from x to y or from y to x in T_e. Let us assume there is a vertex $z \in V(x, y)$ that is in O_e. As z is a leaf of T_e, vertex z must be the end of the path induced by $V(x, y)$, that is, z is x or y. Since x and y are in S_e, we obtain a contradiction. □

Lemma 194. *Let $e \in F$ be a terminus, and $x, y \in V_e$. If there is a directed path from x to y in T_e, then $[x, y]$ is a subbundle pair of H and $V(x, y)$ consists of all vertices of the directed path from x to y in T_e.*

Proof. Let $e \in F$ be a terminus, and $x, y \in V_e$. As there is a path from x to y in T_e, Corollary 179 implies that either $x = y$ or x and y are adjacent in H. Thus, $[x, y]$ is a subbundle pair of H with $x, y \in V_e$. Corollary 192 implies that $V(x, y)$ consists of all vertices of the directed path from x to y in T_e. □

Observation 195. *Let $[x, y]$ be a subbundle pair of H, and let $z \in V(x, y)$. Then $[x, z]$ is a subbundle pair of H and $V(x, z) \subseteq V(x, y)$.*

Proof. Let $[x, y]$ be a subbundle pair of H, and let $z \in V(x, y)$. According to Observation 190 either $x = z$ or x and z are adjacent. Consequently, $[x, z]$ is a subbundle pair. Let $\prec_1$ be one of the inherited strict partial orders for H. Without loss of generality, let

$x \preceq_1 y$. As $z \in V(x,y)$ we have $x \preceq_1 z$ and $z \preceq_1 y$. Let $v \in V(x,z)$. Then $x \preceq_1 v \preceq_1 z$ because $x \preceq_1 z$. Hence, $x \preceq_1 v \preceq_1 z \preceq_1 y$. We obtain $x \preceq_1 v \preceq_1 y$ by transitivity of $\preceq_1$. It follows that $v \in V(x,y)$. Therefore, $V(x,z) \subseteq V(x,y)$. □

Lemma 196. *Let $e \in F$ and $[e,y]$ be a subbundle pair of H. Then $V(e,y) \cap V_e$ is the vertex set of a subtree of T_e that is rooted at e.*

Proof. We show that for each vertex $v \in V(e,y) \cap V_e$ there is a path from e to v in the subgraph of T_e induced by $V(e,y) \cap V_e$. Then the induced subgraph $T_e[V(e,y) \cap V_e]$ is connected, and therefore, a subtree of T_e. Further, e is the root of this subtree. Let $v \in V(e,y) \cap V_e$. By Observation 195 the multiset $[e,v]$ is a subbundle pair and $V(e,v) \subseteq V(e,y)$. As $e, v \in V_e$, we can apply Corollary 192 and obtain that $V(e,v) \subseteq V_e$ and that $V(e,v)$ induces a directed path from e to v or from v to e in T_e. Since e is the root of T_e, the set $V(e,v)$ induces a directed path from e to v. Now, $V(e,v) \subseteq V(e,y) \cap V_e$ implies that there is a path from e to v in the subgraph of T_e induced by $V(e,y) \cap V_e$. □

Lemma 197. *Let $[e_1,e_2]$ be a subbundle pair with $e_1, e_2 \in F$. Then $V(e_1,e_2) \subseteq S_{e_1} \cup V_{e_2}$.*

Proof. Let $[e_1,e_2]$ be a subbundle pair with $e_1, e_2 \in F$. If $e_1 = e_2$, then $V(e_1,e_2) = \{e_2\}$ and $V(e_1,e_2) \subseteq V_{e_2}$. Thus, let $e_1 \neq e_2$. Then e_1 and e_2 are adjacent. Hence, e_1 and e_2 are in different color classes of the 2-coloring $\{F_1, F_2\} = \{\mathtt{C}_{f_1}, \mathtt{C}_{f_2}\}$. Without loss of generality, let $e_1 \in \mathtt{C}_{f_1}$ and $e_2 \in \mathtt{C}_{f_2}$. Let us assume there is a vertex $v \in V(e_1,e_2)$ such that $v \notin S_{e_1} \cup V_{e_2}$. First, let us consider the case where $v \in \mathtt{V}_{f_2}$. As $v \in V(e_1,e_2)$, the multiset $[e_2,v]$ is a subbundle pair of H by Observation 195. Since $e_2 \in \mathtt{C}_{f_2}$, $e_2 \in V_{e_2}$ and $v \in \mathtt{V}_{f_2}$, we can apply Lemma 191 and obtain that $V(e_2,v) \subseteq V_{e_2}$. It follows that $v \in V_{e_2}$, a contradiction. Now, let us consider the case where $v \in V \setminus \mathtt{V}_{f_2}$. Then $v \in \mathtt{V}_{f_1}$, and analogous to the previous case we obtain that $v \in V_{e_1}$. As $v \notin S_{e_1}$, we have $v \in O_e \subseteq O$. According to Observation 182, $O \subseteq \mathtt{O}$. Further, $\mathtt{O} \subseteq \mathtt{V}_{f_2}$. It follows that $v \in \mathtt{V}_{f_2}$, a contradiction. We obtain that $V(e_1,e_2) \subseteq S_{e_1} \cup V_{e_2}$. □

Corollary 198. *Let $[e_1,e_2]$ be a subbundle pair of the graph H with $e_1, e_2 \in F$. Then $V(e_1,e_2) \subseteq S_{e_1} \cup O \cup S_{e_2}$.*

For $e \in F$ we let N_e be the set of neighbors of e in the bundle tree $H[F]$. Thus, $N_e \subseteq F$. Clearly, for each $e \in F$ and every neighbor $e' \in N_e$, the multiset $[e,e']$ is a subbundle pair of H.

Observation 199. *Let $e \in F$. Then $S_e \cap N_e = \emptyset$.*

Proof. Let $e \in F$. Let us suppose there is an $e' \in N_e$ with $e' \in S_e$. Clearly, $e' \in V_{e'}$. Corollary 188 implies that $e' \in O$. Hence, $e' \notin S_e$, a contradiction. □

Lemma 200. *We have $V(e,e_1) \cap V(e,e_2) \subseteq S_e$ for all $e \in F_{\text{in}}$ and all $e_1, e_2 \in N_e$ with $e_1 \neq e_2$.*

Proof. Let $e \in F_{\text{in}}$ and $e_1, e_2 \in N_e$ with $e_1 \neq e_2$. Then $[e,e_1]$ and $[e,e_2]$ are subbundle pairs and e, e_1 and e_2 all are in F. According to Lemma 197 we have $V(e,e_1) \subseteq S_e \cup V_{e_1}$ and $V(e,e_2) \subseteq S_e \cup V_{e_2}$. Since e_1 and e_2 are neighbors of e, the vertices e_1 and e_2 belong to the same color class of the 2-coloring $\{F_1, F_2\}$. Thus, $V_{e_1} \cap V_{e_2} = \emptyset$ by Corollary 187. It follows that $V(e,e_1) \cap V(e,e_2) \subseteq S_e$. □

Corollary 201. *Let $e \in F_{\text{in}}$ be an inner inherited terminus of valid subgraph H. Then e is in the inherited side S_e of e.*

Proof. Let $e \in F_{\text{in}}$. Then e has two distinct neighbors $e_1, e_2 \in F$ in the inherited bundle tree $H[F]$. Hence, $e_1, e_2 \in N_e$, and $[e, e_1]$ and $[e, e_2]$ are subbundle pairs with $e \in V(e, e_1)$ and $e \in V(e, e_2)$. It follows that $e \in S_e$ by Lemma 200. □

Lemma 202. *Let $e \in F_{\text{in}}$, $e' \in N_e$ and $x \in V_e$. If e' and x are adjacent in H, then $x \in V(e, e')$.*

Proof. Let $e \in F_{\text{in}}$, $e' \in N_e$. Let $\prec_1$ be one of the inherited strict partial orders for H. Then $\{F_1^{\prec_1}, F_2^{\prec_1}\}$ is a 2-coloring of $H[F]$ (Lemma 79). Without loss of generality, let $e \in F_1^{\prec_1}$ and $e' \in F_2^{\prec_1}$. Thus, e is $\prec_1$-minimal and e' is $\prec_1$-maximal. As $x \in V_e$, e in an ancestor of x in T_e. By Corollary 179 either $e = x$ or e and x are adjacent in H. Since e is $\prec_1$-minimal, it follows that $e \preceq_1 x$. Further, $x \prec_1 e'$ because x and e' are adjacent in H and e' is $\prec_1$-maximal. It follows that $e \preceq_1 x \preceq_1 e'$, and therefore $x \in V(e, e')$. □

Observation 203. *Let $v \in V$ be a vertex of H. Then there are termini $e, e' \in F$ such that $v \in V(e, e')$.*

Proof. Let $\prec_1$ be one of the inherited strict partial orders for H. Let $e \in V$ be any vertex with $e \preceq_1 v$ that is $\prec$-minimal, and $e' \in V$ be any vertex with $v \preceq_1 e'$ that is $\prec$-maximal. Then $v \in V(e, e')$. By Lemma 185 $e, e' \in F$. □

Observation 204. *Let $\{v, w\} \in E$ be an edge of H. Then there are termini $e, e' \in F$ with $e \neq e'$ such that $v, w \in V(e, e')$.*

Proof. Let $\prec_1$ be one of the inherited strict partial orders for H. Without loss of generality, let $v \prec_1 w$. Let $e \in V$ be any vertex with $e \preceq_1 v$ that is $\prec$-minimal, and $e' \in V$ be any vertex with $w \preceq_1 e'$ that is $\prec$-maximal. By Lemma 185 $e, e' \in F$. As $\prec_1$ is a strict partial order and $\preceq_1$ is its associated partial order, we have $v, w \in V(e, e')$. Further, it follows that $e \prec_1 e'$ and therefore, $e \neq e'$. □

6.5. The Genealogical Decomposition Tree

The genealogical decomposition tree of the underlying graph `H` of a bundle extension `H*` is a directed tree, which is of use in the canonization procedure. It has a recursive structure that is based on decomposition trees of valid subgraphs of `H`. The graph `H` and its valid subgraphs are introduced in the previous section, and the decomposition tree of a valid subgraph is defined in Section 6.5.3.

Every node of a decomposition tree of a valid subgraph H is a subbundle pair of H, and therefore, a subbundle pair of the graph `H`. Each subbundle pair represents a certain subset of vertices of `H`. More precisely, the subbundle pair $[a, b]$ of valid subgraph H represents the vertex set $V(a, b)$. The most important property of a decomposition tree of a valid subgraph H of `H` is that the intersection of the vertex sets represented by a node and its parent node is contained within a max clique of H. Moreover, there is a terminus e of H such that the intersection is contained in the side S_e of e of H and induces a path in the side tree T_e of e of H. In order to introduce the decomposition tree of a valid subgraph of H, We start with defining the simplified decomposition tree, which reflects the basic structure of the decomposition tree.

6.5.1. The Simplified Decomposition Tree

We first define a simplified version of the decomposition tree of H. This simplified decomposition tree is a directed tree. The set of nodes of the simplified decomposition tree is a set of blocks, where almost each block basically consists of all subbundle pairs that contain a common terminus and a neighbor of this terminus in the bundle tree of H. To obtain the actual decomposition tree, we refine this simplified decomposition tree. For each block we construct a directed tree of subbundle pairs. We call this directed tree the *decomposition tree for a block*. It is introduced in the subsequent section. In Section 6.5.3 the directed trees for each block are then attached according to the edge relation of the simplified decomposition tree.

Blocks and the Simplified Decomposition Tree

Let $H' = (V, E, \prec, T)$ be an extended valid subgraph with underlying graph H. Let F be the set of termini of H, and N_e be the set of neighbors of $e \in F$ in the bundle tree $H[F]$. In the following we define the simplified decomposition tree of H. Its definition depends on the number of centroids of the bundle tree $H[F]$. We distinguish between one and two centroids. Remember that if a tree has two centroids, the two centroids are adjacent (Lemma 28), and that each centroid of a tree is an inner node (Observation 29). The set of nodes of the simplified decomposition tree of H is the set of blocks $\mathfrak{B}$. A *block* is a pair $\mathfrak{b} = (\mathfrak{L}, l)$, where l is a subbundle pair of H and $\mathfrak{L}$ is a set of subbundle pairs of H with $l \in \mathfrak{L}$. We call l the *eldest* of block $\mathfrak{b}$. In the following we define the set of blocks $\mathfrak{B}$.

The set of inner termini F_{in} is the set of inner nodes of the bundle tree $H[F]$, that is, F_{in} contains all termini that are not a leaf of $H[F]$. We assume that $F_{\text{in}} \neq \emptyset$. If F_{in} is empty we can define the decomposition tree directly, without help of the simplified decomposition tree. For each $e \in F_{\text{in}}$ we define a block $\mathfrak{b}_e = (\mathfrak{L}_e, l_e)$. Then the set of blocks $\mathfrak{B}$ consists of all blocks $\mathfrak{b}_e$ for $e \in F_{\text{in}}$, and possibly also of an additional block as described below. Let $C \subseteq F_{\text{in}}$ be the set of centroids of $H[F]$.

If $|C| = 1$, we transfer the bundle tree $H[F]$ into a rooted tree T'_{e_r} by fixing $e_r \in C$ as the root of $H[F]$. Let $p(e)$ be the parent node of e in T'_{e_r} for each $e \in F_{\text{in}} \setminus \{e_r\}$. Now we can define the block $\mathfrak{b}_e = (\mathfrak{L}_e, l_e)$ for each $e \in F_{\text{in}}$. For every $e \in F_{\text{in}} \setminus \{e_r\}$ we let $l_e := [e, p(e)]$ be the eldest of the block $\mathfrak{b}_e$ and we let $l_{e_r} := [e_r, e_r]$ be the eldest of the block $\mathfrak{b}_{e_r}$ for the root e_r. Further, for every $e \in F_{\text{in}}$ we let $\mathfrak{L}_e := \{[e, f] \mid f \in N_e\} \cup \{l_e\}$. In the case that $|C| = 1$, the simplified decomposition tree $(\mathfrak{B}, \mathfrak{E})$ is defined as follows: We let $\mathfrak{B}$ be the set $\{\mathfrak{b}_e \mid e \in F_{\text{in}}\}$, and we let $(\mathfrak{b}_e, \mathfrak{b}_{e'}) \in \mathfrak{E}$, if, and only if, (e, e') is an edge in the directed tree equivalent to the rooted tree T'_{e_r}.

Let $|C| = 2$. Let e_{r_1} and e_{r_2} be the two centroids in C. We transfer $H[F]$ into two rooted trees $T'_{e_{r_1}}$ and $T'_{e_{r_2}}$ by removing the edge between e_{r_1} and e_{r_2} (Lemma 28) and fixing e_{r_1} and e_{r_2} as roots of the two emerging subtrees. Again we let $p(e)$ be the parent node of e in the respective rooted tree for each $e \in F_{\text{in}} \setminus \{e_{r_1}, e_{r_2}\}$. Then we can define a block $\mathfrak{b}_e = (\mathfrak{L}_e, l_e)$ for every $e \in F_{\text{in}}$ by letting $l_e := [e, p(e)]$ be the eldest of the block for $e \in F_{\text{in}} \setminus \{e_{r_1}, e_{r_2}\}$ and $l_e := [e_{r_1}, e_{r_2}]$ be the eldest of the block for $e \in \{e_{r_1}, e_{r_2}\}$. For every $e \in F_{\text{in}}$ we let $\mathfrak{L}_e := \{[e, f] \mid f \in N_e\}$. Notice that $l_e \in \mathfrak{L}_e$ for all $e \in F_{\text{in}}$. Further, we need the additional block $\mathfrak{b}_{\text{root}} := ([e_{r_1}, e_{r_2}], \{[e_{r_1}, e_{r_2}]\})$. In the case that $|C| = 2$, we let $\mathfrak{B}$, the set of nodes of the simplified decomposition tree, be the set $\{\mathfrak{b}_e \mid e \in F_{\text{in}}\} \cup \{\mathfrak{b}_{\text{root}}\}$. The set $\mathfrak{E}$ of edges of the simplified decomposition tree $(\mathfrak{B}, \mathfrak{E})$ is defined as follows: We let $(\mathfrak{b}_{\text{root}}, \mathfrak{b}_{e_{r_1}})$ and $(\mathfrak{b}_{\text{root}}, \mathfrak{b}_{e_{r_1}})$ be edges of the simplified decomposition tree. Further, we let $(\mathfrak{b}_e, \mathfrak{b}_{e'}) \in \mathfrak{E}$ if (e, e') is an edge in one of the directed trees equivalent to the rooted trees $T'_{e_{r_1}}$ or $T'_{e_{r_2}}$.

Observation 205. *Let block $\mathfrak{b}_e = (\mathfrak{L}_e, l_e)$ be a block of an inner terminus $e \in F_{\text{in}}$ in the simplified decomposition tree $(\mathfrak{B}, \mathfrak{E})$. Then $l_e \in \mathfrak{L}_e$, and for all subbundle pairs $[e_1, e_2] \in \mathfrak{L}_e$ we have $e \in \{e_1, e_2\}$.*

Observation 206. *Let block $\mathfrak{b} = (\mathfrak{L}, l)$ be a child of a block $\mathfrak{b}' = (\mathfrak{L}', l')$ in the simplified decomposition tree $(\mathfrak{B}, \mathfrak{E})$. Then the subbundle pair l is an element in $\mathfrak{L}'$.*

Defining the Simplified Decomposition Tree in STC+C

Given an extended valid subgraph $H' = (V, E, \prec, T)$ where H denotes the underlying valid subgraph, the set F of tips and the set F_{in} of inner termini of H are FO-definable (cf. (6.15)). In order to define the decomposition tree of H, we will only need the blocks $\mathfrak{b}_e$ of inner termini $e \in F_{\text{in}}$. In the following we describe STC+C formulas $\varphi_{\mathfrak{L}}(x^*, x)$ and $\varphi_l(x^*, x)$ such that for extended valid subgraphs $H' = (V, E, \prec, T)$ and $e, e' \in V$ we have

$$\begin{aligned} H' \models \varphi_{\mathfrak{L}}[e, e'] \quad &\iff \quad e \in F_{\text{in}} \text{ and } [e, e'] \in \mathfrak{L}_e, \\ H' \models \varphi_l[e, e'] \quad &\iff \quad e \in F_{\text{in}} \text{ and } l_e = [e, e']. \end{aligned} \tag{6.17}$$

Let $\mathcal{S}$ be a set of subbundle pairs where $e \in \{a, b\}$ for all $[a, b] \in \mathcal{S}$. The *e-reduct* $\mathcal{S}|_e$ of $\mathcal{S}$ is the set $\{c \mid [e, c] \in \mathcal{S}\}$. Thus there is a one-to-one correspondence between the subbundle pairs in $\mathcal{S}$ and the vertices in $\mathcal{S}|_e$. Note that $e \in \mathcal{S}|_e$ if and only if $[e, e] \in \mathcal{S}$. Then it follows from Observation 205 that $\mathfrak{L}_e|_e = \varphi_{\mathfrak{L}}[H', e; x]$ and $\{l_e\}|_e = \varphi_l[H', e; x]$ for all $e \in F_{\text{in}}$.

For all $e \in F_{\text{in}}$, we have $\mathfrak{L}_e = \{[e, e'] \mid e' \in N_e\} \cup \{l_e\}$ in each of the two cases. Thus, we can easily define $\varphi_{\mathfrak{L}}$ by using formula φ_l. We let

$$\varphi_{\mathfrak{L}}(x^*, x) := \big(\varphi_{F_{\text{in}}}(x^*) \land \varphi_F(x) \land E(x^*, x)\big) \lor \varphi_l(x^*, x),$$

where ψ_F and $\psi_{F_{\text{in}}}$ are the formulas for F and F_{in} from (6.15).

In order the define φ_l, let us first summarize the definition of l_e:

- For all $e \in F_{\text{in}} \setminus C$, we have $l_e = [e, p(e)]$.
- For all $e \in C$, we have
  - $l_e = [e, e]$ if $|C| = 1$, and
  - $l_e = [e, e']$ where $e' \in C \setminus \{e\}$ if $|C| = 2$.

It is not hard to construct the STC+C-formula φ_l, once we have a formula for the centroids of $H[F]$ and a formula that defines the relation $P := \{(e, p(e)) \mid e \in F_{\text{in}} \setminus C\}$. Thus, we confine ourselves to describe how to obtain these formulas.

We obtain a formula for the centroids of $H[F]$ by pulling back the formula $\vartheta_{\text{cen}}(x)$ given in (2.5) under the transduction $\Theta = (\varphi_{F_{\text{in}}}(x), E(x, x'))$. To obtain a formula for the relation P, we transfer the bundle tree $H[F]$ into a rooted tree where we use a centroid as root. Let T_c be the rooted tree that we obtain by fixing $c \in C$ as the root of $H[F]$. If $|C| = 1$, then clearly T_c is the rooted tree T'_{e_r}. If $|C| = 2$, then the two rooted trees $T'_{e_{r_1}}$ and $T'_{e_{r_2}}$ are subtrees of T_c. Thus, for any node $e \in F_{\text{in}} \setminus C$ the parent node of e in T_c corresponds to $p(e)$ independent of our choice for c. We use a parameterized transduction similar to the one from Example 8. We let $\Theta_2(x_r) = (\top, \varphi_{F_{\text{in}}}(x), E(x, x') \wedge \vartheta_{\text{conn}}(x', x_r, x))$. Then we can pull back a formula that defines the parent relation in a rooted tree under this parameterized transduction. We can use the pull back to define P if we bound the parameter variable in such a way that it can only be interpreted by centroids.

6.5.2. The Decomposition Tree of a Block

In this section we create a directed tree $\vartheta(\mathfrak{b})$ for each block $\mathfrak{b} = (\mathfrak{L}, l) \in \mathfrak{B}$ occurring in the simplified decomposition tree. The set of nodes of $\vartheta(\mathfrak{b})$ includes all subbundle pairs in $\mathfrak{L}$ and $\vartheta(\mathfrak{b})$ is rooted at l. For block $\mathfrak{b}_{\text{root}}$ the directed tree $\vartheta(\mathfrak{b}_{\text{root}})$ is $([e_{r_1}, e_{r_2}], \emptyset)$. Thus, we only need to consider the blocks in $\mathfrak{B} \setminus \{\mathfrak{b}_{\text{root}}\}$, which correspond to blocks $\mathfrak{b}_e$ for $e \in F_{\text{in}}$. For each $e \in F_{\text{in}}$ we will construct a rooted tree $\vartheta_e := \vartheta(\mathfrak{b}_e)$ for block $\mathfrak{b}_e$ in this section. The nodes of ϑ_e are subbundle pairs, all of which contain terminus e. We inductively define ϑ_e by successively determining the parent node of each subbundle pair $[e, a]$ in $\mathfrak{L}_e$ and removing $[e, a]$ from $\mathfrak{L}_e$. During this process there might not always be a subbundle pair in $\mathfrak{L}_e$ that is suitable to be the parent node of $[e, a]$. We add new subbundle pairs to $\mathfrak{L}_e$ to make sure this does not happen. In the end we ensure that the eldest l_e becomes the root of ϑ_e.

Construction of ϑ_e

Let $e \in F_{\text{in}}$. We use a sequence of sets $\mathfrak{L}_i$ of subbundle pairs and a sequence of directed graphs ϑ_i to help us define the directed tree ϑ_e. We will maintain the following properties for all $i \geq 0$:

(a) $l_e \in \mathfrak{L}_i$,
(b) $e \in \{a, b\}$ for all $[a, b] \in \mathfrak{L}_i$,
(c) $a \in N_e$ or $a \in S_e$ for all $[e, a] \in \mathfrak{L}_i$,
(d) $\bigcup_{j \leq i} \mathfrak{L}_j = V(\vartheta_i)$,
(e) ϑ_i is a directed forest with $|\mathfrak{L}_i|$ connected components, where $\mathfrak{L}_i$ is the set of roots,
(f) $\vartheta_i \subseteq \vartheta_{i+1}$, that is, $V(\vartheta_i) \subseteq V(\vartheta_{i+1})$ and $E(\vartheta_i) \subseteq E(\vartheta_{i+1})$, and
(g) $|\mathfrak{L}_i| > |\mathfrak{L}_{i+1}|$ if $|\mathfrak{L}_i| > 1$.

We start by letting $\mathfrak{L}_0 := \mathfrak{L}_e$ and $\vartheta_0 := (\mathfrak{L}_e, \emptyset)$. Thus, the vertices of the directed graph ϑ_0 are the subbundle pairs in $\mathfrak{L}_e$. Notice that the blocks are defined in a way that $\mathfrak{L}_e$ cannot be empty and always contains the eldest l_e. Further, the set $\mathfrak{L}_e \setminus \{l_e\}$ is a set of subbundle pairs $[e, a]$ with $a \in N_e$, and subbundle pair l_e is either of the form $[e, c]$ with

$c \in N_e$ or $[e, e]$. We have $e \in S_e$ by Corollary 201 since $e \in F_{\text{in}}$. Thus, $\mathfrak{L}_0$ and ϑ_0 satisfy properties (a)-(e). In the following, as long as $|\mathfrak{L}_i| > 1$ for $i \geq 0$, we recursively add edges and possibly also new vertices to ϑ_i to obtain ϑ_e in the end. The tree ϑ_e is constructed from the bottom to the top by identifying some children and assigning them their parent. After each round, we omit the vertices that already got a parent and again look for new children that can be assigned a parent in the remaining set of vertices. The set $\mathfrak{L}_i$ is the set of vertices we need to consider in round i. We continue with this procedure until $\mathfrak{L}_i$ contains only one subbundle pair, the eldest l_e. It is not hard to see, that this must be the case for some $i \geq 0$ due to property (a) and (g). Then we have reached the root of the tree, which will be l_e, and $\vartheta_i = \vartheta_e$ is our decomposition tree for $e \in F_{\text{in}}$. Notice that property (e) ensures that ϑ_e indeed is a tree.

The following observation is a consequence of properties (b), (c) and (d) and the definition of $\mathfrak{L}_0$.

Observation 207. *Let $e \in F_{\text{in}}$. For all subbundle pairs $p \in V(\vartheta_e)$ there exists a vertex c with $c \in N_e$ or $c \in S_e$ such that $p = [e, c]$. Furthermore, for all $c \in N_e$ the subbundle pair $[e, c] \in V(\vartheta_e)$.*

In the following we explain how to obtain $\mathfrak{L}_{i+1}$ and ϑ_{i+1} from $\mathfrak{L}_i$ and ϑ_i for $i \geq 0$. If $|\mathfrak{L}_i| = 1$, that is, $\mathfrak{L}_i = \{l_e\}$, we let $\mathfrak{L}_{i+1} := \mathfrak{L}_i$ and $\vartheta_{i+1} := \vartheta_i$. Then clearly, all properties are satisfied for $\mathfrak{L}_{i+1}$ and ϑ_{i+1} given that they hold for $\mathfrak{L}_i$ and ϑ_i. In the next couple paragraphs we create the prerequisites to define $\mathfrak{L}_{i+1}$ and ϑ_{i+1} in the more complicated case that $\mathfrak{L}_i$ contains at least two subbundle pairs.

Let $i \geq 0$ and $|\mathfrak{L}_i| > 1$. Notice that $|\mathfrak{L}_i| > 1$ implies $|\mathfrak{L}_j| > 1$ for all $0 \leq j \leq i$. For all $[e, a] \in \mathfrak{L}_i$ let

$$R_i(e, a) := V(e, a) \cap \bigcup_{[e,b] \in \mathfrak{L}_i \setminus \{[e,a]\}} V(e, b).$$

The set $R_i(e, a)$ of vertices is a connected subset of the vertex set of T_e:

Lemma 208. *Let $|\mathfrak{L}_i| > 1$ and $[e, a] \in \mathfrak{L}_i$. Then $R_i(e, a) \subseteq S_e$ and $T_e[R_i(e, a)]$ is a subtree of T_e and contains vertex e.*

Proof. Let $|\mathfrak{L}_i| > 1$ and $[e, a] \in \mathfrak{L}_i$. As an immediate consequence of Corollary 193 and Lemma 200 we obtain $R_i(e, a) \subseteq S_e$. Since $e \in V(e, a)$ and $e \in V(e, b)$ for all $[e, b] \in \mathfrak{L}_i \setminus \{[e, a]\}$, we clearly have $e \in R_i(e, a)$. Now let us assume there exists a vertex $v \in R_i(e, a)$ such that there is no path from e to v in $T_e[R_i(e, a)]$. We know that $v \in V(e, a)$, and let $v \in V(e, b)$ for $[e, b] \in \mathfrak{L}_i \setminus \{[e, a]\}$. As $v \in R_i(e, a) \subseteq S_e$, there exists exactly one path from e to v in T_e. Since $e, v \in V(e, a)$ and $e, v \in V(e, b)$ the vertices on this path must be a subset of $V(e, a)$ and $V(e, b)$ by Lemma 196. Hence, the vertices on this path also belong to $R_i(e, a)$. ☐

Let $L_i(e, a)$ be the set of leaves of the subtree $T_e[R_i(e, a)]$. Clearly, $|L_i(e, a)| \geq 1$. If $|L_i(e, a)| = 1$, then $R_i(e, a)$ are the vertices on the path from e to $a' \in L_i(e, a)$ in T_e, and $R_i(e, a) = V(e, a')$ (Corollary 192). For a subbundle pair $[e, a] \in \mathfrak{L}_i$ let $\mathfrak{R}_i(e, a)$ be the set of subbundle pairs $[e, b] \in \mathfrak{L}_i$ with $R_i(e, a) \subseteq V(e, b)$. Notice that for all subbundle pairs $[e, a]$ we have $[e, a] \in \mathfrak{R}_i(e, a)$. Thus, $|\mathfrak{R}_i(e, a)| \geq 1$ for each subbundle pair $[e, a] \in \mathfrak{L}_i$.

Lemma 209. *Let* $|\mathfrak{L}_i| > 1$ *and* $[e,a] \in \mathfrak{L}_i$*. If* $|L_i(e,a)| = 1$*, then* $|\mathfrak{R}_i(e,a)| \geq 2$*.*

Proof. Let $|\mathfrak{L}_i| > 1$, $[e,a] \in \mathfrak{L}_i$ and $|L_i(e,a)| = 1$. Further, let $a' \in L_i(e,a) \subseteq R_i(e,a)$. By definition of $R_i(e,a)$ there must exist a subbundle pair $[e,c] \in \mathfrak{L}_i$ with $[e,c] \neq [e,a]$ such that $a' \in V(e,c)$. The set $R_i(e,a)$ contains the vertices of the only path from e to a' in T_e (Lemma 208). Since $e, a' \in V(e,c)$, we must also have $R_i(e,a) \subseteq V(e,c)$ by Lemma 196. It follows that $[e,c] \in \mathfrak{R}_i(e,a)$. □

Let $\mathfrak{V}_i$ be the subset of subbundle pairs $[e,a]$ of $\mathfrak{L}_i$ that satisfy all of the following properties:

1. $|L_i(e,a)| = 1$.
2. $[e,a] \neq l_e$.
3. There is at most one subbundle pair $[e,b] \in \mathfrak{R}_i(e,a)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$.
4. For all subbundle pairs $[e,c] \in \mathfrak{R}_i(e,a)$ we have $L_i(e,a) \subseteq L_i(e,c)$.

The following lemma will be used later to show that we assign each node in our tree ϑ_e to only one parent node.

Lemma 210. *Let* $|\mathfrak{L}_i| > 1$*. Let* $[e,a] \in \mathfrak{V}_i$ *and* $[e,c] \in \mathfrak{R}_i(e,a)$ *with* $|L_i(e,c)| = 1$*. Then* $R_i(e,a) = R_i(e,c)$*.*

Proof. Let $|\mathfrak{L}_i| > 1$, $[e,a] \in \mathfrak{V}_i$ and $[e,c] \in \mathfrak{R}_i(e,a)$ with $|L_i(e,c)| = 1$. Since $[e,a] \in \mathfrak{V}_i$ and $[e,c] \in \mathfrak{R}_i(e,a)$, we have $L_i(e,a) \subseteq L_i(e,c)$ (property 4). Moreover, it must hold that $L_i(e,a) = L_i(e,c)$, because $|L_i(e,a)| = 1$ and $|L_i(e,c)| = 1$. By Lemma 208 the sets $R_i(e,a)$ and $R_i(e,c)$ are vertex sets of subtrees of T_e, which both contain the vertex e. As the two subtrees have the same set of leaves, they must be equal. Thus, $R_i(e,a) = R_i(e,c)$. □

We divide the set $\mathfrak{V}_i$ into the subsets $\mathfrak{V}_i^0$ and $\mathfrak{V}_i^1$. We let $\mathfrak{V}_i^0$ be the set of subbundle pairs $[e,a] \in \mathfrak{V}_i$ where there exists no subbundle pair $[e,b] \in \mathfrak{R}_i(e,a)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$, and $\mathfrak{V}_i^1$ be the set of subbundle pairs $[e,a] \in \mathfrak{V}_i$ where there exists exactly one subbundle pair $[e,b] \in \mathfrak{R}_i(e,a)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$.

Now we are ready to define $\mathfrak{L}_{i+1}$ and ϑ_{i+1}. First of all, we add all subbundle pairs in $\mathfrak{L}_i \setminus \mathfrak{V}_i$ to $\mathfrak{L}_{i+1}$, and all vertices and edges from ϑ_i to ϑ_{i+1}. In the following, we will list further subbundle pairs that are added to $\mathfrak{L}_{i+1}$ and more vertices and edges that are added to ϑ_{i+1}.

Let us look at subbundle pairs $[e,a] \in \mathfrak{V}_i^1$. There exists exactly one subbundle pair $[e,b]$ in $\mathfrak{R}_i(e,a)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$. For each subbundle pair $[e,a] \in \mathfrak{V}_i^1$, we add the edge $([e,b],[e,a])$ to $E(\vartheta_{i+1})$.

Before taking a look at the subbundle pairs in $\mathfrak{V}_i^0$, let us add a few remarks. Let $[e,a]$ be a subbundle pair in $\mathfrak{V}_i^1$ and let $[e,b]$ be the only subbundle pair in $\mathfrak{R}_i(e,a)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$. Then each subbundle pair $[e,c] \in \mathfrak{R}_i(e,a) \setminus \{[e,b]\}$ is in $\mathfrak{V}_i^1$, as for each subbundle pair $[e,c] \in \mathfrak{R}_i(e,a) \setminus \{[e,b]\}$ we have $R_i(e,c) = R_i(e,a)$ (Lemma 210), and therefore, $L_i(e,a) = L_i(e,c)$ and $\mathfrak{R}_i(e,c) = \mathfrak{R}_i(e,a)$. Note that $[e,c] \neq l_e$ because $[e,b]$ is the only subbundle pair in $\mathfrak{R}_i(e,a)$ for which $[e,b] = l_e$ might hold. It follows that $[e,b]$ is also the only subbundle pair in $\mathfrak{R}_i(e,c)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$. Thus, we add the edge $([e,b],[e,c])$ for every subbundle pair in $[e,c] \in \mathfrak{R}_i(e,a) \setminus \{[e,b]\}$. Now, the subbundle pairs in $\mathfrak{R}_i(e,a) \setminus \{[e,b]\} \subseteq \mathfrak{V}_i^0$ are not added to $\mathfrak{L}_{i+1}$ by default, but the parent vertex $[e,b]$ is an element of $\mathfrak{L}_i \setminus \mathfrak{V}_i$ and is therefore added to $\mathfrak{L}_{i+1}$. Since $\mathfrak{R}_i(e,a) \subseteq \mathfrak{L}_i \subseteq V(\vartheta_i) \subseteq V(\vartheta_{i+1})$, we know that all subbundle pairs in $\mathfrak{R}_i(e,a)$ are vertices of ϑ_{i+1}.

Next, let us consider subbundle pairs $[e,a] \in \mathfrak{V}_i^0$. As $|L_i(e,a)| = 1$, there is only one element a' in $L_i(e,a)$. Clearly, $[e,a']$ is a subbundle pair (Observation 195), and $a' \in S_e$ (Lemma 208). We add $[e,a']$ to $\mathfrak{L}_{i+1}$ and to $V(\vartheta_{i+1})$, and we add the edge $([e,a'],[e,a])$ to $E(\vartheta_{i+1})$. We say the subbundle pair $[e,a']$ is *generated* by subbundle pair $[e,a]$ in $\mathfrak{L}_i$.

Again we add some remarks. Since for all $[e,c] \in \mathfrak{R}_i(e,a)$ we have $|L_i(e,c)| = 1$ and $[e,c] \neq l_e$, we obtain $R_i(e,c) = R_i(e,a)$ (Lemma 210), and therefore, $L_i(e,c) = L_i(e,a)$ and $\mathfrak{R}_i(e,c) = \mathfrak{R}_i(e,a)$. Hence, every subbundle pair in $\mathfrak{R}_i(e,a)$ is in $\mathfrak{V}_i^0$ as well. As $L_i(e,c) = L_i(e,a)$, we add the edge $([e,a'],[e,c])$ for each subbundle pair $[e,c] \in \mathfrak{R}_i(e,a)$. Notice that the subbundle pairs in $\mathfrak{R}_i(e,a)$ are not added to $\mathfrak{L}_{i+1}$ by default as they are vertices in $\mathfrak{V}_i^0$; but they are vertices of ϑ_{i+1} as $\mathfrak{R}_i(e,a) \subseteq \mathfrak{L}_i \subseteq V(\vartheta_i) \subseteq V(\vartheta_{i+1})$.

Observation 211. *Let $|\mathfrak{L}_i| > 1$. If a subbundle pair $[e,a']$ is generated by a subbundle pair $[e,a]$ in $\mathfrak{L}_i$, then $R_i(e,a) = V(e,a')$.*

Proof. Let $|\mathfrak{L}_i| > 1$ and subbundle pair $[e,a']$ be generated by subbundle pair $[e,a]$ in $\mathfrak{L}_i$. Then $|L_i(e,a)| = 1$, and $a' \in L_i(e,a)$. By Lemma 208, $R_i(e,a)$ is the vertex set of a path from e to a' in T_e. Lemma 194 implies that $R_i(e,a) = V(e,a')$. □

Observation 212. *If subbundle pair $[e,b]$ becomes the parent of $[e,a] \in \mathfrak{L}_i$ in round i of the construction of ϑ_e. Then $[e,b] \in \mathfrak{L}_{i+1}$ and $R_i(e,a) \subseteq V(e,b)$.*

Proof. Let i be the round of the construction of ϑ_e where $[e,b]$ becomes the parent of $[e,a] \in \mathfrak{L}_i$. Only subbundle pairs in $\mathfrak{V}_i$ get a parent in round i of the construction of ϑ_e. Thus, $[e,a] \in \mathfrak{V}_i$. Let $[e,a] \in \mathfrak{V}_i^0$. Then the parent $[e,b]$ of $[e,a]$ is generated by $[e,a]$ in $\mathfrak{L}_i$, and $R_i(e,a) \subseteq V(e,b)$ follows from Observation 211. (Note that $|\mathfrak{L}_i| > 1$, because there cannot be generated any subbundle pairs in $\mathfrak{L}_i$ if $|\mathfrak{L}_i| = 1$.) Further, the generated subbundle pair $[e,b]$ is added to $\mathfrak{L}_{i+1}$. Let $[e,a] \in \mathfrak{V}_i^1$. Then the parent $[e,b]$ of $[e,a]$ is in $\mathfrak{R}_i(e,a)$, and therefore, $R_i(e,a) \subseteq V(e,b)$. As $[e,b]$ is the subbundle pair in $\mathfrak{R}_i(e,a)$ with $|L_i(e,b)| > 1$ or $[e,b] = l_e$, it holds that $[e,b] \notin \mathfrak{V}_i$. Thus, $[e,b] \in \mathfrak{L}_{i+1}$. □

Corollary 213. *Let $[e,b]$ be the parent of $[e,a]$ in ϑ_e. Then $V(e,a) \cap V(e,b)$ is a subset of S_e and induces a path from e to some node c in T_e.*

Proof. Let $[e,b]$ be the parent of $[e,a]$ in the directed tree ϑ_e. Let i be the round of the construction of ϑ_e in which $[e,b]$ becomes the parent of $[e,a] \in \mathfrak{L}_i$. First let us show that $V(e,a) \cap V(e,b) = R_i(e,a)$. Let us consider the case where $[e,b]$ is generated by $[e,a] \in \mathfrak{V}_i^0$. Then according to Observation 211, $R_i(e,a) = V(e,b)$. As $R_i(e,a) \subseteq V(e,a)$, it follows that $R_i(e,a) = V(e,a) \cap V(e,b)$. Now let us consider the case where $[e,a] \in \mathfrak{V}_i^1$. Then $[e,b] \in \mathfrak{L}_i$. Clearly, $V(e,a) \cap V(e,b) \subseteq R_i(e,a)$. Since $R_i(e,a) \subseteq V(e,a)$ and $R_i(e,a) \subseteq V(e,b)$ by Observation 212, we obtain $V(e,a) \cap V(e,b) \supseteq R_i(e,a)$. Thus, $V(e,a) \cap V(e,b) = R_i(e,a)$ in each case. As $[e,a] \in \mathfrak{V}_i$, we have $|L_i(e,a)| = 1$, and Corollary 213 follows from Lemma 208. □

Let us consider $\mathfrak{L}_{i+1}$ and ϑ_{i+1}. Clearly, the set $\mathfrak{L}_{i+1}$ only contains subbundle pairs that are already in $\mathfrak{L}_i$ and subbundle pairs that are generated by subbundle pairs $[e, a] \in \mathfrak{V}_i^0$ in $\mathfrak{L}_i$. It is not hard to see that the way we constructed $\mathfrak{L}_{i+1}$ and ϑ_{i+1} properties (a)-(d) and (f) must be satisfied. In order to prove property (e), which says that ϑ_{i+1} is a directed forest with $|\mathfrak{L}_{i+1}|$ connected components, where $\mathfrak{L}_{i+1}$ is the set of roots, we show that for all $[e, a] \in \mathfrak{V}_i^0$, the subbundle pair $[e, a']$ generated by $[e, a]$ in $\mathfrak{L}_i$ does not occur in $\bigcup_{j \leq i} \mathfrak{L}_j$.

Let us extend our notion of descendant. Let $e \in F_{\text{in}}$. We say a vertex a is a *proper extended descendant* of a vertex b in T_e if $a, b \in V_e$ and a is a proper descendant of b in T_e; or $a \in N_e \setminus V_e$, $b \in V_e$ and $b \in V(e, a)$. Let us denote the "proper extended descendant"-relation in T_e by $<_{\text{p.e.d.}}$.

Observation 214. *The "proper extended descendant"-relation in T_e is a strict partial order.*

Proof. Clearly, $<_{\text{p.e.d.}}$ is irreflexive. Let us show that $<_{\text{p.e.d.}}$ is transitive. Let $a <_{\text{p.e.d.}} b$ and $b <_{\text{p.e.d.}} c$. Let us consider the case where $a, b, c \in V_e$. Then a is a proper descendant of b and b is a proper descendant of c in T_e. Therefore, a is a proper descendant of c in T_e, and $a <_{\text{p.e.d.}} c$. In the case that $b \in N_e \setminus V_e$ or $c \in N_e \setminus V_e$, we cannot have $a <_{\text{p.e.d.}} b$ or $b <_{\text{p.e.d.}} c$. Thus, this case cannot occur. Let us consider the case where $a \in N_e \setminus V_e$ and $b, c \in V_e$. Then $a <_{\text{p.e.d.}} b$ and $b <_{\text{p.e.d.}} c$ implies that $b \in V(e, a)$ and that b is a proper descendant of c in T_e. By Observation 195, $[e, b]$ is a subbundle pair and $V(e, b) \subseteq V(e, a)$. As $e, b \in V_e$ and there is a path from e to b in T_e, Lemma 194 implies that $V(e, b)$ contains all vertices on this path from e to b, which includes c. Hence, $c \in V(e, b) \subseteq V(e, a)$. It follows that $a <_{\text{p.e.d.}} c$. □

We let a be an *extended descendant* of a vertex b in T_e if $a = b$ or a is a proper extended descendant of a vertex b in T_e.

Corollary 215. *The "extended descendant"-relation in T_e is a partial order.*

Observation 216. *Let $|\mathfrak{L}_i| > 1$. If a subbundle pair $[e, a]$ generates a subbundle pair $[e, a']$ in $\mathfrak{L}_i$, then a is an extended descendant of a'.*

Proof. Let $|\mathfrak{L}_i| > 1$. Let $[e, a]$ be a subbundle pair that generates the subbundle pair $[e, a']$ in $\mathfrak{L}_i$. In the case that $a \in N_e \setminus V_e$, we have $a' \in L_i(e, a) \subseteq R_i(e, a) \subseteq V(e, a)$, and by Lemma 208 $a' \in S_e \subseteq V_e$. Thus, if $a \in N_e \setminus V_e$, then a is an extended descendant of a'. Let us consider the case where $a \in V_e$. By Lemma 194 $[e, a]$ is a subbundle pair and $V(e, a)$ consists of all vertices of the path from e to a in T_e. As $R_i(e, a) \subseteq V(e, a)$, the induced subgraph $T_e[R_i(e, a)]$ is a subpath of the path $T_e[V(e, a)]$ by Lemma 208. Thus, the end a of the path $T_e[V(e, a)]$ is a descendant of the end a' of the subpath $T_e[R_i(e, a)]$ in T_e. It follows that, a is an extended descendant of a'. □

Observation 217. *If vertex a is an extended descendant of vertex b in T_e, then $[e, a]$ and $[e, b]$ are subbundle pairs and $V(e, b) \subseteq V(e, a)$.*

Proof. Let a be an extended descendant of vertex b in T_e. Let us consider the case where $a \in N_e \setminus V_e$. Then $[e, a]$ is a subbundle pair. If $a = b$, then, clearly, $[e, b]$ is a subbundle pair as well and $V(e,b) \subseteq V(e,a)$. If $a \neq b$, then $b \in V_e$ and $b \in V(e,a)$ and it follows from Observation 195 that $[e,b]$ is a subbundle pair and $V(e,b) \subseteq V(e,a)$. Now let us consider the case where $a \in V_e$. Then $b \in V_e$, and a is a descendant of b. In this case, it follows from Lemma 194 that $[e,a]$ and $[e,b]$ are subbundles pair and $V(e,b) \subseteq V(e,a)$. ☐

Observation 218. *Let $|\mathfrak{L}_i| > 1$. If $V(e,b) \subseteq V(e,a)$ for subbundle pairs $[e,a],[e,b] \in \mathfrak{L}_i$, then a is an extended descendant of vertex b in T_e.*

Proof. Let $|\mathfrak{L}_i| > 1$. Let $[e,a],[e,b] \in \mathfrak{L}_i$ be subbundle pairs with $V(e,b) \subseteq V(e,a)$. Clearly, a is an extended descendant of vertex b in T_e if $a = b$. Thus, let $a \neq b$. In the case that $a \in N_e \setminus V_e$ and $b \in S_e$, vertex a clearly is an extended descendant of vertex b in T_e. Let us consider the case where $a, b \in N_e$. Then $V(e,a) \cap V(e,b) \subseteq S_e$ by Lemma 200. Thus, $b \in V(e,b) \subseteq S_e$, which is a contradiction according to Observation 199. Now, let us consider the case where $a \in V_e$. Then $V(e,a) \subseteq V_e$ by Corollary 192. As $V(e,b) \subseteq V(e,a)$, it follows that $b \in V_e$. Since $a, b \in V_e$, Lemma 194 implies that $V(e,a)$ and $V(e,b)$ consist of all vertices of the path from e to a and all vertices of the from e to b, respectively. As $V(e,b) \subseteq V(e,a)$, it follows that a is a descendant of vertex b in T_e. Therefore, a is an extended descendant of vertex b in T_e. ☐

Lemma 219. *Let $|\mathfrak{L}_i| > 1$. Vertex a is not an extended descendant of b in T_e for*

- *all $[e,a] \in \mathfrak{L}_i$, $[e,b] \in \mathfrak{L}_i \setminus \{l_e\}$ with $a \neq b$, and*
- *all $[e,a] \in \mathfrak{L}_{i+1} \setminus \mathfrak{L}_i$ and $[e,b] \in \bigcup_{j \leq i} \mathfrak{L}_j \setminus \{l_e\}$.*

Proof. Let $|\mathfrak{L}_i| > 1$. Then $|\mathfrak{L}_j| > 1$ for all $j \leq i$. We prove this lemma by induction. Let $i = 0$. Let $[e,b] \in \mathfrak{L}_0 \setminus \{l_e\}$. Then $b \in N_e$, and there is no vertex a with $a \neq b$ such that a is an extended descendant of b in T_e. Now let $i > 0$. As inductive assumption, we suppose that for each $i' < i$ vertex a is not an extended descendant of b in T_e for

- all $[e,a] \in \mathfrak{L}_{i'}$, $[e,b] \in \mathfrak{L}_{i'} \setminus \{l_e\}$ with $a \neq b$ and
- all $[e,a] \in \mathfrak{L}_{i'+1} \setminus \mathfrak{L}_{i'}$ and $[e,b] \in \bigcup_{j \leq i'} \mathfrak{L}_j \setminus \{l_e\}$.

In the following we show the above for $i' = i$.

First, let us show that a is not an extended descendant of b in T_e for all $[e,a] \in \mathfrak{L}_i$ and all $[e,b] \in \mathfrak{L}_i \setminus \{l_e\}$ with $a \neq b$. We only need to consider the case where $[e,b] \in \mathfrak{L}_i \setminus \mathfrak{L}_{i-1}$ as otherwise a is not an extended descendant of b by inductive assumption. Let us assume a is an extended descendant of b. Since $[e,b] \notin \mathfrak{L}_{i-1}$, it has to be generated by a subbundle pair $[e,b']$ in $\mathfrak{L}_{i-1}$. Furthermore, we know $[e,b'] \in \mathfrak{V}^0_{i-1}$. Thus, $[e,b'] \neq l_e$. By Observation 211, we have $R_{i-1}(e,b') = V(e,b)$.

Let us consider the case where $[e,a] \in \mathfrak{L}_{i-1}$. According to Observation 217, the set $R_{i-1}(e,b') = V(e,b)$ is a subset of $V(e,a)$ as a is an extended descendant of b. Therefore, $[e,a] \in \mathfrak{R}_{i-1}(e,b')$, and $[e,a]$ is in $\mathfrak{V}^0_{i-1}$ as well. Thus, $[e,a]$ generates $[e,b]$ in $\mathfrak{L}_{i-1}$, and $[e,a]$ is not added to $\mathfrak{L}_i$ per se. Yet, $[e,a]$ is contained in $\mathfrak{L}_i$, and as $[e,a] \in \mathfrak{V}^0_{i-1}$, the only way for $[e,a]$ to be in $\mathfrak{L}_i$ is if it is generated by a subbundle pair $[e,a']$ in $\mathfrak{L}_{i-1}$. Then a' is an extended descendant of a (Observation 216). Further, $[e,a]$ and $[e,a']$ are distinct subbundle pairs in $\mathfrak{L}_{i-1}$ because $[e,a] \in \mathfrak{R}_{i-1}(e,b')$ generates $[e,b]$ with $a \neq b$, and $[e,a']$ generates $[e,a]$ in $\mathfrak{L}_{i-1}$. As $[e,a] \in \mathfrak{V}^0_{i-1}$, we have $[e,a] \neq l_e$. Thus, we have a contradiction to the first part of the inductive assumption.

Now let us consider the case where $[e,a] \notin \mathfrak{L}_{i-1}$. Then $[e,a]$ is generated by a subbundle pair $[e,a']$ in $\mathfrak{L}_{i-1}$. Hence, a' is an extended descendant of a (Observation 216). It follows that a' is an extended descendant of b by transitivity of the "extended descendant"-relation (Corollary 215). Observation 217 implies that $V(e,b) \subseteq V(e,a')$. As $R_{i-1}(e,b') = V(e,b)$, it follows that $[e,a'] \in \mathfrak{R}_{i-1}(e,b')$. Since $[e,b'],[e,a'] \in \mathfrak{V}^0_{i-1}$, and therefore, $|L_{i-1}(e,a')| = 1$, Lemma 210 implies that $R_{i-1}(e,a') = R_{i-1}(e,b')$. Consequently, $[e,a] = [e,b]$, a contradiction.

Next, we show that a is not an extended descendant of b in T_e for all $[e,a] \in \mathfrak{L}_{i+1} \setminus \mathfrak{L}_i$ and $[e,b] \in \bigcup_{j\leq i} \mathfrak{L}_j \setminus \{l_e\}$. Let $[e,a] \in \mathfrak{L}_{i+1} \setminus \mathfrak{L}_i$ and $[e,b] \in \mathfrak{L}_j \setminus \{l_e\}$ for $j \leq i$. Again, let us assume a is an extended descendant of b. As $[e,a] \notin \mathfrak{L}_i$, it has to be generated by a subbundle pair $[e,a']$ in $\mathfrak{L}_i$. Thus, a' is an extended descendant of a in T_e (Observation 216), and by transitivity of the "extended descendant"-relation (Corollary 215), also of b in T_e. Further, $[e,a'] \in \mathfrak{V}^0_i$, and therefore, $[e,a'] \neq l_e$. By Observation 211, we have $R_i(e,a') = V(e,a)$. Vertex a' must be different from b: Suppose $a' = b$. Then $a' = a$ as the "extended descendant"-relation is a partial order (Corollary 215). Therefore, $R_i(e,a') = V(e,a')$. According to the definition of $R_i(e,a')$, there exists a subbundle pair $[e,c] \in \mathfrak{L}_i \setminus \{[e,a']\}$ with $R_i(e,a') \subseteq V(e,c)$. Thus, $V(e,a') \subseteq V(e,c)$. Consequently, c is an extended descendant of a' by Observation 218, and we have a contradiction to the first part of the proof. It follows that $a' \neq b$. Let $i' \leq i$ be minimal such that for all k with $i' \leq k \leq i$ we have $[e,a'] \in \mathfrak{L}_k$. If $j \geq i'$, then the distinct subbundle pairs $[e,a']$ and $[e,b]$ are in $\mathfrak{L}_j$, a contradiction to the first part of our inductive assumption. If $j < i'$, we obtain a contradiction to the second part of the inductive assumption since $[e,a'] \in \mathfrak{L}_{i'} \setminus \mathfrak{L}_{i'-1}$ and $[e,b] \in \mathfrak{L}_j \setminus \{l_e\}$ for $j \leq i'-1$. □

Corollary 220. *Let $|\mathfrak{L}_i| > 1$ and $[e,a] \in \mathfrak{V}^0_i$. The subbundle pair $[e,a']$ generated by $[e,a]$ in $\mathfrak{L}_i$ does not occur in $\bigcup_{j<i} L_j$.*

Proof. Let $|\mathfrak{L}_i| > 1$. Then $|\mathfrak{L}_j| > 1$ for all $j \leq i$. Let $[e,a] \in \mathfrak{V}^0_i$ and $[e,a']$ be the subbundle pair generated by $[e,a]$ in $\mathfrak{L}_i$. First, let us show that $a' \neq a$. Suppose $a' = a$. Then $V(e,a) = R_i(e,a)$ by Observation 211. According to the definition of $R_i(e,a)$, there exists a subbundle pair $[e,c] \in \mathfrak{L}_i \setminus \{[e,a]\}$ with $R_i(e,a) \subseteq V(e,c)$. Therefore, $V(e,a) \subseteq V(e,c)$, and it follows from Observation 218 that c is an extended descendant of a in T_e. Further, $[e,a] \neq l_e$ because $[e,a] \in \mathfrak{V}^0_i$. Hence, we have a contradiction to the first part of Lemma 219.

Next, we prove that $[e,a'] \neq l_e$. Let us suppose $[e,a'] = l_e$. Then $[e,a'] \in \mathfrak{L}_i$ by property (a). Since $[e,a]$ generates $[e,a']$ in $\mathfrak{L}_i$, we have $R_i(e,a) = V(e,a')$, and $l_e = [e,a'] \in \mathfrak{R}_i(e,a)$. It follows that $[e,a]$ is not in $\mathfrak{V}^0_i$, and therefore, $[e,a]$ cannot generate $[e,a']$ in $\mathfrak{L}_i$, a contradiction.

Now, let us assume the subbundle pair $[e,a']$ is in $\mathfrak{L}_j$ for $j \leq i$. Let us show that $[e,a] \in \mathfrak{L}_k$ for all k with $j \leq k \leq i$. Suppose $[e,a] \notin \mathfrak{L}_{k'}$ for a k' with $j \leq k' \leq i$, and let k' be maximal with that property. Since $[e,a] \in \mathfrak{L}_i$ we know $k' < i$. Thus, $[e,a] \in \mathfrak{L}_{k'+1} \setminus \mathfrak{L}_{k'}$ and $[e,a'] \in \mathfrak{L}_j \setminus \{l_e\}$ for a $j \leq k'$. By Observation 216, vertex a is an extended descendant of a' in T_e. Hence, we obtain a contradiction to the second part of Lemma 219. Consequently, $[e,a] \in \mathfrak{L}_k$ for all k with $j \leq k \leq i$. Therefore, we have $[e,a] \in \mathfrak{L}_j$, $[e,a'] \in \mathfrak{L}_j \setminus \{l_e\}$ and $a' \neq a$. As a is an extended descendant of a' in T_e, we have a contradiction to the first part of Lemma 219. □

We can conclude: Once a subbundle pair is assigned a parent in round i of the construction of ϑ_e it will not occur again in $\mathfrak{L}_j$ for all $j > i$, as the only way to add it to $\mathfrak{L}_j$ for $j > i$ is by generating it from another subbundle pair which Corollary 220 tells us is not possible. Hence, we obtain the following corollary.

Corollary 221. *If $[e, a] \in \mathfrak{L}_i$ becomes the child of a subbundle pair in round i of the construction of ϑ_e. Then $[e, a] \notin \mathfrak{L}_j$ for all $j > i$.*

It follows immediately from the properties of ϑ_i, the construction of ϑ_{i+1} and Corollary 221 that ϑ_{i+1} is a directed forest with $|\mathfrak{L}_{i+1}|$ connected components, where $\mathfrak{L}_{i+1}$ is the set of roots. Thus, we have shown properties (a)-(f). It remains to show property (g). The following proposition shows that as long as $\mathfrak{L}_i$ contains more than one element there always exists a subbundle pair that satisfies properties 1-4 from above, that is, $\mathfrak{V}_i \neq \emptyset$ if $|\mathfrak{L}_i| > 1$.

Proposition 222. *Let $|\mathfrak{L}_i| > 1$. If $\mathfrak{V}_i = \emptyset$, then there exists a cordless cycle of length ≥ 4 in H.*

The application of this proposition in combination with Lemma 209 yields that property (g) must be satisfied as well: Let $|\mathfrak{L}_i| > 1$. Then Proposition 222 implies that $\mathfrak{V}_i \neq \emptyset$. Hence, there exists a subbundle pair $[e, a] \in \mathfrak{V}_i$. For this subbundle $[e, a]$ it holds that $|L_i(e, a)| = 1$. Consequently, $|\mathfrak{R}_i(e, a)| \geq 2$ according to Lemma 209. If $[e, a] \in \mathfrak{V}_i^0$, then all subbundle pairs in $\mathfrak{R}_i(e, a)$ are in $\mathfrak{V}_i^0$ and therefore not added to $\mathfrak{L}_{i+1}$; in place of the subbundle pairs in $\mathfrak{R}_i(e, a)$ one new subbundle pair is added to $\mathfrak{L}_{i+1}$. If $[e, a] \in \mathfrak{V}_i^1$, then only one subbundle pair from $\mathfrak{R}_i(e, a)$ is added to $\mathfrak{L}_{i+1}$, the other one are not added. Hence, in any case it follows that $|\mathfrak{L}_{i+1}| < |\mathfrak{L}_i|$.

Thus, the given sequences $\mathfrak{L}_i$ and ϑ_i for $i \geq 0$ indeed satisfy all the necessary properties.

Now it remains to proof Proposition 222. First we need to lay foundations.

Lemma 223. *Let $|\mathfrak{L}_i| > 1$. The set $\mathfrak{L}_i$ contains only subbundle pairs $[e, c]$ such that either $c \in N_e$ or $|L_i(e, c)| = 1$.*

Proof. Let $[e, c] \in \mathfrak{L}_i$. According to property (c) either $c \in N_e$ or $c \in S_e$. If $c \in S_e$, then $V(e, c)$ is the vertex set of a path from e to c in T_e (Corollary 192). If $V(e, c)$ is the vertex set of a path from e to c in T_e, then $R_i(e, c) \subseteq V(e, c)$ is the vertex set of a path from e to a vertex a' in T_e by Lemma 208. Therefore, $|L_i(e, c)| = 1$. □

Let $|\mathfrak{L}_i| > 1$. In the following we consider the case that there exists no subbundle pair that satisfies the above four properties. Thus, let $\mathfrak{V}_i = \emptyset$. Hence, each subbundle pair in $\mathfrak{L}_i$ does not satisfy at least one of the four properties. We call all subbundle pairs $[e, a] \in \mathfrak{L}_i \setminus \{l_e\}$ with $|L_i(e, a)| = 1$ *leaf pairs*. A leaf pair $[e, a]$ has to not satisfy property 3 or 4. We show that if $[e, a]$ is a leaf pair, then $[e, a]$ does not satisfy property 3:

Lemma 224. *Let $i \geq 0$ and $|\mathfrak{L}_i| > 1$. Further, let $\mathfrak{V}_i = \emptyset$. If $[e, a]$ is a leaf pair, then $[e, a] \in \mathfrak{L}_i$ does not satisfy property 3.*

Proof. Let $|\mathfrak{L}_i| > 1$, and let there be no subbundle pair that satisfies the above four properties. Let us assume $[e, a] \in \mathfrak{L}_i$ is a leaf pair that satisfies property 3. Then, we can give a sequence $([e, a_n])_{n\in\mathbb{N}}$ of leaf pairs $[e, a_n] \in \mathfrak{L}_i \setminus \{l_e\}$ with $\mathfrak{R}_i(e, a_n) \subseteq \mathfrak{R}_i(e, a)$ and $R_i(e, a_n) \subset R_i(e, a_{n+1})$ for all $n \in \mathbb{N}$. In such an infinite sequence of leaf pairs $([e, a_n])_{n\in\mathbb{N}}$ no two leaf pairs are the same. As the cardinality of $\mathfrak{L}_i \setminus \{l_e\}$ is finite, we obtain a contradiction.

Let $[e, a_1] := [e, a]$. Now let $[e, a_1], \ldots, [e, a_m]$ be the first m leaf pairs of the above sequence. Let us determine a leaf pair $[e, a_{m+1}]$ such that all properties are satisfied for the first $m+1$ elements of the sequence. If the leaf pair $[e, a_m]$ does not satisfy property 3, then $[e, a]$ cannot satisfy property 3 either, because $\mathfrak{R}_i(e, a_m) \subseteq \mathfrak{R}_i(e, a)$. Consequently, $[e, a_m]$ satisfies property 3, and $[e, a_m]$ does not satisfy property 4. Thus, there exists a subbundle pair $[e, a_m^1] \in \mathfrak{R}_i(e, a_m)$ such that $L_i(e, a_m) \not\subseteq L_i(e, a_m^1)$. As $[e, a_m]$ is a leaf pair, $|L_i(e, a_m)| = 1$ and $R_i(e, a_m)$ is a path from e to $a' \in L_i(e, a_m)$ in T_e (Lemma 208). Since $[e, a_m^1] \in \mathfrak{R}_i(e, a_m)$, the vertices of the path are in $V(e, a_m^1)$. Furthermore, they must belong to $R_i(e, a_m^1)$ as they are in $V(e, a_m) \cap V(e, a_m^1)$. Now $L_i(e, a_m) \not\subseteq L_i(e, a_m^1)$ implies that a' is not a leaf of the subtree induced by $R_i(e, a_m^1)$ in T_e (Lemma 208). Let $a'' \in V_e$ be a child of a' such that a'' belongs to the subtree induced by $R_i(e, a_m^1)$ in T_e. Then $a'' \in R_i(e, a_m^1)$. Thus, $a'' \in V(e, a_m^1)$, but $a'' \notin V(e, a_m)$ because $a'' \notin R_i(e, a_m)$. Hence, there must exist a subbundle pair $[e, a_m^2]$ different from $[e, a_m^1]$ and $[e, a_m]$ such that $a'' \in V(e, a_m^2)$. Since $e, a'' \in V_e$ and $e, a'' \in V(e, a_m^2)$, the vertices of the path from e to a'' in T_e are in $V(e, a_m^2)$ by Lemma 196. Thus, we have $R_i(e, a_m) \,\dot\cup \{a''\} \subseteq V(e, a_m^2)$. As a consequence, $[e, a_m^1]$ and $[e, a_m^2]$ are distinct subbundle pairs with $[e, a_m^1], [e, a_m^2] \in \mathfrak{R}_i(e, a_m)$ and $R_i(e, a_m) \,\dot\cup \{a''\} \subseteq V(e, a_m^1) \cap V(e, a_m^2)$. Since $[e, a_m]$ satisfies property 3, there exist a $j \in \{1, 2\}$ such that $|L_i(e, a_m^j)| = 1$ and $[e, a_m^j] \neq l_e$. We let $[e, a_{m+1}] := [e, a_m^j]$. Of course, subbundle pair $[e, a_{m+1}]$ is a leaf pair. Further, $[e, a_{m+1}]$ was chosen such that $R_i(e, a_m) \subset R_i(e, a_{m+1})$, because $R_i(e, a_m) \subset V(e, a_m^1) \cap V(e, a_m^2) \subseteq R_i(e, a_{m+1})$. As a consequence, $\mathfrak{R}_i(e, a_{m+1}) \subseteq \mathfrak{R}_i(e, a_m)$, and therefore, $\mathfrak{R}_i(e, a_{m+1}) \subseteq \mathfrak{R}_i(e, a)$. ☐

Proof (Proposition 222). Let $i \geq 0$, $|\mathfrak{L}_i| > 1$ and $\mathfrak{V}_i = \emptyset$. We consider sequences $(b_1, w_1), (b_2, w_2), \ldots, (b_m, w_m)$ of pairs of vertices of length $m \geq 2$ with the properties:

A) $[e, b_k] \in \mathfrak{L}_i$ for all $k \in [m]$,
B) $w_k \in L_i(e, b_k)$ for all $k \in [m]$,
C) $w_k \in R_i(e, b_{k+1})$ for all $k \in [m-1]$ and $w_m \in R_i(e, b_1)$, and
D) $b_k \neq b_{k'}$ and $w_k \neq w_{k'}$ for all $k \neq k'$ with $k, k' \in [m]$.

We show that such sequences exist. Then we take a shortest one, and show that we can find a cordless cycle of length ≥ 4.

In order to obtain a sequence of pairs with the above properties, we first construct another sequence $(a_1, v_1), (a_2, v_2), \ldots, (a_n, v_n)$ of pairs of vertices, where $n \geq 2$. This new sequence still satisfies property A and B, that is, for all $k \in [n]$ we have $(e, a_k) \in \mathfrak{L}_i$ and $v_k \in L_i(e, a_k)$, but we relax the other properties. Instead of property C, we only require that

C*) $v_k \in R_i(e, a_{k+1})$ for all $k \in [n-1]$, and instead of property D we ask for the following properties:

D1) $a_k \neq a_{k'}$ and $v_k \neq v_{k'}$ for all $k, k' \in [n-1]$ with $k \neq k'$,
D2) $a_k \neq a_{k+1}$ and $v_k \neq v_{k+1}$ for all $k \in [n-1]$, and
D3) there exists a $k \in [n-1]$ such that $a_n = a_k$ or $v_n = v_k$.

We construct the sequence $(a_1, v_1), (a_2, v_2), \dots, (a_n, v_n)$ inductively. Hence, for all subsequences $(a_1, v_1), \dots, (a_j, v_j)$ with $1 \leq j \leq n$, we maintain properties A, B (for all $k \in [j]$) and C*, D1, D2 (for all $k, k' \in [j-1]$), and we append pairs to the subsequence until property D3 is satisfied.

Let vertex a_1 be such that $[e, a_1] = l_e$, and let v_1 be any vertex in $L_i(e, a_1)$, then subsequence (a_1, v_1) satisfies all required properties and does not satisfy property D3. Now let us assume we have given a subsequence $(a_1, v_1), \dots, (a_j, v_j)$ that satisfies properties A, B, C*, D1 and D2 but not property D3. We want to determine (a_{j+1}, v_{j+1}). It is not hard to see that if properties D1 and D2 are satisfied for all $k, k' \in [j-1]$, but property D3 is not, that is, there does not exist a $k \in [j-1]$ such that $a_j = a_k$ or $v_j = v_k$, then $a_k \neq a_{k'}$ and $v_k \neq v_{k'}$ for all $k, k' \in [j]$. Thus, property D1 is satisfied for the sequence extended by (a_{j+1}, v_{j+1}) independent of our choice for a_{j+1} and v_{j+1}. As vertex v_j is in $L_i(e, a_j) \subseteq R_i(e, a_j)$ according to property B, there must exist a subbundle pair $[e, a'_j] \in \mathfrak{L}_i \setminus \{[e, a_j]\}$ such that $v_j \in V(e, a_j) \cap V(e, a'_j)$. Thus, $v_j \in R_i(e, a'_j)$.

If $|L_i(e, a'_j)| = 1$ and $[e, a'_j] \neq l_e$, then by Lemma 224 we know that $[e, a'_j]$ does not satisfy property 3. Therefore, there exist at least two subbundle pairs $[e, b] \in \mathfrak{L}_i$ such that $[e, b]$ is in $\mathfrak{R}_i(e, a'_j)$ and additionally $|L_i(e, b)| > 1$ or $[e, b] = l_e$. For each such subbundle pair $[e, b]$ we know that $v_j \in R_i(e, b)$, because $v_j \in V(e, b)$ (as $v_j \in R_i(e, a'_j) \subseteq V(e, b)$) and v_j is also contained in the vertex sets $V(e, a_j)$ and $V(e, a'_j)$, and at least one of the vertices a_j, a'_j is different from b. As there are at least two vertices b with the described properties, we pick one of them that is not equal to a_j and define it to be a_{j+1}. If $|L_i(e, a'_j)| > 1$ or $[e, a'_j] = l_e$, then we let a_{j+1} be a'_j. Clearly, this way $a_{j+1} \neq a_j$ and $v_j \in R_i(e, a_{j+1})$.

Now we have chosen a_{j+1} such that $[e, a_{j+1}] \in \mathfrak{L}_i$ (property A), $v_j \in R_i(e, a_{j+1})$ (property C*) and $|L_i(e, a_{j+1})| > 1$ or $[e, a_{j+1}] = l_e$. If $|L_i(e, a_{j+1})| > 1$, then we choose v_{j+1} arbitrarily from $L_i(e, a_{j+1}) \setminus \{v_j\}$. This way, $v_{j+1} \neq v_j$. Thus, properties B and D2 are satisfied. Let $|L_i(e, a_{j+1})| = 1$. Then $[e, a_{j+1}]$ must be $l_e = [e, a_1]$. We let v_{j+1} be v_1, the only vertex in $L_i(e, a_{j+1})$. Then property B holds trivially. Since $a_{j+1} \neq a_j$, we have $j \neq 1$. As $v_j \neq v_1$ (property D1), it follows that $v_{j+1} \neq v_j$. Hence, property D2 holds.

Now our sequence is extended by (a_{j+1}, v_{j+1}) such that properties A, B, C*, D1 and D2 are satisfied. If property D3 is satisfied as well, we are done. Otherwise, we continue with this recursive construction. As there are only finitely many subbundle pairs in $\mathfrak{L}_i$ the recursion must terminate at some point.

Now we use sequence $S = (a_1, v_1), (a_2, v_2), \dots, (a_n, v_n)$ to construct a sequence S' of the first form. Let $k < n$ be maximal such that $a_n = a_k$ or $v_n = v_k$. By property D2, $k \leq n-2$.

If we have $a_n = a_k$, we let $S' = (a_k, v_k), \dots, (a_{n-1}, v_{n-1})$. Then all properties are satisfied: Clearly, properties A and B hold. Property C is satisfied, because $v_{n-1} \in R_i(e, a_n)$ (property C*) implies $v_{n-1} \in R_i(e, a_k)$; and property D is satisfied for S' as property D1 holds for S. It remains to consider the case where $v_n = v_k$ but $a_n \neq a_k$. In this case we use $S' = (a_{k+1}, v_{k+1}), \dots, (a_n, v_n)$. Then again properties A and B are satisfied. Since v_k

is in $R_i(e, a_{k+1})$ (property C*), we obtain that $v_n \in R_i(e, a_{k+1})$. Thus, property C holds. By the choice of k and property D1, it follows that property D is satisfied. As $k \leq n-2$, S' is a sequence of length at least 2.

Now let $S'' = (b_1, w_1), (b_2, w_2), \ldots, (b_m, w_m)$ with $m \geq 2$ be a shortest sequence that satisfies properties A-D. In the following we prove that $C = b_1, w_1, b_2, w_2, \ldots, b_m, w_m, b_1$ is a cycle of length ≥ 4 without any chords. Since $b_k \neq b_{k'}$ and $w_k \neq w_{k'}$ for all $k \neq k'$ with $k, k' \in [m]$ by property D, we only need to show that $b_k \neq w_{k'}$ for all $k, k' \in [m]$ and that there is no chord between any non-consecutive vertices in C.

First, we show that there is no chord between w_j and $w_{j'}$ for $j, j' \in [m]$ and $j \neq j'$. Let us assume there exist $j \neq j'$ such that there is a chord between w_j and $w_{j'}$. By Lemma 208 the vertices w_j and $w_{j'}$ are nodes of T_e. As they are adjacent in H, vertex w_j is a proper ancestor of $w_{j'}$ in T_e or the other way around, according to Corollary 179. Without loss of generality let us assume that w_j is a proper ancestor of $w_{j'}$ and that $j = 1$. Then $j' \neq m$, because $w_{j'} \in R_i(e, b_1)$ (property C) implies that $w_1 \in L_i(e, b_1)$ cannot be a proper ancestor of $W_{j'}$ in T_e. Now let us remove the pairs $(b_2, w_2), \ldots, (b_{j'}, w_{j'})$ from S''. Note that the resulting sequence has length at most 2. According to property C vertex $w_{j'}$ is in $R_i(e, b_{j'+1})$. Further, all ancestors of $w_{j'}$ are in $R_i(e, b_{j'+1})$ (Lemma 208), and as w_1 is a proper ancestor of $w_{j'}$, we know that w_1 is contained in $R_i(e, b_{j'+1})$. Thus, we after removing the pairs $(b_2, w_2), \ldots, (b_{j'}, w_{j'})$ from S'', we obtain an even shorter sequence that satisfies properties A-D and has length at least 2, which is a contradiction.

Claim 225. *For all $k \in [m]$ we have $b_k \in N_e$.*

Proof. According to Lemma 223 for all subbundle pairs $[e, b] \in \mathfrak{L}_i$. We have either $|L_i(e, b)| = 1$ or $b \in N_e$. Let us assume there is a $k \in [m]$ such that $|L_i(e, b_k)| = 1$. Without loss of generality, let $k = 1$. Then $w_m, w_1 \in R_i(e, b_1)$ (property C). As $w_1 \in L_i(e, b_1)$ and $w_m \neq w_1$ (property D), it follows from Lemma 208 that w_1 is a proper descendant of w_m in T_e. Corollary 179 yields that there is an edge between w_1 and w_m, which is not possible as shown above. Hence, we obtain a contradiction, and $b_k \in N_e$. ⌟

Next we show that there are no chords between any vertices b_j and $b_{j'}$ for $j \neq j'$. By Lemma 225 vertex b_k is in N_e for all $k \in [m]$. As all vertices b_k with $k \in [m]$ are adjacent to e in $H[F]$ and $H[F]$ is a bipartite graph (Lemmas 80 and 185), there cannot be an edge between b_j and $b_{j'}$ for $j, j' \in [m]$ with $j \neq j'$.

Now we show that there are no chords between any non-consecutive vertices b_j and $w_{j'}$ of C. Let us assume there is a chord between the non-consecutive vertices b_j and $w_{j'}$ of C. Without loss of generality let $j = 1$. Then $j' \neq 1$ and $j' \neq m$. We prove that $S^* = (b_1, w_1), \ldots, (b_{j'}, w_{j'})$ is a shorter sequence that satisfies properties A-D and has length at least 2. Clearly, S^* satisfy properties A, B and D and the length of S^* is at least 2. By Claim 225 we have $b_1 \in N_e$, and Lemma 208 implies that $w_{j'} \in S_e$ as $w_{j'} \in L_i(e, b_{j'}) \subseteq R_i(e, b_{j'})$ (property B). Since b_1 and $w_{j'}$ are adjacent, it follows that $w_{j'} \in V(e, b_1)$ by Lemma 202. As $w_{j'}$ is also contained in $R_i(e, b_{j'}) \subseteq V(e, b_{j'})$, and $b_1 \neq b_{j'}$ by property D, vertex $w_{j'}$ is in $R_i(e, b_1)$. Hence, S^* also satisfies property C. Since S^* is shorter than S'', we obtain a contradiction.

Now we proved that there is no chord between any non-consecutive vertices in C. It remains to show that $b_k \neq w_{k'}$ for all $k, k' \in [m]$. By Claim 225 we have $b_k \in N_e$ for all

$k \in [m]$. Further, $w_{k'} \in L_i(e, b_{k'}) \subseteq R_i(e, b_{k'}) \subseteq S_e$ for all $k' \in [m]$ by Lemma 208. It follows from Observation 199 that $b_k \neq w_{k'}$ for all $k, k' \in [m]$.

Hence, C is a chordless cycle of length ≥ 4 in H. $\square$

Defining the Decomposition Tree of a Block in FP+C

In this section we show that the decomposition tree ϑ_e of a block $\mathfrak{b}_e$ for inner termini $e \in F_{\text{in}}$ is definable in fixed-point logic with counting.

We only considered one terminus $e \in F_{\text{in}}$ in the last section and defined the sets $\mathfrak{L}_i$, R_i, L_i, $\mathfrak{R}_i$, $\mathfrak{V}_i^0$, $\mathfrak{V}_i^1$ and the tree ϑ_i for each $i \geq 0$ for this terminus e. As they actually depend on the terminus $e \in F_{\text{in}}$ we denote them by $\mathfrak{L}_i^e$, R_i^e, L_i^e, $\mathfrak{R}_i^e$, $(\mathfrak{V}_i^0)^e$, $(\mathfrak{V}_i^1)^e$ and ϑ_i^e in the following.

The terminus $e \in F_{\text{in}}$ can be considered as a parameter. Within the following formulas, the variable x^* represents the parameter $e \in F_{\text{in}}$. Further, we constantly use the ono-to-one-correspondence between a set P of subbundle pairs that each contain terminus e and the set of vertices $P|_e$. Thus, when considering a set P of subbundle pairs $[e, a]$ that all contain the terminus $e \in F_{\text{in}}$, we use the vertex a to represent the subbundle pair $[e, a]$.

Let us suppose there is an FP+C-formula $\psi_{\mathfrak{L}_i}(x^*, x)$ such that for all vertices $e, a \in V(H')$ of an extended valid subgraph H' we have

$$H' \models \psi_{\mathfrak{L}_i}[e, a] \iff e \in F_{\text{in}} \text{ and } [e, a] \in \mathfrak{L}_i^e.$$

Note that $e \in [a, b]$ for each $[a, b] \in \mathfrak{L}_i^e$ for all $i \geq 0$ and $e \in F_{\text{in}}$ (property (c)). Thus, we have $\mathfrak{L}_i^e|_e = \psi_{\mathfrak{L}_i}[H', e; x]$ for $e \in F_{\text{in}}$.

Remember that there are TC-formulas for F, F_{in} and the edge relation $E(T_e)$ of the side tree T_e of e of H (see (6.15)); and for the subbundle pairs of H and the set $V(v, w)$ for vertices $v, w \in V$ (see (6.16)). Further, there is an STC+C-formula for l_e (see (6.17)). It is not hard to see that with these formulas and $\psi_{\mathfrak{L}_i}(x^*, x)$, we can construct FP+C-formulas $\psi_{R_i}(x^*, x, y)$, $\psi_{L_i}(x^*, x, y)$, $\psi_{\mathfrak{R}_i}(x^*, x, x')$, $\psi_{\mathfrak{V}_i^0}(x^*, x, x')$ and $\psi_{\mathfrak{V}_i^0}(x^*, x, x')$ such that for all vertices $e, a, b \in V(H')$ of an extended valid subgraph H' it holds that

$$\begin{aligned}
H' \models \psi_{R_i}[e, a, b] &\iff e \in F_{\text{in}}, [e, a] \in \mathfrak{L}_i^e \text{ and } b \in R_i^e(e, a),\\
H' \models \psi_{L_i}[e, a, b] &\iff e \in F_{\text{in}}, [e, a] \in \mathfrak{L}_i^e \text{ and } b \in L_i^e(e, a),\\
H' \models \psi_{\mathfrak{R}_i}[e, a, b] &\iff e \in F_{\text{in}}, [e, a] \in \mathfrak{L}_i^e \text{ and } [e, b] \in \mathfrak{R}_i^e(e, a),\\
H' \models \psi_{\mathfrak{V}_i^0}[e, a, b] &\iff e \in F_{\text{in}}, [e, a] \in \mathfrak{L}_i^e \text{ and } [e, b] \in (\mathfrak{V}_i^0)^e(e, a),\\
H' \models \psi_{\mathfrak{V}_i^1}[e, a, b] &\iff e \in F_{\text{in}}, [e, a] \in \mathfrak{L}_i^e \text{ and } [e, b] \in (\mathfrak{V}_i^1)^e(e, a).
\end{aligned}$$

We need formulas that define the sets $V(\vartheta_{i+1}) \setminus V(\vartheta_i)$ and $E(\vartheta_{i+1}) \setminus E(\vartheta_i)$. Except of the vertices of ϑ_i^e, the vertex set of ϑ_{i+1}^e contains all subbundle pairs $[e, b]$ that are generated by a subbundle pair $[e, a] \in (\mathfrak{V}_i^0)^e$ in $\mathfrak{L}_i^e$, that is, all subbundle pairs $[e, b]$ where $b \in L_i^e(e, a)$ for a subbundle pair $[e, a] \in (\mathfrak{V}_i^0)^e$. The set $E(\vartheta_{i+1}) \setminus E(\vartheta_i)$ contains all edges $([e, b], [e, a])$ where $[e, a] \in (\mathfrak{V}_i^0)^e$ and $[e, b]$ is the subbundle pair generated by $[e, a]$ in $\mathfrak{L}_i^e$, or where $[e, a] \in (\mathfrak{V}_i^1)^e$ and $[e, b]$ is the one subbundle pair $[e, b] \in \mathfrak{R}_i^e(e, a)$ with $|L_i^e(e, b)| > 1$ or $[e, b] = l_e$.

Clearly, we can define the sets $V(\vartheta_{i+1}) \setminus V(\vartheta_i)$ and $E(\vartheta_{i+1}) \setminus E(\vartheta_i)$ with the above formulas. Hence, there are FP+C-formulas $\psi_{V(\vartheta_{i+1})\setminus V(\vartheta_i)}$ and $\psi_{E(\vartheta_{i+1})\setminus E(\vartheta_i)}$ such that for all vertices $e, a, b \in V(H')$ of an extended valid subgraph H' we have

$$H' \models \psi_{V(\vartheta_{i+1})\setminus V(\vartheta_i)}[e,a] \iff e \in F_{\text{in}} \text{ and } [e,a] \in V(\vartheta_{i+1}) \setminus V(\vartheta_i), \tag{6.18}$$
$$H' \models \psi_{E(\vartheta_{i+1})\setminus E(\vartheta_i)}[e,a,b] \iff e \in F_{\text{in}} \text{ and } ([e,a],[e,b]) \in E(\vartheta_{i+1}) \setminus E(\vartheta_i).$$

The goal are FP+C-formulas that define the vertex set and the edge set of the decomposition tree ϑ_e of a block $\mathfrak{b}_e$ for $e \in F_{\text{in}}$. In the following we describe how to obtain them. The vertex set $V(\vartheta_e)$ and the edge set $E(\vartheta_e)$ can be defined in a similar way. We use a simultaneous inflationary fixed-point operator to define each of the sets. Within this simultaneous inflationary fixed-point operator we use two relational variables X_V and X_E such that $X_V^{i+1} = V(\vartheta_i^e)$ and $X_E^{i+1} = E(\vartheta_i^e)$, where X_V^i and X_E^i are the relations we obtain in the ith round of the recursion when interpreting the formula. Thus, in round $i+1$ of the recursion we define the directed forest ϑ_i^e. As there is an STC+C-formula for the set $\mathfrak{L}_e$ (see (6.17)), we can easily define $V(\vartheta_0^e) = \mathfrak{L}_e$ and $E(\vartheta_0^e) = \emptyset$ and initiate the recursion such that $X_V^1 = V(\vartheta_0^e)$ and $X_E^1 = E(\vartheta_0^e)$. Once we have X_V^{i+1} and X_E^{i+1} for $i \geq 0$, that is, the vertex set and the edge set of the directed forest ϑ_i^e, we can determine the set of roots of ϑ_i^e. According to property (e) the set of roots of ϑ_i^e is the set $\mathfrak{L}_i^e$. We let the formula $\psi_{\mathfrak{L}_i}$ from the beginning, be the formula that uses the relational variables X_V and X_E to defines the roots of the directed forest that the pair (X_V, X_E) of relational variables will be interpreted with. Then $\psi_{V(\vartheta_{i+1})\setminus V(\vartheta_i)}$ and $\psi_{E(\vartheta_{i+1})\setminus E(\vartheta_i)}$ define the set of vertices and the set of edges that we need to add in each round.

We can conclude the following.

Lemma 226. *There are* FP+C*-formulas* $\psi_{V(\vartheta_e)}(x^*, x)$ *and* $\psi_{E(\vartheta_e)}(x^*, x, x')$ *such that for all vertices* $e, a, b \in V(H')$ *of an extended valid subgraph* H' *we have*

$$H' \models \psi_{V(\vartheta_e)}[e,a] \iff e \in F_{\text{in}} \textit{ and } [e,a] \in V(\vartheta_e),$$
$$H' \models \psi_{E(\vartheta_e)}[e,a,b] \iff e \in F_{\text{in}} \textit{ and } ([e,a],[e,b]) \in E(\vartheta_e).$$

6.5.3. The Decomposition Tree of a Valid Subgraph

In the last section we created the directed tree $\vartheta(\mathfrak{b})$ for each block $\mathfrak{b} = (\mathfrak{L}, l)$ occurring in the simplified decomposition tree of a valid subgraph $H = (V, E)$ of bundle extension $\mathtt{H}^*$. In this section we attach the directed trees $\vartheta(\mathfrak{b})$ for $\mathfrak{b} \in \mathfrak{B}$ according to the simplified decomposition tree from Section 6.5.1 to obtain the actual decomposition tree of H.

Definition

The directed tree $\vartheta(\mathfrak{b})$ for each block $\mathfrak{b} = (\mathfrak{L}, l)$ has the property that l is the root of it and $\mathfrak{L}$ is a subset of its set of nodes. Further, if block $\mathfrak{b} = (\mathfrak{L}, l)$ is a child of a block $\mathfrak{b}' = (\mathfrak{L}', l')$ in the simplified decomposition tree, then the subbundle pair l is an element in $\mathfrak{L}'$(Observation 206). We can attach the directed tree $\vartheta(\mathfrak{b})$ to the directed tree $\vartheta(\mathfrak{b}')$ by gluing them together at the node l, the root of $\vartheta(\mathfrak{b})$, which also occurs in $\vartheta(\mathfrak{b}')$. We obtain the complete decomposition tree by attaching all directed trees for the blocks occurring in the simplified decomposition tree like that.

First of all, we show the following property:

Lemma 227. *Let $\mathfrak{b}, \mathfrak{b}' \in \mathfrak{B}$ be two blocks with $\mathfrak{b} \neq \mathfrak{b}'$. Then $V(\vartheta(\mathfrak{b})) \cap V(\vartheta(\mathfrak{b}')) \neq \emptyset$ if, and only if, $\mathfrak{b}'$ is the parent of $\mathfrak{b}$, $\mathfrak{b}$ is the parent of $\mathfrak{b}'$, or $\mathfrak{b}$ and $\mathfrak{b}'$ are children of the block $\mathfrak{b}_{root}$ in the simplified decomposition tree. Let $p \in V(\vartheta(\mathfrak{b})) \cap V(\vartheta(\mathfrak{b}'))$. Then*

- *p is the root of $\vartheta(\mathfrak{b})$ if $\mathfrak{b}'$ is the parent of $\mathfrak{b}$,*
- *p is the root of $\vartheta(\mathfrak{b}')$ if $\mathfrak{b}$ is the parent of $\mathfrak{b}'$, or*
- *p is the root of $\vartheta(\mathfrak{b})$ and of $\vartheta(\mathfrak{b}')$ if $\mathfrak{b}$ and $\mathfrak{b}'$ are children of the block $\mathfrak{b}_{root}$. (Then $\{p\} = V(\vartheta(\mathfrak{b}_{root}))$.)*

Proof. First, let us consider blocks of inner termini. Thus, let us consider $\mathfrak{b}_{e_1}, \mathfrak{b}_{e_2} \in \mathfrak{B}$ where $e_1, e_2 \in F_{\text{in}}$ with $e_1 \neq e_2$. By Observation 207 every subbundle pair $p_{e_1} \in V(\vartheta_{e_1})$ contains the terminus e_1 and a vertex from S_{e_1} or N_{e_1}. Similarly, every subbundle pair $p_{e_2} \in V(\vartheta_{e_2})$ contains the terminus e_2 and a vertex from S_{e_2} or N_{e_2}. By Lemma 189 we have $S_{e_1} \cap S_{e_2} = \emptyset$. Thus, we have $p_{e_1} = p_{e_2}$ for $p_{e_1} \in V(\vartheta_{e_1})$ and $p_{e_2} \in V(\vartheta_{e_2})$ if, and only if, e_1 and e_2 are adjacent in H and $p_{e_1} = p_{e_2} = [e_1, e_2]$. Hence, if there is no edge between e_1 and e_2 in $H[F]$, then the sets of nodes of ϑ_{e_1} and ϑ_{e_2} are disjoint. If there is an edge between e_1 and e_2 in $H[F]$, then $V(\vartheta_{e_1}) \cap V(\vartheta_{e_2}) = \{[e_1, e_2]\}$. Now, let $V(\vartheta_{e_1}) \cap V(\vartheta_{e_2}) \neq \emptyset$. Then there is an edge between e_1 and e_2 in $H[F]$, and due to the construction of the simplified decomposition tree either $\mathfrak{b}_{e_1}$ is the parent of $\mathfrak{b}_{e_2}$, $\mathfrak{b}_{e_2}$ is the parent of $\mathfrak{b}_{e_1}$, or $\mathfrak{b}_{e_1}$ and $\mathfrak{b}_{e_2}$ are the children of the block $\mathfrak{b}_{\text{root}}$ (then e_1 and e_2 are the two (adjacent) centroids of $H(F)$, Lemma 28). Clearly, if $\mathfrak{b}_{e_1}$ is the parent of $\mathfrak{b}_{e_2}$, then $[e_1, e_2]$ is the eldest of $\mathfrak{b}_{e_2}$ and therefore the root of ϑ_{e_2}. Analogously, $[e_1, e_2]$ is the root of ϑ_{e_2} if $\mathfrak{b}_{e_2}$ is the parent of $\mathfrak{b}_{e_1}$. Let $\mathfrak{b}_{e_1}$ and $\mathfrak{b}_{e_2}$ be the children of the block $\mathfrak{b}_{\text{root}}$. If the simplified decomposition tree contains the node $\mathfrak{b}_{\text{root}}$, then $H[F]$ has two centroids e_{r_1} and e_{r_2}. It follows that $\{e_1, e_2\} = \{e_{r_1}, e_{r_2}\}$. Thus, $[e_1, e_2] = [e_{r_1}, e_{r_2}]$ is the root of ϑ_{e_1} and ϑ_{e_2}. Further, the vertex set of $\vartheta(\mathfrak{b}_{\text{root}}) = (\{[e_{r_1}, e_{r_2}]\}, \emptyset)$ is $\{[e_1, e_2]\}$. Now we have shown Lemma 227 for distinct blocks $\mathfrak{b}_{e_1}$ and $\mathfrak{b}_{e_2}$ of inner termini $e_1, e_2 \in F_{\text{in}}$.

Next, let us suppose the block $\mathfrak{b}_{\text{root}} = ([e_{r_1}, e_{r_2}], \{[e_{r_1}, e_{r_2}]\})$ is involved. The children of $\mathfrak{b}_{\text{root}}$ are the blocks $\mathfrak{b}_{e_{r_1}}$ and $\mathfrak{b}_{e_{r_2}}$ of the two centroids e_{r_1} and e_{r_2} whose eldest is $[e_{r_1}, e_{r_2}]$. Thus, $[e_{r_1}, e_{r_2}]$, which is the only node of $\vartheta(\mathfrak{b}_{\text{root}})$, is the root of $\vartheta_{e_{r_1}}$ and $\vartheta_{e_{r_1}}$. Note that e_{r_1} and e_{r_2} are adjacent (Lemma 28). As there is no inner terminus $e \in F_{\text{in}}$ that is adjacent to e_{r_1} and e_{r_2} ($H(F)$ is a tree), $\mathfrak{b}_{e_{r_1}}$ and $\mathfrak{b}_{e_{r_2}}$ are the only blocks of inner termini whose directed tree contains the node $[e_{r_1}, e_{r_2}]$. □

Now we can define the *decomposition tree* $\mathcal{T} = (V_{\mathcal{T}}, E_{\mathcal{T}})$ of valid subgraph $H = (V, E)$. We also denote the tree by $\mathcal{T}(H) = (V_{\mathcal{T}(H)}, E_{\mathcal{T}(H)})$ if it is not clear from the context what valid subgraph we are referring to.

Let $H = (V, E)$ be a valid subgraph. Let F be the set of termini of H and F_{in} be the set of inner nodes of $H[F]$. If $F_{\text{in}} = \emptyset$, then $H[F]$ consists of at most two nodes. If $H[F]$ consists of only one node e, we let $\mathcal{T} := (\{[e, e]\}, \emptyset)$. If it consists of two nodes e, f, we define $\mathcal{T} := (\{[e, f]\}, \emptyset)$. Let $F_{\text{in}} \neq \emptyset$. Then, let $(\mathfrak{B}, \mathfrak{E})$ be the simplified decomposition tree of H. For each block $\mathfrak{b} \in \mathfrak{B}$ we constructed a tree $\vartheta(\mathfrak{b})$ of subbundle pairs in the previous section. We let $V_{\mathcal{T}} := \bigcup_{\mathfrak{b} \in \mathfrak{B}} V(\vartheta(\mathfrak{b}))$ be the set of nodes and $E_{\mathcal{T}} := \bigcup_{\mathfrak{b} \in \mathfrak{B}} E(\vartheta(\mathfrak{b}))$ be the set of edges of $\mathcal{T}$. It follows from Lemma 227 that $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ is a directed tree. Each block $\mathfrak{b} \in \mathfrak{B}$ is a block $\mathfrak{b}_e$ for an inner terminus $e \in F_{\text{in}}$ except if there are two centroids of $H[F]$, then $\mathfrak{B}$ additionally contains the block $\mathfrak{b}_{\text{root}}$. We know that $V(\vartheta(\mathfrak{b}_{\text{root}})) = \{[e_{r_1}, e_{r_2}]\}$ and $E(\vartheta(\mathfrak{b}_{\text{root}})) = \emptyset$. Further, subbundle pair $[e_{r_1}, e_{r_2}]$ is the root of the directed tree $\vartheta(\mathfrak{b}_{e_{r_1}})$ for the terminus $e_{r_1} \in F_{\text{in}}$. Thus, $V_\mathcal{T} = \bigcup_{e \in F_{\text{in}}} V(\vartheta_e)$ and $E_\mathcal{T} = \bigcup_{e \in F_{\text{in}}} E(\vartheta_e)$. Further, note that the root $r_\mathcal{T}$ of the decomposition tree is $[e_r, e_r]$ for $e_r \in C$ if $|C| = 1$, and $[e_{r_1}, e_{r_2}]$ for $e_{r_1}, e_{r_2} \in C$ with $e_{r_1} \neq e_{r_2}$ if $|C| = 2$, where C is the set of centroids of the bundle tree $H[F]$. Clearly, every directed tree ϑ_e for $e \in F_{\text{in}}$ is a subtree of $\mathcal{T}$.

Observation 207 implies the following observation.

Observation 228. *Let $F_{\text{in}} \neq \emptyset$. For all subbundle pairs $p \in V_\mathcal{T}$ there exists a terminus $e \in F_{\text{in}}$ and a vertex c with $c \in N_e$ or $c \in S_e$ such that $p = [e, c]$.*

Defining the Decomposition Tree in FP+C

The vertex set $V_\mathcal{T}$ and the edge set $E_\mathcal{T}$ of the decomposition tree $\mathcal{T}$ of a valid subgraph H can be defined in FP+C.

Since $V_\mathcal{T} = \bigcup_{e \in F_{\text{in}}} V(\vartheta_e)$ and $E_\mathcal{T} = \bigcup_{e \in F_{\text{in}}} E(\vartheta_e)$ for $|F_{\text{in}}| \geq 2$, we can use the FP+C-formulas $\psi_{V(\vartheta_e)}(x^*, x)$ and $\psi_{E(\vartheta_e)}(x^*, x, x')$ from Lemma 226 to define the vertex set $V_\mathcal{T}$ and the edge set $E_\mathcal{T}$, respectively. It is not hard to see that there are FP+C-formulas $\psi_{V_\mathcal{T}}(y_1, y_2)$ and $\psi_{E_\mathcal{T}}(y_1, y_2, z_1, z_2)$ such that for all extended valid subgraphs H' and all vertices $o_1, o_2, p_1, p_2 \in V(H')$ it holds that

$$\begin{aligned} H' \models \psi_{V_\mathcal{T}}[o_1, o_2] &\iff [o_1, o_2] \in V_\mathcal{T}, \\ H' \models \psi_{E_\mathcal{T}}[o_1, o_2, p_1, p_2] &\iff ([o_1, o_2], [p_1, p_2]) \in E_\mathcal{T}. \end{aligned} \tag{6.19}$$

Properties of the Decomposition Tree

Let $H = (V, E)$ be a valid subgraph and $\mathcal{T}$ be its decomposition tree.

Lemma 229. *Let $[u, v], [x, y] \in V_\mathcal{T}$ be subbundle pairs of H with $[u, v] \neq [x, y]$. If $V(u, v) \cap V(x, y) \neq \emptyset$, then there exists a terminus $e \in F_{\text{in}}$ such that $e \in \{u, v\} \cap \{x, y\}$.*

Proof. Let $[u, v], [x, y] \in V_\mathcal{T}$ be subbundle pairs with $[u, v] \neq [x, y]$. From $|V_\mathcal{T}| \geq 2$ it follows that $F_{\text{in}} \neq \emptyset$. According to Observation 228 we can suppose, without loss of generality, $u \in F_{\text{in}}$ and $v \in S_u$ or $v \in N_u$, and $x \in F_{\text{in}}$ and $y \in S_x$ or $y \in N_x$. Note that $u \in S_u$ and $x \in S_x$ by Corollary 201. Let there be a vertex $w \in V(u, v) \cap V(x, y)$.

First, let us assume that w is in the middle of H, that is, $w \in O$. If $v \in S_u$, then $w \in V(u, v) \subseteq S_u$ by Corollary 193. As $w \in O$, we cannot have $w \in S_u$. Hence, $v \in N_u$, and similarly we obtain that $y \in N_x$. Consequently, we have $u, v, x, y \in F$ where u and v are adjacent and x and y are adjacent. Thus, u and v are in different color classes of the 2-coloring $\{F_1, F_2\}$ of the bundle tree $H[F]$. The same holds for x and y. Without loss of generality, let $u, x \in F_1$ and $v, y \in F_2$. According to Lemma 197, $V(u, v) \subseteq S_u \cup V_v$ and $V(x, y) \subseteq S_x \cup V_y$. Since $w \in O$, it follows that $w \in V_v$ and $w \in V_y$. Then Corollary 187 implies that $v = y$. Thus, there is a terminus $e \in F$ such that $e \in \{u, v\} \cap \{x, y\}$ if $w \in O$.

Now let $w \in S_e$ for a terminus $e \in F$. Let us consider the subbundle pair $[u, v]$. First, let $v \in S_u$. Then $w \in V(u, v) \subseteq S_u$ by Corollary 193. Since $w \in S_u \cap S_e$, Lemma 189 implies that $u = e$. Now let $v \in N_u$. Then $w \in V(u, v) \subseteq S_u \cup O \cup S_v$ by Corollary 198. As $w \in S_e$, we have $e = u$ or $e = v$ by Lemma 189. Thus, in both cases we obtain $e \in \{u, v\}$. Analogously, we obtain $e \in \{x, y\}$ for the subbundle pair $[x, y]$. Thus, if $w \in S_e$ for a terminus $e \in F$, then $e \in \{u, v\} \cap \{x, y\}$.

Hence, there exists a terminus $e \in F$ such that $e \in \{u, v\} \cap \{x, y\}$. Let $e \in F$ be this terminus. We show that $e \in F_{\text{in}}$. Let us suppose $e \in F \setminus F_{\text{in}}$. Then by Observation 228 the vertex $f_{u,v}$ in $\{u, v\} \setminus \{e\}$ and the vertex $f_{x,y}$ in $\{x, y\} \setminus \{e\}$ must be in F_{in}. Note that $[u, v] = [e, f_{u,v}]$ and $[x, y] = [e, f_{x,y}]$. Since $[e, f_{u,v}]$ and $[e, f_{x,y}]$ are subbundle pairs and $e \neq f_{u,v}$ and $e \neq f_{x,y}$, vertex e is adjacent to $f_{u,v}$ and $f_{x,y}$. As $e \in F \setminus F_{\text{in}}$, which means e has only one neighbor in $H[F]$, we obtain $f_{u,v} = f_{x,y}$. Consequently, $[u, v] = [x, y]$, a contradiction. □

Corollary 230. *Let $[u, v], [x, y] \in V_{\mathcal{T}}$ be subbundle pairs of H with $[u, v] \neq [x, y]$. If $V(u, v) \cap V(x, y) \neq \emptyset$, then there exists a terminus $e \in F_{\text{in}}$ such that $[u, v], [x, y] \in V(\vartheta_e)$.*

Proof. Let $[u, v], [x, y] \in V_{\mathcal{T}}$ be subbundle pairs with $[u, v] \neq [x, y]$. Further, let $V(u, v) \cap V(x, y) \neq \emptyset$. By Lemma 229 there exists a terminus $e \in F_{\text{in}}$ such that $e \in \{u, v\} \cap \{x, y\}$. We prove that $[u, v]$ and $[x, y]$ are nodes of ϑ_e. For a contradiction, let us assume $[u, v] \notin V(\vartheta_e)$. Then there exists an $f \in F_{\text{in}}$ with $f \neq e$ and $[u, v] \in V(\vartheta_f)$. According to Observation 207, $f \in \{u, v\}$. Thus, $[u, v] = [e, f]$. As $[e, f]$ is a subbundle pair and $e \neq f$, there is an edge between e and f. Consequently, $f \in N_e$. Observation 207 implies that $[e, f] \in V(\vartheta_e)$. Hence, $[u, v] \in V(\vartheta_e)$. Analogously, it can be shown that $[x, y] \in V(\vartheta_e)$. □

Lemma 231. *Let $e \in F_{\text{in}}$ and let $[a, b]$ be a subbundle pair in ϑ_e. Further, let $k \geq 0$ be such that $[a, b] \in \mathfrak{L}_k$. Then for each $i \leq k$ there exists a subbundle pair $[a_i, b_i] \in \mathfrak{L}_i$ with $V(a, b) \subseteq V(a_i, b_i)$. Further, $[a_i, b_i]$ is a descendant of $[a, b]$ in ϑ_e.*

Proof. Let $e \in F_{\text{in}}$ and let $[a, b] \in V(\vartheta_e)$. Let $k \geq 0$ be such that $[a, b] \in \mathfrak{L}_k$. We prove inductively, that there exists a subbundle pair $[a_i, b_i] \in \mathfrak{L}_i$ with $V(a, b) \subseteq V(a_i, b_i)$ such that $[a_i, b_i]$ is a descendant of $[a, b]$ in ϑ_e for all $0 \leq i \leq k$. For $i = k$ we let $[a_k, b_k] := [a, b]$. Now let $i < k$, and let there a subbundle pair $[a_{i+1}, b_{i+1}] \in \mathfrak{L}_{i+1}$ that is a descendant of $[a, b]$ with $V(a, b) \subseteq V(a_{i+1}, b_{i+1})$. If $[a_{i+1}, b_{i+1}]$ is in $\mathfrak{L}_i$, we let $[a_i, b_i] := [a_{i+1}, b_{i+1}]$. Then $V(a, b) \subseteq V(a_i, b_i)$ and $[a_i, b_i]$ is a descendant of $[a, b]$ by inductive assumption. If $[a_{i+1}, b_{i+1}]$ is not in $\mathfrak{L}_i$, it is generated by a subbundle pair $[x, y]$ in $\mathfrak{L}_i$, and we let $[a_i, b_i] := [x, y]$. Then $([a_{i+1}, b_{i+1}], [a_i, b_i])$ is an edge in ϑ_e. By Observation 211, it follows that $R_i(a_i, b_i) = V(a_{i+1}, b_{i+1})$. (Note that $|\mathfrak{L}_i| > 1$, because there cannot be generated any subbundle pairs in $\mathfrak{L}_i$ if $|\mathfrak{L}_i| = 1$.) Thus, $V(a_{i+1}, b_{i+1}) \subseteq V(a_i, b_i)$. By inductive assumption, we have $V(a, b) \subseteq V(a_i, b_i)$ and $[a_i, b_i]$ is a descendant of $[a, b]$ in ϑ_e. □

Let $p = [a, b]$ be a node of the decomposition tree $\mathcal{T} = (V_{\mathcal{T}}, E_{\mathcal{T}})$ of valid subgraph H. By $W(a, b)$ we denote the union of all sets $V(x, y)$ where $[x, y]$ is a descendant of $[a, b]$ in $\mathcal{T}$. Note that $V(a, b) \subseteq W(a, b)$ for all subbundle pairs $[a, b] \in V_{\mathcal{T}}$.

Lemma 232. *Let $p = [a, b]$ be a node of the decomposition tree $\mathcal{T}$. For all children $[u, v]$ of p, it holds that $W(u, v) \cap V(a, b) = V(u, v) \cap V(a, b)$.*

Proof. Let $[a,b] \in V_{\mathcal{T}}$. Let $[u,v]$ be a child of $[a,b]$ in $\mathcal{T}$. Clearly, $V(u,v) \cap V(a,b) \subseteq W(u,v) \cap V(a,b)$. Let us assume there exists a vertex $w \in W(u,v) \cap V(a,b)$ that is not in $V(u,v) \cap V(a,b)$. Let $[x,y]$ be a subbundle pair of minimal depth such that $V(x,y)$ contains w, and $[x,y]$ is a descendant of $[u,v]$ in $\mathcal{T}$. According to Corollary 230 there exists a terminus $e \in F_{\text{in}}$ such that $[x,y]$ and $[a,b]$ are nodes of the directed tree ϑ_e. As $[u,v]$ lies on the unique path from $[a,b]$ to $[x,y]$ in $\mathcal{T}$, and ϑ_e is a subtree of $\mathcal{T}$, $[u,v]$ is a node of ϑ_e as well. Since $w \notin V(u,v)$, subbundle pair $[x,y]$ is a proper descendant of $[u,v]$ in ϑ_e. Let $[x',y']$ be the parent of $[x,y]$ in ϑ_e. Since $[x,y]$ is of minimal depth such that $V(x,y)$ contains w, it follows that $w \notin V(x',y')$. Let i be the round during the construction of ϑ_e, where $[x,y] \in \mathfrak{L}_i$ but $[x,y] \notin \mathfrak{L}_{i+1}$, that is, where we assign $[x,y]$ to be the child of $[x',y'] \in \mathfrak{L}_{i+1}$. Then $R_i(x,y) \subseteq V(x',y')$ by Observation 212. Let us assume there exist a subbundle pair $[c,d] \in \mathfrak{L}_i$ with $[x,y] \neq [c,d]$ and $w \in V(c,d)$. Then $w \in R_i(x,y)$. As $R_i(x,y) \subseteq V(x',y')$ but w is not in $V(x',y')$, we obtain a contradiction. Thus, $[x,y]$ is the only subbundle pair in $\mathfrak{L}_i$ such that $V(x,y)$ contains vertex w. Since $[x',y'] \in \mathfrak{L}_{i+1}$ is a descendant of $[a,b]$ in ϑ_e, there exists a $j \geq i+1$ such that $[a,b] \in \mathfrak{L}_j$ (which follows from Observation 212 and Corollary 221). By Lemma 231 there must exist a subbundle pair $[a^*,b^*]$ such that $V(a,b) \subseteq V(a^*,b^*)$ and $[a^*,b^*] \in \mathfrak{L}_{i+1}$. As $[x,y]$ is the only subbundle pair in $\mathfrak{L}_i$ such that $w \in V(x,y)$ and $[a^*,b^*] \neq [x,y]$ ($[x,y] \notin \mathfrak{L}_{i+1}$), the valid $[a^*,b^*]$ is not in $\mathfrak{L}_i$ and must therefore be generated by a subbundle pair $[c,d]$ in $\mathfrak{L}_i$. Then $R_i(c,d) = V(a^*,b^*)$ and $V(a^*,b^*) \subseteq V(c,d)$. Consequently, $w \in V(c,d)$ and $[c,d] = [x,y]$. Hence, $[a^*,b^*]$ is the parent $[x',y']$ of $[x,y]$, which is a contradiction to the choice of $[x,y]$ since $w \in V(a^*,b^*)$. □

Lemma 233. *Let $p = [a,b]$ be a vertex of decomposition tree $\mathcal{T}$. For all children $[u_1,v_1]$ and $[u_2,v_2]$ of p, with $[u_1,v_1] \neq [u_2,v_2]$ we have $W(u_1,v_1) \cap W(u_2,v_2) \subseteq V(a,b)$.*

Proof. Let $[a,b] \in V_{\mathcal{T}}$, and let $[u_1,v_1]$ and $[u_2,v_2]$ be children of $[a,b]$ with $[u_1,v_1] \neq [u_2,v_2]$. Let us assume that there exists a vertex $w \in W(u_1,v_1) \cap W(u_2,v_2)$ that is not in $V(a,b)$. Let $[x_j,y_j]$ be a subbundle pair of minimal depth such that $V(x_j,y_j)$ contains w, and $[x_j,y_j]$ is a descendant of $[u_j,v_j]$ in $\mathcal{T}$ for $j \in \{1,2\}$. According to Corollary 230 there exists a terminus $e \in F_{\text{in}}$ such that $[x_1,y_1]$ and $[x_2,y_2]$ are nodes of the directed tree ϑ_e, and so must be $[a,b]$, as it is the least common ancestor of $[x_1,y_1]$ and $[x_2,y_2]$. For $j \in [2]$ let $[x'_j,y'_j]$ be the parent of $[x_j,y_j]$. Then $w \notin V(x'_j,y'_j)$ for $j \in [2]$. Further, let i_j be the round during the construction of ϑ_e, where $[x_j,y_j] \in \mathfrak{L}_{i_j}$ becomes the child of $[x'_j,y'_j]$. Without loss of generality let $i_1 \leq i_2$. Then $[x_1,y_1] \in \mathfrak{L}_{i_1}$ and $[x_2,y_2] \in \mathfrak{L}_{i_2}$. By Lemma 231 there must be a subbundle pair $[x_2^*,y_2^*] \in \mathfrak{L}_{i_1}$ with $V(x_2,y_2) \subseteq V(x_2^*,y_2^*)$. As $w \in V(x_2^*,y_2^*)$, vertex w must be in $R_{i_1}(x_1,y_1)$. Since $R_{i_1}(x_1,y_1) \subseteq V(x'_1,y'_1)$ (Observation 212) and w is not in $V(x'_1,y'_1)$, we obtain a contradiction. □

Observation 234. *Let $v \in V$ be a vertex of H. Then there is a node $[x,y] \in V_{\mathcal{T}}$ such that $v \in V(x,y)$.*

Proof. Let $v \in V$ be a vertex of H. According to Observation 203 there are termini $e,e' \in F$ such that $v \in V(e,e')$. First, let us consider the case where $e \neq e'$. Then $e \in N_{e'}$ by Observation 190. It follows from the construction of the decomposition tree $\mathcal{T}$ and Observation 207 that $[e,e']$ is a node of $\mathcal{T}$. Now, let us consider the case where $e = e'$. If e is the only terminus of H, then $[e,e]$ is a node of the decomposition $\mathcal{T}$. Let $|F| > 1$. Then there exists a terminus $e'' \in N_e$. Clearly, $v \in V(e,e'')$. Again, it follows from the construction of the decomposition tree $\mathcal{T}$ and Observation 207 that $[e, e'']$ is a node of $\mathcal{T}$. □

Observation 235. *Let $\{v, w\} \in E$ be an edge of H. Then there is a node $[x, y] \in V_{\mathcal{T}}$ such that $v, w \in V(x, y)$.*

Proof. Let $\{v, w\} \in E$ be an edge of H. According to Observation 204 there are termini $e, e' \in F$ with $e \neq e'$ such that $v, w \in V(e, e')$. Then $e \in N_{e'}$ by Observation 190. It follows from the construction of the decomposition tree $\mathcal{T}$ and Observation 207 that $[e, e']$ is a node of $\mathcal{T}$. □

Lemma 236. *Let $[a, b]$ be a vertex of decomposition tree $\mathcal{T}$. Let $[x, y]$ be a child of $[a, b]$ in $\mathcal{T}$. Let $v \in W(x, y)$ and $w \in V(a, b)$ be adjacent vertices of H. Then $w \in W(x, y)$ or $v \in V(a, b)$.*

Proof. Let $[x, y]$ be a child of $[a, b]$ in decomposition tree $\mathcal{T}$. Let $v \in W(x, y)$ and $w \in V(a, b)$ be adjacent vertices of H. According to Observation 235, there exists a node $[s, t] \in V_{\mathcal{T}}$ such that $v, w \in V(s, t)$. If $[s, t] = [a, b]$, then clearly $v \in V(a, b)$. If $[s, t]$ is a descendant of $[x, y]$, then $w \in W(x, y)$. In the case that there is a child $[x', y']$ of $[a, b]$ with $[x', y'] \neq [x, y]$ and $[s, t]$ is a descendant of $[x', y']$, we have $v \in W(x', y')$, and Lemma 233 implies that $v \in V(a, b)$. It remains to consider the case, where $[s, t]$ is a proper ancestor of $[a, b]$. Then there is a path from $[s, t]$ to $[a, b]$ in $\mathcal{T}$. Let $[s, t] = [s_0, t_0], [s_1, t_1], \dots, [s_k, t_k] = [a, b]$ be this path. We inductively show that $v \in V(s_i, t_i)$ for all $i \in [0, k]$. Then it follows that $v \in [a, b]$. Clearly, $v \in V(s_0, t_0)$. Let $i \in [k]$. Suppose $v \in V(s_{i-1}, t_{i-1})$ and let us show that $v \in V(s_i, t_i)$. As $v \in W(x, y)$ and $[x, y]$ is a descendant of $[s_i, t_i]$, we have $v \in W(s_i, t_i)$. Then Lemma 232 implies that $v \in V(s_i, t_i) \cap V(s_{i-1}, t_{i-1})$. Thus, $v \in V(s_i, t_i)$. □

Lemma 237. *Let $[a, b]$ be a vertex of decomposition tree $\mathcal{T}$. Let $[x, y]$ and $[x', y']$ be distinct children of $[a, b]$ in $\mathcal{T}$. Let $v \in W(x, y)$ and $v' \in W(x', y')$ be adjacent vertices of H. Then $v' \in W(x, y)$, $v \in W(x', y')$ or $v, v' \in V(a, b)$.*

Proof. Let $[x, y]$ and $[x', y']$ be distinct children of $[a, b]$ in $\mathcal{T}$. Let $v \in W(x, y)$ and $v' \in W(x', y')$ be adjacent vertices of H. According to Observation 235, there exists a node $[s, t] \in V_{\mathcal{T}}$ such that $v, v' \in V(s, t)$. If $[s, t] = [a, b]$, then clearly $v, v' \in V(a, b)$. If $[s, t]$ is a descendant of $[x, y]$ or $[x', y']$, then $v' \in W(x, y)$ or $v \in W(x', y')$. Let us consider the case where there is a child $[z, z']$ of $[a, b]$ with $[z, z'] \neq [x, y]$ and $[z, z'] \neq [x', y']$ and $[s, t]$ is a descendant of $[z, z']$. Then $v, v' \in W(z, z')$, and Lemma 233 implies that $v, v' \in V(a, b)$. Finally let us consider the case where $[s, t]$ is a proper ancestor of $[a, b]$. Let $[s, t] = [s_0, t_0], [s_1, t_1], \dots, [s_k, t_k] = [a, b]$ be a path from $[s, t]$ to $[a, b]$ in $\mathcal{T}$. As in the proof of Lemma 236, we can show that $v, v' \in V(s_i, t_i)$ for all $i \in [0, k]$. It follows that $v, v' \in V(a, b)$. □

Lemma 238. *Let $[a, b]$ be a vertex of decomposition tree $\mathcal{T}$. Let $[x, y]$ be a child of $[a, b]$, and $[u, v]$ be the parent of $[a, b]$ in $\mathcal{T}$. Then $V(a, b) \cap V(x, y)$ is not a subset of $V(a, b) \cap V(u, v)$.*

Proof. Let $[a, b]$ be a vertex of decomposition tree $\mathcal{T}$. Let $[x, y]$ be a child of $[a, b]$, and $[u, v]$ be the parent of $[a, b]$. From $|V_{\mathcal{T}}| \geq 2$ it follows that $F_{\text{in}} \neq \emptyset$.

Let us consider the case where there does not exist an $e \in F_{\text{in}}$ such that $[x, y], [a, b], [u, v] \in V(\vartheta_e)$. There exists an $e_1 \in F_{\text{in}}$ such that $[x, y], [a, b] \in V(\vartheta_{e_1})$, and there exists an $e_2 \in F_{\text{in}}$ such that $[a, b], [u, v] \in V(\vartheta_{e_2})$. Clearly, $e_1 \neq e_2$. Observation 207 implies that $e_1 \in \{x, y\}, \{a, b\}$ and $e_2 \in \{a, b\}, \{u, v\}$. Thus, $[a, b] = [e_1, e_2]$. Observation 207 further implies that $[x, y] = [e_1, c]$ where $c \in N_{e_1}$ or $c \in S_{e_1}$. According to Lemma 200 or Corollary 193, we have $V(x, y) \cap V(a, b) \subseteq S_{e_1}$. Analogously, it can be shown that $V(u, v) \cap V(a, b) \subseteq S_{e_2}$. Since $V(a, b) \cap V(x, y)$ is not empty, it follows from Lemma 189 that $V(a, b) \cap V(x, y)$ is not a subset of $V(u, v) \cap V(x, y)$.

Now let us consider the case where there exists an $e \in F_{\text{in}}$ such that $[x, y], [a, b], [u, v] \in V(\vartheta_e)$. Observation 207 implies that, without loss of generality, $x = e$, $a = e$ and $u = e$. Let $[e, b]$ become the parent of $[e, y]$ in round i of the construction of ϑ_e. Then $[e, y] \in \mathfrak{L}_i$, and $R_i(e, y) \subseteq V(e, b)$ by Observation 212. In the following we show that $[e, b]$ is the only subbundle pair in $\mathfrak{L}_{i+1}$ with $R_i(e, y) \subseteq V(e, b)$. Suppose there is another one $[e, b']$.

In the case that $[e, b'] \in \mathfrak{L}_i$, we have $[e, b'] \in \mathfrak{R}_i(e, y)$. Assume $[e, y] \in \mathfrak{V}_i^0$. Then all subbundle pairs in $\mathfrak{R}_i(e, y)$ become children of the same node. Thus, like $[e, y]$ the subbundle pair $[e, b']$ becomes the child of $[e, b]$ and is not contained in $\mathfrak{L}_{i+1}$, a contradiction. Assume $[e, y] \in \mathfrak{V}_i^1$. Then there is one subbundle pair in $\mathfrak{R}_i(e, y)$ that becomes the parent of all other subbundle pairs in $\mathfrak{R}_i(e, y)$. As all subbundle pairs that become a child of a subbundle pair are not in $\mathfrak{L}_{i+1}$, the subbundle pair $[e, b']$ must be the parent of $[e, y]$. Hence $[e, b'] = [e, b]$, a contradiction.

Now let us consider the case where $[e, b'] \notin \mathfrak{L}_i$. Then $[e, b']$ has to be generated by a subbundle pair $[e, y']$ in $\mathfrak{L}_i$. According to Observation 211, we have $R_i(e, y') = V(e, b')$. Since $R_i(e, y) \subseteq V(e, b')$, it follows that $R_i(e, y) \subseteq R_i(e, y')$. Hence, $R_i(e, y) \subseteq V(e, y')$. Consequently, $[e, y'] \in \mathfrak{R}_i(e, y)$. As $[e, y'] \in \mathfrak{V}_i^0$, it follows that $[e, y']$ and $[e, y]$ become children of the same node. Thus, $[e, b] = [e, b']$ a contradiction.

Hence, $[e, b]$ is the only subbundle pair in $\mathfrak{L}_{i+1}$ with $R_i(e, y) \subseteq V(e, b)$. Let j be the round of the construction of ϑ_e where $[e, v]$ becomes the parent of $[e, b]$. It follows from Observation 212 and Corollary 221 that $j > i$. It holds that $[e, b] \in \mathfrak{L}_k$ for all $k \in \{i + 1, \ldots, j\}$. Inductively we can show for all $k \in \{i + 1, \ldots, j\}$ that $[e, b]$ is the only subbundle in $\mathfrak{L}_k$ with $R_i(e, y) \subseteq V(e, b)$. Clearly, this is the case for $k = i + 1$. Let $k \in \{i + 2, \ldots, j\}$. Let $[e, c] \in \mathfrak{L}_k \setminus [e, b]$. If $[e, c] \in \mathfrak{L}_{k-1}$, then $R_i(e, y)$ is not a subset of $[e, c]$ by inductive assumption. If $[e, c]$ is generated by a subbundle pair $[e, c'] \in \mathfrak{L}_{k-1} \setminus [e, b]$, then $V(e, c) = R_{k-1}(e, c') \subseteq V(e, c')$, and as $R_i(e, y)$ is not a subset of $V(e, c')$ by inductive assumption, it is not a subset of $V(e, c)$ either. Consequently, $[e, b]$ is the only subbundle in $\mathfrak{L}_j$ with $R_i(e, y) \subseteq V(e, b)$.

Since $R_i(e, y) \subseteq V(e, y)$ and $R_i(e, y) \subseteq V(e, b)$, it follows that $R_i(e, y) \subseteq V(e, y) \cap V(e, b)$, and therefore, $R_i(e, y) \subseteq V(e, v) \cap V(e, b)$. As $R_i(e, y) \in V(e, v)$, it holds that $[e, v] \notin \mathfrak{L}_j$. Thus, $[e, v]$ has to be generated by a subbundle pair $[e, d]$ in $\mathfrak{L}_j$. Then $R_i(e, y) \subseteq V(e, v) = R_j(e, d)$ (Observation 211). Hence, there must exist at least two subbundle pairs, $[e, d]$ and another one, in $\mathfrak{L}_j$ that contain $R_i(e, y)$, a contradiction. □

6.5.4. The Genealogical Decomposition Tree

We define the genealogical decomposition tree in this section.

Preliminaries

Let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be a bundle extension with underlying graph $\mathtt{H} = (\mathtt{V}, \mathtt{E})$. Let $\mathtt{F}$ be the set of termini of $\mathtt{H}$, and $\mathtt{O}$ be the middle of $\mathtt{H}$. Further, for $f \in \mathtt{F}$ let $\mathtt{S}_f$, $\mathtt{T}_f$ and $\mathtt{V}_f$ be the side of f, the side tree of f and the vertices of the side tree of f, respectively. For $f \in \mathtt{F}$ and $v \in \mathtt{V}_f$, let $\mathtt{anc}_f(v)$ be set of ancestors of v in $\mathtt{T}_f$, and $\mathtt{dec}_f(v)$ be the set of descendants of v in $\mathtt{T}_f$. Let $\prec_f$ be the strict partial order for $\mathtt{H}$ from Section 6.4.2 for $f \in \mathtt{F}$. We have already seen in Section 6.4.2 that for bundle extensions all of these sets/relations are definable in transitive closure logic.

For $f \in \mathtt{F}$ and $v \in \mathtt{V}_f$ we define the *side depth* $\mathtt{sd}_f(v)$ of v regarding f as the number of proper ancestors of v in $\mathtt{T}_f$, that is, $\mathtt{sd}_f(v) := |\mathtt{anc}_f(v) \setminus \{v\}|$. Then $\mathtt{sd}_f(f) = 0$ for each $f \in \mathtt{F}$. Surely, we can construct a TC+C-formula $\varphi_{\mathtt{sd}}(x^*, x, p)$ that is satisfied for $\mathtt{H}^*$ and $(f, v, l) \in \mathtt{U}^2 \times N(\mathtt{U})$ exactly if $f \in \mathtt{F}$, $v \in \mathtt{V}_f$ and $l = \mathtt{sd}_f(v)$ by using formulas $\varphi_{\mathtt{anc}}$ and $\varphi_{\mathtt{V}}$ from (6.10) and (6.11). We let $\mathtt{sd}([a, b]) := \min\{\mathtt{sd}_f(v) \mid f \in \mathtt{F}, v \in \{a, b\} \cap \mathtt{V}_f\}$ be the *side depth* of a subbundle pair $[a, b] \in \mathtt{P}_{\text{sub}}$ of $\mathtt{H}$. Thus, in order to determine the side depth of a subbundle pair $[a, b]$, we look which of the two vertices a and b are in $\mathtt{V}_{f_1}$ and for these we determine the side depth in $\mathtt{T}_{f_1}$; and we do the same for f_2. Then the side depth of $[a, b]$ is the minimum of the determined values. We also denote the side depth of a subbundle pair by $\mathtt{sd}(a, b)$.

Observation 239. *Let $[a, b]$ be a subbundle pair of $\mathtt{H}$. Let $\mathtt{sd}(a, b) = d$. Further, let $f \in \mathtt{F}$ and $v \in \mathtt{V}_f$. If $v \in \mathtt{V}(a, b)$, then $\mathtt{sd}_f(v) \geq d$.*

Proof. Let $[a, b]$ be a subbundle pair of $\mathtt{H}$. Let $\mathtt{sd}(a, b) = d$. Let $v \in \mathtt{V}(a, b)$. Without loss of generality, let $a \preceq_{f_1} b$. Then $a \preceq_{f_1} v \preceq_{f_1} b$. Let $v \in \mathtt{V}_{f_1}$. Then a is an ancestor of v in $\mathtt{T}_{f_1}$ according to Observation 165. Hence, $\mathtt{sd}_{f_1}(v) \geq \mathtt{sd}_{f_1}(a) \geq \mathtt{sd}(a, b) = d$. Analogously, we can show that $\mathtt{sd}_{f_2}(v) \geq d$ if $v \in \mathtt{V}_{f_2}$. □

Observation 240. *Let $[a, b]$ be a subbundle pair of H with $a \in \mathtt{V}_{f_1}$ and $b \in \mathtt{V}_{f_2}$. If $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(b) = l$, then $\mathtt{sd}(a, b) = l$.*

Proof. Let $[a, b]$ be a subbundle pair of H with $a \in \mathtt{V}_{f_1}$ and $b \in \mathtt{V}_{f_2}$. Let $\mathtt{sd}_{f_1}(a) = l$ and $\mathtt{sd}_{f_2}(b) = l$. Then $\mathtt{sd}(a, b) \leq l$. Suppose that $b \in \mathtt{V}_{f_1}$ and $\mathtt{sd}_{f_1}(b) < l$. Then $a, b \in \mathtt{V}_{f_1}$. By Observation 170 there is a path from b to a or from a to b in $\mathtt{T}_{f_1}$. As $\mathtt{sd}_{f_1}(a) > \mathtt{sd}_{f_1}(b)$, vertex b is proper ancestor of a in $\mathtt{T}_{f_1}$. Since $b \in \mathtt{V}_{f_1} \cap \mathtt{V}_{f_2} = \mathtt{O}$ (Observation 158), b is a leaf of $\mathtt{T}_{f_1}$, a contradiction. Similarly, we can show that there is no $a \in \mathtt{V}_{f_2}$ with $\mathtt{sd}_{f_2}(a) < k$. Hence, $\mathtt{sd}(a, b) = k$. □

A subbundle pair $[a, b]$ of $\mathtt{H}$ is *consistent* if there exists an $i \in [2]$ such that $a \in \mathtt{V}_{f_i}$, $b \in \mathtt{V}_{f_{3-i}}$ and $\mathtt{sd}_{f_i}(a) = \mathtt{sd}_{f_{3-i}}(b)$. We call a consistent subbundle pair of $\mathtt{H}$ a *consistent pair* of $\mathtt{H}$. If, in addition, a is a leaf of $\mathtt{T}_{f_i}$ or b is a leaf of $\mathtt{T}_{f_{3-i}}$, we say the consistent pair $[a, b]$ is *minimal*. We let $\mathtt{P}_{\text{con}}$ and $\mathtt{P}_{\text{con}}^{\text{nm}}$, respectively, be the set of consistent pairs and non-minimal consistent pairs of $\mathtt{H}$. It is not hard to see that minimal consistent pairs are TC+C-definable. Hence, there is a formula $\varphi_{\mathtt{P}_{\text{con}}^{\text{nm}}}(y_1, y_2)$ such that for each bundle extension $\mathtt{H}^*$ with underlying graph $\mathtt{H}$ and all $a, b \in \mathtt{U}(\mathtt{H}^*)$ we have

$$\mathtt{H}^* \models \varphi_{\mathtt{P}_{\text{con}}^{\text{nm}}}[a, b] \quad \Longleftrightarrow \quad [a, b] \text{ is non-minimal consistent pair of } \mathtt{H}. \tag{6.20}$$

Observation 241. *A minimal consistent pair is a trivial subbundle pair.*

Proof. Let $[a, b]$ be a minimal consistent pair of $\mathtt{H}$. Then there exists an $i \in [2]$ such that $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{3-i}}$, and a is a leaf of $\mathtt{T}_{f_i}$ or b is a leaf of $\mathtt{T}_{f_{3-i}}$. As $\mathtt{O} = \mathtt{O}_{f_1} = \mathtt{O}_{f_2}$, either $a, b \in \mathtt{V}_{f_1}$ or $a, b \in \mathtt{V}_{f_2}$. Thus, $[a, b]$ is a trivial subbundle pair. □

Observation 242. *A non-minimal consistent pair* $[a, b]$ *of* $\mathtt{H}$ *is a non-trivial subbundle pair of* $\mathtt{H}$.

Proof. Let $[a, b]$ be a consistent pair of $\mathtt{H}$ that is not minimal. As $[a, b]$ is consistent, there exists an $i \in [2]$ such that $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{3-i}}$. Let us assume $[a, b]$ is a trivial subbundle pair. Then $a, b \in \mathtt{V}_{f_i}$ or $a, b \in \mathtt{V}_{f_{3-i}}$. It follows that a or b is in $\mathtt{V}_{f_1} \cap \mathtt{V}_{f_2} = \mathtt{O}$ (Observation 158). Since $\mathtt{O} = \mathtt{O}_{f_1} = \mathtt{O}_{f_2}$, vertex a is a leaf of $\mathtt{V}_{f_i}$ or vertex b is a leaf of $\mathtt{V}_{f_{3-i}}$, a contradiction. □

An *affiliation* of a subbundle pair $[a, b] \in \mathtt{P}_{\text{sub}}$ of $\mathtt{H}$ can have two possible forms: If the side depth $\mathtt{sd}(a, b) = 0$, that is, if $a \in \mathtt{F}$ or $b \in \mathtt{F}$, then $[\,]$ is an affiliation of $[a, b]$. If $\mathtt{sd}(a, b) > 0$, then an affiliation of $[a, b]$ is a consistent pair $[a', b'] \in \mathtt{P}_{\text{con}}$, such that $\mathtt{sd}(a, b) = \mathtt{sd}(a', b') + 1$ and $\mathtt{V}(a, b) \subseteq \mathtt{V}(a', b')$. We denote a subbundle pair $p \in \mathtt{P}_{\text{sub}}$ with affiliation p' by $p_{p'}$, and we name it an *affiliated subbundle pair*. If the affiliation p' is a consistent pair $[a', b']$, then we denote the affiliated subbundle pair $p_{[a',b']}$ also by $p_{a',b'}$. We call p the *(underlying) subbundle pair* and p' the *affiliation* of an affiliated subbundle pair $p_{p'}$. If p is a consistent pair, then it has a unique affiliation, which can be determined given p.

Observation 243. *Let* $[a, b]$ *be a consistent pair of* $\mathtt{H}$ *where* $a \in \mathtt{V}_{f_1}$ *and* $b \in \mathtt{V}_{f_2}$. *Then there exists a (unique) affiliation* p' *of* $[a, b]$. *Moreover* $p' = [\,]$ *if* $[a, b] = [f_1, f_2]$, *otherwise* $p' = [a', b']$ *where* a' *is the parent of* a *in* $\mathtt{T}_{f_1}$ *and* b' *is the parent of* b *in* $\mathtt{T}_{f_2}$.

Proof. Let $[a, b]$ be a consistent pair of $\mathtt{H}$ where $a \in \mathtt{V}_{f_1}$ and $b \in \mathtt{V}_{f_2}$. As $[a, b]$ is consistent we have $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(b)$, or $a \in \mathtt{V}_{f_2}$, $b \in \mathtt{V}_{f_1}$ and $\mathtt{sd}_{f_2}(a) = \mathtt{sd}_{f_1}(b)$. First let us consider the latter case. If $a \in \mathtt{V}_{f_2}$ and $b \in \mathtt{V}_{f_1}$, then $a, b \in \mathtt{O}$ by Observation 158. As $\mathtt{O}$ is an independent set (Corollary 167), there cannot be an edge between a and b. Thus, we have $a = b$. Then $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_1}(b)$ and $\mathtt{sd}_{f_2}(a) = \mathtt{sd}_{f_2}(b)$. Hence, $\mathtt{sd}_{f_2}(a) = \mathtt{sd}_{f_1}(b)$ implies $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(b)$. It follows that $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(b)$ for all consistent pairs $[a, b]$ with $a \in \mathtt{V}_{f_1}$ and $b \in \mathtt{V}_{f_2}$. Let $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(b) = l$. Then $\mathtt{sd}(a, b) = l$ by Observation 240.

First, let $l = 0$. Then $\mathtt{sd}_{f_1}(a) = 0$ and $\mathtt{sd}_{f_2}(b) = 0$, and therefore, $a = f_1$ and $b = f_2$. It follows that $[a, b] = [f_1, f_2]$ and $[\,]$ is the (only) affiliation of $[a, b]$.

Now, let $l > 0$. Then $\{f_1, f_2\} \cap \{a, b\} = \emptyset$ because $\mathtt{sd}(a, b) > 0$. Thus, $[a, b]$ cannot have the affiliation $[\,]$. Let a' be the parent of a in $\mathtt{T}_{f_1}$, and b' be the parent of b in $\mathtt{T}_{f_2}$. Then $\mathtt{sd}_{f_1}(a') = \mathtt{sd}_{f_1}(a) - 1 = l - 1 = \mathtt{sd}_{f_2}(b) - 1 = \mathtt{sd}_{f_2}(b')$. Thus, $\mathtt{sd}(a', b') = l - 1$ according to Observation 240. Further $a' \preceq_{f_1} a$ and $b \preceq_{f_1} b'$. Since, $a \in \mathtt{V}_{f_1}$ and $b \in \mathtt{V}_{f_2}$, we have $a \preceq_{f_1} b$ (Observation 171). By transitivity of $\prec_{f_1}$, $a \preceq_{f_1} v \preceq_{f_1} b$ implies $a' \preceq_{f_1} v \preceq_{f_1} b'$ for all $v \in \mathtt{V}$. Thus, $\mathtt{V}(a, b) \subseteq \mathtt{V}(a', b')$. It follows that $[a', b']$ is an affiliation of $[a, b]$.

Let us assume there exist another affiliation $[a'', b'']$ of $[a, b]$. Since $[a'', b'']$ is consistent, there exists an $i \in [2]$ such that $a'' \in \mathtt{V}_{f_i}$, $b'' \in \mathtt{V}_{f_{3-i}}$ and $\mathtt{sd}_{f_i}(a'') = \mathtt{sd}_{f_{3-i}}(b'')$. Without loss of generality, let $i = 1$. Then $a'' \preceq_{f_1} b''$ by Observation 171. As $[a'', b'']$ is an affiliation of $[a, b]$, we have $\mathtt{sd}(a'', b'') = l - 1$. From Observation 240 it follows that $\mathtt{sd}_{f_1}(a'') = \mathtt{sd}_{f_2}(b'') = l - 1$. Since $\mathtt{V}(a, b) \subseteq \mathtt{V}(a'', b'')$, we have $a \in \mathtt{V}(a'', b'')$. Thus, $a'' \preceq_{f_1} a \preceq_{f_1} b''$. According to Observation 164, $a'' \prec_{f_1} a$ implies that a'' is an ancestor of a in $\mathtt{T}_{f_1}$. The only vertex $a'' \in V_{f_1}$ with $\mathtt{sd}_{f_1}(a'') + 1 = \mathtt{sd}_{f_1}(a)$ and a'' is an ancestor of a in $\mathtt{T}_{f_1}$, is the parent of a. Similarly, we can show that b'' must be the parent of b. □

Let p' be the affiliation of a subbundle pair $p \in \mathtt{P}_{\text{sub}}$. If p is not consistent and $p' \neq [\,]$, then we still can determine at least one vertex of the affiliation p'.

Observation 244. *Let $[a, b]$ be a subbundle pair of* H *where $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{i'}}$ with $i, i' \in [2]$. Let p' be an affiliation of $[a, b]$. If $\{f_1, f_2\} \cap \{a, b\} = \emptyset$, then p' contains the parent of a in $\mathtt{T}_{f_i}$ or p' contains the parent of b in $\mathtt{T}_{f_{i'}}$.*

Proof. Let $[a, b]$ be a subbundle pair of H where $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{i'}}$ with $i, i' \in [2]$. Let $\{f_1, f_2\} \cap \{a, b\} = \emptyset$. Then $[a, b]$ cannot have the affiliation $[\,]$. Thus, let $p' = [a', b'] \in \mathtt{P}_{\text{con}}$ be an affiliation of $[a, b]$. Let $\mathtt{sd}(a, b) = l$. Then, without loss of generality, $a \in \mathtt{V}_{f_1}$ and $\mathtt{sd}_{f_1}(a) = l$. As p' is consistent, we have, without loss of generality, $a' \in \mathtt{V}_{f_1}$, $b' \in \mathtt{V}_{f_2}$ and $\mathtt{sd}_{f_1}(a') = \mathtt{sd}_{f_2}(b')$. Then $a' \prec_{f_1} b'$ according to Observation 171. Since p' is an affiliation, we have $\mathtt{sd}(a', b') = l - 1$ and $\mathtt{V}(a, b) \subseteq \mathtt{V}(a', b')$. Observation 240 implies that $\mathtt{sd}_{f_1}(a') = l - 1$. Further, $a \in \mathtt{V}(a', b')$. Thus, $a' \preceq_{f_1} a \preceq_{f_1} b$. From Observation 164 it follows that a' is an ancestor of a in $\mathtt{T}_{f_1}$ As $\mathtt{sd}_{f_1}(a') + 1 = \mathtt{sd}_{f_1}(a)$, vertex a' is the parent of a. □

Observation 245. *Let $[a, b]$ be a trivial subbundle pair of* H *where $a, b \in \mathtt{V}_f$ and $f \in \mathtt{F}$. Then $V(a, b)$ is the vertex set of a directed path with ends a and b in $\mathtt{T}_f$. Let $p' \neq [\,]$ be an affiliation of $[a, b]$. If a (or b) is the first vertex of the path, then p' contains the parent of a (or b) in $\mathtt{T}_f$.*

Proof. Let $[a, b]$ be a trivial subbundle pair of H where $a, b \in \mathtt{V}_f$ and $f \in \mathtt{F}$. Then it follows from Observation 170 that $V(a, b)$ is the vertex set of a directed path with ends a and b in $\mathtt{T}_f$. Let $p' \neq [\,]$ be an affiliation of $[a, b]$. Without loss of generality, let a be the first vertex of the path. As $p' \neq [\,]$, we have $\{f_1, f_2\} \cap \{a, b\} = \emptyset$. According to Observation 244, the affiliation p' contains the parent of a or the parent of b in $\mathtt{T}_f$. If $a = b$, then clearly p' contains the parent of a in $\mathtt{T}_f$. Let $a \neq b$, and let us assume p' contains the parent b' of b in $\mathtt{T}_f$, but not the parent of a. Then $b' \in \mathtt{S}_f$. Since p' is consistent, it follows that $\mathtt{sd}_f(b') = \mathtt{sd}(p')$. As b' is a descendant of a in $\mathtt{T}_f$, we have $\mathtt{sd}_f(a) \leq \mathtt{sd}_f(b')$. It follows that $\mathtt{sd}(a, b) \leq \mathtt{sd}_f(a) \leq \mathtt{sd}(p')$. Thus, p' cannot be an affiliation of $[a, b]$, and we obtain a contradiction. □

Usually we assume that for an affiliated subbundle pair $[a, b]_{a',b'}$ where $a \in \mathtt{V}_{f_i}$ and $b \in \mathtt{V}_{f_{i'}}$, vertex a' is the parent of a in $\mathtt{T}_{f_i}$, or vertex b' is the parent of b in $\mathtt{T}_{f_{i'}}$. If $[a, b]$ is a consistent subbundle pair and it is possible to determine both vertices of its affiliation then we can omit denoting the determinable vertices in the affiliation and just write $[a, b]$ for the affiliated subbundle pair. An affiliated subbundle pair $p_{p'}$ is *trivial* if p is trivial. We call an affiliated subbundle pair $p_{p'}$ *consistent* if p is consistent. An *affiliated consistent pair* is a consistent affiliated subbundle pair. An affiliated consistent pair $p_{p'}$

is *minimal*, if p is minimal. We define the *side depth* $\mathtt{sd}(p_{p'})$ of an affiliated subbundle pair as the side depth $\mathtt{sd}(p)$ of the underlying subbundle pair p.

Let $[a, b]$ be a non-trivial subbundle pair of $\mathtt{H}$. $\mathtt{H}^{-}(a, b)$ is the subgraph of $\mathtt{H}$ induced by $\mathtt{V}(a, b) \setminus \{a, b\}$. The graph $\mathtt{H}^{-}(a, b)$ consist if connected components $C_1, \ldots, C_k$. For $i \in [k]$, let H_i be the subgraph induced by C_i. Note that H_i is a valid subgraph of $\mathtt{H}$. More precisely, H_i is the valid subgraph $H_{([a,b],c_i)}$ of $\mathtt{H}$ defined by the valid triple $([a, b], c_i)$ for $c_i \in C_i$. We define the *decomposition forest* $\mathcal{F}([a, b])$ (or $\mathcal{F}(a, b)$) as the disjoint union of the decomposition trees $\mathcal{T}(H_1), \ldots, \mathcal{T}(H_k)$ of the valid subgraphs $H_1, \ldots, H_k$. The set of roots $R_{\mathcal{F}(a,b)}$ of the decomposition forest $\mathcal{F}(a, b)$ is the set $\{r_{\mathcal{T}(H_1)}, \ldots, r_{\mathcal{T}(H_k)}\}$ of roots of the decomposition trees of $H_1, \ldots, H_k$. We let $V_{\mathcal{F}(a,b)}$ and $E_{\mathcal{F}(a,b)}$, respectively, be the set of nodes and the set of edges of the decomposition forest $\mathcal{F}(a, b)$.

Note that each subbundle pair of a valid subgraph of H is a subbundle pair of $\mathtt{H}$. Thus, the nodes of $\mathcal{F}(a, b)$ are subbundle pairs of $\mathtt{H}$. Further note that each non-minimal consistent pair $[a, b] \in \mathsf{P}_{\text{con}}$ of $\mathtt{H}$ is a non-trivial subbundle pair of $\mathtt{H}$ by Observation 242. Hence, the decomposition forest $\mathcal{F}(a, b)$ is defined for non-minimal consistent pairs $[a, b]$ of $\mathtt{H}$.

The following corollary is a consequence of Observation 234 and Lemma 184.

Corollary 246. *Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$. Let $v \in \mathtt{V}(a, b) \setminus \{a, b\}$ be a vertex of $\mathtt{H}^{-}(a, b)$ Then there is a node $[x, y] \in V_{\mathcal{F}(a,b)}$ such that $v \in \mathtt{V}(x, y)$.*

Observation 247. *Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$. If $\mathtt{sd}(a, b) = j$, then $\mathtt{sd}(x, y) = j + 1$ for all for all nodes $[x, y]$ of the decomposition forest $\mathcal{F}(a, b)$.*

Proof. Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$ where $a \in \mathtt{V}_{f_1}$ and $b \in \mathtt{V}_{f_2}$ and $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(b)$. Since neither a nor b is a leaf of $\mathtt{H}$, we have $a \in \mathtt{S}_{f_1}$ and $b \in \mathtt{S}_{f_2}$. Thus, $[a, b]$ is a non-trivial subbundle pair. Further, $a \prec_{f_1} b$ according to Observation 171. Let $\mathtt{sd}(a, b) = j$. Then $\mathtt{sd}_{f_1}(a) = j$ and $\mathtt{sd}_{f_2}(b) = j$ by Observation 240.

For each node $[x, y]$ of the decomposition forest $\mathcal{F}(a, b)$, there exists a $c \in \mathtt{V}^{-}(a, b)$ such that $[x, y]$ is a node of the decomposition tree of the valid subgraph $H_{([a,b],c)}$ of $\mathtt{H}$ defined by a valid triple $([a, b], c)$. Each node of the decomposition tree of the valid subgraph $H_{([a,b],c)}$ contains at least one inherited terminus. This follows from the definition of the decomposition tree of $H_{([a,b],c)}$ if the set of inner termini is empty, and from Observation 228 otherwise. Since the set of inherited termini $F_{([a,b],c)}$ of $H_{([a,b],c)}$ contains only children of a in $\mathtt{T}_{f_1}$ and children of b in $\mathtt{T}_{f_2}$, it follows that x or y is a child of a in $\mathtt{T}_{f_1}$, or that x or y is a child of b in $\mathtt{T}_{f_2}$ Without loss of generality, let x be a child of a in $\mathtt{T}_{f_1}$. Then $\mathtt{sd}_{f_1}(x) = j + 1$. Thus, $\mathtt{sd}(x, y) \leq j + 1$.

Since $[x, y]$ is a subbundle pair of $H_{([a,b],c)}$, we have $x, y \in V_{([a,b],c)}$. Thus, $x, y \in \mathtt{V}^{-}(a, b)$ and $a \prec_{f_1} x, y \prec_{f_1} b$. Let us assume there is a $z \in \{x, y\}$ and an $i \in [2]$ such that $z \in \mathtt{V}_{f_i}$ and $\mathtt{sd}_{f_i}(z) \leq j$. Without loss of generality let $i = 1$. As $a \prec_{f_1} z$, vertex a is a proper ancestor of z in $\mathtt{T}_{f_1}$ by Observation 164. Thus, $\mathtt{sd}_{f_1}(a) < \mathtt{sd}_{f_1}(z)$, a contradiction. It follows that $\mathtt{sd}(x, y) = j + 1$. □

Observation 248. *Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$. Each node $[x, y]$ of the decomposition forest $\mathcal{F}(a, b)$ is either a consistent or a trivial subbundle pair.*

Proof. Observation 248 can be proved similar to Observation 247. It follows from Observation 228 that each node $[x, y]$ of the decomposition forest $\mathcal{F}(a, b)$ is either a consistent or a trivial subbundle pair if $[a, b]$ is a non-minimal consistent pair of $\mathtt{H}$. □

Observation 249. *Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$. Let $[p_1, p_2] \in V_{\mathcal{F}(a,b)}$. Then there exists a descendant $[o_1, o_2] \in V_{\mathcal{F}(a,b)}$ of $[p_1, p_2]$ in $\mathcal{F}(a, b)$ such that $[o_1, o_2]$ is consistent and $\mathtt{V}(p_1, p_2) \subseteq \mathtt{V}(o_1, o_2)$.*

Proof. Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$. Let $[p_1, p_2] \in V_{\mathcal{F}(a,b)}$. Then there is a $c \in \mathtt{V}^{\text{-}}(a, b)$ such that $[p_1, p_2]$ is a node of the decomposition tree $\mathcal{T}(H_{([a,b],c)})$ of valid subgraph $H_{([a,b],c)}$. Let $H := H_{([a,b],c)}$ and $\mathcal{T} := \mathcal{T}(H)$. Let us look at the construction of the decomposition tree $\mathcal{T}$.

Let us consider the case where $|F| = 1$. Let $F = \{e\}$. Then $\mathcal{T} = (\{[e, e]\}, \emptyset)$, and $[p_1, p_2] = [e, e]$. It follows from Corollary 181, that e is a child of a and of b in the respective side trees. As $[a, b]$ is consistent, $[e, e] = [p_1, p_2]$ is also consistent.

Next let us consider the case where $|F| = 2$. Let $F = \{e, f\}$. Then $\mathcal{T} = (\{[e, f]\}, \emptyset)$. Corollary 181 implies that in the respective side trees either e is a child of a and f is a child of b or the other way around. Since $[a, b]$ is consistent, it follows that $[e, f] = [p_1, p_2]$ is consistent as well.

Now let us consider the case where $F_{\text{in}} \neq \emptyset$. Then there is a terminus $e \in F_{\text{in}}$ of H such that $[p_1, p_2]$ is a node of the decomposition tree ϑ_e of block $\mathfrak{b}_e$. Let us look at the construction of ϑ_e. By property (d) there exists a $k \geq 0$ such that $[p_1, p_2] \in \mathfrak{L}_k$. According to Lemma 231, there is a subbundle pair $[r_1, r_2] \in \mathfrak{L}_0$ such that $V(p_1, p_2) \subseteq V(r_1, r_2)$ and $[r_1, r_2]$ is a descendant of $[p_1, p_2]$ in ϑ_e. The set $\mathfrak{L}_0$ is the set of all subbundle pairs $[e, a']$ where $a' \in N_e$ and possibly $\mathfrak{L}_0$ also contains $[e, e]$.

Let us consider the case where $[r_1, r_2] \neq [e, e]$. Then $[r_1, r_2]$ is a subbundle pair $[e, a']$ such that $a' \in N_e$, $V(p_1, p_2) \subseteq V(e, a')$ and $[e, a']$ is a descendant of $[p_1, p_2]$ in ϑ_e. As $e \in F$ and $a' \in N_e$, the vertices e and a' are in different color classes of the 2-coloring $\{F_1, F_2\} = \{\mathtt{C}_{f_1}, \mathtt{C}_{f_2}\}$ of $H[F]$. It follows that in the respective side trees e is a child of a and a' is a child of b or the other way around. Hence, $[e, a']$ is consistent. According to Lemma 184 we have $\mathtt{V}(p_1, p_2) \subseteq \mathtt{V}(e, a')$. Hence, $[o_1, o_2] := [e, a']$ is a consistent pair where $[o_1, o_2] \in V_{\mathcal{F}(a,b)}$ is a descendant of $[p_1, p_2]$ in ϑ_e and $\mathtt{V}(p_1, p_2) \subseteq \mathtt{V}(o_1, o_2)$.

Next let consider the case where $[r_1, r_2] = [e, e]$. Then $[r_1, r_2] = l_e$. It follows that $[r_1, r_2]$ is the root of ϑ_e, and $[r_1, r_2] = [p_1, p_2]$. Thus, $v = e$, and there exists a subbundle pair $[e, a']$ with $a' \in N_e$ with $v \in V(e, a')$. Note that $N_e \neq \emptyset$ as $e \in F_{\text{in}}$. It follows from property (d) that $[e, a']$ is a node of ϑ_e, and therefore also a node of $\mathcal{T}$. Since $[p_1, p_2]$ is the root of ϑ_e, $[e, a']$ is a descendant of $[p_1, p_2]$. As in the case above, we can show that $[o_1, o_2] := [e, a']$ is a consistent pair where $[o_1, o_2] \in V_{\mathcal{F}(a,b)}$ is a descendant of $[p_1, p_2]$ in ϑ_e and $\mathtt{V}(p_1, p_2) \subseteq \mathtt{V}(o_1, o_2)$. □

Observation 250. *Let $[a, b]$ be a non-minimal consistent pair of $\mathtt{H}$. Let $[o_1, o_2]$ and $[p_1, p_2]$ be nodes of the decomposition forest $\mathcal{F}(a, b)$ and let $[o_1, o_2]$ be the parent of $[p_1, p_2]$. Then there is an $f \in \mathtt{F}$ such that $\mathtt{V}(o_1, o_2) \cap \mathtt{V}(p_1, p_2)$ is a subset of $\mathtt{S}_f$ and $\mathtt{V}(o_1, o_2) \cap \mathtt{V}(p_1, p_2)$ induces a path in $\mathtt{T}_f$.*

Proof. Let $[a,b]$ be a non-minimal consistent pair of $\mathtt{H}$. Let $[o_1,o_2],[p_1,p_2] \in V_{\mathcal{F}(a,b)}$, and let $[o_1,o_2]$ be the parent of $[p_1,p_2]$. Then there is a $c \in \mathtt{V}^{-}(a,b)$ such that $[o_1,o_2]$ and $[p_1,p_2]$ are nodes of the decomposition tree $\mathcal{T}(H_{([a,b],c)})$ of valid subgraph $H_{([a,b],c)}$. Let $H := H_{([a,b],c)}$ and $\mathcal{T} := \mathcal{T}(H)$. It follows from the construction of the decomposition tree $\mathcal{T}$ that there is a terminus $e \in F_{\text{in}}$ of H such that $[o_1,o_2]$ is the parent of $[p_1,p_2]$ in the decomposition tree ϑ_e of block $\mathfrak{b}_e$. Corollary 213 implies that $V(o_1,o_2) \cap V(p_1,p_2)$ is a subset of S_e and induces a path from e to some node c in T_e. According to Lemma 184, $\mathtt{V}(o_1,o_2) \cap \mathtt{V}(p_1,p_2) = V(o_1,o_2) \cap V(p_1,p_2)$. As there is an $f \in \mathtt{F}$ such that $S_e \subseteq \mathtt{S}_f$ (Observation 182) and T_e is a subtree of $\mathtt{T}_f$, we obtain that $\mathtt{V}(o_1,o_2) \cap \mathtt{V}(p_1,p_2)$ is a subset of $\mathtt{S}_f$ and $\mathtt{V}(o_1,o_2) \cap \mathtt{V}(p_1,p_2)$ induces a path in $\mathtt{T}_f$. □

Definition

Let us define the genealogical decomposition tree $\mathcal{T}_{\text{gen}} = (V_{\mathcal{T}_{\text{gen}}}, E_{\mathcal{T}_{\text{gen}}})$ of $\mathtt{H}$. The nodes of $\mathcal{T}_{\text{gen}}$ are affiliated subbundle pairs. We construct $\mathcal{T}_{\text{gen}}$ recursively. We let the root $r_{\mathcal{T}_{\text{gen}}}$ of the genealogical decomposition tree $\mathcal{T}_{\text{gen}}$ be the affiliated subbundle pair $[f_1,f_2]_{[\,]}$. We let $V_0 := \{[f_1,f_2]_{[\,]}\}$ and $E_0 := \emptyset$. Then V_{j+1} and E_{j+1} are constructed from V_j and E_j as follows: Let $\mathcal{C}_j$ be the set of affiliated consistent pairs in V_j that are not minimal. For every affiliated consistent pair $[a,b] \in \mathcal{C}_j$ we add to V_{j+1} all affiliated subbundle pairs $p_{[a,b]}$ with $p \in V_{\mathcal{F}(a,b)}$, and to E_{j+1} the edges $\{(o_{[a,b]}, p_{[a,b]}) \mid (o,p) \in E_{\mathcal{F}(a,b)}\}$. Further, for all roots $r \in R_{\mathcal{F}(a,b)}$ of decomposition forest $\mathcal{F}(a,b)$ we add the edge $\{[a,b], r_{[a,b]}\}$ to E_{j+1}.

Let $m \in \mathbb{N}$ be maximal such that $V_m \neq \emptyset$. Then

$$V_{\mathcal{T}_{\text{gen}}} := \bigcup_{i \in [m]} V_m \quad \text{and} \quad E_{\mathcal{T}_{\text{gen}}} := \bigcup_{i \in [m]} E_m.$$

Observation 251. *Let $0 \leq j \leq m$. Then $\mathtt{sd}(q) = j$ for all $q \in V_j$.*

Proof. We show Observation 251 by induction. Since $\mathtt{sd}([f_1,f_2]_{[\,]}) = 0$, we have $\mathtt{sd}(q) = 0$ for all $q \in V_0$. Let $j \geq 0$ and $\mathtt{sd}(q) = j$ for all $q \in V_j$. Let $q' \in V_{j+1}$. Then $q' = p_{[a,b]}$ with $p \in V_{\mathcal{F}(a,b)}$ for an affiliated consistent pair $[a,b] \in V_j$ that is not minimal. Since $[a,b] \in V_j$, we have $\mathtt{sd}([a,b]) = j$ by inductive assumption. Observation 247 implies that $\mathtt{sd}(p) = j+1$. Thus, $\mathtt{sd}(q') = j+1$. □

Corollary 252. *Let $0 \leq j \leq m$. Then $\mathtt{sd}(q') = j$ for all $(q,q') \in E_j$.*

Proof. Corollary 252 follows from Observation 251 and the fact that for all edges $(q,q') \in E_j$ with $0 \leq j \leq m$, we have $q' \in V_j$. □

Corollary 253. *The node $[f_1,f_2]_{[\,]}$ has no incoming edges in $\mathcal{T}_{\text{gen}}$.*

Proof. Let us assume $[f_1,f_2]_{[\,]}$ has an incoming edge $e \in E_{\mathcal{T}_{\text{gen}}}$. It follows that there is a $j \geq 0$ such that $e \in E_j$. As the end of the directed edge e is $[f_1,f_2]_{[\,]}$, j must be 0 according to Corollary 252. We obtain a contradiction because $E_0 = \emptyset$. □

Observation 254. *Let $p_{p'}$ be a node of the genealogical decomposition tree $\mathcal{T}_{\text{gen}}$ with $p' \neq [\,]$. Then the affiliation p' is a consistent pair that is not minimal, and the affiliated subbundle pair p' is a node of $\mathcal{T}_{\text{gen}}$.*

Proof. Let $p_{p'}$ be a node of the genealogical decomposition tree $\mathcal{T}_{\text{gen}}$ with $p' \neq [\,]$. Then there is a $j > 0$ such that $p_{p'} \in V_j$. The construction of V_j implies that p' is a consistent pair, and that the affiliated consistent pair p' is in C_{j-1} and therefore in V_{j-1}. Hence, the affiliated subbundle pair p' is a node of $\mathcal{T}_{\text{gen}}$. □

Observation 255. *Let $p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair with $p' \neq [\,]$. Then p is a node of the decomposition forest $\mathcal{F}(p')$. Further, $o_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ for each node o of the decomposition forest $\mathcal{F}(p')$.*

Observations 254, 255 and 248 imply the following corollary.

Corollary 256. *Let $p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair. Then p is a consistent pair or a trivial subbundle pair.*

As minimal consistent pairs are trivial subbundle pairs (Observation 241) and non-minimal consistent pairs are non-trivial subbundle pairs (Observation 242), we obtain the following corollary.

Corollary 257. *Let $p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair. Then p is either a non-minimal consistent pair or a trivial subbundle pair.*

Observation 258. *For all $(o_{o'}, p_{p'}) \in E_{\mathcal{T}_{\text{gen}}}$, we have $p' \neq [\,]$ and either*

- *$p' = o'$, (o, p) is an edge of the decomposition forest $\mathcal{F}(p')$, and $\texttt{sd}(o) = \texttt{sd}(p)$; or*
- *$p' = o$, p is a root of the decomposition forest $\mathcal{F}(p')$, and $\texttt{sd}(o) = \texttt{sd}(p) - 1$.*

Proof. Let $(o_{o'}, p_{p'}) \in E_{\mathcal{T}_{\text{gen}}}$. The only node $[f_1, f_2]_{[\,]}$ with affiliation $[\,]$ has no incoming edges (Corollary 253). Thus, $p' \neq [\,]$. Clearly, we have either $p' = o'$ and $(o, p) \in E_{(\mathcal{F}(p')}$, or $p' = o$ and $p \in R_{\mathcal{F}(a,b)}$. Since $p' \neq [\,]$, p' is a non-minimal consistent pair (Observation 254). Then it follows from Observation 247 that $\texttt{sd}(o) = \texttt{sd}(p)$ in the first case. Since $p' \neq [\,]$ is an affiliation of p, we have $\texttt{sd}(p') = \texttt{sd}(p) - 1$, and therefore, $\texttt{sd}(o) = \texttt{sd}(p) - 1$, in the second case. □

Corollary 259. *For all $(q, q') \in E_{\mathcal{T}_{\text{gen}}}$ we have $\texttt{sd}(q) \leq \texttt{sd}(q')$.*

Observation 260. *Let $o_{o'}, p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$. Then $(o_{o'}, p_{p'}) \in E_{\mathcal{T}_{\text{gen}}}$ if $p' \neq [\,]$ and either*

- *$p' = o'$ and (o, p) is an edge of the decomposition forest $\mathcal{F}(p')$, or*
- *$p' = o$ and p is a root of the decomposition forest $\mathcal{F}(p')$.*

Proof. Let $o_{o'}, p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ and $p' \neq [\,]$. Let $p_{p'} \in V_j$ for $j \geq 0$. Since $p' \neq [\,]$, we have $j > 0$. Let us consider the case where $p' = o'$ and $(o, p) \in E_{\mathcal{F}(p')}$. As $j > 0$, it follows from $p_{p'} \in V_j$ that the set $\{(s_{p'}, t_{p'}) \mid (s, t) \in E_{\mathcal{F}(p')}\}$ of edges is a subset of E_j. Thus, $(o_{p'}, p_{p'}) \in E_j$, and therefore, $(o_{o'}, p_{p'}) \in E_{\mathcal{T}_{\text{gen}}}$. Now let us consider the case where $p' = o$ and $p \in R_{\mathcal{F}(p')}$. Since $j > 0$, $p_{p'} \in V_j$ yields that $\{(p', r_{p'}) \mid r \in R_{\mathcal{F}(p')}\}$ is a subset of E_j. Hence, $(p', p_{p'}) \in E_j$, and consequently, $(o_{o'}, p_{p'}) \in E_{\mathcal{T}_{\text{gen}}}$. □

Lemma 261. *The genealogical decomposition tree $\mathcal{T}_{\text{gen}}$ is a directed tree.*

Proof. Clearly, $\mathcal{T}_{\text{gen}}$ is connected. We first show that $\mathcal{T}_{\text{gen}}$ is an acyclic directed graph. For a contradiction let us assume there exists a directed cycle $q_1, \dots, q_m, q_1$ with $m \geq 1$ in $\mathcal{T}_{\text{gen}}$. Thus, $(q_i, q_{i+1}) \in E_{\mathcal{T}_{\text{gen}}}$ for $i \in [m-1]$ and $(q_m, q_1) \in E_{\mathcal{T}_{\text{gen}}}$. For all $(q, q') \in E_{\mathcal{T}_{\text{gen}}}$ we have $\mathtt{sd}(q) \geq \mathtt{sd}(q')$ according to Corollary 259. Therefore, there exists a $d \geq 0$ such that $d = \mathtt{sd}(q_i)$ for all $i \in [m]$. Let p_i be the underlying subbundle pair of q_i for $i \in [m]$. It follows from Observation 258 that $(p_1, \dots, p_m)$ is a directed cycle in $\mathcal{F}(a, b)$, a contradiction.

Now, let us suppose there exist affiliated subbundle pairs $q, q_1, q_2 \in V_{\mathcal{T}_{\text{gen}}}$ such that $(q_1, q), (q_2, q)$ are edges in $E_{\mathcal{T}_{\text{gen}}}$ and $q_1 \neq q_2$. Let $q = p_{p'}$ and $q_i = p_{i p'_i}$ for $i \in [2]$. By Observation 258 we have $p' \neq [\,]$. Now, we cannot have the case that $p'_i = p'$ for all $i \in [2]$ as this yields the existence of a subbundle pair with more than one incoming edge in the decomposition forest $\mathcal{F}(p')$ by Observation 258, a contradiction. Next, let us assume $p'_1 = p'$ and $p'_2 \neq p'$. (The case $p'_1 \neq p'$ and $p'_2 = p'$ can dealt with analogously.) Then $p_2 = p'$ and p is a root of the decomposition forest $\mathcal{F}(p')$ (Observation 258). At the same time, Observation 258 implies that (p_1, p) is an incoming edge of p in $\mathcal{F}(p')$, a contradiction. It remains to consider the case where $p'_1 \neq p'$ and $p'_2 \neq p'$. Then $p_1 = p'$ and $p_2 = p'$ by Observation 258. As $p' \neq [\,]$, p' is a consistent pair. By Observation 243 there exists only one affiliation for a consistent pair. Thus, $p'_1 = p'_2$, again a contradiction. $\square$

Defining the Genealogical Decomposition Tree in FP+C

In the following we show that the genealogical decomposition tree $\mathcal{T}_{\text{gen}}$ of the underlying graph $\mathtt{H}$ of bundle extension $\mathtt{H}^*$ is definable in fixed-point logic with counting.

Let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be a bundle extension and let $\mathtt{H}$ be its underlying graph. For a pair $\bar{p} = (p_1, p_2) \in \mathtt{U}^2$ of elements of $\mathtt{H}^*$, we let $\tilde{p}$ denote the multiset $[p_1, p_2]$. Let $\bar{y}, \bar{y}', \bar{z}, \bar{z}'$ be pairs of variables.

For each extended valid subgraph H' with underlying graph H, there are FP+C-formulas $\psi_{V_\mathcal{T}}(\bar{y})$ and $\psi_{E_\mathcal{T}}(\bar{y}, \bar{z})$ that define the vertex set $V_{\mathcal{T}(H)}$ and the edge set $E_{\mathcal{T}(H)}$ of the decomposition tree of H (see (6.19)) We pull back these formulas under the parameterized transduction $\Theta_{\text{val}}(x_1, x_2, x_3)$ presented in (6.14) that maps each bundle extension $\mathtt{H}^*$ and every valid triple $([a, b], c)$ of the underlying graph $\mathtt{H}$ of $\mathtt{H}^*$ to the extended valid subgraph $H'_{([a,b],c)}$ of $\mathtt{H}$ defined by $([a, b], c)$. We obtain FP+C-formulas

$$\begin{aligned} \varphi_{V_\mathcal{T}} &:= \psi_{V_\mathcal{T}}^{-\Theta_{\text{val}}}(x_1, x_2, x_3, \bar{y}) \quad \text{and} \\ \varphi_{E_\mathcal{T}} &:= \psi_{E_\mathcal{T}}^{-\Theta_{\text{val}}}(x_1, x_2, x_3, \bar{y}, \bar{z}) \end{aligned}$$

such that for all elements $a, b, c \in \mathtt{U}(\mathtt{H}^*)$ and all pairs $\bar{o}, \bar{p} \in \mathtt{U}(\mathtt{H}^*)^2$ of elements of each bundle extension $\mathtt{H}^* \in \mathtt{C}^*_{\text{ChCo}}$ we have

$\mathtt{H}^* \models \varphi_{V_\mathcal{T}}[a, b, c, \bar{p}] \iff$ $([a, b], c)$ is valid triple of $\mathtt{H}$ and $\tilde{p}$ is a node of the decomposition tree of the valid subgraph $H_{([a,b],c)}$ of $\mathtt{H}$, and

$\mathtt{H}^* \models \varphi_{E_\mathcal{T}}[a, b, c, \bar{o}, \bar{p}] \iff$ $([a, b], c)$ is valid triple of $\mathtt{H}$ and $(\tilde{o}, \tilde{p})$ is an edge of the decomposition tree of the valid subgraph $H_{([a,b],c)}$ of $\mathtt{H}$.

We can use $\varphi_{V_\mathcal{T}}$ and $\varphi_{E_\mathcal{T}}$ to obtain FP+C-formulas that define the vertices and edges of the decomposition forests we let

$$\begin{aligned}\varphi_{V_\mathcal{F}}(x_1,x_2,\bar{y}) &:= \exists x\, \varphi_{V_\mathcal{T}}(x_1,x_2,x,\bar{y}) \quad \text{and}\\ \varphi_{E_\mathcal{F}}(x_1,x_2,\bar{y},\bar{z}) &:= \exists x\, \varphi_{V_\mathcal{T}}(x_1,x_2,x,\bar{y},\bar{z}).\end{aligned}$$

Further, we can use these formulas to construct an FP+C-formula $\varphi_{R_\mathcal{F}}(x_1,x_2,\bar{y})$ that defines the roots of the decomposition forest. Then for all elements $a,b \in \mathtt{U}(\mathtt{H}^*)$ and all pairs $\bar{o},\bar{p} \in \mathtt{U}(\mathtt{H}^*)^2$ of every bundle extension $\mathtt{H}^* \in \mathtt{C}^*_{\mathrm{ChCo}}$ we have

$$\begin{aligned}\mathtt{H}^* \models \varphi_{V_\mathcal{F}}[a,b,\bar{p}] &\iff [a,b] \in \mathtt{P}_{\mathrm{sub}} \text{ is non-trivial and } \bar{o} \text{ is a node of the decomposition forest } \mathcal{F}(a,b),\\ \mathtt{H}^* \models \varphi_{E_\mathcal{F}}[a,b,\bar{o},\bar{p}] &\iff [a,b] \in \mathtt{P}_{\mathrm{sub}} \text{ is non-trivial and } (\bar{o},\bar{p}) \text{ is an edge of the decomposition forest } \mathcal{F}(a,b), \text{ and}\\ \mathtt{H}^* \models \varphi_{R_\mathcal{F}}[a,b,\bar{p}] &\iff [a,b] \in \mathtt{P}_{\mathrm{sub}} \text{ is non-trivial and } \bar{p} \text{ is a root of the decomposition forest } \mathcal{F}(a,b).\end{aligned}$$

Now we define FP+C-formulas $\varphi_{V_{\mathcal{T}_{\mathrm{gen}}}}$ and $\varphi_{E_{\mathcal{T}_{\mathrm{gen}}}}$, respectively, for the vertex set and the edge set of the genealogical decomposition tree of $\mathtt{H}$. We define $\varphi_{V_{\mathcal{T}_{\mathrm{gen}}}}(\bar{y},\bar{y}')$ and $\varphi_{E_{\mathcal{T}_{\mathrm{gen}}}}(\bar{y},\bar{y}',\bar{z},\bar{z}')$ such that for all pairs $\bar{o},\bar{o}',\bar{p},\bar{p}' \in \mathtt{U}(\mathtt{H}^*)^2$ of elements of every bundle extension $\mathtt{H}^* \in \mathtt{C}^*_{\mathrm{ChCo}}$ we have

$$\begin{aligned}\mathtt{H}^* \models \varphi_{V_{\mathcal{T}_{\mathrm{gen}}}}[\bar{p},\bar{p}'] &\iff \bar{p}_{\bar{p}'} \in V_{\mathcal{T}_{\mathrm{gen}}}, \quad \text{and}\\ \mathtt{H}^* \models \varphi_{E_{\mathcal{T}_{\mathrm{gen}}}}[\bar{o},\bar{o}',\bar{p},\bar{p}'] &\iff (\bar{o}_{\bar{o}'},\bar{p}_{\bar{p}'}) \in E_{\mathcal{T}_{\mathrm{gen}}}.\end{aligned}$$

In order to be able to define all affiliated subbundle pairs, we encode the empty multiset $[\,]$ by $[v,v]$ for elements $v \in \mathtt{U}$. Note that for all nodes $p_{p'}$ of the genealogical decomposition tree $\mathcal{T}_{\mathrm{gen}}$ with $p' \neq [\,]$ the affiliation p' is a consistent pair that is not minimal (Observation 254), and for non-minimal consistent pairs $[a,b]$ we clearly we have $a \neq b$.

Let $\bar{y} = (y_1,y_2)$ and $\bar{y}' = (y_1',y_2')$. Further, let $\varphi_\mathtt{F}$ and $\varphi_{\mathtt{P}^{\mathrm{nm}}_{\mathrm{con}}}$ be the formulas from (6.10) and (6.20), respectively. Then we let

$$\varphi_{V_{\mathcal{T}_{\mathrm{gen}}}}(\bar{y},\bar{y}') \;:=\; \mathrm{ifp}\left(X(\bar{y},\bar{y}') \leftarrow\; \psi_0 \vee \psi_{V_{\mathcal{T}_{\mathrm{gen}}}}\right)(\bar{y},\bar{y}')$$

where

$$\begin{aligned}\psi_0(\bar{y},\bar{y}') &:= \varphi_\mathtt{F}(y_1) \wedge \varphi_\mathtt{F}(y_2) \wedge y_1 \neq y_2 \wedge y_1' = y_2'\\ \psi_{V_{\mathcal{T}_{\mathrm{gen}}}}(\bar{y},\bar{y}',X) &:= \exists \bar{z}\, X(\bar{y}',\bar{z}) \wedge \varphi_{\mathtt{P}^{\mathrm{nm}}_{\mathrm{con}}}(\bar{y}') \wedge \varphi_{V_\mathcal{F}}(\bar{y}',\bar{y})\end{aligned}$$

For $\bar{p},\bar{p}' \in \mathtt{U}(\mathtt{H}^*)^2$ of bundle extension $\mathtt{H}^* \in \mathtt{C}^*_{\mathrm{ChCo}}$, let X^j be the relation defined in round j of the recursion within the fixed-point operator of $\varphi_{V_{\mathcal{T}_{\mathrm{gen}}}}$. We show that for $j \geq 0$, we have $(\bar{s},\bar{s}') \in X^j$ if and only if $\bar{s}_{\bar{s}'} \in \bigcup_{i<j} V_i$. As $X^0 = \emptyset$, this is clearly true for $j = 0$. It is not hard to see that $\psi_0(\bar{y},\bar{y}')$ is satisfied for $\bar{s},\bar{s}' \in \mathtt{U}(\mathtt{H}^*)^2$ of each bundle extension $\mathtt{H}^*$ if and only if $\bar{s} = [f_1,f_2]$ and $\bar{s}' = [v,v]$ where $v \in \mathtt{U}$. Hence, for $j = 1$ we obtain $(\bar{s},\bar{s}') \in X^1$ if and only if $\bar{s}_{\bar{s}'} \in V_0$. Now let $j \geq 1$ and suppose we have $(\bar{s},\bar{s}') \in X^j$ if and only if

$\bar{s}_{\bar{s}'} \in \bigcup_{i<j} V_i$. The relation X^{j+1} contains all $(\bar{s}, \bar{s}') \in X^j$, and for all $(\bar{s}', \bar{t}) \in X^j$ where $\bar{s}'$ is a non-minimal consistent pair and $\bar{s}$ is a node of the decomposition forest $\mathcal{F}(\bar{s}')$ the tuple $(\bar{s}, \bar{s}')$ is in X^{j+1}. Hence, $(\bar{s}, \bar{s}') \in X^{j+1}$ if and only if $\bar{s}_{\bar{s}'} \in \bigcup_{i<j+1} V_i$. It follows that $\varphi_{V_{\mathcal{T}_{\text{gen}}}}$ is satisfied for $\bar{p}, \bar{p}' \in \mathtt{U}(\mathtt{H}^*)^2$ of bundle extension $\mathtt{H}^*$ if and only if $\bar{p}_{\bar{p}'} \in V_{\mathcal{T}_{\text{gen}}}$.

Let $\bar{z}' = (z_1', z_2')$. We let

$$\begin{aligned}\varphi_{E_{\mathcal{T}_{\text{gen}}}}(\bar{y}, \bar{y}', \bar{z}, \bar{z}') := \; & \varphi_{V_{\mathcal{T}_{\text{gen}}}}(\bar{y}, \bar{y}') \wedge \varphi_{V_{\mathcal{T}_{\text{gen}}}}(\bar{z}, \bar{z}') \wedge z_1' \neq z_2' \wedge \\ & \Big((\text{“}\bar{y}' = \bar{z}'\text{”} \wedge \varphi_{E_{\mathcal{F}}}(\bar{z}', \bar{y}, \bar{z})) \vee (\text{“}\bar{y} = \bar{z}'\text{”} \wedge \varphi_{R_{\mathcal{F}}}(\bar{z}', \bar{z})) \Big).\end{aligned}$$

It follows from Observations 258 and 260 that $\varphi_{E_{\mathcal{T}_{\text{gen}}}}$ is satisfied for $\bar{o}, \bar{o}', \bar{p}, \bar{p}' \in \mathtt{U}(\mathtt{H}^*)^2$ of bundle extension $\mathtt{H}^*$ if and only if $(\bar{o}_{\bar{o}'}, \bar{p}_{\bar{p}'}) \in E_{\mathcal{T}_{\text{gen}}}$.

6.6. Canonization

Let $G^* \in \mathcal{C}^*_{\text{ChCo}}$ be an LO-colored graph and let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be the bundle extension of the O-extension of G^*. Further let $\mathtt{H} = (\mathtt{V}, \mathtt{E})$ be the underlying graph of $\mathtt{H}^*$ and $\mathcal{T}_{\text{gen}}$ be the genealogical decomposition tree of $\mathtt{H}$. In this section, we show that we can use $\mathtt{H}^*$ and $\mathcal{T}_{\text{gen}}$ to define a canon of the LO-colored graph $G^* \in \mathcal{C}^*_{\text{ChCo}}$.

For every affiliated subbundle pair $q \in V_{\mathcal{T}_{\text{gen}}}$ we recursively define a subset $\mathcal{W}(q)$ of vertices of $\mathtt{H}$. For each $q \in V_{\mathcal{T}_{\text{gen}}}$, we show that there exists a set $R(q)$ of nodes of $\mathcal{T}_{\text{gen}}$ such that $\mathcal{W}(q)$ can be decomposed into $\{a, b\}$ or $\mathtt{V}(a, b)$, and the sets $\mathcal{W}(r)$ for $r \in R(q)$. For distinct $r_1, r_2 \in R(q)$ the intersection of $\mathcal{W}(r_1)$ and $\mathcal{W}(r_2)$ is contained in a subset of $\mathtt{V}$ which can easily be defined by q, actually this subset is contained in a max clique of $\mathtt{H}$.

Later we extend the set $\mathcal{W}(q)$ to the set $\mathcal{W}_{\mathtt{anc}}(q)$ for all $q \in V_{\mathcal{T}_{\text{gen}}}$. We define the extended height of each node $q \in V_{\mathcal{T}_{\text{gen}}}$ in $\mathcal{T}_{\text{gen}}$. For all $r \in R(q)$ the extended height of r turns out to be less than the extended height of q. Based on the extended height of q in $\mathcal{T}_{\text{gen}}$, we recursively define an isomorphic copy on the number sort of the subgraph $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$ induced by $\mathcal{W}_{\mathtt{anc}}(q)$ for each $q \in V_{\mathcal{T}_{\text{gen}}}$. Note that we also maintain the side tree relation for this isomorphic copy and use various colorings of the vertices.

6.6.1. Preliminaries

Let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be a bundle extension with underlying graph $\mathtt{H} = (\mathtt{V}, \mathtt{E})$. We define the *ancestors* $\mathtt{anc}(v)$ and *descendants* $\mathtt{dec}(v)$ of vertex $v \in \mathtt{V}$ in $\mathtt{H}$ as follows we let

$$\mathtt{anc}(v) := \bigcup_{f \in \mathtt{F}(v)} \mathtt{anc}_f(v) \quad \text{and} \quad \mathtt{dec}(v) := \bigcup_{f \in \mathtt{F}(v)} \mathtt{dec}_f(v),$$

where, $\mathtt{F}(v) := \{f \in \mathtt{F} \mid v \in \mathtt{V}_f\}$ for $v \in \mathtt{V}$. This means for $v \in \mathtt{S}_f$ with $f \in \mathtt{F}$, the sets $\mathtt{anc}(v)$ and $\mathtt{dec}(v)$ are the ancestors $\mathtt{anc}_f(v)$ of v in $\mathtt{T}_f$ and the descendants $\mathtt{dec}_f(v)$ of v in $\mathtt{T}_f$, respectively. If $v \in \mathtt{O}$, then $\mathtt{dec}(v) = \{v\}$ and $\mathtt{anc}(v)$ is the set of all ancestors of v in $\mathtt{T}_{f_1}$ and of all ancestors of v in $\mathtt{T}_{f_2}$, that is, $\mathtt{anc}(v)$ is the max clique $M_v \in \mathcal{M}_{\mathtt{H}}$ spanned by v (Lemmas 161 and 168). Let $v \in \mathtt{V}$. The *side depth* $\mathtt{sd}(v)$ of v is the minimum of $\mathtt{sd}_f(v)$ for all $f \in \mathtt{F}(v)$.

Observation 262. *Let $v \in \mathtt{V} \setminus \mathtt{O}$ and $\mathtt{sd}(v) = j$. Then for each $i \in \{0, \dots, j\}$ there exists a unique ancestor w of v with $\mathtt{sd}(w) = i$.*

Observation 263. *Let $[a, b]$ be a consistent pair of side depth d. Let $v \in \mathtt{V}(a, b)$. Then $w \in \mathtt{V}(a, b)$ for each ancestor w of v with $\mathtt{sd}(w) \geq d$.*

Observation 264. *Let $[a, b]$ be a trivial subbundle pair of side depth d. Let $v \in \mathtt{V}(a, b)$. Then $w \in \mathtt{V}(a, b)$ for each ancestor w of v with $\mathtt{sd}(w) \geq d$.*

Corollary 265. *Let $[a, b]$ be a consistent pair of side depth d. Let $v \in \mathtt{V}(a, b)$. If w is an ancestor of v with $\mathtt{sd}(w) = d$, then $w \in \{a, b\}$.*

Let q be an affiliated subbundle pair in $V_{\mathcal{T}_{\text{gen}}}$ of side depth $d \geq 0$. Let $[a, b]$ be the underlying subbundle pair of q. We define the *side ancestors* $a_d, b_d, a_{d-1}, b_{d-1}, \dots, a_0, b_0$ of q as follows: We let $a_d := a$ and $b_d := b$. If $d = 0$, then all side ancestors of q are defined. Let $d > 0$. Then the affiliation of q is a consistent pair $[a', b']$ of side depth $d - 1$. Observation 254 implies that $[a', b']$ is not minimal. Thus, $a', b' \notin \mathtt{O}$, and it follows that there is only one ancestor of a' of depth i and one ancestor of b' of depth i for each $i \in \{0, \dots, d - 1\}$ (Observation 262). Recall that for an affiliated subbundle pair $[a, b]_{a',b'}$ we assume vertex a' is the parent of a, or vertex b' is the parent of b in T. Now, for every $i \in \{0, \dots, d - 1\}$, we let a_i be the ancestor of a' of depth i, and b_i be the ancestor of b' of depth i in T. We denote the set $\{a_d, b_d, a_{d-1}, b_{d-1}, \dots, a_0, b_0\}$ of side ancestors of the affiliated subbundle pair q by $\mathtt{anc}(q)$. We let $a_{d-1}, b_{d-1}, \dots, a_0, b_0$ be the *proper side ancestors* of q.

Note that for all affiliated subbundle pairs q of side depth d with side ancestors $a_d, b_d, \dots, a_0, b_0$, we have $\{a_0, b_0\} = \{f_1, f_2\}$ and $a_d = a$ and $b_d = b$. If $d > 0$, then q is the affiliated subbundle pair $[a_d, b_d]_{a_{d-1}, b_{d-1}}$. By Observation 243, $[a_{i-1}, b_{i-1}]$ is the affiliation of $[a_i, b_i]$ for all $i \in [d]$ and $[\,]$ is the affiliation of $[a_0, b_0]$. It follows that $[a_{i-1}, b_{i-1}]$ is a non-minimal consistent pair of $\mathtt{H}$ of side depth $i - 1$ for all $i \in [d]$. Observation 254 implies that $[a_0, b_0]_{[\,]}$ and $[a_i, b_i]_{a_{i-1}, b_{i-1}}$ for all $i \in \{1, \dots, d\}$ are nodes of $\mathcal{T}_{\text{gen}}$.

Observation 266. *Let $q, r \in V_{\mathcal{T}_{\text{gen}}}$. Let d and d' be the side depth of q and r, respectively. Further, let a_i, b_i for $i \in \{0, \dots, d\}$ be the side ancestors of q and a'_i, b'_i for $i \in \{0, \dots, d'\}$ be the side ancestors of r. Let $0 \leq j \leq \min\{d, d'\}$. If $a_j = a'_j$ and $b_j = b'_j$, then $a'_i = a_i$ and $b'_i = b_i$ for all $i \in \{0, \dots, j\}$.*

Observation 267. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Then $\mathtt{V}(a_j, b_j) \subseteq \mathtt{V}(a_{j'}, b_{j'})$ for all $j, j' \in \{0, \dots, d\}$ with $j \geq j'$.*

Next we define the decomposition forests $\mathcal{F}_i(q)$ for $i \in \{0, \dots, d\}$ of q. Since $[a_{i-1}, b_{i-1}]$ is a non-minimal consistent pair for $i \in [d]$, the decomposition forest $\mathcal{F}(a_{i-1}, b_{i-1})$ is defined. We let $\mathcal{F}_i(q) := \mathcal{F}(a_{i-1}, b_{i-1})$ for $i \in [d]$. Further, we let $\mathcal{F}_0(q)$ be the directed tree $([f_1, f_2], \emptyset)$ and if $[a, b]$ is consistent and not minimal we let $\mathcal{F}_{d+1}(q) := \mathcal{F}(a, b)$. According to Observation 247, the nodes in $\mathcal{F}_i(q)$ are subbundle pairs of side depth i for each i. Further, $[a_i, b_i]$ is a node of $\mathcal{F}_i(q)$ (Observation 255). For all nodes $[u, v]$ of $\mathcal{F}_i(q)$ we denote by $\mathtt{W}^q_i(u, v)$ the union of all sets $\mathtt{V}(x, y)$ where $[x, y]$ is a descendent of $[u, v]$ in $\mathcal{F}_i(q)$. We let $\mathrm{C}_i(q)$ be the set of all children of $[a_i, b_i]$ in $\mathcal{F}_i(q)$ for $i \in \{0, \dots, d\}$. If $[a, b]$ is consistent and not minimal, we let $\mathrm{C}_{d+1}(q)$ be the set of roots of $\mathcal{F}_{d+1}(q)$, otherwise we let $\mathrm{C}_{d+1}(q)$ be empty. Clearly, $\mathrm{C}_0(q) = \emptyset$.

Observation 268. *Let $q, r \in V_{\mathcal{T}_{\text{gen}}}$. Let d and d' be the side depth of q and r, respectively. Further, let a_i, b_i for $i \in \{0, \dots, d\}$ be the side ancestors of q and a'_i, b'_i for $i \in \{0, \dots, d'\}$ be the side ancestors of r. Let $0 \le j < \min\{d, d'\}$. If $a_j = a'_j$ and $b_j = b'_j$, then $\mathcal{F}_i(q) = \mathcal{F}_i(r)$ for all $i \le j + 1$.*

Observation 269. *Let $q, r \in V_{\mathcal{T}_{\text{gen}}}$. Let d and d' be the side depth of q and r, respectively. Further, let a_i, b_i for $i \in \{0, \dots, d\}$ be the side ancestors of q and a'_i, b'_i for $i \in \{0, \dots, d'\}$ be the side ancestors of r. Let $0 \le j \le \min\{d, d'\}$. If $a_j = a'_j$ and $b_j = b'_j$, then $\mathrm{C}_i(q) = \mathrm{C}_i(r)$ for all $i \le j$.*

Let $[u, v]$ and $[u', v']$ be adjacent nodes in $\mathcal{F}_i(q)$. According to Observation 250 there is an $f \in \mathtt{F}$ such that $\mathtt{V}(u, v) \cap \mathtt{V}(u', v')$ is a subset of $\mathtt{S}_f$ and induces a path in $\mathtt{T}_f$. Let $z([u, v], [u', v'])$ be the vertex from $\mathtt{V}(u, v) \cap \mathtt{V}(u', v')$ that has maximal depth in $\mathtt{T}_f$.

Let $J(q)$ be the set of all indices $j \in [0, d]$ such that subbundle pair $[a_j, b_j]$ has a parent node $[a^*_j, b^*_j]$ in $\mathcal{F}_j(q)$. Notice that for $j = 0$ node $[a, b] = [a_0, b_0]$ is the only node in $\mathcal{F}_0(q)$. Thus, $0 \notin J(q)$. For $j \in J(q)$, let $p_j(q) := z([a_j, b_j], [a^*_j, b^*_j])$. Let $d > 0$. Then for all $i \in [0, d]$ we let $P_i(q) := \{a_{d-1}, b_{d-1}\} \cup \{p_j(q) \mid j \in J(q), j \ge i\}$. For $d = 0$, node $[a, b] = [a_0, b_0]$ has the affiliation $[\,]$ and $J(q) = \emptyset$. We let $P_0(q) := \emptyset$ in this case.

Observation 270. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$. Then $P_i(q) \cap \mathtt{O} = \emptyset$ for all $i \in [0, d]$.*

Observation 271. *Let $q, r \in V_{\mathcal{T}_{\text{gen}}}$. Let d and d' be the side depth of q and r, respectively. Further, let a_i, b_i for $i \in \{0, \dots, d\}$ be the side ancestors of q and a'_i, b'_i for $i \in \{0, \dots, d'\}$ be the side ancestors of r. Let $0 \le j \le \min\{d, d'\}$. Let $a_j = a'_j$ and $b_j = b'_j$. Then for all $i \le j$, we have $i \in J(r) \iff i \in J(q)$, and $p_i(q) = p_i(r)$ if $i \in J(q)$.*

6.6.2. The Set $\mathcal{W}(q)$

It the following let q always be an affiliated subbundle pair in $V_{\mathcal{T}_{\text{gen}}}$ of side depth $d \ge 0$ with underlying subbundle pair $[a, b]$ and side ancestors a_i, b_i for $i \in \{0, \dots, d\}$.

We define a set $\mathcal{W}(q)$ of vertices of $\mathtt{H}$ recursively. In order to do this we define the set $\mathcal{W}_i(q)$ for all $i \in [d]$ and the set $\mathrm{C}^P_i(q)$, which is a subset of $\mathrm{C}_i(q)$, for all $i \in [d+1]$. We let $\mathrm{C}^P_{d+1}(q) := \mathrm{C}_{d+1}(q)$ and $\mathrm{C}^P_d(q) := \mathrm{C}_d(q)$. Further, we let $\mathcal{W}_d(q) := \mathtt{W}^q_d(a, b)$ where $[a, b] = [a_d, b_d]$ for the underlying subbundle pair of q. Now let $i < d$. We let $\mathrm{C}^P_i(q)$ be the subset of $\mathrm{C}_i(q)$ that contains all subbundle pairs $[u, v] \in \mathrm{C}_i(q)$ where the vertex $z([u, v], [a_i, b_i])$ is contained in $\mathcal{W}_{i+1}(q)$ and is not an ancestor of any vertex in $P_i(q)$. For each $[u, v] \in \mathrm{C}^P_i(q)$ we add all vertices in $\mathtt{W}^q_i(u, v)$ to $\mathcal{W}_i(q)$. Finally, we let $\mathcal{W}(q) := \bigcup_{i \le d} \mathcal{W}_i(q)$. Then

$$\mathcal{W}(q) = \mathtt{W}^q_d(a, b) \cup \bigcup_{i<d} \bigcup_{[u,v] \in \mathrm{C}^P_i(q)} \mathtt{W}^q_i(u, v).$$

Observation 272. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair with underlying subbundle pair $[a, b]$. Then $\mathtt{V}(a, b) \subseteq \mathcal{W}(q)$.*

Proof. Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair with underlying subbundle pair $[a, b]$ of side depth d. We have $\mathtt{V}(a, b) \subseteq \mathtt{W}^q_d(a, b)$ and $\mathtt{W}^q_d(a, b) = \mathcal{W}_d(q)$. Thus, $\mathtt{V}(a, b) \subseteq \mathcal{W}(q)$. □

Observation 273. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. Let $j \in [d+1]$ and $[u,v] \in \mathrm{C}_j^P(q)$. Then $[u,v]_{a_{j-1},b_{j-1}} \in V_{\mathcal{T}_{\text{gen}}}$.*

Proof. Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. Let $j \in [d]$ and $[u,v] \in \mathrm{C}_j^P(q)$. It follows inductively from Observation 254 and from Observation 255 that $[u,v]_{[a_{j-1},b_{j-1}]} \in V_{\mathcal{T}_{\text{gen}}}$. Let $j = d+1$ and $[u,v] \in \mathrm{C}_j^P(q)$. Then $[a_d, b_d]$ is consistent and not minimal. It follows from the construction of $\mathcal{T}_{\text{gen}}$. That $[u,v]_{a_d,b_d} \in V_{\mathcal{T}_{\text{gen}}}$. □

Observation 274. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. Let $j \in [d+1]$ and $[u,v] \in \mathrm{C}_j^P(q)$. Let $r := [u,v]_{a_{j-1},b_{j-1}} \in V_{\mathcal{T}_{\text{gen}}}$. Then $\mathtt{W}_j^q(u,v) = \mathcal{W}_j(r)$.*

Proof. Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. Let $j \in [d+1]$ and $[u,v] \in \mathrm{C}_j^P(q)$. Let $r := [u,v]_{a_{j-1},b_{j-1}}$. Then $r \in V_{\mathcal{T}_{\text{gen}}}$ (Observation 273). According to Observation 268 we have $\mathcal{F}_j(q) = \mathcal{F}_j(r)$. Thus, $\mathtt{W}_j^q(u,v) = \mathtt{W}_j^r(u,v)$. As $\mathcal{W}_j(r) = \mathtt{W}_j^r(u,v)$, it follows that $\mathtt{W}_j^q(u,v) = \mathcal{W}_j(r)$. □

Observation 275. *Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be minimal consistent. If $a = b$, then $\mathcal{W}(q) = \{a\}$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be minimal consistent, and $a = b$. Then $a \in \mathtt{O}$. Let d be the side depth of q and a_i, b_i for $i \in \{0, \ldots, d\}$ be the side ancestors of q. Let us show that $\mathtt{W}_d^q([a,b]) = \{a\}$. As $a = b$, it follows that $[a,b] \neq [f_1, f_2]$. Thus, $p' = [a_{d-1}, b_{d-1}]$. As $[a,b]$ is consistent, it follows from Observations 243 and 165 that $\{a\}$ is a connected component of $\mathtt{H}^{\text{-}(a_{d-1},b_{d-1})}$. Hence, the decomposition tree of $H_{([a_{d-1},b_{d-1}],a)}$ is $(\{[a,b]\}, \emptyset)$. Thus, $\mathtt{W}_d^q([a,b]) = \{a\}$. Therefore, $\mathcal{W}_d(q) = \{a\}$.

Let us show that $\mathrm{C}_{d-1}^P(q) = \emptyset$. Assume $[u,v] \in \mathrm{C}_{d-1}^P(q)$. Then $z := z([u,v],[a_{d-1},b_{d-1}]) \in \mathcal{W}_d(q)$. Hence, $z \in \{a\}$. It follows from Observation 250 that $z \notin \mathtt{O}$, a contradiction. Hence, $\mathrm{C}_{d-1}^P(q) = \emptyset$. Therefore, $\mathcal{W}_{d-1}(q) = \emptyset$. Inductively, it follows that $\mathrm{C}_j^P(q) = \emptyset$ and $\mathcal{W}_j(q) = \emptyset$ for all $j \in [0, d-1]$. Consequently, $\mathcal{W}(q) = \{a\}$. □

Observation 276. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in [0,d]$. Then $\mathcal{W}_j(q) \subseteq \mathtt{V}^{\text{-}}(a_{j'}, b_{j'})$ for all $j, j' \in [0,d]$ with $j > j'$.*

Observation 277. *Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in [0,d]$. Let $j \in [d]$ and $w \in \mathcal{W}_j(q)$. Let $z \in \mathtt{anc}(w)$ be an ancestor of w with $\mathtt{sd}(z) \geq j$. Then $z \in \mathcal{W}_j(q)$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in [0,d]$. Let $j \in [d]$ and $w \in \mathcal{W}_j(q)$. First let us show that there is a subbundle pair $[x,y] \in \mathcal{F}_j(q)$ such that $w \in \mathtt{W}_j^q(x,y)$. If $j = d$, then $w \in \mathtt{W}_j^q(a,b)$, and we let $[x,y] = [a,b]$. If $j < d$, then w is contained in a set $\mathtt{W}_j^q(u,w)$ for a subbundle pair $[u,w] \in \mathrm{C}_j^P(q)$, and we let $[x,y] := [u,w]$. Thus, there is a subbundle pair $[x,y] \in \mathcal{F}_j(q)$ such that $w \in \mathtt{W}_j^q(x,y)$. As $[a_{j-1}, b_{j-1}]$ is consistent and not minimal and $\mathcal{F}_j(q) = \mathcal{F}(a_{j-1}, b_{j-1})$, it follows from Observation 249 that there exists a consistent pair $[s,t]$ which is a descendant of $[x,y]$ in $\mathcal{F}_j(q)$ such that $w \in \mathtt{V}(x,y) \subseteq \mathtt{V}(s,t)$. Then $\mathtt{V}(s,t) \subseteq \mathtt{W}_j^q(x,y)$. According to

Observation 247 the side depth of $[s,t]$ is j. By Observation 263, each ancestor of w of side depth at least j must also be in $\mathtt{V}(s,t) \subseteq \mathtt{W}_j^q(x,y) \subseteq \mathcal{W}_j(q)$. Let $z \in \mathtt{anc}(w)$ be an ancestor of w with $\mathtt{sd}(z) \geq j$. Then vertex z is in $\mathcal{W}_j(q)$. □

Lemma 278. *Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0,\dots,d\}$. For all $j \in [d]$ and all $[u,v] \in \mathrm{C}_j^P(q)$, the affiliated subbundle pair $r := [u,v]_{[a_{j-1},b_{j-1}]}$ is a node of $\mathcal{T}_{\text{gen}}$ and it holds that $\mathcal{W}(r) \cap \mathtt{V}(a_j,b_j) \subseteq \mathtt{V}(u,v) \cap \mathtt{V}(a_j,b_j)$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0,\dots,d\}$. Let $j \in [d]$ and $[u,v] \in \mathrm{C}_j^P(q)$. It follows from Observation 273 that $r := [u,v]_{[a_{j-1},b_{j-1}]} \in V_{\mathcal{T}_{\text{gen}}}$. Let a_i', b_i' for $i \in \{0,\dots,j\}$ be the side ancestors of r. Then $a_i' = a_i$ and $b_i' = b_i$ for all $i \in \{0,\dots,j-1\}$ by Observation 266. Let w be an arbitrary vertex in $\mathcal{W}(r) \cap \mathtt{V}(a_j,b_j)$, and let $k \leq j$ be maximal such that $w \in \mathcal{W}_k(r)$.

First let us consider the case where $k = j$. Then w is in $\mathcal{W}_j(r) = \mathtt{W}_j^q(u,v)$ (Observation 274). Thus, $w \in \mathtt{W}_j^q(u,v) \cap \mathtt{V}(a_j,b_j)$. Since $[u,v]$ is a child of $[a_j,b_j]$ in the decomposition forest $\mathcal{F}_j(q)$, it follows from Lemma 232 and Lemma 184 that $\mathtt{W}_j^q(u,v) \cap \mathtt{V}(a_j,b_j) \subseteq \mathtt{V}(u,v) \cap \mathtt{V}(a_j,b_j)$. Thus, $w \in \mathtt{V}(u,v) \cap \mathtt{V}(a_j,b_j)$.

Next let us consider the case where $k < j$. As $w \in \mathtt{V}(a_j,b_j)$, vertex w must also be contained in $\mathtt{V}(a_k,b_k)$ (Observation 267). Let $[x,y] \in \mathrm{C}_k^P(r)$ be such that $w \in \mathtt{W}_k^r(x,y)$. Then $[x,y]$ is a child of $[a_k',b_k']$ in the decomposition forest $\mathcal{F}_k(r)$. According to Observation 268 we have $\mathcal{F}_k(q) = \mathcal{F}_k(r)$. Therefore, $\mathtt{W}_k^r(x,y) = \mathtt{W}_k^q(x,y)$. Thus, $w \in \mathtt{W}_k^q(x,y) \cap \mathtt{V}(a_k,b_k)$, and $[x,y]$ is a child of $[a_k,b_k]$ in the decomposition forest $\mathcal{F}_k(q)$. Lemmas 232 and 184 yield that $\mathtt{W}_k^q(x,y) \cap \mathtt{V}(a_k,b_k) \subseteq \mathtt{V}(x,y) \cap \mathtt{V}(a_k,b_k)$. Hence, $w \in \mathtt{V}(x,y) \cap \mathtt{V}(a_k,b_k)$.

According to Observation 250, there is an $f \in \mathtt{F}$ such that $\mathtt{V}(x,y) \cap \mathtt{V}(a_k,b_k)$ is a subset of $\mathtt{S}_f$ and $\mathtt{V}(x,y) \cap \mathtt{V}(a_k,b_k)$ induces a path in $\mathtt{T}_f$. Let $d_w := \mathtt{sd}_f(w)$ be the depth of w in $\mathtt{T}_f$. Clearly, $d_w \geq j$ since $w \in \mathtt{V}(a_j,b_j)$ and $\mathtt{V}(a_j,b_j)$ contains only vertices $w' \in \mathtt{V}_f$ with $\mathtt{sd}_f(w') \leq j$ (Observation 239). Let $w_k := z([x,y],[a_k,b_k])$. Then vertex w is an ancestor of w_k in $\mathtt{V}_f$. Since $[a_k,b_k] = [a_k',b_k']$, we have $w_k = z([x,y],[a_k',b_k'])$ and vertex w_k must be in $\mathcal{W}_{k+1}(r)$ by definition of $\mathrm{C}_k^P(r)$. As vertex w has side depth $d_w \geq j \geq k+1$, Observation 277 implies that $w \in \mathcal{W}_{k+1}(r)$, which is a contradiction to the choice of k. □

Lemma 279. *Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0,\dots,d\}$. For all $j \in \{1,\dots,d+1\}$ and all $[u,v] \in \mathrm{C}_j^P(q)$, it holds that $\mathcal{W}(r) \subseteq \mathcal{W}(q)$ where $r := [u,v]_{[a_{j-1},b_{j-1}]}$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0,\dots,d\}$. Let $j \in [0,d+1]$ and $[u,v] \in \mathrm{C}_j^P(q)$. Let $r := [u,v]_{[a_{j-1},b_{j-1}]}$. By Observation 273 we have $r \in V_{\mathcal{T}_{\text{gen}}}$. Let a_i', b_i' for $i' \in \{0,\dots,j\}$ be the side ancestors of r. Observation 266 implies $a_i' = a_i$ and $b_i' = b_i$ for all $i \in \{0,\dots,j-1\}$.

Let $j = d+1$. Then $r = [u,v]_{a,b}$ and $[u,v]$ is the root of $\mathcal{F}(a,b) = \mathcal{F}(a_d',b_d') = \mathcal{F}_{d+1}(r)$. Thus, $[u,v]$ does not have a parent in $\mathcal{F}_{d+1}(r)$, and $d+1 \notin J(r)$. As $a_d' = a_d$ and $b_d' = b_d$ for all $i \in \{0,\dots,d\}$, we have $J(r) = J(q)$ and $p_i(q) = p_i(r)$ for all $i \in J(q)$ by Observation 271. Since, $a_d' = a$ and $b_d' = b$, we obtain $P_i(r) \setminus \{a,b\} = P_i(q) \setminus \{a_{d-1},b_{d-1}\}$

for all $i \in \{0, \dots, d\}$. As $[a, b]$ is the affiliation of $[u, v]$, $[a, b]$ is non-minimal consistent by Observation 254. Thus, a_{d-1} and b_{d-1} are ancestors of a and b, respectively. Hence, each ancestor of some vertex in $P_i(q)$ is an ancestor of some vertex in $P_i(r)$ for all $i \in \{0, \dots, d\}$.

Let us show that $\mathcal{W}_d(r) \subseteq \mathcal{W}_d(q)$: We have $\mathcal{W}_{d+1}(r) \subseteq \mathcal{W}_d(q)$, because $\mathcal{W}_{d+1}(r) \subseteq \mathtt{V}(a, b)$ by Observation 276 and $\mathtt{V}(a, b) \subseteq \mathtt{W}_d^q(a, b) = \mathcal{W}_d(q)$. All subbundle pairs $[x, y]$ in $\mathrm{C}_d^P(r)$ are children of $[a'_d, b'_d] = [a, b]$ in $\mathcal{F}_d(r) = \mathcal{F}_d(q)$ (Observation 268). Thus, $\mathtt{W}_d^r(x, y) = \mathtt{W}_d^q(x, y) \subseteq \mathtt{W}_d^q(a, b)$. It follows that $\mathcal{W}_d(r) \subseteq \mathcal{W}_d(q)$. Claim 1 yields that $\mathcal{W}(r) \subseteq \mathcal{W}(q)$.

Claim 1. Let $l \leq \min\{j, d\}$. Let $\mathcal{W}_l(r) \subseteq \mathcal{W}_l(q)$, Further, let each ancestor of some vertex in $P_k(q)$ be an ancestor of some vertex in $P_k(r)$ or not be in $\mathcal{W}(r)$ for all $k \leq l$. Then $\mathcal{W}(r) \subseteq \mathcal{W}(q)$.

Proof. Let $l \leq \min\{j, d\}$. Let $\mathcal{W}_l(r) \subseteq \mathcal{W}_l(q)$, and let each ancestor of some vertex in $P_k(q)$ be an ancestor of some vertex in $P_k(r)$ or not be in $\mathcal{W}(r)$ for all $k \leq l$. Let us prove by induction that $\mathcal{W}_m(r) \subseteq \mathcal{W}_m(q)$ for all $m \in \{l, \dots, 0\}$. Clearly, we have $\mathcal{W}_m(r) \subseteq \mathcal{W}_m(q)$ for $m = l$. Let $m < l$ and let us assume $\mathcal{W}_{m+1}(r) \subseteq \mathcal{W}_{m+1}(q)$. Since $a'_{j-1} = a_{j-1}$ and $b'_{j-1} = b_{j-1}$ and $m < l \leq j$, we have $\mathrm{C}_m(r) = \mathrm{C}_m(q)$ (Observation 269). Let $[x, y] \in \mathrm{C}_m^P(r)$. Then $[x, y] \in \mathrm{C}_m(r)$. Further, the vertex $z := z([x, y], [a'_m, b'_m]) = z([x, y], [a_m, b_m])$ is contained in $\mathcal{W}_{m+1}(r)$, and therefore, in $\mathcal{W}_{m+1}(q)$ by inductive assumption. Moreover, $z \in \mathcal{W}(r)$ is not an ancestor of any vertex in $P_m(r)$. It follows that z is not an ancestor of any vertex in $P_m(q)$. Hence, $[x, y] \in \mathrm{C}_m^P(q)$, and $\mathtt{W}_m^q(x, y) \subseteq \mathcal{W}_m(q)$. Since $\mathcal{F}_m(q) = \mathcal{F}_m(r)$ (Observation 268), we have $\mathtt{W}_m^r(x, y) = \mathtt{W}_m^q(x, y)$. Thus, $\mathtt{W}_m^r(x, y) \subseteq \mathcal{W}_m(q)$. It follows that $\mathcal{W}_m(r) \subseteq \mathcal{W}_m(q)$. ⌟

Next, let $j = d$. As $a'_{d-1} = a_{d-1}$ and $b'_{d-1} = b_{d-1}$, we have $J(r) \setminus \{d\} = J(q) \setminus \{d\}$ and $p_i(q) = p_i(r)$ for all $i \in J(q) \setminus \{d\}$ by Observation 271. If $[a, b]$ has no parent in $\mathcal{F}_d(q)$, then $P_k(r) \setminus \{p_d(r)\} = P_k(q)$ for all $k \in \{0, \dots, d\}$, and each ancestor of some vertex in $P_k(q)$ is an ancestor of some vertex in $P_k(r)$. If $[a, b]$ has a parent in $\mathcal{F}_d(q)$, then there exists a vertex $p_d(q)$, and $P_k(r) \setminus \{p_d(r)\} = P_k(q) \setminus \{p_d(q)\}$ for all $k \in \{0, \dots, d\}$. According to Lemma 278, $\mathcal{W}(r) \cap \mathtt{V}(a, b) \subseteq \mathtt{V}(u, v) \cap \mathtt{V}(a, b)$. Therefore, all vertices in $\mathcal{W}(r) \cap \mathtt{V}(a, b)$ are ancestors of $p_d(r) = z([u, v], [a, b])$. Since $p_d(q) \in \mathtt{V}(a, b)$, every ancestor $w \in \mathtt{V}(a, b)$ of $p_d(q)$ is an ancestor of $p_d(r)$ or not in $\mathcal{W}(r)$. It follows that every ancestor w of $p_d(q)$ is an ancestor of a, b or $p_d(r)$, or not in $\mathcal{W}(r)$. Hence, each ancestor of some vertex in $P_k(q)$ is an ancestor of some vertex in $P_k(r)$ or not in $\mathcal{W}(r)$ for all $k \in \{0, \dots, d\}$.

Let us show that $\mathcal{W}_d(r) \subseteq \mathcal{W}_d(q)$: Since $a'_{d-1} = a_{d-1}$ and $b'_{d-1} = b_{d-1}$, it follows that $\mathcal{F}_d(q) = \mathcal{F}_d(r)$ (Observation 268). As $[u, v]$ is a child of $[a, b]$ in $\mathcal{F}_d(q)$, we obtain $\mathtt{W}_d^r(u, v) \subseteq \mathtt{W}_d^r(a, b)$. Hence, $\mathcal{W}_d(r) \subseteq \mathcal{W}_d(q)$. Claim 1 implies that $\mathcal{W}(r) \subseteq \mathcal{W}(q)$.

Finally, let $j < d$. Since $a'_{j-1} = a_{j-1}$ and $b'_{j-1} = b_{j-1}$, we have $J(r) \setminus \{j\} = J(q) \setminus \{j, \dots, d\}$ and $p_i(q) = p_i(r)$ for all $i \in J(r) \setminus \{j\}$ by Observation 271. Hence, for all $k \in \{0, \dots, j-1\}$, it holds that

$$P_k(r) \setminus \{a_{j-1}, b_{j-1}, p_j(r)\} = P_k(q) \setminus (\{a_{d-1}, b_{d-1}\} \cup \{p_l(q) \mid l \in [j, d] \cap J(q)\}).$$

Let $i \in \{0, \dots, j-1\}$ and $[x, y] \in \mathrm{C}_i^P(r)$. Let the vertex $z_i := z([x, y], [a_i, b_i])$ be an ancestor of a_{d-1} (or b_{d-1}). In the following, we prove that z_i also is an ancestor of

a_{j-1} or $p_j(r)$ (or b_{j-1} or $p_j(r)$). Let us assume that z_i is a proper descendant of a_{j-1} (or b_{j-1}). Then $z_i \in \{a_{d-1}, \ldots, a_j\}$ or $z_i \in \{b_{d-1}, \ldots, b_j\}$. Hence, $z_i \in \mathtt{V}(a_j, b_j)$ according to Observation 267. Since $[x, y] \in \mathrm{C}_i^P(r)$, we have $z_i \in \mathcal{W}_{i+1}(r)$. Thus, $z_i \in \mathcal{W}(r) \cap \mathtt{V}(a_j, b_j)$. By Lemma 278 we know that $\mathcal{W}(r) \cap \mathtt{V}(a_j, b_j) \subseteq \mathtt{V}(u, v) \cap \mathtt{V}(a_j, b_j)$. Therefore, the intersection of $\mathtt{V}(u, v) \cap \mathtt{V}(a_j, b_j)$ must contain z_i. Then z_i is an ancestor of $p_j(r) = z([u, v], [a_j, b_j])$.

Let $i \in \{0, \ldots, j-1\}$ and $[x, y] \in \mathrm{C}_i^P(r)$. Let the vertex $z_i := z([x, y], [a_i, b_i])$ be an ancestor of $p_l(q)$ with $l \in [j, d] \cap J(q)$. We show that z_i is also an ancestor of some vertex in $\{a_{j-1}, b_{j-1}, p_j(r)\}$. Since $p_l(q) \in V(a_l, b_l)$, we have $p_l(q) \in V(a_j, b_j)$ according to Observation 267. Further, $\mathtt{sd}(p_l(q)) \geq j$ according to Observation 239. As $p_l(q) \notin \mathtt{O}$ (Observation 270), there is a unique ancestor of $p_l(q)$ of side depth k for all $k \leq j$ (Observation 262). By Corollary 265 the unique ancestor of $p_l(q)$ of side depth j is a_j or b_j. If $\mathtt{sd}(z_i) < j$, then z_i is an ancestor of a_{j-1} or b_{j-1}. Let $\mathtt{sd}(z_i) \geq j$. According to Observation 263, $z_i \in \mathtt{V}(a_j, b_j)$. Since $[x, y] \in \mathrm{C}_i^P(r)$, we have $z_i \in \mathcal{W}_{i+1}(r)$. Hence, $z_i \in \mathcal{W}(r) \cap \mathtt{V}(a_j, b_j)$. It follows from Lemma 278 that $z_i \in \mathtt{V}(u, v) \cap \mathtt{V}(a_j, b_j)$. Consequently, z_i is an ancestor of $p_j(r) = z([u, v], [a_j, b_j])$.

Let $i \in \{0, \ldots, j-1\}$. Suppose $\mathcal{W}_{i+1}(r) \subseteq \mathcal{W}_{i+1}(q)$. Since $a'_{j-1} = a_{j-1}$ and $b'_{j-1} = b_{j-1}$, we have $\mathrm{C}_i(r) = \mathrm{C}_i(q)$ (Observation 269). Let $[x, y] \in \mathrm{C}_i^P(r)$. Then $[x, y] \in \mathrm{C}_i(q)$. The vertex $z_i := z([x, y], [a_i, b_i])$ is in $\mathcal{W}_{i+1}(r)$, and z_i is not an ancestor of any vertex in $P_i(r)$. Let us show that $[x, y] \in \mathrm{C}_i^P(q)$. Assume that $[x, y] \notin \mathrm{C}_i^P(q)$. Since $z_i \in \mathcal{W}_{i+1}(r) \subseteq \mathcal{W}_{i+1}(q)$, it follows that z_i is an ancestor of some vertex in $P_i(q)$. As z_i is not an ancestor of any vertex in $P_i(r)$, the vertex z_i cannot be an ancestor of any vertex in $P_i(q) \setminus (\{a_{d-1}, b_{d-1}\} \cup \{p_l(q) \mid l \in [j, d] \cap J(q)\})$. Further, it follows from the above two paragraphs that z_i also cannot be an ancestor of a_{d-1} or b_{d-1}, or of any vertex in $\{p_l(q) \mid l \in [j, d] \cap J(q)\}$. We obtain a contradiction. Consequently, $\mathcal{W}_{i+1}(r) \subseteq \mathcal{W}_{i+1}(q)$ implies that $\mathrm{C}_i^P(r) \subseteq \mathrm{C}_i^P(q)$.

Let us show that $\mathcal{W}_j(r) \subseteq \mathcal{W}_j(q)$: According to Observation 274, we have $\mathcal{W}_j(r) = \mathtt{W}_j^q(u, v)$. As $[u, v] \in \mathrm{C}_j^P(q)$, we have $\mathtt{W}_j^q(u, v) \subseteq \mathcal{W}_j(q)$. Thus, $\mathcal{W}_j(r) \subseteq \mathcal{W}_j(q)$.

Since $\mathcal{W}_j(r) \subseteq \mathcal{W}_j(q)$ and $\mathcal{W}_{i+1}(r) \subseteq \mathcal{W}_{i+1}(q)$ implies that $\mathrm{C}_i^P(r) \subseteq \mathrm{C}_i^P(q)$ for every $i \in \{0, \ldots, j-1\}$, we can show inductively that $\mathcal{W}_i(r) \subseteq \mathcal{W}_i(q)$ for all $i \in \{0, \ldots, j\}$ (cf. the proof of Claim 1). Hence, $\mathcal{W}(r) \subseteq \mathcal{W}(q)$. □

6.6.3. Decomposing $\mathcal{W}(q)$ for Non-Minimal Affiliated Consistent Pairs q

All affiliated subbundle pairs $q \in V_{\mathcal{T}_{\text{gen}}}$ are either non-minimal consistent or trivial according to Corollary 257. We first consider non-minimal consistent affiliated subbundle pairs $q \in V_{\mathcal{T}_{\text{gen}}}$.

Let q be a non-minimal affiliated consistent pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. Let us consider specific subsets $\mathrm{C}_j^{\text{con}}(q)$ of $\mathrm{C}_j^P(q)$ for all $j \in \{0, \ldots, d+1\}$. Let $\mathrm{C}_{d+1}^{\text{con}}(q) := \mathrm{C}_{d+1}^P(q)$, and for all $j \in \{0, \ldots, d\}$ let $\mathrm{C}_j^{\text{con}}(q)$ be the set of all subbundle pairs $[u, v] \in \mathrm{C}_j^P(q)$ where $z([u, v], [a_j, b_j])$ is a or b. For $j \in [d+1]$ we let $R_j(q)$ be the set of all $[u, v]_{a_{j-1}, b_{j-1}}$ where $[u, v] \in \mathrm{C}_j^{\text{con}}(q)$. We let $R(q) := \bigcup_{j \in [d+1]} R_j(q)$.

Lemma 280. *Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair. Then $\mathcal{W}(q)$ is the union U of the set $\{a, b\}$ and of all sets $\mathcal{W}(r)$ for $r \in R(q)$.*

Proof. Let q be a non-minimal affiliated consistent pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. By Lemma 279 we already know that U is a subset of $\mathcal{W}(q)$. We now prove inductively that U contains all vertices in $\mathcal{W}(q)$. For all $j \in \{0, \ldots, d\}$ let U_j be the union of $\{a, b\}$ and of all sets $\mathcal{W}_j([u, v]_{a_{k-1}, b_{k-1}})$ for $k \in \{j, \ldots, d+1\} \setminus \{0\}$ and $[u, v] \in \mathrm{C}_k^{\mathrm{con}}(q)$. Then $\bigcup_{j \le d} U_j = U$. We use induction to show that $\mathcal{W}_j(q) \subseteq U_j$ for all $j \in \{0, \ldots, d\}$.

Let us show that $\mathcal{W}_d(q) \subseteq U_d$. The set $\mathcal{W}_d(q) = \mathtt{W}_d^q(a, b)$ is the union of $\mathtt{V}(a, b)$ and $\mathtt{W}_d^q(x, y)$ for all children $[x, y]$ of $[a, b]$ in $\mathcal{F}_d(q)$.

First let us prove that $\mathtt{V}(a, b) \subseteq U_d$. As q is consistent and not minimal, the set $\mathrm{C}_{d+1}^P(q) = \mathrm{C}_{d+1}^{\mathrm{con}}(q)$ is not empty. We have $[u', v'] \in \mathrm{C}_{d+1}^P(q)$ if and only if $[u', v']$ is a root of the forest $\mathcal{F}_{d+1}(q) = \mathcal{F}(a, b)$. By Corollary 246, the union of all sets $\mathtt{W}_{d+1}^q(u', v')$ for $[u', v'] \in \mathrm{C}_{d+1}^P(q)$ is exactly the set $\mathtt{V}(a, b) \setminus \{a, b\}$. Let $[u', v'] \in \mathrm{C}_{d+1}^P(q)$ and $r' = [u', v']_{a,b}$. Then $\mathcal{W}_{d+1}(r') = \mathtt{W}_{d+1}^q(u', v')$ by Observation 274. As a consequence, the union of all sets $\mathcal{W}_{d+1}(r')$ for $r' = [u', v']_{a,b}$ with $[u', v'] \in \mathrm{C}_{d+1}^P(q) = \mathrm{C}_{d+1}^{\mathrm{con}}(q)$ is the set $\mathtt{V}^-(a, b)$. Thus, we have $\mathtt{V}(a, b) \subseteq U_d$.

If $d = 0$, then $[a, b]$ has no children in $\mathcal{F}_d(q)$, and this already proves that $\mathcal{W}_0(q) \subseteq U_0$. In the following let $d > 0$.

Now let $[x, y]$ be a child of $[a, b]$ in $\mathcal{F}_d(q)$, that is, let $[x, y] \in \mathrm{C}_d^P(q)$. Let us show that $\mathtt{W}_d^q(x, y) \subseteq U_d$. Let $r := [x, y]_{a_{d-1}, b_{d-1}}$. According to Observation 274, we have $\mathtt{W}_d^q(x, y) = \mathcal{W}_d(r)$. If $z := z([x, y], [a_d, b_d])$ is a or b, then $[x, y] \in \mathrm{C}_d^{\mathrm{con}}(q)$, and $\mathcal{W}_d(r) \subseteq U_d$ according to the definition of U_d, and therefore, $\mathtt{W}_d^q(x, y) \subseteq U_d$. Let z be not a or b. Then z is in $\mathtt{V}^-(a, b)$, and there is a subbundle pair $[u', v'] \in \mathrm{C}_{d+1}^P(q) = \mathrm{C}_{d+1}^{\mathrm{con}}(q)$ such that $z \in \mathtt{W}_{d+1}^q(u', v')$. Let $r' := [u', v']_{a,b}$. Then, $\mathcal{W}_d(r') \subseteq U_d$ according to the definition of U_d. If $\mathtt{W}_d^q(x, y) \subseteq \mathcal{W}_d(r')$, then $\mathtt{W}_d^q(x, y) \subseteq U_d$. Thus, let us show that $\mathtt{W}_d^q(x, y) \subseteq \mathcal{W}_d(r')$. Since $\mathcal{F}_d(q) = \mathcal{F}_d(r')$ (Observations 268), we have $\mathtt{W}_d^q(x, y) = \mathtt{W}_d^{r'}(x, y)$. We prove that $[x, y] \in \mathrm{C}_d^P(r')$. Then it follows that $\mathtt{W}_d^{r'}(x, y) \subseteq \mathcal{W}_d(r')$, and therefore, $\mathtt{W}_d^q(x, y) \subseteq \mathcal{W}_d(r')$.

Observation 269 implies that $\mathrm{C}_d(q) = \mathrm{C}_d(r')$. Consequently, $[x, y] \in \mathrm{C}_d(r')$. It follows from Observation 274 that $\mathtt{W}_{d+1}^q(u', v') = \mathcal{W}_{d+1}(r')$. Since $z \in \mathtt{W}_{d+1}^q(u', v')$, we have $z \in \mathcal{W}_{d+1}(r')$. In the following paragraph we show that z is not an ancestor of any vertex in $P_d(r') = \{a, b\} \cup \{p_d(r') \mid d \in J(r')\}$. Then it follows that $[x, y] \in \mathrm{C}_d^P(r')$, and we are done.

Let us show that z is not an ancestor of any vertex in $P_d(r') = \{a, b\} \cup \{p_d(r') \mid d \in J(r')\}$. As $[u', v']$ is a root of the forest $\mathcal{F}_{d+1}(q) = \mathcal{F}_{d+1}(r')$ (Observation 268), we have $d+1 \notin J(r')$. Suppose $d \in J(r')$, and assume z is an ancestor of $p_d(r')$. Let $[a^*, b^*]$ be the parent of $[a, b]$ in $\mathcal{F}_d(r')$. Since $[x, y]$ is a child of $[a, b]$ in $\mathcal{F}_d(q) = \mathcal{F}_d(r')$ (Observation 268), it follows from Lemma 238 that $\mathtt{V}(x, y) \cap \mathtt{V}(a, b)$ is a subset of $\mathtt{V}(a, b) \cap \mathtt{V}(a^*, b^*)$. As $z = z([x, y], [a, b])$ and $p_d(r') = z([a, b], [a^*, b^*])$, we obtain a contradiction according to Observation 250 and Lemma 184. Thus, if $d \in J(r')$, then z is not an ancestor of $p_d(r')$. Since z is not a or b, vertex z is not an ancestor of any vertex in $P_d(r') = \{a, b\} \cup \{p_d(r') \mid d \in J(r')\}$.

Next let us assume that $\mathcal{W}_{j+1}(q) \subseteq U_{j+1}$ for $j \in \{0, \ldots, d-1\}$. We prove that $\mathcal{W}_j(q) \subseteq U_j$. In order to do this, we have to show that $\mathtt{W}_j^q(u, v) \subseteq U_j$ for all $[u, v] \in \mathrm{C}_j^P(q)$. As $\mathrm{C}_0^P(q) = \emptyset$, this clearly is the case for $j = 0$. Thus, let $j > 0$. Let $[u, v] \in \mathrm{C}_j^P(q)$ and let $r := [u, v]_{a_{j-1}, b_{j-1}}$. Observation 274 implies that $\mathcal{W}_j(r) = \mathtt{W}_j^q(u, v)$. Let $z := z([u, v], [a_j, b_j])$. According to the definition of U_j we have $\mathcal{W}_j(r) \subseteq U_j$ if z is a or b. Hence, if z is a or b, then $\mathtt{W}_j^q(u,v) \subseteq U_j$. Let z be neither a nor b for the rest of the proof. As $[u,v] \in \mathrm{C}_j^P(q)$, vertex z is in $\mathcal{W}_{j+1}(q)$. Hence, it also is in U_{j+1} by inductive assumption.

Let us consider the case where $z \in \mathcal{W}_{j+1}(r')$ for $r' := [u',v']_{a,b}$ with $[u',v'] \in \mathrm{C}_{d+1}^P(q) = \mathrm{C}_{d+1}^{\mathrm{con}}(q)$. As $[u',v']$ is a root of the forest $\mathcal{F}_{d+1}(q) = \mathcal{F}_{d+1}(r')$ (Observation 268), we have $d+1 \notin J(r')$. Hence, Observation 271 implies that $P_j(r') \setminus \{a,b\} = P_j(q) \setminus \{a_{d-1}, b_{d-1}\}$. Since $[u,v] \in \mathrm{C}_j^P(q)$, vertex z is not an ancestor of an vertex in $P_j(q)$. As z is neither a nor b, vertex z is not an ancestor of an vertex in $P_j(r')$. According to Observation 269, we have $\mathrm{C}_j(q) = \mathrm{C}_j(r')$. Hence, $[u,v] \in \mathrm{C}_j(r')$. Since $[u,v] \in \mathrm{C}_j(r')$, $z \in \mathcal{W}_{j+1}(r')$ and vertex z is not an ancestor of an vertex in $P_j(r')$, we obtain that $[u,v]$ is contained in $\mathrm{C}_j^P(r')$. Consequently, $\mathtt{W}_j^{r'}(u,v) \subseteq \mathcal{W}_j(r')$. As $[u',v'] \in \mathrm{C}_{d+1}^{\mathrm{con}}(q)$, we have $\mathcal{W}_j(r') \subseteq U_j$ according to the definition of U_j. Since $\mathcal{F}_j(q) = \mathcal{F}_j(r')$ (Observations 268), we obtain $\mathtt{W}_j^q(u,v) = \mathtt{W}_j^{r'}(u,v)$. It follows that $\mathtt{W}_j^q(u,v) \subseteq U_j$.

Finally, let vertex z be in the set $\mathcal{W}_{j+1}(s)$ where $s = [x,y]_{a_{k-1},b_{k-1}}$ with $k \in \{j+1,\dots,d\}$ and $[x,y] \in \mathrm{C}_k^{\mathrm{con}}(q) \subseteq \mathrm{C}_k^P(q)$. Thus, $z([x,y],[a_k,b_k])$ is a or b. By Observation 271, we have $P_j(s) \setminus \{a_{k-1}, b_{k-1}, p_k(s)\} = P_j(q) \setminus (\{a_{d-1}, b_{d-1}\} \cup \{p_l(q) \mid k \le l \le d\})$. As vertex $p_k(s) = z([x,y],[a_k,b_k])$ is either a or b, the only vertex that is an ancestor of some vertex in $P_j(s)$ but not an ancestor of any vertex in $P_j(q)$ is $p_k(s)$, that is, a or b. Note that $a,b \notin \mathtt{O}$ because q is consistent and not minimal. Since $[u,v] \in \mathrm{C}_j^P(q)$, vertex z is not an ancestor of any vertex in $P_j(q)$. As we assumed that z is not a or b, it follows that vertex z is not an ancestor of any vertex in $P_j(s)$. According to Observation 269, $\mathrm{C}_j(q) = \mathrm{C}_j(s)$. Therefore, $[u,v] \in \mathrm{C}_j(s)$. Since $[u,v] \in \mathrm{C}_j(s)$, $z \in \mathcal{W}_{j+1}(s)$ and z is not an ancestor of any vertex in $P_j(s)$, we obtain that $[u,v] \in \mathrm{C}_j^P(s)$. Consequently, $\mathtt{W}_j^s(u,v) \subseteq \mathcal{W}_j(s) \subseteq U_j$. As $\mathcal{F}_j(q) = \mathcal{F}_j(s)$ (Observations 268), it follows that $\mathtt{W}_j^q(u,v) = \mathtt{W}_j^s(u,v)$. Hence, $\mathtt{W}_j^q(u,v) \subseteq U_j$. □

Lemma 281. *Let $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ be a non-minimal affiliated consistent pair. Let $r_1, r_2 \in R(q)$ with $r_1 \neq r_2$. Then $\mathcal{W}(r_1) \cap \mathcal{W}(r_2) \subseteq \mathtt{anc}(q)$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\mathrm{gen}}}$ be a non-minimal affiliated consistent pair of side depth d with side ancestors a_i, b_i for $i \in \{0,\dots,d\}$. Let $r_1, r_2 \in R(q)$ with $r_1 \neq r_2$. Let $k_1, k_2 \in \{1,\dots,d+1\}$ and let $[u_1,v_1] \in \mathrm{C}_{k_1}^{\mathrm{con}}(q)$ and $[u_2,v_2] \in \mathrm{C}_{k_2}^{\mathrm{con}}(q)$ be such that $r_1 := [u_1,v_1]_{a_{k_1-1},b_{k_1-1}}$ and $r_2 := [u_2,v_2]_{a_{k_2-1},b_{k_2-1}}$.

Let us assume there exists a vertex w such that $w \in \mathcal{W}(r_1) \cap \mathcal{W}(r_2)$ and $w \notin \mathtt{anc}(q)$. For each $w \in (\mathcal{W}(r_1) \cap \mathcal{W}(r_2)) \setminus \mathtt{anc}(q)$ let $l_j(w)$ be the maximal $l \le k_j$ such that $w \in \mathcal{W}_l(r_j)$. Let $w \in (\mathcal{W}(r_1) \cap \mathcal{W}(r_2)) \setminus \mathtt{anc}(q)$ be such that for all $w' \in (\mathcal{W}(r_1) \cap \mathcal{W}(r_2)) \setminus \mathtt{anc}(q)$ we have

- $l_j(w') \le l_j(w)$ for each $j \in \{1,2\}$ or
- there exists a $j' \in \{1,2\}$ such that $l_j(w') < l_j(w)$.

From now on we denote $l_1(w)$ and $l_2(w)$ by l_1 and l_2, respectively. Without loss of generality, let $l_1 \le l_2$, and if $l_1 = l_2$, then let $k_1 \le k_2$.

We check multiple cases. The first case is that $l_1 = k_1$ and $l_1 < l_2$. We know that $w \in \mathcal{W}_{l_1}(r_1)$ and $w \in \mathcal{W}_{l_2}(r_2)$. As $l_1 < l_2 \le k_2$, Observations 266 and 276 imply that $\mathcal{W}_{l_2}(r_2) \subseteq \mathtt{V}(a_{l_1}, b_{l_1})$. We obtain that $w \in \mathtt{V}(a_{l_1}, b_{l_1})$. Thus, w is contained in $\mathcal{W}_{l_1}(r_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$. Since $l_1 < l_2$, we have $l_1 \le d$, and according to Lemma 278 we have $\mathcal{W}(r_1) \cap \mathtt{V}(a_{l_1}, b_{l_1}) \subseteq \mathtt{V}(u_1,v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$. Thus, w is contained in $\mathtt{V}(u_1,v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$.

As $l_1 = k_1$, we have $k_1 \leq d$. Hence, the vertex $z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is a or b. By Observation 250, each vertex in $\mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$ is an ancestor of a or of b. As $k_1 = l_1$, vertex w is an ancestor of a or of b. Thus, $w \in \mathtt{anc}(q)$, a contradiction.

Next, let $l_1 = k_1$, $l_1 = l_2$ and $l_2 < k_2$. As $l_1 = k_1$ and $w \in \mathcal{W}_{l_1}(r_1)$, we have $w \in \mathtt{W}^q_{l_1}(u_1, v_1)$ according to Observation 274. Subbundle pair $[u_1, v_1]$ is a child of $[a_{l_1}, b_{l_1}]$ in $\mathcal{F}_{l_1}(q)$. Further, $w \in \mathcal{W}_{l_2}(r_2)$ and $w \notin \mathcal{W}_{l_2+1}(r_2)$. Since $l_2 < k_2$, there exists a child $[u'_2, v'_2]$ in $\mathrm{C}^P_{l_2}(r_2)$ such that $w \in \mathtt{W}^{r_2}_{l_2}(u'_2, v'_2)$. By Observation 268, $\mathcal{F}_{l_2}(r_2) = \mathcal{F}_{l_2}(q)$. Hence, $\mathtt{W}^{r_2}_{l_2}(u'_2, v'_2) = \mathtt{W}^q_{l_2}(u'_2, v'_2)$. Observation 269 yields that $\mathrm{C}_{l_2}(r_2) = \mathrm{C}_{l_2}(q)$. As $l_1 = l_2$, the subbundle pair $[u'_2, v'_2]$ is a child of $[a_{l_1}, b_{l_1}]$ in $\mathcal{F}_{l_1}(q)$ as well. According to Claim 1, $[u_1, v_1] \neq [u'_2, v'_2]$. By Lemmas 233 and 184, we have $\mathtt{W}^q_{l_1}(u_1, v_1) \cap \mathtt{W}^q_{l_1}(u'_2, v'_2) \subseteq \mathtt{V}(a_{l_1}, b_{l_1})$. Therefore, w is contained in $\mathtt{W}^q_{l_1}(u_1, v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$, which is a subset of $\mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$ according to Lemmas 232 and 184. Since $l_1 = k_1$, we have $w \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. As $k_1 < k_2$, we have $k_1 \leq d$, and $z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is a or b. By Observation 250, it follows that w is an ancestor of a or b, a contradiction.

Claim 1. Let $l_1 = k_1$, $l_1 = l_2$ and $l_2 < k_2$. Then $[u_1, v_1] \neq [u'_2, v'_2]$.

Proof. Let $l_1 = k_1$, $l_1 = l_2$ and $l_2 < k_2$. Assume $[u_1, v_1] = [u'_2, v'_2]$. As $k_1 < k_2$, we have $k_1 \leq d$, and we know that $z_1 := z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is a or b. Without loss of generality, let $z_1 = a$.

First let us consider the case where $k_2 = d + 1$ and $k_1 < d$. Then $a, b \in P_{k_1}(r_2)$. Since $l_2 = k_1 < d$ and $[u'_2, v'_2] \in \mathrm{C}^P_{l_2}(r_2)$, vertex $z_1 = z([u'_2, v'_2], [a_{l_2}, b_{l_2}])$ cannot be an ancestor of any vertex in $P_{k_1}(r_2)$, a contradiction.

Next let us consider the case where $k_2 = d + 1$ and $k_1 = d$. We have $\mathcal{W}_{k_2}(r_2) \subseteq \mathtt{V}^-(a, b)$ according to Observation 276. Since $[u'_2, v'_2] \in \mathrm{C}^P_{l_2}(r_2)$, that is, $[u_1, v_1] \in \mathrm{C}^P_{k_1}(r_2)$, the vertex $z_1 = z([u_1, v_1], [a_{k_1}, b_{k_1}])$ must be in $\mathcal{W}_{k_1+1}(r_2) = \mathcal{W}_{k_2}(r_2)$. We obtain a contradiction, because $z_1 = a$ and $a \notin \mathcal{W}_{k_2}(r_2)$.

Finally let us consider the case where $k_2 \leq d$. Then $z_2 := z([u_2, v_2], [a_{k_2}, b_{k_2}])$ is a or b. First, let $z_2 = a$. It follows that $p_{k_2}(r_2) = a$. As $l_2 < k_2$, and $[u'_2, v'_2] \in \mathrm{C}^P_{l_2}(r_2)$, that is, $[u_1, v_1] \in \mathrm{C}^P_{k_1}(r_2)$, the vertex $z_1 = z([u_1, v_1], [a_{k_1}, b_{k_1}])$ cannot be an ancestor of any vertex in $P_{k_1}(r_2)$. As $z_1 = a$ and $p_{k_2}(r_2) = a \in P_{k_1}(r_2)$, we obtain a contradiction. Now, let $z_2 = b$. Since $a \in \mathtt{V}(a, b)$, we have $a \in \mathtt{V}(a_{k_2}, b_{k_2})$ by Observation 267. As $z_1 = a$, we further have $a \in \mathtt{V}(u_1, v_1)$. Thus, $a \in \mathtt{V}(u'_2, v'_2)$. Hence, $a \in \mathtt{W}^{r_2}_{l_2}(u'_2, v'_2) \subseteq \mathcal{W}(r_2)$. According to Lemma 278, $\mathcal{W}(r_2) \cap \mathtt{V}(a_{k_2}, b_{k_2}) \subseteq \mathtt{V}(u_2, v_2) \cap \mathtt{V}(a_{k_2}, b_{k_2})$. It follows that $a \in \mathtt{V}(u_2, v_2) \cap \mathtt{V}(a_{k_2}, b_{k_2})$. As $z_2 = z([u_2, v_2], [a_{k_2}, b_{k_2}])$ is b, we have $a, b \in \mathtt{V}(u_2, v_2) \cap \mathtt{V}(a_{k_2}, b_{k_2})$. Since q is consistent and not minimal, we obtain a contradiction according to Observation 250. ⌟

Now let $l_1 = k_1$ and $l_1 = l_2$ and $l_2 = k_2$. Then $w \in \mathtt{W}^q_{k_1}(u_1, v_1)$ and $w \in \mathtt{W}^q_{k_1}(u_2, v_2)$. If $k_1 = d + 1$, then $\mathtt{W}^q_{k_1}(u_1, v_1)$ and $\mathtt{W}^q_{k_1}(u_2, v_2)$ must be disjoint as $[u_1, v_1]$ and $[u_2, v_2]$ are different roots of the decomposition forest $\mathcal{F}_{d+1}$, and therefore disjoint subsets of $\mathtt{V}^-(a, b)$. Let $k_1 \leq d$. Then $[u_1, v_1]$ and $[u_2, v_2]$ are distinct children of $[a_{k_1}, b_{k_1}]$ in $\mathcal{F}_{k_1}(q)$, and similar to the case above we can apply Lemma 233 and Lemma 232 (and Lemma 184) to show that $w \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. Further, we have $k_1 \leq d$ and therefore $z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is a or b. Consequently, w is an ancestor of a or b (Observation 250), a contradiction.

Finally, let $l_1 < k_1$ (and $l_1 \leq l_2$). We know that $w \in \mathcal{W}_{l_1}(r_1)$ and $w \in \mathcal{W}_{l_2}(r_2)$. Let $[u_1', v_1']$ be a child in $\mathrm{C}^P_{l_1}(r_1)$ such that $w \in \mathtt{W}^{r_1}_{l_1}(u_1', v_1')$. First we show that $w \in \mathtt{V}(a_{l_1}, b_{l_1})$. If $l_1 < l_2$, then $\mathcal{W}_{l_2}(r_2) \subseteq \mathtt{V}(a_{l_1}, b_{l_1})$ by Observation 276 and $w \in \mathtt{V}(a_{l_1}, b_{l_1})$. Now let $l_1 = l_2$. Then $k_1 \leq k_2$ and $l_1 = l_2 < k_2$. Thus, let $[u_2', v_2']$ be a child in $\mathrm{C}^P_{l_1}(r_2)$ such that $w \in \mathtt{W}^{r_2}_{l_1}(u_2', v_2')$. From Observation 268 it follows that $\mathcal{F}_{l_1}(r_1) = \mathcal{F}_{l_1}(q) = \mathcal{F}_{l_1}(r_2)$. Hence, $\mathtt{W}^{r_2}_{l_1}(u_2', v_2') = \mathtt{W}^{r_1}_{l_1}(u_2', v_2')$. The subbundle pairs $[u_1', v_1']$ and $[u_2', v_2']$ are children of $[a_{l_1}, b_{l_1}]$ in $\mathcal{F}_{l_1}(r_1)$. According to Claim 2, we have $[u_1', v_1'] \neq [u_2', v_2']$. By Lemmas 233 and 184 it holds that $\mathtt{W}^{r_1}_{l_1}(u_1', v_1') \cap \mathtt{W}^{r_1}_{l_1}(u_2', v_2') \subseteq \mathtt{V}(a_{l_1}, b_{l_1})$. Hence, $w \in \mathtt{V}(a_{l_1}, b_{l_1})$.

Claim 2. Let $l_1 = l_2$, $l_1 < k_1$ and $l_2 < k_2$. Then $[u_1', v_1'] \neq [u_2', v_2']$.

Proof. Let $l_1 = l_2$, $l_1 < k_1$ and $l_2 < k_2$. Let us assume that $[u_1', v_1'] = [u_2', v_2']$. Let $z := z([u_1', v_1'], [a_{l_1}, b_{l_1}])$. As $[u_1', v_1'] \in \mathrm{C}^P_{l_1}(r_1)$ and $[u_2', v_2'] \in \mathrm{C}^P_{l_1}(r_2)$, we have $z \in \mathcal{W}_{l_1+1}(r_1)$ and $z \in \mathcal{W}_{l_1+1}(r_2)$. Thus, $l_1(z) = l_2(z) > l_1 = l_2$. It follows from the choice of w that $z \in \mathtt{anc}(q)$. Note that z is not an ancestor of any vertex in $P_{l_1}(r_1)$. If $k_1 = d + 1$, then $a, b \in P_{l_1}(r_1)$, and we obtain a contradiction. Let $k_1 \leq d$. Then $z_1 := z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is a or b. Without loss of generality, let $z_1 = a$. Since $z_1 = p_{k_1}(r_1)$, we have $a \in P_{l_1}(r_1)$. Further $a_{k_1-1}, b_{k_1-1} \in P_{l_1}(r_1)$. Consequently, vertex z must be an ancestor of b and a descendant of b_{k_1}. By Observation 267, $z \in \mathtt{V}(a_{k_1}, b_{k_1})$. According to Lemma 278, $\mathcal{W}(r_1) \cap \mathtt{V}(a_{k_1}, b_{k_1}) \subseteq \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. It follows that $z \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. Thus $a, z \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. Since q is consistent and not minimal, we obtain a contradiction according to Observation 250. ⌟

According to Lemmas 232 and 184, the set $\mathtt{W}^{r_1}_{l_1}(u_1', v_1') \cap \mathtt{V}(a_{l_1}, b_{l_1})$ is a subset of $\mathtt{V}(u_1', v_1') \cap \mathtt{V}(a_{l_1}, b_{l_1})$. Thus, $w \in \mathtt{V}(u_1', v_1') \cap \mathtt{V}(a_{l_1}, b_{l_1})$, and w is an ancestor of the vertex $w_1' := z([u_1', v_1'], [a_{l_1}, b_{l_1}])$ (cf. Observation 250). We know that w_1' is contained in $\mathcal{W}_{l_1+1}(r_1)$, because $[u_1', v_1'] \in \mathrm{C}^P_{l_1}(r_1)$ and $l_1 < k_1$. Let $m := \mathtt{sd}(w)$. If $m \geq l_1 + 1$, then $w \in \mathcal{W}_{l_1+1}(r_1)$ according to Observation 277, and we obtain a contradiction because l_1 is the maximal $l \leq k_1$ with $w \in \mathcal{W}_l(r_1)$. Let $m < l_1 + 1$. Then $\mathcal{W}_{l_1+1}(r_1) \subseteq \mathtt{V}(a_m, b_m)$ (Observation 276) and each ancestor of w_1' of side depth m is a_m or b_m (Corollary 265). Thus, $w \in \{a_m, b_m\}$ is a side ancestor of q, a contradiction. ☐

Lemma 282. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair. Let $r_1, r_1 \in R(q)$. Let $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathcal{W}(r_2)$ be adjacent vertices of* H. *Then $w_2 \in \mathcal{W}(r_1) \cup \mathtt{anc}(r_1)$, $w_1 \in \mathcal{W}(r_2) \cup \mathtt{anc}(r_2)$ or $w_1, w_2 \in \mathtt{anc}(q)$.*

Proof. Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \ldots, d\}$. Let $r_1, r_1 \in R(q)$. Let $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathcal{W}(r_2)$ be adjacent vertices of H. Let $k_i \in \{1, \ldots, d+1\}$ and $[u_i, v_i] \in \mathrm{C}^{\text{con}}_{k_i}(q)$ be such that $r_i := [u_i, v_i]_{a_{k_i-1}, b_{k_i-1}}$ for $i \in [2]$.

Let $j_i \in [0, k_i]$ be maximal such that $w_i \in \mathcal{W}_{j_i}(r_i)$ for $i \in [2]$. Let $i \in [2]$. Let us show that there exist an $[x_i, y_i] \in \mathrm{C}_{j_i}(q)$ such that $w_i \in \mathtt{W}^q_{j_i}(x_i, y_i) \subseteq \mathcal{W}_{j_i}(r_i)$. If $j_i = k_i$, then $w \in \mathtt{W}^q_{j_i}(u_i, v_i) = \mathcal{W}_{j_i}(r_i)$ according to Observation 274, and clearly $[u_i, v_i] \in \mathrm{C}_{j_i}(q)$. Thus, we let $[x_i, y_i] := [u_i, v_i]$ if $j_i = k_i$. Let $j_i < k_i$. Then $w_i \in \mathtt{W}^{r_i}_{j_i}(u_i', v_i') \subseteq \mathcal{W}_{j_i}(r_i)$ for a child $[u_i', v_i'] \in \mathrm{C}^P_{j_i}(r_i)$. According to Observations 269 and 268 we have $\mathrm{C}_{j_i}(r_i) = \mathrm{C}_{j_i}(q)$ and $\mathcal{F}_{j_i}(r_i) = \mathcal{F}_{j_i}(q)$. Consequently, $[u_i', v_i'] \in \mathrm{C}_{j_i}(q)$ and $w_i \in \mathtt{W}^q_{j_i}(u_i', v_i') \subseteq \mathcal{W}_{j_i}(r_i)$. We let $[x_i, y_i] := [u_i', v_i']$ if $j_i < k_i$.

Clearly, if $[x_1, y_1] = [x_2, y_2]$, then $w_2 \in \mathtt{W}^q_{j_1}(x_1, y_1) \subseteq \mathcal{W}_{j_1}(r_1) \subseteq \mathcal{W}(r_1) \cup \mathtt{anc}(r_1)$. Thus, let $[x_1, y_1] \neq [x_2, y_2]$.

First let us consider the case where $j := \boldsymbol{j_1 = j_2}$ and $\boldsymbol{j = d + 1}$. Then $k_1 = k_2 = d + 1$, and $[x_i, y_i] = [u_i, v_i]$ and $[u_i, v_i] \in \mathrm{C}_{d+1}(q)$ for each $i \in [2]$. By Observation 276, $\mathcal{W}_{d+1}(r_i) \subseteq \mathtt{V}^{\text{-}}(a, b)$. Thus $w_1, w_2 \in \mathtt{V}^{\text{-}}(a, b)$. Since $[x_1, y_1] \neq [x_2, y_2]$, $[x_1, y_1]$ and $[x_2, y_2]$ are distinct roots of the decomposition forest $\mathcal{F}_{d+1}(q)$. It follows from Observation 234 and Lemma 184 that $\mathtt{W}^q_j(x_1, y_1)$ and $\mathtt{W}^q_j(x_2, y_2)$ are two connected components of $\mathtt{H}^{\text{-}}(a, b)$. Thus $w_1 \in \mathtt{W}^q_j(x_1, y_1)$ and $w_2 \in \mathtt{W}^q_j(x_2, y_2)$ cannot be connected, a contradiction.

Next let us consider the case where $j := \boldsymbol{j_1 = j_2}$ and $\boldsymbol{j \leq d}$. As $[x_1, y_1]$ and $[x_2, y_2]$ are children of $[a_j, b_j]$ in $\mathcal{F}_j(q)$, Lemmas 237 and 184 imply that $w_2 \in \mathtt{W}^q_j(x_1, y_1)$, $w_1 \in \mathtt{W}^q_j(x_2, y_2)$ or $w_1, w_2 \in \mathtt{V}(a_j, b_j)$. Hence, $w_2 \in \mathcal{W}_{j_1}(r_1) \subseteq \mathcal{W}(r_1)$, $w_1 \in \mathcal{W}_{j_2}(r_2) \subseteq \mathcal{W}(r_1)$ or $w_1, w_2 \in \mathtt{V}(a_j, b_j)$. If $w_2 \in \mathcal{W}(r_1)$ or $w_1 \in \mathcal{W}(r_2)$, then we are done. Thus, let $w_1, w_2 \in \mathtt{V}(a_j, b_j)$. Since $w_i \in \mathtt{W}^q_j(x_i, y_i)$, it follows from Lemmas 232 and 184 that $w_i \in \mathtt{V}(x_i, y_i) \cap \mathtt{V}(a_j, b_j)$ for all $i \in [2]$. According to Observation 250, w_i is an ancestor of the vertex $z_i := z([x_i, y_i], [a_j, b_j])$.

Let $i \in [2]$ and let $j < k_i$. Then $[x_i, y_i] = [u'_i, v'_i]$ where $[u'_i, v'_i] \in \mathrm{C}^P_j(r_i)$. It follows that z_i is in $\mathcal{W}_{j+1}(r_i)$. As w_i is an ancestor of z_i and $w_i \notin \mathcal{W}_{j+1}(r_i)$ according to the choice of j_i, Observation 277 implies that $\mathtt{sd}(w_i) \leq j$. Since $w_i \in \mathtt{V}(a_j, b_j)$, it follows from Observation 239 and Corollary 265 that w_i is a_j or b_j. Hence, $w_i \in \mathtt{anc}(q)$.

Let $i \in [2]$ and let $j = k_i$. Then $[x_i, y_i] = [u_i, v_i]$. Since $[u_i, v_i] \in \mathrm{C}^{\mathrm{con}}_{k_i}(q)$ and $k_i = j \leq d$, the vertex z_i is a or b. It follows that w_i is an ancestor of a or b. Thus, $w_i \in \mathtt{anc}(q)$.

It follows that $w_1, w_2 \in \mathtt{anc}(q)$.

Now let us consider the case where $j_1 \neq j_2$. Without loss of generality, let $\boldsymbol{j_1 < j_2}$. Then $\boldsymbol{j_1 \leq d}$. According to Observation 276, $\mathcal{W}_{j_2}(r_2) \subseteq \mathtt{V}(a_{j_1}, b_{j_1})$. Thus, $w_2 \in \mathtt{V}(a_{j_1}, b_{j_1})$. Since $w_1 \in \mathtt{W}^q_{j_1}(x_1, y_1)$, we have $w_2 \in \mathtt{W}^q_{j_1}(x_1, y_1)$ or $w_1 \in \mathtt{V}(a_{j_1}, b_{j_1})$ according to Lemmas 236 and 184. Hence, $w_2 \in \mathcal{W}_{j_1}(r_1) \subseteq \mathcal{W}(r_1)$ or $w_1 \in \mathtt{V}(a_{j_1}, b_{j_1})$. If $w_2 \in \mathcal{W}(r_1)$, we are done. Thus, let $w_1 \in \mathtt{V}(a_{j_1}, b_{j_1})$. Since $w_1 \in \mathtt{W}^q_{j_1}(x_1, y_1)$, it follows from Lemmas 232 and 184 that $w_1 \in \mathtt{V}(x_1, y_1) \cap \mathtt{V}(a_{j_1}, b_{j_1})$. According to Observation 250, w_1 is an ancestor of the vertex $z_1 := z([x_1, y_1], [a_{j_1}, b_{j_1}])$.

Let $j_1 < k_1$. Then similar to the corresponding case for $j = j_1 = j_2$ and $j \leq d$, we can show that w_1 is a_{j_1} or b_{j_1}. Since $j_1 < j_2 \leq k_2$, we have $w_1 \in \mathtt{anc}(r_2)$.

Let $j_1 = k_1$. Then similar to the corresponding case for $j = j_1 = j_2$ and $j \leq d$, we obtain that w_1 is an ancestor of a or b. If $w_1 \in \mathtt{anc}(r_2)$, we are done. Thus, let $w_1 \in \{a_d, \ldots, a_{k_2}\} \cup \{b_d, \ldots, b_{k_2}\}$. Then $k_2 \leq d$. Hence, $j_2 \leq k_2 \leq d$. According to Observation 267, $w_1 \in \mathtt{V}(a_{j_2}, b_{j_2})$. Since $w_2 \in \mathtt{W}^q_{j_2}(x_2, y_2)$, it follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$ or $w_1 \in \mathtt{W}^q_{j_2}(x_2, y_2)$. Consequently, $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$ or $w_1 \in \mathcal{W}_{j_2}(r_2) \subseteq \mathcal{W}(r_2)$. If $w_1 \in \mathcal{W}(r_2)$, we are done. Therefore, let $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$. Since $w_2 \in \mathtt{W}^q_{j_2}(x_2, y_2)$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(x_2, y_2) \cap \mathtt{V}(a_{j_2}, b_{j_2})$. According to Observation 250, w_2 is an ancestor of the vertex $z_2 = z([x_2, y_2], [a_{j_2}, b_{j_2}])$.

Let $j_2 < k_2$. Then similar to the corresponding case for $j = j_1 = j_2$ and $j \leq d$, we can show that w_2 is a_{j_2} or b_{j_2}. Hence, $w_1, w_2 \in \mathtt{anc}(q)$.

Let $j_2 = k_2$. Then similar to the corresponding case for $j = j_1 = j_2$ and $j \leq d$, we obtain that w_2 is an ancestor of a or b. Then $w_1, w_2 \in \mathtt{anc}(q)$. ☐

6.6.4. Decomposing $\mathcal{W}(q)$ for Trivial Affiliated Subbundle Pairs q

In the following let us consider affiliated subbundle pairs $q \in V_{\mathcal{T}_{\text{gen}}}$ that are trivial. For a trivial affiliated subbundle pair q let us define subsets $\mathrm{C}_j^{\text{triv}}(q)$ of $\mathrm{C}_j^P(q)$ for all $j \in \{0,\dots,d\}$. For all $j \in \{0,\dots,d\}$ we let $\mathrm{C}_j^{\text{triv}}(q)$ be the set of all subbundle pairs $[u,v] \in \mathrm{C}_j^P(q)$ where $z([u,v],[a_j,b_j])$ is in $\mathtt{V}(a,b)$. For $j \in [d]$ we let $R_j(q)$ be the set of all $[u,v]_{a_{j-1},b_{j-1}}$ where $[u,v] \in \mathrm{C}_j^{\text{triv}}(q)$. We let $R(q) := \bigcup_{j\in[d]} R_j(q)$.

Lemma 283. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair. Then $\mathcal{W}(q)$ is the union U of the set $\mathtt{V}(a,b)$ and of all sets $\mathcal{W}(r)$ where $r \in R(q)$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d with ancestors a_i, b_i for $i \in \{0,\dots,d\}$. According to Observation 272, we have $\mathtt{V}(a,b) \subseteq \mathcal{W}(q)$. By Lemma 279 it follows that U is a subset of $\mathcal{W}(q)$. We now prove inductively that U contains all vertices in $\mathcal{W}(q)$. For all $j \in \{0,\dots,d\}$ let U_j be the union of $\mathtt{V}(a,b)$ and of all sets $\mathcal{W}_j([u,v]_{a_{k-1},b_{k-1}})$ where $k \in \{j,\dots,d\} \setminus \{0\}$ and $[u,v] \in \mathrm{C}_k^{\text{triv}}(q)$. Then $\bigcup_{j\le d} U_j = U$. We use induction to show that $\mathcal{W}_j(q) \subseteq U_j$ for all $j \in \{0,\dots,d\}$.

Let us show that $\mathcal{W}_d(q) = \mathtt{W}_d^q(a,b) \subseteq U_d$. Clearly, we have $\mathtt{V}(a,b) \subseteq U_d$. Let us show that $\mathtt{W}_d^q(u,v) \subseteq U_d$ for all children $[u,v]$ of $[a,b]$ in $\mathcal{F}_d(q)$, that is, for all $[u,v] \in \mathrm{C}_d^P(q)$. As $\mathrm{C}_0^P(q) = \emptyset$, there is nothing to show for $d = 0$. Let $d > 0$. For all $[u,v] \in \mathrm{C}_d^P(q)$ the vertex $z([u,v],[a,b])$ is in $\mathtt{V}(a,b)$. Thus, $[u,v] \in \mathrm{C}_d^{\text{triv}}(q)$ and $\mathcal{W}_d([u,v]_{a_{d-1},b_{d-1}}) \subseteq U_d$ according to the definition of U_d. Since $\mathtt{W}_d^q(u,v) = \mathcal{W}_d([u,v]_{a_{d-1},b_{d-1}})$ (Observation 274), the set $\mathtt{W}_d^q(u,v)$ is in U_d for all $[u,v] \in \mathrm{C}_d^P(q)$.

Next let us assume $\mathcal{W}_{j+1}(q) \subseteq U_{j+1}$ for $j \in \{0,\dots,d-1\}$. We show that $\mathcal{W}_j(q) \subseteq U_j$. In order to do this, we have to show that $\mathtt{W}_j^q(x,y) \subseteq U_j$ for all $[x,y] \in \mathrm{C}_j^P(q)$. As $\mathrm{C}_0^P(q) = \emptyset$, this clearly is the case for $j = 0$. Thus, let $j > 0$. Let $[x,y] \in \mathrm{C}_j^P(q)$ and let $z := z([x,y],[a_j,b_j])$. As z is in $\mathcal{W}_{j+1}(q)$, it also is in U_{j+1} by inductive assumption.

If z is in $\mathtt{V}(a,b)$, then $[x,y] \in \mathrm{C}_j^{\text{triv}}(q)$ and $\mathcal{W}_j([x,y]_{a_{j-1},b_{j-1}})$ is a subset of U_j. Hence, $\mathtt{W}_j^q(x,y) = \mathcal{W}_j([x,y]_{a_{j-1},b_{j-1}})$ (Observation 274) is a subset of U_j. Thus, let z be not in $\mathtt{V}(a,b)$ for the rest of this proof.

Let z be in $\mathcal{W}_{j+1}(r)$ for $r = [u,v]_{a_{k-1},b_{k-1}}$ with $k \in \{j+1,\dots,d\}$ and $[u,v] \in \mathrm{C}_k^{\text{triv}}(q) \subseteq \mathrm{C}_k^P(q)$. Thus, $z([u,v],[a_k,b_k])$ is in $\mathtt{V}(a,b)$. In the following we show that $[x,y] \in \mathrm{C}_j^P(r)$. According to Observation 269, we have $\mathrm{C}_j(q) = \mathrm{C}_j(r)$, and therefore $[x,y] \in \mathrm{C}_j(r)$. Further, $z \in \mathcal{W}_{j+1}(r)$. Hence, it remains to show that z is not an ancestor of any vertex in $P_j(r)$. As $[x,y] \in \mathrm{C}_j^P(q)$, vertex z is not an ancestor of any vertex in $P_j(q)$. By Observation 271, we have

$$P_j(r) \setminus \{a_{k-1}, b_{k-1}, p_k(r)\} = P_j(q) \setminus (\{a_{d-1}, b_{d-1}\} \cup \{p_l(q) \mid k \le l \le d,\, l \in J(q)\}).$$

Clearly, z is not an ancestor of any vertex in $P_j(r) \setminus \{a_{k-1}, b_{k-1}, p_k(r)\}$. As z is not an ancestor of a_{d-1} or b_{d-1}, and $k \le d$, it follows that z is not an ancestor of a_{k-1} or b_{k-1}. The vertex $p_k(r) = z([u,v],[a_k,b_k])$ is in $\mathtt{V}(a,b)$. We assumed that z is not in $\mathtt{V}(a,b)$. As z is not an ancestor of a_{d-1} or b_{d-1}, Observation 245 implies that z is not an ancestor of $p_k(r) = z([u,v],[a_k,b_k])$. Consequently, vertex z is not an ancestor of any vertex in $P_j(r)$. Hence, $[x,y] \in \mathrm{C}_j^P(r)$. It follows that $\mathtt{W}_j^r(x,y) \subseteq \mathcal{W}_j(r)$ is a subset of U_j. By Observation 268 we have $\mathcal{F}_j(q) = \mathcal{F}_j(r)$. Thus, $\mathtt{W}_j^q(x,y) = \mathtt{W}_j^r(x,y)$. As a consequence, $\mathtt{W}_j^q(x,y) \subseteq U_j$. □

Observation 284. *Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $j \in \{1, \dots, d\}$, and let $[u, v] \in \mathrm{C}_j^{\text{triv}}(q)$. Further, let $r := [u, v]_{a_{j-1}, b_{j-1}}$. Then $\mathcal{W}(r) \cap \mathtt{V}(a, b)$ does not contain any proper descendants of $z([u, v], [a_j, b_j]) \in \mathtt{V}(a, b)$.*

Proof. Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $j \in \{1, \dots, d\}$, and let $[u, v] \in \mathrm{C}_j^{\text{triv}}(q)$. Further, let $r := [u, v]_{a_{j-1}, b_{j-1}}$. Let us assume there exists a proper descendant $c \in \mathtt{V}(a, b)$ of $z := z([u, v], [a_j, b_j]) \in \mathtt{V}(a, b)$ in $\mathcal{W}(r)$. Suppose l is maximal with $c \in \mathcal{W}_l(r)$.

Let us consider the case where $l = j$. As $c \in \mathtt{V}(a, b)$, we have $c \in \mathtt{V}(a_j, b_j)$ by Observation 267. According to Lemma 278, it holds that $\mathcal{W}(r) \cap \mathtt{V}(a_j, b_j) \subseteq \mathtt{V}(u, v) \cap \mathtt{V}(a_j, b_j)$. Hence, $c \in \mathtt{V}(u, v) \cap \mathtt{V}(a_j, b_j)$, and by Observation 250 vertex c must be an ancestor of $z = z([u, v], [a_j, b_j])$, a contradiction.

Now let us consider the case where $l < j$. Then there is a subbundle pair $[x, y] \in \mathrm{C}_l^P(r)$ such that $c \in \mathtt{W}_l^r(x, y)$. We have $z' := z([x, y], [a_l, b_l]) \in \mathcal{W}_{l+1}(r)$. As $c \in \mathtt{V}(a, b)$, it follows that $c \in \mathtt{V}(a_l, b_l)$ by Observation 267. According to Lemmas 232 and 184, it holds that $\mathtt{W}_l^r(x, y) \cap \mathtt{V}(a_l, b_l) \subseteq \mathtt{V}(x, y) \cap \mathtt{V}(a_l, b_l)$. Thus, $c \in \mathtt{V}(x, y) \cap \mathtt{V}(a_l, b_l)$. By Observation 250, vertex c is an ancestor of $z' = z([x, y], [a_l, b_l])$. As $c \in V(a, b)$, we have $\mathtt{sd}(c) \geq d$ according to Observation 239. Since $z' \in \mathcal{W}_{l+1}(r)$ we obtain $c \in \mathcal{W}_{l+1}(r)$ by Observation 277, a contradiction to the choice of l. ☐

Lemma 285. *Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair. Let $r_1, r_2 \in R(q)$ with $r_1 \neq r_2$. Then $\mathcal{W}(r_1) \cap \mathcal{W}(r_2) \subseteq \mathtt{anc}(q) \cup \mathtt{V}(a, b)$.*

Proof. Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $r_1, r_2 \in R(q)$ with $r_1 \neq r_2$. Let $k_1, k_2 \in \{1, \dots, d\}$ and let $[u_1, v_1] \in \mathrm{C}_{k_1}^{\text{triv}}(q)$ and $[u_2, v_2] \in \mathrm{C}_{k_2}^{\text{triv}}(q)$ be such that $r_1 := [u_1, v_1]_{a_{k_1-1}, b_{k_1-1}}$ and $r_2 := [u_2, v_2]_{a_{k_2-1}, b_{k_2-1}}$.

Let us assume there exists a vertex w such that $w \in \mathcal{W}(r_1) \cap \mathcal{W}(r_2)$ and $w \notin \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. For each $w \in (\mathcal{W}(r_1) \cap \mathcal{W}(r_2)) \setminus (\mathtt{anc}(q) \cup \mathtt{V}(a, b))$ let $l_j(w)$ be the maximal $l \leq k_j$ such that $w \in \mathcal{W}_l(r_j)$. Let $w \in (\mathcal{W}(r_1) \cap \mathcal{W}(r_2)) \setminus (\mathtt{anc}(q) \cup \mathtt{V}(a, b))$ be such that for all $w' \in (\mathcal{W}(r_1) \cap \mathcal{W}(r_2)) \setminus (\mathtt{anc}(q) \cup \mathtt{V}(a, b))$ we have

- $l_j(w') \leq l_j(w)$ for each $j \in \{1, 2\}$ or
- there exists a $j' \in \{1, 2\}$ such that $l_j(w') < l_j(w)$.

From now on we denote $l_1(w)$ and $l_2(w)$ by l_1 and l_2, respectively. Without loss of generality, let $l_1 \leq l_2$, and if $l_1 = l_2$, then let $k_1 \leq k_2$.

We check multiple cases. The first case is that $l_1 = k_1$ and $l_1 < l_2$. We know that $w \in \mathcal{W}_{l_1}(r_1)$ and $w \in \mathcal{W}_{l_2}(r_2)$. As $l_1 < l_2 \leq k_2$, Observations 266 and 276 imply that $\mathcal{W}_{l_2}(r_2) \subseteq \mathtt{V}(a_{l_1}, b_{l_1})$. We obtain that $w \in \mathtt{V}(a_{l_1}, b_{l_1})$. Consequently, w is contained in $\mathcal{W}_{l_1}(r_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$. As $l_1 \leq d$, we have $\mathcal{W}(r_1) \cap \mathtt{V}(a_{l_1}, b_{l_1}) \subseteq \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$ according to Lemma 278. Thus, w is contained in $\mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1}) = \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. Since $[u_1, v_1] \in \mathrm{C}_{k_1}^{\text{triv}}(q)$, the vertex $z_1 := z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is in $\mathtt{V}(a, b)$. It follows from Observation 250 that $z_1 \notin \mathtt{O}$ and that each vertex in $\mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$ is an ancestor of $z_1 \in V(a, b)$. As $[a, b]$ is trivial, Observation 245 implies that each ancestor of z_1 is either in $\mathtt{V}(a, b)$ or an ancestor of a or b. Hence, $w \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$, a contradiction.

Next, let $l_1 = k_1$ and $l_1 = l_2$. As $l_1 = k_1$ and $w \in \mathcal{W}_{l_1}(r_1)$, we have $w \in \mathtt{W}_{l_1}^q(u_1, v_1)$ according to Observation 274. Subbundle pair $[u_1, v_1]$ is a child of $[a_{l_1}, b_{l_1}]$ in $\mathcal{F}_{l_1}(q)$. Further, $w \in \mathcal{W}_{l_2}(r_2)$ and $w \notin \mathcal{W}_{l_2+1}(r_2)$. Let us show that there is a subbundle pair $[x, y] \in \mathrm{C}_{l_1}^P(q)$ with $[x, y] \neq [u_1, v_1]$ and $w \in \mathtt{W}_{l_1}^q(x, y)$. Let $l_2 = k_2$. Then $[u_2, v_2] \in \mathrm{C}_{l_1}^P(q)$ and $w \in \mathcal{W}_{l_2}(r_2) = \mathtt{W}_{l_1}^q(u_2, v_2)$ (Observation 274). As $r_1 \neq r_1$ and r_1 and r_2 have the same affiliations, it follows that $[u_1, v_1] \neq [u_2, v_2]$. Thus, we let $[x, y] := [u_2, v_2]$ in this case. Let $l_2 < k_2$. Then there exists a subbundle pair $[u_2', v_2']$ in $\mathrm{C}_{l_2}^P(r_2)$ such that $w \in \mathtt{W}_{l_2}^{r_2}(u_2', v_2')$. By Observations 269 and 268, we have $\mathrm{C}_{l_2}(r_2) = \mathrm{C}_{l_2}(q)$ and $\mathcal{F}_{l_2}(r_2) = \mathcal{F}_{l_2}(q)$. Hence, $[u_2', v_2']$ in $\mathrm{C}_{l_2}^P(q)$ and $\mathtt{W}_{l_2}^{r_2}(u_2', v_2') = \mathtt{W}_{l_2}^q(u_2', v_2')$. Since $l_1 = l_2$, we have $[u_2', v_2']$ in $\mathrm{C}_{l_1}^P(q)$ and $w \in \mathtt{W}_{l_1}^q(u_2', v_2')$. Further, $[u_2', v_2'] \neq [u_1, v_1]$ by Claim 1. Thus, we let $[x, y] := [u_2', v_2']$. Then subbundle pairs $[u_1, v_1]$ and $[x, y]$ are children of $[a_{l_1}, b_{l_1}]$ in $\mathcal{F}_{l_1}(q)$. By Lemmas 233 and 184, we have $\mathtt{W}_{l_1}^q(u_1, v_1) \cap \mathtt{W}_{l_1}^q(x, y) \subseteq \mathtt{V}(a_{l_1}, b_{l_1})$. Therefore, w is contained in $\mathtt{W}_{l_1}^q(u_1, v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$, which is a subset of $\mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{l_1}, b_{l_1})$ according to Lemmas 232 and 184. Since $l_1 = k_1$, we have $w \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. As $[u_1, v_1] \in \mathrm{C}_{k_1}^{\mathrm{triv}}(q)$, the vertex $z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is in $V(a, b)$. According to Observation 250, w is an ancestor of $z([u_1, v_1], [a_{k_1}, b_{k_1}]) \notin \mathtt{O}$. Since $[a, b]$ is trivial, Observation 245 implies that w is in $\mathtt{anc}(q) \cup \mathtt{V}(a, b)$, a contradiction.

Claim 1. Let $l_1 = k_1$, $l_1 = l_2$ and $l_2 < k_2$. Then $[u_1, v_1] \neq [u_2', v_2']$.

Proof. Let $l_1 = k_1$, $l_1 = l_2$ and $l_2 < k_2$. Assume $[u_1, v_1] = [u_2', v_2']$. We know that $z_1 := z([u_1, v_1], [a_{k_1}, b_{k_1}])$ and $z_2 := z([u_2, v_2], [a_{k_2}, b_{k_2}])$ are in $\mathtt{V}(a, b)$. As $[u_2', v_2'] \in \mathrm{C}_{l_2}^P(r_2)$, that is, $[u_1, v_1] \in \mathrm{C}_{k_1}^P(r_2)$, the vertex $z_1 = z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is not an ancestor of any vertex in $P_{k_1}(r_2)$. Hence, the vertex z_1 is not an ancestor of $z_2 = p_{k_2}(r_2)$. Since $[a, b]$ is trivial and $z_1, z_2 \in \mathtt{V}(a, b)$, Observation 170 implies that the vertex z_1 is a proper descendant of z_2. It follows from $[u_1, v_1] = [u_2', v_2']$, that vertex z_1 is in $\mathtt{V}(u_2', v_2') \subseteq \mathtt{W}_{l_2}^{r_2}(u_2', v_2') \subseteq \mathcal{W}(r_2)$. According to Observation 284, the set $\mathcal{W}(r_2) \cap \mathtt{V}(a, b)$ does not contain any proper descendants of $z_2 = z([u_2, v_2], [a_{k_2}, b_{k_2}])$. Since $z_1 \in \mathcal{W}(r_2) \cap \mathtt{V}(a, b)$ is a proper descendant of z_2, we obtain a contradiction. ⌟

Finally, let $l_1 < k_1$ (and $l_1 \leq l_2$). The proof of this case is identical to the proof of the same case in Lemma 281. Only Claim 2 needs to be proved differently.

Claim 2. Let $l_1 = l_2$, $l_1 < k_1$ and $l_2 < k_2$. Let $[u_1', v_1'] \in \mathrm{C}_{l_1}^P(r_1)$ be such that $w \in \mathtt{W}_{l_1}^{r_1}(u_1', v_1')$, and $[u_2', v_2'] \in \mathrm{C}_{l_1}^P(r_2)$ be such that $w \in \mathtt{W}_{l_1}^{r_2}(u_2', v_2')$. Then $[u_1', v_1'] \neq [u_2', v_2']$.

Proof. Let $l_1 = l_2$, $l_1 < k_1$ and $l_2 < k_2$. Let $[u_1', v_1'] \in \mathrm{C}_{l_1}^P(r_1)$ be such that $w \in \mathtt{W}_{l_1}^{r_1}(u_1', v_1')$, and $[u_2', v_2'] \in \mathrm{C}_{l_1}^P(r_2)$ be such that $w \in \mathtt{W}_{l_2}^{r_2}(u_2', v_2')$. Assume $[u_1', v_1'] = [u_2', v_2']$. Let $z := z([u_1', v_1'], [a_{l_1}, b_{l_1}])$. As $[u_1', v_1'] \in \mathrm{C}_{l_1}^P(r_1)$ and $[u_2', v_2'] \in \mathrm{C}_{l_1}^P(r_2)$, we have $z \in \mathcal{W}_{l_1+1}(r_1)$ and $z \in \mathcal{W}_{l_1+1}(r_2)$. Thus, $l_1(z) = l_2(z) > l_1 = l_2$. It follows from the choice of w that $z \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. Note that z is not an ancestor of any vertex in $P_{l_1}(r_1)$.

According to Observation 170 there is an $f \in \mathtt{F}$ such that $V(a, b)$ is the vertex set of a directed path in $\mathtt{T}_f$. Without loss of generality, let $f = f_1$ and let a be the first vertex of this path and b the last one. Then it follows from Observation 245 that $a_d, \ldots, a_0$ are the ancestors of a in $\mathtt{T}_{f_1}$. As $[u_1, v_1] \in \mathrm{C}_{k_1}^{\mathrm{triv}}(q)$, the vertex $z_1 := z([u_1, v_1], [a_{k_1}, b_{k_1}])$ is in $V(a, b)$. According to Observation 250, $z_1 \notin \mathtt{O}$. Thus, $z \in \mathtt{S}_{f_1}$. Since $z_1 = p_{k_1}(r_1)$, we have $z_1 \in P_{l_1}(r_1)$. Further $a_{k_1-1}, b_{k_1-1} \in P_{l_1}(r_1)$. Consequently, vertex z must be either a proper descendant of z_1 and in $V(a,b)$, or an ancestor of b_{d-1} and a descendant of b_{k_1}.

First let us assume z is an ancestor of b_{d-1} and a descendant of b_{k_1}. As $[a_{d-1}, b_{d-1}]$ is non-minimal consistent, this implies that $k_1 < d$ and $z \in \mathtt{S}_{f_2}$. By Observation 267, $z \in \mathtt{V}(a_{k_1}, b_{k_1})$. According to Lemma 278, $\mathcal{W}(r_1) \cap \mathtt{V}(a_{k_1}, b_{k_1}) \subseteq \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. Since $z \in \mathcal{W}_{l_1+1}(r_1) \subseteq \mathcal{W}(r_1)$, it follows that $z \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. Consequently, $z_1, z \in \mathtt{V}(u_1, v_1) \cap \mathtt{V}(a_{k_1}, b_{k_1})$. As $z_1 \in \mathtt{S}_{f_1}$ and $z \in \mathtt{S}_{f_2}$, we obtain a contradiction according to Observation 250.

Now let us assume z is a proper descendant of z_1 and in $V(a,b)$. We know that z is in $\mathcal{W}_{l_1+1}(r_1) \subseteq \mathcal{W}(r_1)$. According to Observation 284, the set $\mathcal{W}(r_1) \cap \mathtt{V}(a,b)$ does not contain any proper descendants of $z_1 = z([u_1, v_1], [a_{k_1}, b_{k_1}])$. Since $z \in \mathcal{W}(r_1) \cap \mathtt{V}(a,b)$ is a proper descendant of z_1, we obtain a contradiction. ⌟

□

Observation 286. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d and with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $k \in [d]$ and $r \in R_k(q)$. Let $j < k$ and $[x,y] \in \mathrm{C}_j^P(r)$. Let $w \in \mathtt{V}(x,y) \cap \mathtt{V}(a_j, b_j)$ and $w \notin \mathcal{W}_{j+1}(r)$. Then $w \in \{a_j, b_j\}$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d and with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $k \in [d]$ and $r \in R_k(q)$. Let $j < k$ and $[x,y] \in \mathrm{C}_j^P(r)$. Let $w \in \mathtt{V}(x,y) \cap \mathtt{V}(a_j, b_j)$ and $w \notin \mathcal{W}_{j+1}(r)$. As $j < k$ and $[x,y] \in \mathrm{C}_j^P(r)$, the vertex $z := z([x,y],[a_j,b_j])$ is in $\mathcal{W}_{j+1}(r)$. According to Observation 250, w is an ancestor of z. Since $w \notin \mathcal{W}_{j+1}(r)$, Observation 277 implies that $\mathtt{sd}(w) \leq j$. As $w \in \mathtt{V}(a_j, b_j)$, we have $\mathtt{sd}(w) = j$ by Observation 239.

If $j < d$, then $[a_j, b_j]$ is consistent and it follows from Observation 239 and Corollary 265 that w is a_j or b_j. If $j = d$, then $w \in \mathtt{V}(a,b)$. According to Observation 245, $V(a,b)$ is the vertex set of a directed path with ends a and b in $\mathtt{T}_f$. Without loss of generality, let a be the first vertex of the path and $f = f_1$ Then Observation 245 also implies that $\mathtt{sd}_{f_1}(a) = d$ and $\mathtt{sd}_{f_1}(v) > d$ for all $v \in \mathtt{V}(a,b) \setminus \{a\}$. Since vertex b is the only vertex that might be in $\mathtt{V}_{f_2}$, it follows that w is a or b. Hence, $w \in \{a_j, b_j\}$. □

Observation 287. *Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d and with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $k \in [d]$ and $r = [u,v]_{a_{k-1}, b_{k-1}} \in R_k(q)$. Let $w \in \mathtt{V}(u,v) \cap \mathtt{V}(a_k, b_k)$. Then $w \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d and with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $k \in [d]$ and $r = [u,v]_{a_{k-1}, b_{k-1}} \in R_k(q)$. Let $w \in \mathtt{V}(u,v) \cap \mathtt{V}(a_k, b_k)$. Since $[u,v] \in \mathrm{C}_k^{\text{triv}}(q)$, the vertex $z := z([u,v],[a_k,b_k])$ is in $\mathtt{V}(a,b)$. According to Observation 250, vertex w is an ancestor of z. Observation 245 implies that $w \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$. □

Lemma 288. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair. Let $r_1, r_1 \in R(q)$. Let $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathcal{W}(r_2)$ be adjacent vertices of $\mathtt{H}$. Then $w_2 \in \mathcal{W}(r_1) \cup \mathtt{anc}(r_1)$, $w_1 \in \mathcal{W}(r_2) \cup \mathtt{anc}(r_2)$ or $w_1, w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$.*

Proof. Let $q = [a,b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $r_1, r_1 \in R(q)$. Let $w_1 \in \mathcal{W}(r_1)$ and

$w_2 \in \mathcal{W}(r_2)$ be adjacent vertices of $\mathtt{H}$. Let $k_i \in \{1, \ldots, d\}$ and $[u_i, v_i] \in \mathrm{C}^{\mathrm{triv}}_{k_i}(q)$ be such that $r_i := [u_i, v_i]_{a_{k_i - 1}, b_{k_i - 1}}$ for $i \in [2]$.

Let $j_i \in [0, k_i]$ be maximal such that $w_i \in \mathcal{W}_{j_i}(r_i)$ for $i \in [2]$. Let $i \in [2]$. Let us show that there exist an $[x_i, y_i] \in \mathrm{C}_{j_i}(q)$ such that $w_i \in \mathtt{W}^q_{j_i}(x_i, y_i) \subseteq \mathcal{W}_{j_i}(r_i)$. If $j_i = k_i$, then $w \in \mathtt{W}^q_{j_i}(u_i, v_i) = \mathcal{W}_{j_i}(r_i)$ according to Observation 274, and clearly $[u_i, v_i] \in \mathrm{C}_{j_i}(q)$. Thus, we let $[x_i, y_i] := [u_i, v_i]$ if $j_i = k_i$. Let $j_i < k_i$. Then $w_i \in \mathtt{W}^{r_i}_{j_i}(u'_i, v'_i) \subseteq \mathcal{W}_{j_i}(r_i)$ for a child $[u'_i, v'_i] \in \mathrm{C}^P_{j_i}(r_i)$. According to Observations 269 and 268 we have $\mathrm{C}_{j_i}(r_i) = \mathrm{C}_{j_i}(q)$ and $\mathcal{F}_{j_i}(r_i) = \mathcal{F}_{j_i}(q)$. Consequently, $[u'_i, v'_i] \in \mathrm{C}_{j_i}(q)$ and $w_i \in \mathtt{W}^q_{j_i}(u'_i, v'_i) \subseteq \mathcal{W}_{j_i}(r_i)$. We let $[x_i, y_i] := [u'_i, v'_i]$ if $j_i < k_i$.

Clearly, if $[x_1, y_1] = [x_2, y_2]$, then $w_2 \in \mathtt{W}^q_{j_1}(x_1, y_1) \subseteq \mathcal{W}_{j_1}(r_1) \subseteq \mathcal{W}(r_1) \cup \mathtt{anc}(r_1)$. Thus, let $[x_1, y_1] \neq [x_2, y_2]$.

First let us consider the case where $j := j_1 = j_2$. As $[x_1, y_1]$ and $[x_2, y_2]$ are children of $[a_j, b_j]$ in $\mathcal{F}_j(q)$, Lemmas 237 and 184 imply that $w_2 \in \mathtt{W}^q_j(x_1, y_1)$, $w_1 \in \mathtt{W}^q_j(x_2, y_2)$ or $w_1, w_2 \in \mathtt{V}(a_j, b_j)$. Consequently, $w_2 \in \mathcal{W}_{j_1}(r_1) \subseteq \mathcal{W}(r_1)$, $w_1 \in \mathcal{W}_{j_2}(r_2) \subseteq \mathcal{W}(r_1)$ or $w_1, w_2 \in \mathtt{V}(a_j, b_j)$. If $w_2 \in \mathcal{W}(r_1)$ or $w_1 \in \mathcal{W}(r_2)$, then we are done. Thus, let $w_1, w_2 \in \mathtt{V}(a_j, b_j)$. Since $w_i \in \mathtt{W}^q_j(x_i, y_i)$, it follows from Lemmas 232 and 184 that $w_i \in \mathtt{V}(x_i, y_i) \cap \mathtt{V}(a_j, b_j)$ for all $i \in [2]$.

Let $i \in [2]$ and $j < k_i$. Then $[x_i, y_i] = [u'_i, v'_i]$ where $[u'_i, v'_i] \in \mathrm{C}^P_j(r_i)$. According to the choice of j_i, we have $w_i \notin \mathcal{W}_{j+1}(r_i)$. It follows from Observation 286 that $w_i \in \{a_j, b_j\}$. Consequently, $w_i \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. If $i \in [2]$ and $j = k_i$, then $[x_i, y_i] = [u_i, v_i]$, and it follows from Observation 287 that $w_i \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. It follows that $w_1, w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$.

Now let us consider the case where $j_1 \neq j_2$. Without loss of generality, let $j_1 < j_2$. According to Observation 276, $\mathcal{W}_{j_2}(r_2) \subseteq \mathtt{V}(a_{j_1}, b_{j_1})$. Thus, $w_2 \in \mathtt{V}(a_{j_1}, b_{j_1})$. Since $w_1 \in \mathtt{W}^q_{j_1}(x_1, y_1)$, we have $w_2 \in \mathtt{W}^q_{j_1}(x_1, y_1)$ or $w_1 \in \mathtt{V}(a_{j_1}, b_{j_1})$ according to Lemmas 236 and 184. Hence, $w_2 \in \mathcal{W}_{j_1}(r_1) \subseteq \mathcal{W}(r_1)$ or $w_1 \in \mathtt{V}(a_{j_1}, b_{j_1})$. If $w_2 \in \mathcal{W}(r_1)$, we are done. Thus, let $w_1 \in \mathtt{V}(a_{j_1}, b_{j_1})$. Since $w_1 \in \mathtt{W}^q_{j_1}(x_1, y_1)$, it follows from Lemmas 232 and 184 that $w_1 \in \mathtt{V}(x_1, y_1) \cap \mathtt{V}(a_{j_1}, b_{j_1})$.

Let $j_1 < k_1$. Then $[x_1, y_1] = [u'_1, v'_1]$ where $[u'_1, v'_1] \in \mathrm{C}^P_{j_1}(r_1)$. According to the choice of j_1, we have $w_1 \notin \mathcal{W}_{j_1+1}(r_1)$. Observation 286 implies that $w_1 \in \{a_{j_1}, b_{j_1}\}$. Since $j_1 < j_2 \leq k_2$, we have $w_1 \in \mathtt{anc}(r_2)$.

Let $j_1 = k_1$. Then $[x_1, y_1] = [u_1, v_1]$ and Observation 287 yields that $w_1 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. If $w_1 \in \mathtt{anc}(r_2)$, we are done. Thus, let $w_1 \in \mathtt{V}(a, b) \cup \{a_{d-1}, \ldots, a_{k_2}\} \cup \{b_{d-1}, \ldots, b_{k_2}\}$. Clearly, $j_2 \leq k_2$. According to Observation 267, $w_1 \in \mathtt{V}(a_{j_2}, b_{j_2})$. Since $w_2 \in \mathtt{W}^q_{j_2}(x_2, y_2)$, it follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$ or $w_1 \in \mathtt{W}^q_{j_2}(x_2, y_2)$. Consequently, $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$ or $w_1 \in \mathcal{W}_{j_2}(r_2) \subseteq \mathcal{W}(r_2)$. If $w_1 \in \mathcal{W}(r_2)$, we are done. Therefore, let $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$. Since $w_2 \in \mathtt{W}^q_{j_2}(x_2, y_2)$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(x_2, y_2) \cap \mathtt{V}(a_{j_2}, b_{j_2})$.

Let $j_2 < k_2$. Then $[x_2, y_2] = [u'_2, v'_2]$ where $[u'_2, v'_2] \in \mathrm{C}^P_{j_2}(r_2)$. According to the choice of j_2, we have $w_2 \notin \mathcal{W}_{j_2+1}(r_2)$. It follows from Observation 286 that $w_2 \in \{a_{j_2}, b_{j_2}\}$. Hence, $w_1, w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$.

Let $j_2 = k_2$. Then $[x_2, y_2] = [u_2, v_2]$, and it follows from Observation 287 that $w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. Consequently, $w_1, w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. □

6.6.5. The Set $\mathcal{W}_{\mathtt{anc}}(q)$ and its Decomposition

For an affiliated subbundle pair q with underlying subbundle pair $[a, b]$ let $\mathcal{W}_{\mathtt{anc}}(q)$ be the set $\mathcal{W}(q) \cup \mathtt{anc}(q)$. As a and b are already in $\mathcal{W}(q)$ we only need to add the proper side ancestors of q to $\mathcal{W}(q)$ to obtain $\mathcal{W}_{\mathtt{anc}}(q)$.

Lemma 289. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair. For all $r \in R(q)$ it holds that $\mathcal{W}_{\mathtt{anc}}(r) \subseteq \mathcal{W}_{\mathtt{anc}}(q)$.*

Proof. Let $q = [a, b]_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $i \in [d+1]$ and $r := [u, v]_{[a_{i-1}, b_{i-1}]} \in V_{\mathcal{T}_{\text{gen}}}$ where $[u, v] \in \mathrm{C}_i^P(q)$. We show that all proper side ancestors of r are in $\mathcal{W}_{\mathtt{anc}}(q)$. Then it follows from Lemma 279 that $\mathcal{W}_{\mathtt{anc}}(r) \subseteq \mathcal{W}_{\mathtt{anc}}(q)$. If $i \leq d$, then all proper side ancestors of r are proper side ancestors of q by Observation 266, and therefore in $\mathcal{W}_{\mathtt{anc}}(q)$. Let $i = d+1$. Then a and b are the proper side ancestors of r of side depth d. Clearly, a and b are in $\mathcal{W}_{\mathtt{anc}}(q)$. According to Observation 266 all proper side ancestors of side depth at most $d-1$ are in $\mathcal{W}_{\mathtt{anc}}(q)$. □

Lemma 290. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair. Then $\mathcal{W}_{\mathtt{anc}}(q)$ is the union U of the set $\mathtt{anc}(q)$ and of all sets $\mathcal{W}_{\mathtt{anc}}(r)$ where $r \in R(q)$. Further, let $r_1, r_2 \in R(q)$ with $r_1 \neq r_1$. Then $\mathcal{W}_{\mathtt{anc}}(r_1) \cap \mathcal{W}_{\mathtt{anc}}(r_2) \subseteq \mathtt{anc}(q)$.*

Proof. Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. As $\mathtt{anc}(q) \subseteq \mathcal{W}_{\mathtt{anc}}(q)$, it follows from Lemma 289 that the set U is a subset of $\mathcal{W}_{\mathtt{anc}}(q)$. Further, $\mathcal{W}(q)$ is a subset of U (Lemma 280). As all side ancestors of q are also in U, we obtain $\mathcal{W}_{\mathtt{anc}}(q) \subseteq U$, and therefore $\mathcal{W}_{\mathtt{anc}}(q) = U$. For every subbundle pair $[u, v] \in \mathrm{C}_k^{\text{con}}(q)$ where $k \in \{1, \dots, d+1\}$, the proper side ancestors of $r = [u, v]_{a_{k-1}, b_{k-1}}$ are also side ancestors of q. Hence, it follows from Lemma 281 that $\mathcal{W}_{\mathtt{anc}}(r_1) \cap \mathcal{W}_{\mathtt{anc}}(r_2) \subseteq \mathtt{anc}(q)$ for all $r_1, r_2 \in R(q)$ with $r_1 \neq r_1$ □

Lemma 291. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair. Let $\{w_1, w_2\}$ be an edge of $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$. Then $w_1, w_2 \in \mathtt{anc}(q)$ or there is an $r \in R(q)$ such that $w_1, w_2 \in \mathcal{W}_{\mathtt{anc}}(r)$.*

Proof. Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a non-minimal affiliated consistent pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $\{w_1, w_2\}$ be an edge of $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$. As q is non-minimal consistent, $\mathrm{C}_{d+1}(q) \neq \emptyset$. Let $r_{d+1} \in R_{d+1}(q)$. Then $\mathtt{anc}(q) \subseteq \mathtt{anc}(r_{d+1})$. Thus, according to Lemma 290 we can assume there exist $r_1, r_2 \in R(q)$ such that $w_1 \in \mathcal{W}_{\mathtt{anc}}(r_1)$ and $w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$. There is nothing to show if $r_1 = r_2$. Thus, let $r_1 \neq r_2$. For $i \in [2]$ let $k_i \in \{1, \dots, d+1\}$ and $[u_i, v_i] \in \mathrm{C}_{k_i}^{\text{con}}(q)$ be such that $r_i = [u_i, v_i]_{a_{k_i-1}, b_{k_i-1}}$. Note that $u_i, v_i \in \mathcal{W}(r_i)$ by Observation 272. Consequently, $\mathcal{W}_{\mathtt{anc}}(r_i) = \mathcal{W}(r_i) \cup (\mathtt{anc}(r_i) \setminus \{u_i, v_i\})$

Let $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathcal{W}(r_2)$. Then it follows from Lemma 282 that $w_2 \in \mathcal{W}_{\mathtt{anc}}(r_1)$, $w_1 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ or $w_1, w_2 \in \mathtt{anc}(q)$. Hence, $w_1, w_2 \in \mathtt{anc}(q)$ or there is an $r \in R(q)$ such that $w_1, w_2 \in \mathcal{W}_{\mathtt{anc}}(r)$.

Let $w_1 \in \mathtt{anc}(r_1) \setminus \{u_1, v_1\}$ and $w_2 \in \mathtt{anc}(r_2) \setminus \{u_2, v_2\}$. Then $w_1, w_2 \in \mathtt{anc}(q)$.

Let $w_1 \in \mathtt{anc}(r_1) \setminus \{u_1, v_1\}$ and $w_2 \in \mathcal{W}(r_2)$. The case where $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathtt{anc}(r_2) \setminus \{u_2, v_2\}$ can be shown analogously. (The proof of this case is similar to the proof of Lemma 282.) If w_1 is an ancestor of a_{k_2-1} or b_{k_2-1}, then $w_1, w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$. In the following let w_1 be a proper descendant of a_{k_2-1} or b_{k_2-1}. Then $k_2 < k_1$ and $w_1 \in \{a_{k_1-1}, \dots, a_{k_2}\} \cup \{b_{k_1-1}, \dots, b_{k_2}\}$. Thus $k_2 \leq d$. Let $j_2 \leq k_2$ be maximal with $w_2 \in \mathcal{W}_{j_2}(r_2)$. According to Observation 267, $w_1 \in \mathtt{V}(a_{j_2}, b_{j_2})$.

Let $j_2 = k_2$. Then $w_2 \in \mathtt{W}^q_{k_2}(u_2, v_2)$ according to Observation 274. It follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_{k_2}, b_{k_2})$ or $w_1 \in \mathtt{W}^q_{k_2}(u_2, v_2)$. Consequently, $w_2 \in \mathtt{V}(a_{k_2}, b_{k_2})$ or $w_1 \in \mathcal{W}_{k_2}(r_2) \subseteq \mathcal{W}(r_2)$. If $w_1 \in \mathcal{W}(r_2)$, then $w_1, w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ and we are done. Therefore, let $w_2 \in \mathtt{V}(a_{k_2}, b_{k_2})$. Since $w_2 \in \mathtt{W}^q_{k_2}(u_2, v_2)$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(u_2, v_2) \cap \mathtt{V}(a_{k_2}, b_{k_2})$. According to Observation 250, w_2 is an ancestor of the vertex $z_2 = z([u_2, v_2], [a_{k_2}, b_{k_2}])$. Since $[u_2, v_2] \in \mathrm{C}^{\mathrm{con}}_{k_2}(q)$ and $k_2 \leq d$, the vertex z_2 is a or b. It follows that w_2 is an ancestor of a or b. Thus, $w_1, w_2 \in \mathtt{anc}(q)$.

Let $j_2 < k_2$. Then there exists a $[u'_2, v'_2] \in \mathrm{C}^P_{j_2}(r_2)$ such that $w_2 \in \mathtt{W}^{r_2}_{j_2}(u'_2, v'_2) \subseteq \mathcal{W}_{j_2}(r_2)$. Since $w_1 \in \mathtt{V}(a_{j_2}, b_{j_2})$, it follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$ or $w_1 \in \mathtt{W}^{r_2}_{j_2}(u'_2, v'_2)$. Consequently, $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$ or $w_1 \in \mathcal{W}_{j_2}(r_2) \subseteq \mathcal{W}(r_2)$. If $w_1 \in \mathcal{W}(r_2)$, then $w_1, w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ and we are done. Therefore, let $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$. Since $w_2 \in \mathtt{W}^{r_2}_{j_2}(u'_2, v'_2)$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(u'_2, v'_2) \cap \mathtt{V}(a_{j_2}, b_{j_2})$. According to Observation 250, w_2 is an ancestor of the vertex $z_2 = z([u'_2, v'_2], [a_{j_2}, b_{j_2}])$. It follows that z_2 is in $\mathcal{W}_{j_2+1}(r_2)$. As w_2 is an ancestor of z_2 and $w_2 \notin \mathcal{W}_{j_2+1}(r_2)$ according to the choice of j_2, Observation 277 implies that $\mathtt{sd}(w_2) \leq j_2$. Since $w_2 \in \mathtt{V}(a_{j_2}, b_{j_2})$, it follows from Observation 239 and Corollary 265 that w_2 is a_{j_2} or b_{j_2}. Hence, $w_1, w_2 \in \mathtt{anc}(q)$. □

Lemma 292. *Let $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ be a trivial affiliated subbundle pair with underlying subbundle pair $[a, b]$. Then $\mathcal{W}_{\mathtt{anc}}(q)$ is the union U of the set $\mathtt{anc}(q) \cup \mathtt{V}(a, b)$ and of all sets $\mathcal{W}_{\mathtt{anc}}(r)$ where $r \in R(q)$. Further, let $r_1, r_2 \in R(q)$ with $r_1 \neq r_1$. Then $\mathcal{W}_{\mathtt{anc}}(r_1) \cap \mathcal{W}_{\mathtt{anc}}(r_2) \subseteq \mathtt{anc}(q) \cup \mathtt{V}(a, b)$.*

Proof. Let $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ be a trivial affiliated subbundle pair of side depth d with underlying subbundle pair $[a, b]$ and side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Since $\mathtt{anc}(q) \subseteq \mathcal{W}_{\mathtt{anc}}(q)$, it follows from Lemma 289 and Observation 272 that the set U is a subset of $\mathcal{W}_{\mathtt{anc}}(q)$. Further, $\mathcal{W}(q)$ is a subset of U (Lemma 283). As all side ancestors of q are also in U, we obtain $\mathcal{W}_{\mathtt{anc}}(q) \subseteq U$, and therefore $\mathcal{W}_{\mathtt{anc}}(q) = U$. For every subbundle pair $[u, v] \in \mathrm{C}^{\mathrm{triv}}_k(q)$ where $k \in \{1, \dots, d\}$, the proper side ancestors of $[u, v]_{a_{k-1}, b_{k-1}}$ are also side ancestors of q. Hence, it follows from Lemma 285 that $\mathcal{W}_{\mathtt{anc}}(r_1) \cap \mathcal{W}_{\mathtt{anc}}(r_2) \subseteq \mathtt{anc}(q) \cup \mathtt{V}(a, b)$ for all $r_1, r_2 \in R(q)$ with $r_1 \neq r_1$. □

Lemma 293. *Let $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ be a trivial affiliated subbundle pair. Let $\{w_1, w_2\}$ be an edge of $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$. Then $w_1, w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$ or there is an $r \in R(q)$ such that $w_1, w_2 \in \mathcal{W}_{\mathtt{anc}}(r)$.*

Proof. Let $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ be a trivial affiliated subbundle pair of side depth d with side ancestors a_i, b_i for $i \in \{0, \dots, d\}$. Let $\{w_1, w_2\}$ be an edge of $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$. By Lemma 292, $\mathcal{W}_{\mathtt{anc}}(q)$ is the union U of the set $\mathtt{anc}(q) \cup \mathtt{V}(a, b)$ and of all sets $\mathcal{W}_{\mathtt{anc}}(r)$ where $r \in R(q)$.

If $w_1, w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$, there is nothing to show.

Let $w_1 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$ and $w_2 \in \mathcal{W}_{\mathtt{anc}}(r)$ where $r \in R(q)$. (The case where $w_1 \in \mathcal{W}_{\mathtt{anc}}(r)$ for an $r \in R(q)$ and $w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a, b)$ can be shown analogously.) Let $k \in [d]$ and $[u, v] \in \mathrm{C}^{\mathrm{triv}}_k(q)$ be such that $r = [u, v]_{a_{k-1}, b_{k-1}}$. If $w_1 \in \mathtt{anc}(r)$ or $w_2 \in \mathtt{anc}(r) \setminus \{u, v\} \subseteq$

$\mathtt{anc}(q)$, then we are done. Thus, let $w_1 \in \mathtt{V}(a,b) \cup \{a_{d-1},\dots,a_k\} \cup \{b_{d-1},\dots,b_k\}$ and $w_2 \in \mathcal{W}(r)$. Let j be maximal such that $w_2 \in \mathcal{W}_j(r)$. As $j \leq k$, we have $w_1 \in \mathtt{V}(a_j,b_j)$ according to Observation 267.

Let us show that there exist an $[x,y] \in \mathrm{C}_j(q)$ such that $w_2 \in \mathtt{W}^q_j(x,y) \subseteq \mathcal{W}_j(r)$. If $j=k$, then $w \in \mathtt{W}^q_j(u,v) = \mathcal{W}_j(r)$ according to Observation 274, and clearly $[u,v] \in \mathrm{C}_j(q)$. Thus, we let $[x,y] := [u,v]$ if $j=k$. Let $j<k$. Then $w \in \mathtt{W}^r_j(u',v') \subseteq \mathcal{W}_j(r)$ for a child $[u',v'] \in \mathrm{C}^P_j(r)$. According to Observations 269 and 268, we have $\mathrm{C}_j(r) = \mathrm{C}_j(q)$ and $\mathcal{F}_j(r) = \mathcal{F}_j(q)$. Consequently, $[u',v'] \in \mathrm{C}_j(q)$ and $w \in \mathtt{W}^q_j(u',v') \subseteq \mathcal{W}_j(r)$. We let $[x,y] := [u',v']$ if $j<k$.

Since $w_1 \in \mathtt{V}(a_j,b_j)$ and $w_2 \in \mathtt{W}^q_j(x,y)$, it follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_j,b_j)$ or $w_1 \in \mathtt{W}^q_j(x,y)$. Consequently, $w_2 \in \mathtt{V}(a_j,b_j)$ or $w_1 \in \mathcal{W}_j(r) \subseteq \mathcal{W}(r)$. If $w_1 \in \mathcal{W}(r)$, we are done. Therefore, let $w_2 \in \mathtt{V}(a_j,b_j)$. Since $w_2 \in \mathtt{W}^q_j(x,y)$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(x,y) \cap \mathtt{V}(a_j,b_j)$.

Let $j<k$. Then $[x,y] = [u',v']$ where $[u',v'] \in \mathrm{C}^P_j(r)$. According to the choice of j, we have $w_2 \notin \mathcal{W}_{j+1}(r)$. It follows from Observation 286 that $w_2 \in \{a_j,b_j\}$. Hence, $w_1,w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$.

Let $j=k$. Then $[x,y]=[u,v]$, and it follows from Observation 287 that $w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$. Consequently, $w_1,w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$.

Now, let $w_1 \in \mathcal{W}_{\mathtt{anc}}(r_1)$ and $w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ where $r_1,r_2 \in R(q)$. There is nothing to show if $r_1 = r_2$. Thus, let $r_1 \neq r_2$. For $i \in [2]$ let $k_i \in \{1,\dots,d\}$ and $[u_i,v_i] \in \mathrm{C}^{\mathrm{triv}}_{k_i}(q)$ be such that $r_i = [u_i,v_i]_{a_{k_i-1},b_{k_i-1}}$.

Let $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathcal{W}(r_2)$. Then it follows from Lemma 288 that $w_2 \in \mathcal{W}_{\mathtt{anc}}(r_1)$, $w_1 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ or $w_1,w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$. Hence, $w_1,w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$ or there is an $r \in R(q)$ such that $w_1,w_2 \in \mathcal{W}_{\mathtt{anc}}(r)$.

Let $w_1 \in \mathtt{anc}(r_1) \setminus \{u_1,v_1\}$ and $w_2 \in \mathtt{anc}(r_2) \setminus \{u_2,v_2\}$. Then $w_1,w_2 \in \mathtt{anc}(q)$.

Let $w_1 \in \mathtt{anc}(r_1) \setminus \{u_1,v_1\}$ and $w_2 \in \mathcal{W}(r_2)$. The case where $w_1 \in \mathcal{W}(r_1)$ and $w_2 \in \mathtt{anc}(r_2) \setminus \{u_2,v_2\}$ can be shown analogously. If w_1 is an ancestor of a_{k_2-1} or b_{k_2-1}, then $w_1,w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$. In the following let w_1 be a proper descendant of a_{k_2-1} or b_{k_2-1}. Then $k_2 < k_1$ and $w_1 \in \{a_{k_1-1},\dots,a_{k_2}\} \cup \{b_{k_1-1},\dots,b_{k_2}\}$. Let $j_2 \leq k_2$ be maximal with $w_2 \in \mathcal{W}_{j_2}(r_2)$. According to Observation 267, $w_1 \in \mathtt{V}(a_{j_2},b_{j_2})$.

Let $j_2 = k_2$. Then $w_2 \in \mathtt{W}^q_{k_2}(u_2,v_2)$ according to Observation 274. It follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_{k_2},b_{k_2})$ or $w_1 \in \mathtt{W}^q_{k_2}(u_2,v_2)$. Consequently, $w_2 \in \mathtt{V}(a_{k_2},b_{k_2})$ or $w_1 \in \mathcal{W}_{k_2}(r_2) \subseteq \mathcal{W}(r_2)$. If $w_1 \in \mathcal{W}(r_2)$, then $w_1,w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ and we are done. Therefore, let $w_2 \in \mathtt{V}(a_{k_2},b_{k_2})$. Since $w_2 \in \mathtt{W}^q_{k_2}(u_2,v_2)$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(u_2,v_2) \cap \mathtt{V}(a_{k_2},b_{k_2})$. According to Observation 287, we have $w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$. Hence, $w_1,w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$.

Let $j_2 < k_2$. Then there exists a $[u_2',v_2'] \in \mathrm{C}^P_{j_2}(r_2)$ such that $w_2 \in \mathtt{W}^{r_2}_{j_2}(u_2',v_2') \subseteq \mathcal{W}_{j_2}(r_2)$. Since $w_1 \in \mathtt{V}(a_{j_2},b_{j_2})$, it follows from Lemmas 236 and 184 that $w_2 \in \mathtt{V}(a_{j_2},b_{j_2})$ or $w_1 \in \mathtt{W}^{r_2}_{j_2}(u_2',v_2')$. Consequently, $w_2 \in \mathtt{V}(a_{j_2},b_{j_2})$ or $w_1 \in \mathcal{W}_{j_2}(r_2) \subseteq \mathcal{W}(r_2)$. If $w_1 \in \mathcal{W}(r_2)$, then $w_1,w_2 \in \mathcal{W}_{\mathtt{anc}}(r_2)$ and we are done. Therefore, let $w_2 \in \mathtt{V}(a_{j_2},b_{j_2})$. Since $w_2 \in \mathtt{W}^{r_2}_{j_2}(u_2',v_2')$, it follows from Lemmas 232 and 184 that $w_2 \in \mathtt{V}(u_2',v_2') \cap \mathtt{V}(a_{j_2},b_{j_2})$. According to the choice of j_2, we have $w_2 \notin \mathcal{W}_{j_2+1}(r_2)$. It follows from Observation 286 that $w_2 \in \{a_{j_2},b_{j_2}\}$. Hence, $w_1,w_2 \in \mathtt{anc}(q) \cup \mathtt{V}(a,b)$. □

6.6.6. Canonization

Let G^* be an LO-colored graph, and let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be the bundle extension of the O-extension of G^*. Let $\mathtt{H} = (\mathtt{V}, \mathtt{E})$ be the underlying graph of $\mathtt{H}^*$. We use the genealogical decomposition tree of $\mathtt{H}$ to define a canon of the LO-colored graph G^* in this section.

In this section, we first define the extended height of each node $q \in V_{\mathcal{T}_{\text{gen}}}$ in $\mathcal{T}_{\text{gen}}$. For all $r \in R(q)$ the extended height of r turns out to be less than the extended height of q. Based on the extended height of q in $\mathcal{T}_{\text{gen}}$, we recursively define an isomorphic copy on the number sort of the induced subgraph $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$ for each $q \in V_{\mathcal{T}_{\text{gen}}}$. We also maintain the edge relation of the side trees for this isomorphic copy and color the vertices of this isomorphic copy with different types of colors. We use module colors, inclusion colors, integration colors and affiliation colors. The module colors correspond to the colors of the color relation $\mathtt{L}$ of $\mathtt{H}^*$. The inclusion color of a vertex tells us whether the vertex is a vertex of G^* or not. We obtain the inclusion color from the unary relation $\mathtt{Z}$ of $\mathtt{H}^*$. The integration colors mark the vertices $p_j(q)$ for $j \in J(q)$ and the affiliation colors mark the vertices that are contained in the affiliation p' of q if $p' \neq [\,]$. Integration colors and affiliation colors are of importance during the recursive construction of isomorphic copies.

For every affiliated subbundle pair $q \in V_{\mathcal{T}_{\text{gen}}}$ let us define a string of numbers, the extended height $h_{\text{ex}}(q)$ of q. Let d be the side depth and a_i, b_i for $i \in \{0, \ldots, d\}$ be the side ancestors of q. First let us consider the decomposition forests $\mathcal{F}_i(q)$ for $i \in \{0, \ldots, d\}$. Let $[x, y]$ be a node in $\mathcal{F}_i(q)$. We let $h_i^q(x, y)$ be the height of $[x, y]$ in $\mathcal{F}_i(q)$. We define the *extended height* of q as the string $h_{\text{ex}}(q) = h_0(a_0, b_0)h_1(a_1, b_1)\ldots h_d(a_d, b_d)$. Note that the decomposition forest $\mathcal{F}_0(q)$ is $([f_1, f_2]_{[\,]}, \emptyset)$ for all $q \in V_{\mathcal{T}_{\text{gen}}}$. Thus, the only node $[f_1, f_2]_{[\,]}$ in $\mathcal{F}_0(q)$ has height 0. Hence, for all $q \in V_{\mathcal{T}_{\text{gen}}}$ the first character of the extended height is 0.

Let q_1 and q_2 be two affiliated subbundle pairs in $V_{\mathcal{T}_{\text{gen}}}$, and let $h_{\text{ex}}(q_1) = k_0^1 k_1^1 \ldots k_{d_1}^1$ and $h_{\text{ex}}(q_2) = k_0^2 k_1^2 \ldots k_{d_2}^2$ be their respective extended heights. We define a linear order on the extended height as follows. We let $h_{\text{ex}}(q_1) < h_{\text{ex}}(q_2)$ if either

- $d_1 > d_2$ and $k_i^1 = k_i^2$ for all $i \in \{0, \ldots, d_2\}$, or
- there exists a $j \leq \min\{d_1, d_2\}$ such that $k_j^1 < k_j^2$ and $k_i^1 = k_i^2$ for all $i \in \{0 \ldots, j-1\}$.

Clearly, the root $[f_1, f_2]_{[\,]}$ of $\mathcal{T}_{\text{gen}}$ is the only node of $\mathcal{T}_{\text{gen}}$ that has the maximal extended height 0. Let $q \in V_{\mathcal{T}_{\text{gen}}}$. It is not hard to see that for each $r \in R(q)$ we have $h_{\text{ex}}(r) < h_{\text{ex}}(q)$. Further, we observe the following.

Observation 294. *Let $q \in V_{\mathcal{T}_{\text{gen}}}$ be of minimal extended height. Then q is minimal consistent.*

Proof. Let $q = p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be of minimal extended height. First let us show that q is consistent. Suppose q is not consistent. As $[f_1, f_1]_{[\,]}$ is consistent, we have $p' \neq [\,]$ and p' is a non-minimal consistent pair by Observation 254. According to Observation 249 there exists a (proper) descendant o of p in $\mathcal{F}(p')$ such that o is consistent. It follows from Observation 255 that $o_{p'}$ is a node of $\mathcal{T}_{\text{gen}}$. Let $a_d, b_d, a_{d-1}, b_{d-1}, \ldots, a_0, b_0$ be the side ancestors of q. Then $h_{\text{ex}}(o_p) = h_0(a_0, b_0) \ldots h_{d-1}(a_{d-1}, b_{d-1})h_d(a_d, b_d)$. Note that $p = [a_d, b_d]$ and $p' = [a_{d-1}, b_{d-1}]$. Let $o = [a_d', b_d']$, then $a_d', b_d', a_{d-1}, b_{d-1}, \ldots, a_0, b_0$ are the side ancestors of $o_{p'}$ (Observation 266) and $h_{\mathrm{ex}}(o_p) = h_0(a_0, b_0) \dots h_{d-1}(a_{d-1}, b_{d-1}) h_d(a'_d, b'_d)$. As o is a proper descendant of p, we have $h_d(a'_d, b'_d) < h_d(a_d, b_d)$. Hence, $h_{\mathrm{ex}}(o_p) < h_{\mathrm{ex}}(q)$.

Now let us show that $q = p_{p'}$ is minimal consistent. Suppose it is non-minimal consistent. Then the decomposition forest $\mathcal{F}(p)$ is defined. Let o be a root of $\mathcal{F}(p)$. According to the definition of $\mathcal{T}_{\mathrm{gen}}$, the affiliated subbundle pair q is the parent of o_p in $\mathcal{T}_{\mathrm{gen}}$. Let the side depth of q be d and $a_d, b_d, \dots, a_0, b_0$ be the side ancestors of q. Let $o = [a^*, b^*]$. As $p = [a_d, b_d]$ is the affiliation of o_p, the side depth of o_p is $d+1$ (Observation 258) and $a^*, b^*, a_d, b_d, \dots, a_0, b_0$ are the side ancestors of o_p (Observation 266). Thus, $h_{\mathrm{ex}}(o_p) = h_{\mathrm{ex}}(q) h_{d+1}(a^*, b^*)$, and $h_{\mathrm{ex}}(o_p) < h_{\mathrm{ex}}(q)$. □

For every affiliated subbundle pair q in $V_{\mathcal{T}_{\mathrm{gen}}}$ we define its *canonization template* $\mathcal{H}^*_q$, which is a 6-tuple $\mathcal{H}^*_q = (V_q, E_q, T_q, L_q, Z_q, I_q)$ that consists of the following sets:

- V_q is the subset $\mathcal{W}_{\mathtt{anc}}(q)$ of vertices of $\mathtt{H}$ defined in the previous section.
- E_q is the edge relation $\mathtt{E}$ of $\mathtt{H}$ restricted to the set $\mathcal{W}_{\mathtt{anc}}(q)$, that is, $E_q := E \cap V_q^2$.
- T_q is the relation $\mathtt{T}$ of $\mathtt{H}$ restricted to the set $\mathcal{W}_{\mathtt{anc}}(q)$. We let $T_q := T \cap V_q^2$. We call T_q the *parent relation* of $\mathcal{H}^*_q$.
- L_q is the color relation $\mathtt{L}$ of bundle extension $\mathtt{H}^*$ restricted to the vertex set $\mathcal{W}_{\mathtt{anc}}(q)$, that is, $L_q := \mathtt{L} \cap (V_q \times \mathtt{M}^2)$. For every vertex $v \in V_q$, the relation $L_q(v) := \{(m,n) \in \mathtt{M}^2 \mid (v,m,n) \in L_q\}$ is the representation of the module that vertex v stands for. We call $L_q(v)$ the *module color* of v.
- $Z_q \subseteq V_q \times \{0,1\}$ is a binary relation that colors all vertices of $\mathcal{W}_{\mathtt{anc}}(q)$ with two colors, which indicate whether the vertex is included in the vertex set of G^* or was added later on. We let $Z_q := \{(v,i) \in V_q \times \{0,1\} \mid \mathbf{1}_{\mathtt{Z}}(v) = i\}$ where $\mathbf{1}_{\mathtt{Z}}$ is the characteristic function of the unary relation $\mathtt{Z}$ of $\mathtt{H}^*$. We call the set $Z_q(v) := \{i \in \{0,1\} \mid (v,i) \in Z_q\} = \{\mathbf{1}_{\mathtt{Z}}(v)\}$ the *inclusion color* of v.
- $I_q \subseteq V_q \times [d]$ is a binary relation used to mark the vertices $p_j(q)$ for $j \in J(q)$. We let d be the side depth of q. Remember that $0 \notin J(q)$. Thus, there does not exist a vertex $p_0(q)$. We let $I_q := \{(v,j) \in V_q \times [0,d] \mid I_q(v) = j\}$, and for all $v \in V_q$, we let $I_q(v) := \{j \in J(q) \mid p_j(q) = v\}$. We call $I_q(v)$ the *integration color* of v for all $v \in V_q$. Note that for $j \in J(q)$ the vertex $p_j(q)$ is not necessarily in V_q.

Note that there are linear orders on the module colors, the inclusion colors and the integration colors: The linear order $\trianglelefteq$ on the set $\mathtt{M}$ of basic color elements of $\mathtt{H}^*$ can be extended to a linear order on the module colors, and the natural linear order $\leq$ can be extended to a linear order on the inclusion colors and the integration colors.

In the following we define a total preorder $\preceq$ on d-tuples of vertices of V_q. We use lexicographic extensions $\leq_{\mathrm{lex}}$ and $\trianglelefteq_{\mathrm{lex}}$ of the natural linear order $\leq$ and of the linear order $\trianglelefteq$ on the set $\mathtt{M}$ of basic color elements of $\mathtt{H}^*$. Let $(a_1, \dots, a_d), (b_1, \dots, b_d) \in V_q^d$. We define $(a_1, \dots, a_d) \prec (b_1, \dots, b_d)$, if

$$(L_q(a_1), \dots, L_q(a_d)) \triangleleft_{\mathrm{lex}} (L_q(b_1), \dots, L_q(b_d)\text{, or}$$

$$(L_q(a_1), \dots, L_q(a_d)) \;=\; (L_q(b_1), \dots, L_q(b_d)) \text{ and}$$
$$(Z_q(a_1), \dots, Z_q(a_d)) <_{\mathrm{lex}} (Z_q(b_1), \dots, Z_q(b_d))\text{, or}$$

$$\begin{aligned}(L_q(a_1), \dots, L_q(a_d)) &= (L_q(b_1), \dots, L_q(b_d)) \text{ and} \\ (Z_q(a_1), \dots, Z_q(a_d)) &= (Z_q(b_1), \dots, Z_q(b_d)) \text{ and} \\ (I_q(a_1), \dots, I_q(a_d)) &<_{\text{lex}} (I_q(b_1), \dots, I_q(b_d)).\end{aligned}$$

In the following, we define what it means that $\mathcal{K}_q^*$ is an extended copy of $\mathcal{H}_q^*$ (on the number sort) for $q \in V_{\mathcal{T}_{\text{gen}}}$. In short, an extended copy of $\mathcal{H}_q^*$ consists of an isomorphic copy on the number sort of the structure (V_q, E_q, T_q) and of the adapted color relations from $\mathcal{H}_q^*$; though the form of the integration colors is slightly changed. Moreover, an extended copy of $\mathcal{H}_q^*$ is equipped with an additional color relation used to mark the vertices that correspond to the vertices of the affiliation of q.

For an affiliated subbundle pair $q \in V_{\mathcal{T}_{\text{gen}}}$, let $\mathfrak{K}_q^*$ be the set of all 7-tuples of relations $\mathcal{K}_q^* := (V_q^{\mathcal{K}}, E_q^{\mathcal{K}}, T_q^{\mathcal{K}}, L_q^{\mathcal{K}}, Z_q^{\mathcal{K}}, I_q^{\mathcal{K}}, A_q^{\mathcal{K}})$ where

$$\begin{aligned} &V_q^{\mathcal{K}} = [|V_q|], && L_q^{\mathcal{K}} \subseteq V_q^{\mathcal{K}} \times \mathtt{M}^2, \\ &E_q^{\mathcal{K}} \subseteq (V_q^{\mathcal{K}})^2, && Z_q^{\mathcal{K}} \subseteq V_q^{\mathcal{K}} \times \{0,1\}, \\ &T_q^{\mathcal{K}} \subseteq (V_q^{\mathcal{K}})^2, && I_q^{\mathcal{K}} \subseteq V_q^{\mathcal{K}} \times [|V_q^{\mathcal{K}}|] \times V_q, \\ & && A_q^{\mathcal{K}} \subseteq V_q^{\mathcal{K}} \times \mathtt{V}. \end{aligned}$$

For all $k \in V_q^{\mathcal{K}}$, we let

$$\begin{aligned} L_q^{\mathcal{K}}(k) &:= \{(m,n) \in \mathtt{M}^2 \mid (k,m,n) \in L_q^{\mathcal{K}}\}, \\ Z_q^{\mathcal{K}}(k) &:= \{i \in \{0,1\} \mid (k,i) \in Z_q^{\mathcal{K}}\}, \\ I_q^{\mathcal{K}}(k) &:= \{(i,v) \in [|V_q^{\mathcal{K}}|] \times V_q \mid (k,i,v) \in I_q^{\mathcal{K}}\}, \\ A_q^{\mathcal{K}}(k) &:= \{v \in \mathtt{V} \mid (w,v) \in A_q^{\mathcal{K}}\}. \end{aligned}$$

Similar to the canonization template we call $L_q^{\mathcal{K}}(k)$ the *module color* of k, $Z_q^{\mathcal{K}}(k)$ the *inclusion color* of k and $I_q^{\mathcal{K}}(k)$ the *integration color* of k for each $k \in V_q^{\mathcal{K}}$. Further, $A_q^{\mathcal{K}}(k)$ is the *affiliation color* of $k \in V_q^{\mathcal{K}}$.

A *semi-extended isomorphism* between $\mathcal{H}_q^*$ and $\mathcal{K}_q^*$ is an isomorphism $h\colon V_q \to [|V_q|]$ between (V_q, E_q, T_q) and $(V_q^{\mathcal{K}}, E_q^{\mathcal{K}}, T_q^{\mathcal{K}})$, so that for all $v \in V_q$,

$$\begin{aligned} L_q^{\mathcal{K}}(h(v)) &= L_q(v), \\ Z_q^{\mathcal{K}}(h(v)) &= Z_q(v), \\ I_q^{\mathcal{K}}(h(v)) &= \{(i,v) \mid i \in I_q(v)\}. \end{aligned} \tag{6.21}$$

Let $q = p_{p'} \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair with $p' \neq [\,]$. Let $p' = [a', b']$. We call $\mathcal{H}_q^*$ *symmetric in a' and b'* if there exists an automorphism f on the structure (V_q, E_q, T_q) such that for all $v \in V_q$ the vertices v and $f(v)$ have the same colors.

Let $q \in V_{\mathcal{T}_{\text{gen}}}$ and let p' be the affiliation of q. Let $\emptyset_{\text{aff}} := \mathtt{V}$. We all call $\mathcal{K}_q^*$ *affiliation-correct* if $p' = [\,]$ and $A_q^{\mathcal{K}} = \emptyset$, or $p' \neq [\,]$ and the following holds: Let $p' = [a', b']$. There exist $k_1, k_2 \in V_q^{\mathcal{K}}$ with $k_1 \neq k_2$ such that $A_q^{\mathcal{K}}(k) = \emptyset$ for all $k \in V_q^{\mathcal{K}} \setminus \{k_1, k_2\}$, and

- $A_q^{\mathcal{K}}(k_1) = \{a'\}$ and $A_q^{\mathcal{K}}(k_2) = \{b'\}$, or
- $A_q^{\mathcal{K}}(k_1) = \emptyset_{\text{aff}}$ and $A_q^{\mathcal{K}}(k_2) = \{a', b'\}$.

Let $\mathcal{K}_q^*$ be affiliation-correct. Let p' be the affiliation of q. An *extended isomorphism* between $\mathcal{H}_q^*$ and $\mathcal{K}_q^*$ is a semi-extended isomorphism h between $\mathcal{H}_q^*$ and $\mathcal{K}_q^*$ where $p' = [a', b'] \neq [\,]$ implies that either

- $A_q^{\mathcal{K}}(h(a')) = \{a'\}$ and $A_q^{\mathcal{K}}(h(b')) = \{b'\}$,

or

- $A_q^{\mathcal{K}}(h(a')) = \{a', b'\}$ and $A_q^{\mathcal{K}}(h(b')) = \emptyset_{\text{aff}}$ or
- $A_q^{\mathcal{K}}(h(a')) = \emptyset_{\text{aff}}$ and $A_q^{\mathcal{K}}(h(b')) = \{a', b'\}$

only if $\mathcal{H}_q^*$ is symmetric in a' and b'.[2]

For each $q \in V_{\mathcal{T}_{\text{gen}}}$, we call $\mathcal{K}_q^*$ an *extended copy* of $\mathcal{H}_q^*$ *(on the number sort)*, if $\mathcal{K}_q^*$ is affiliation-correct and there is an extended isomorphism between $\mathcal{H}_q^*$ and $\mathcal{K}_q^*$.

In the subsequent section, we recursively define an extended copy $\mathcal{K}_q^*$ of $\mathcal{H}_q^*$ on the number sort for all affiliated subbundle pairs $q \in V_{\mathcal{T}_{\text{gen}}}$. The extended copy $\mathcal{K}_q^*$, that is the relations $V_q^{\mathcal{K}}, E_q^{\mathcal{K}}, T_q^{\mathcal{K}}, L_q^{\mathcal{K}}, Z_q^{\mathcal{K}}, I_q^{\mathcal{K}}, A_q^{\mathcal{K}}$, can be defined in FP+C. We describe the construction of $\mathcal{K}_q^*$ in a way that illustrates how the necessary FP+C-formulas can be defined. We also add a few notes to the end of the following section that sketch how the extended copy $\mathcal{K}_q^*$ can be defined in FP+C.

Let G^* be an LO-colored graph, and let $\mathtt{H}^* = (\mathtt{U}, \mathtt{V}, \mathtt{E}, \mathtt{M}, \trianglelefteq, \mathtt{L}, \mathtt{T}, \mathtt{Z})$ be the bundle extension of the O-extension of G^*. For the affiliated subbundle pair $q = [f_1, f_2]_{[\,]}$, it holds that $\mathcal{W}_{\mathtt{anc}}(q) = \mathtt{V}(f_1, f_2) = \mathtt{V}$ (Observation 272). Thus, we can use the FP+C-definable extended copy $\mathcal{K}_q^* = (V_q^{\mathcal{K}}, E_q^{\mathcal{K}}, T_q^{\mathcal{K}}, L_q^{\mathcal{K}}, Z_q^{\mathcal{K}}, I_q^{\mathcal{K}}, A_q^{\mathcal{K}})$ of $\mathcal{H}_q^*$ for $q = [f_1, f_2]_{[\,]}$ to define the canon K^* of LO-colored graph G^*. Let $Z^* := \{k \in V_q^{\mathcal{K}} \mid Z_q^{\mathcal{K}}(k) = \{1\}\}$. Then we define $K^* := (U^*, V^*, E^*, \mathtt{M}, \trianglelefteq, L^*, \leq^*)$ where

$$
\begin{aligned}
U^* &:= V^* \cup \mathtt{M}, \\
V^* &:= V_q^{\mathcal{K}} \setminus Z^*, \\
E^* &:= E_q^{\mathcal{K}} \cap (V^* \times V^*), \\
L^* &:= L_q^{\mathcal{K}}, \\
\leq^* &:= \leq_{[|V^*|]} \cup \trianglelefteq \cup ([|V^*|] \times \mathtt{M}).
\end{aligned}
$$

It is not hard to see that we can use FP+C-formulas for the sets $V_q^{\mathcal{K}}, E_q^{\mathcal{K}}, L_q^{\mathcal{K}}, Z_q^{\mathcal{K}}$ to define an FP+C-transduction $\Theta_{\mathcal{K}}$ that maps the bundle extension $\mathtt{H}^*$ of the O-extension of an LO-colored graph $G^* \in \mathcal{C}_{\text{ChCo}}^*$ to the canon of K^* of G^*. In order to obtain an FP+C-canonization of the class $\mathcal{C}_{\text{ChCo}}^*$, we apply Proposition 12 and compose the transduction Θ' from Section 6.4.1, which maps each LO-colored graph $G^* \in \mathcal{C}_{\text{ChCo}}^*$ to an isomorphic copy of its O-extension, the transduction $\mathtt{\Theta}$ from Section 6.4.2, which maps each O-extension to an isomorphic copy of its bundle extension, and the transduction $\Theta_{\mathcal{K}}$.

[2] Note that if $\mathcal{H}_q^*$ is symmetric in a' and b', we can also have $A_q^{\mathcal{K}}(h(a')) = \{a'\}$ and $A_q^{\mathcal{K}}(h(b')) = \{b'\}$ for an extended isomorphism h between $\mathcal{H}_q^*$ and $\mathcal{K}_q^*$.

6.6.7. Defining the Extended Copy

Now we inductively define the extended copy $\mathcal{K}_q^*$ of $\mathcal{H}_q^*$ for every $q \in V_{\mathcal{T}_{\text{gen}}}$. The induction is based on the extended height. We start with all $q \in V_{\mathcal{T}_{\text{gen}}}$ of minimal extended height.

Affiliated Pairs of Minimal Extended Height

First of all, we show that we can define the extended copy $\mathcal{K}_q^*$ of $\mathcal{H}_q^*$ for all $q \in V_{\mathcal{T}_{\text{gen}}}$ of minimal extended height $h_{\text{ex}}(q) = 0 \dots 0$. Then $q \neq [f_1, f_2]_{[\,]}$. Let $[a, b]$ be the underlying subbundle pair, d be the side depth and a_i, b_i for $i \in [0, d]$ be the side ancestors of q. As q is of minimal extended height, q must be minimal consistent according to Observation 294. Because of the minimal extended height, $\mathrm{C}_i(q)$ is empty for all $i \leq d$, and $[a, b]$ is a leaf of $\mathcal{F}_d(q)$. Consequently, we have $\mathcal{W}(q) = \mathtt{W}_d^q(a, b) = \mathtt{V}(a, b)$ and $\mathcal{W}_{\mathtt{anc}}(q) = \mathtt{V}(a, b) \cup \mathtt{anc}(q)$. As $[a, b]$ is minimal consistent, $[a, b]$ is trivial (Observation 241) and a or b is a leaf of $\mathtt{H}$. Without loss of generality, let $b \in \mathtt{O}$. By Observation 170, $V(a, b)$ is the vertex set of a directed path with ends a and b in $\mathtt{T}_f$ for an $f \in \mathtt{F}$. Since $b \in \mathtt{O}$, vertex a is the first vertex of the path. It follows from Observation 243 that $V(a, b) \cup \mathtt{anc}(q)$ is the set of all ancestors of b. By Corollary 169, vertex b spans a max clique M, which is equal to $V(a, b) \cup \mathtt{anc}(q)$. Thus, $V_q = M$ and $E_q = \binom{M}{2}$. We let the vertex set $V_q^{\mathcal{K}}$ and edge set $E_q^{\mathcal{K}}$ of the extended copy $\mathcal{K}_q^*$ be the set $[|M|]$ and the set $\{\{k, l\} \mid k, l \in [|M|], k \neq l\}$, respectively. In order to define the remaining relations, we assign the vertices of V_q to the numbers in $V_q^{\mathcal{K}}$. We distinguish between the following cases.

First let us consider the case where $a = b$. Then both a and b are leaves. In order to define the remaining relations of $\mathcal{K}_q^*$, we compare the module, inclusion and integration colors of $a_0, \dots, a_{d-1}$ and $b_0, \dots, b_{d-1}$. Let $\bar{a} := (a_0, \dots, a_{d-1})$ and $\bar{b} := (b_0, \dots, b_{d-1})$. Let us assume $\bar{a} \lessdot \bar{b}$. Then $\mathcal{H}_q^*$ is not symmetric in a_{d-1} and b_{d-1}. We assign a_i to $h(a_i) = i+1$ for all $i \in [0, d-1]$ and b_i to $h(b_i) = d+1+i$ for all $i \in [0, d]$. Further, we let $(k, l) \in T_q^{\mathcal{K}}$ if and only if $(h^{-1}(k), h^{-1}(l)) \in T_q$ for all $k, l \in V_q^{\mathcal{K}}$. We transfer the module, inclusion and integration colors of the vertices in V_q to the numbers in $V_q^{\mathcal{K}}$ according to assignment h (cf. the rules in (6.21)). Then h is a semi-extended isomorphism between $\mathcal{H}_q^*$ and the extended copy $\mathcal{K}_q^*$ that we define. To complete the definition of the extended copy K_q^* of $\mathcal{H}_q^*$, we let the affiliation color of d be $\{a_{d-1}\}$ and the affiliation color of $2d$ be $\{b_{d-1}\}$. Then K_q^* is affiliation-correct, and h is an extended isomorphism between H_q^* and K_q^*. Thus, K_q^* is an extended copy $\mathcal{H}_q^*$. Let us denote the extended copy K_q^* defined in this case by $K_q^*(\bar{a}, \bar{b})$. If $\bar{b} \lessdot \bar{a}$, then we define K_q^* analogously, that is, we let $K_q^* := K_q^*(\bar{b}, \bar{a})$. Now, let neither $\bar{a} \lessdot \bar{b}$ nor $\bar{b} \lessdot \bar{a}$. Then $\mathcal{H}_q^*$ is symmetric in a_{d-1} and b_{d-1}. As a consequence, $K_q^*(\bar{a}, \bar{b})$ and $K_q^*(\bar{b}, \bar{a})$ are equal if the affiliation colors are removed. In this case, we use the assignment h from above to define the parent relation and to transfer all module, inclusion and integration colors according to the rules in (6.21). Finally, we let the affiliation color of d be $\{a_{d-1}, b_{d-1}\}$ and the one of $2d$ be $\emptyset_{\text{aff}}$. Then we obtain an extended copy K_q^* of $\mathcal{H}_q^*$.

Next let us consider the case where $a \neq b$. Without loss of generality, let b be the leaf. Then there is an $f \in \mathtt{F}$ such that $V(a, b)$ is the vertex set of the directed path from a to b in $\mathtt{T}_f$ (Observation 245). Let the length of the path be l and let this path from a to b be $a = a_d, a_{d+1}, \dots, a_{d+l} = b$. Then we assign b_i to $h(b_i) = i + 1$ for all $i \in \{0, \dots, d-1\}$

and a_i to $h(a_i) = d + 1 + i$ for all $i \in \{0, \dots, d + l\}$. As above we use h to define the parent relation and transfer all module, inclusion and integration colors according to the rules in (6.21). We let the affiliation color of d be $\{b_{d-1}\}$ and the affiliation color of $2d$ be $\{a_{d-1}\}$. Again we obtain an extended copy K_q^* of H_q^*.

Affiliated Pairs of Non-Minimal Extended Height

Now let $q \in V_{\mathcal{T}_{\text{gen}}}$ be an affiliated subbundle pair whose extended height $h_{\text{ex}}(q)$ is not minimal, and let us assume that the extended copy $K_{q'}^*$ of $\mathcal{H}_q^*$ for each $q' \in V_{\mathcal{T}_{\text{gen}}}$ of extended height $h_{\text{ex}}(q') < h_{\text{ex}}(q)$ is defined. Let $[a, b]$ be the underlying subbundle pair, d be the side depth and a_i, b_i for $i \in [0, d]$ be the side ancestors of q. In the following we define the extended copy $\mathcal{K}_q^*$.

Non-Minimal Consistent Affiliated Pairs

First let us consider affiliated subbundle pairs $q \in V_{\mathcal{T}_{\text{gen}}}$ that are consistent and not minimal. To define $\mathcal{K}_q^*$ we construct two candidates $\mathcal{K}_q^a$ and $\mathcal{K}_q^b$ for the extended copy of $\mathcal{H}_q^*$, which do not have affiliation colors.

Let $c \in \{a, b\}$. We illustrate the construction of $\mathcal{K}_q^c$. Without loss of generality, let $c = a$. The set $\mathcal{W}_{\texttt{anc}}(q)$ is the union U of the set $\texttt{anc}(q)$ and of all sets $\mathcal{W}_{\texttt{anc}}(r)$ for $r \in R(q)$, and the pairwise intersection of these sets forming U is a subset of $\texttt{anc}(q)$ according to Lemma 290. Furthermore, each edge of $\texttt{H}[\mathcal{W}_{\texttt{anc}}(q)]$ is an edge of $\texttt{H}[\texttt{anc}(q)]$ or $\texttt{H}[\mathcal{W}_{\texttt{anc}}(r)]$ for an $r \in R(q)$ by Lemma 291. For all affiliated subbundle pairs $r \in R(q)$ there is a $j \in [d + 1]$ and a $[u, v] \in \mathrm{C}_j^{\text{con}}(q)$ such that $r = [u, v]_{a_{j-1}, b_{j-1}}$. Thus, we have $h_{\text{ex}}(r) < h_{\text{ex}}(q)$. Consequently, for these affiliated subbundle pairs r we already constructed the extended copies $\mathcal{K}_r^*$ and we can use them to construct $\mathcal{K}_q^a$.

Let us start to construct $\mathcal{K}_q^a$. First we construct the part $\mathcal{K}_{\texttt{anc}(q)}^a$ of $\mathcal{K}_q^a$ that is based on the set of number vertices corresponding to $\texttt{anc}(q)$. We assign numbers to all vertices in $\texttt{anc}(q)$ by mapping a_i to $h(a_i) = i + 1$ for all $i \in \{0, \dots, d\}$ and b_i to $h(b_i) = d + 2 + i$ for all $i \in \{0, \dots, d\}$. Here, we assign the smaller numbers $1, \dots, d + 1$ to $a_d, \dots, a_0$, because we fixed a. When constructing $\mathcal{K}_{\texttt{anc}(q)}^b$, we assign them to $b_d, \dots, b_0$. Note that $\texttt{anc}(q)$ is a clique in $\texttt{H}$, which follows from Observation 267 and the fact that each $z \in \texttt{V}(a_i, b_i)$ with $z \neq a_i$ and $z \neq b_i$ is adjacent to a_i and b_i. (cf. Observation 190). Thus, we let the numbers $1, \dots, 2d + 2$ be the vertices of $\mathcal{K}_{\texttt{anc}(q)}^a$, and we add all edges between these numbers to $\mathcal{K}_{\texttt{anc}(q)}^a$. We use the assignment h to transfer all module, inclusion and integration colors from the vertices in $\texttt{anc}(q)$ to their corresponding numbers (cf. the rules in (6.21)), and to define the parent relation on $\{1, \dots, 2d+2\}$ in $\mathcal{K}_{\texttt{anc}(q)}^a$ accordingly. We do not add affiliation colors to $\mathcal{K}_{\texttt{anc}(q)}^a$.

Let $k \in [d + 1]$. We show that there is a linear order $\leq_k$ on the set of all extended copies $\mathcal{K}_r^*$ where $r \in R_k(q)$. Suppose there are linear orders $\leq_I$ and $\leq_A$, respectively, on the set of all integration colors and on the set of all affiliation colors occurring in any extended copy $\mathcal{K}_r^*$ for $r \in R_k(q)$. We know that $\trianglelefteq_{\text{lex}}$ and $\leq$ are linear orders on the set of all module colors and the set of all inclusion colors, respectively. As the natural linear order $\leq$ is a linear order on the vertices of each extended copy $\mathcal{K}_r^*$ for $r \in R_k(q)$, it is not hard to see that we can define a linear order $\leq_k$ on the set of all extended copies $\mathcal{K}_r^*$ with $r \in R_k(q)$ if linear orders $\leq_I$ and $\leq_A$ exist.

First let us show that there is a linear order $\leq_A$ on the set of all affiliation colors occurring in extended copies $\mathcal{K}_r^*$ where $r \in R_k(q)$. Each $r \in R_k(q)$ has the affiliation $[a_{k-1}, b_{k-1}]$. As a consequence, in each extended copy $\mathcal{K}_r^*$ where $r \in R_k(q)$ there can only occur the affiliation colors $\emptyset_{\text{aff}}$, $\{a_{k-1}\}$, $\{b_{k-1}\}$ and $\{a_{k-1}, b_{k-1}\}$. As we fixed a we let $\emptyset_{\text{aff}} <_A \{a_{k-1}\} <_A \{b_{k-1}\} <_A \{a_{k-1}, b_{k-1}\}$. (In the case that b is fixed, we define $\emptyset_{\text{aff}} <_A \{b_{k-1}\} <_A \{a_{k-1}\} <_A \{a_{k-1}, b_{k-1}\}$.)

Next let us show that there is a linear order $\leq_I$ on the set of all integration colors occurring in extended copies $\mathcal{K}_r^*$ where $r \in R_k(q)$. As all affiliated subbundle pairs $r \in R_k(q)$ have the same affiliation $[a_{k-1}, b_{k-1}]$, Observation 271 implies that there is at most one integration color that contains l as its first component for every $l \leq k-1$. For $l = k$ there might be multiple integration colors with l as its first component. If $l = k = d+1$, then there is no integration color with l as its first component in any extended copy $\mathcal{K}_r^*$ with $r \in R_k(q)$, since the underlying pair $[u, v]$ of r is a root in $\mathcal{F}_{d+1}$ for all $r \in R_k(q)$. Let $l = k < d+1$. The set $R_k(q)$ contains only affiliated subbundle pairs r where $r = [u, v]_{[a_{k-1}, b_{k-1}]}$ and $[u, v] \in \mathrm{C}_k^{\mathrm{con}}(q)$. For all $[u, v] \in \mathrm{C}_k^{\mathrm{con}}(q)$ the vertex $z([u, v], [a_k, b_k])$ is a or b. Thus, the integration colors with $l = k$ as first component are either $\{(k, a)\}$ or $\{(k, b)\}$. As we fixed a, we let $\{(k, a)\} <_I \{(k, b)\}$. For fixed b, we have $\{(k, b)\} <_I \{(k, a)\}$. If we additionally let $(i, v) <_I (i', v')$ whenever $i < i'$ for all integration colors $(i, v), (i', v') \in \mathbb{N} \times \mathtt{V}$ occurring in any extended copy $\mathcal{K}_r^*$ where $r \in R_k(q)$, then $\leq_I$ is a linear order on the set of all integration colors occurring in extended copies $\mathcal{K}_r^*$ where $r \in R_k(q)$.

In the following we describe the construction of $\mathcal{K}_q^a$. We already assigned number vertices to the vertices in $\mathtt{anc}(q)$ and defined $\mathcal{K}_{\mathtt{anc}(q)}^a$. Now we basically attach a copy of each extended copy $\mathcal{K}_r^*$ for $r \in R(q)$ to $\mathcal{K}_{\mathtt{anc}(q)}^a$. We start with $k = d+1$ and attach copies of the extended copies $\mathcal{K}_r^*$ for $r \in R_k(q)$ for each $k \in [d+1]$ in the order given by the linear order $\leq_k$.

Let $k \in [d+1]$. In order to attach a copy of the extended copy $\mathcal{K}_r^*$ for $r \in R_k(q)$ to $\mathcal{K}_{\mathtt{anc}(q)}^a$, we first identify the number vertices of $\mathcal{K}_r^*$ that correspond to vertices in $\mathtt{anc}(q)$.

Let $k = d+1$. Then $[a_d, b_d]$ is the affiliation of r and $\mathtt{anc}(q) \subseteq \mathtt{anc}(r)$. Thus, $\mathcal{K}_r^*$ contains vertices corresponding to $a_d, \ldots, a_0$ and $b_d, \ldots, b_0$. We can easily find the number vertices in $\mathcal{K}_r^*$ for $a = a_d$ and $b = b_d$ in $\mathcal{K}_r^*$ using the affiliation color: There exist either two number vertices whose respective affiliation colors are $\{a\}$ and $\{b\}$, then the vertex colored with $\{a\}$ corresponds to a and the one colored with $\{b\}$ corresponds to b, or two number vertices whose respective affiliation colors are $\{\emptyset_{\text{aff}}\}$ and $\{a, b\}$. In the second case $\mathcal{H}_r^*$ is symmetric in a and b, and as we fixed a, we suppose the vertex colored with $\{\emptyset_{\text{aff}}\}$ corresponds to a and the one colored with $\{a, b\}$ corresponds to b. We can use the parent relation to determine the number vertices corresponding to the ancestors of a and b. Note that $[a, b]$ is non-minimal consistent. Thus, $a_d, \ldots, a_0$ are the ancestors of a and $b_d, \ldots, b_0$ the ancestors of b. Hence, if $k = d+1$, we can identify the number vertices corresponding to the vertices in $\mathtt{anc}(q)$.

Let $k < d+1$. Let $r = [u, v]_{a_{k-1}, b_{k-1}}$ where $[u, v] \in \mathrm{C}_k^{\mathrm{con}}(q)$. Thus, the vertex $z := z([u, v], [a_k, b_k])$ is a or b. Let us assume $z = a$. The set $\mathcal{W}_{\mathtt{anc}}(r)$ contains the set $\mathtt{anc}(r)$ and therefore the side ancestors $a_{k-1}, \dots, a_0$ and $b_{k-1}, \dots, b_0$. Since $z \in V(u,v)$, we have $a \in V(u,v) \subseteq W_k^r(u,v) \subseteq \mathcal{W}_k(r)$. It follows from Observation 277 that $\mathcal{W}_k(r)$, and therefore also $\mathcal{W}_{\mathtt{anc}}(r)$, contains $a = a_d, \dots, a_k$. Since $z([u,v],[a_k,b_k]) = a$, Observation 250 implies that $b_k, \dots, b_d$ are not in $V(u,v)$. Consequently, $b_k, \dots, b_d$ are also not in $\mathcal{W}(r) \subseteq \mathcal{W}_{\mathtt{anc}}(r)$ by Lemma 278. It follows that $\mathcal{W}_{\mathtt{anc}}(r) \cap \mathtt{anc}(q) = \{a_d, \dots, a_0, b_{k+1}, \dots, b_0\}$. We can identify the number vertex corresponding to a_d in $\mathcal{K}_r^*$ with help of the integration color. The integration color of the vertex corresponding to a_d is (k, a_d). We can identify the number vertex corresponding to b_{k-1} with help of the affiliation color. Either there exists a number vertex with affiliation color $\{b_{k-1}\}$ in $\mathcal{K}_r^*$ or $\mathcal{H}_r^*$ is symmetric in a_{k-1} and b_{k-1}. Since there is (only) one vertex with integration color (k, a_d) in $\mathcal{K}_r^*$, which is a descendant of the number vertex corresponding to a_{k-1} but not of the number vertex corresponding to b_{k-1} ($a_d = p_k(r) \notin \mathtt{O}$, Observation 270), $\mathcal{H}_r^*$ cannot be symmetric in a_{k-1} and b_{k-1}. Thus, there exists a number vertex with affiliation color $\{b_{k-1}\}$ in $\mathcal{K}_r^*$. Using the parent relation we can determine all number vertices corresponding to ancestors of a and b_{k-1}. Consequently, if $k < d+1$, we can also identify the number vertices corresponding to the vertices in $\mathtt{anc}(q)$.

Now that we can identify the number vertices of $\mathcal{K}_r^*$ that correspond to vertices in $\mathtt{anc}(q)$, we describe how the extended copies $\mathcal{K}_r^*$ are attached to $\mathcal{K}_{\mathtt{anc}(q)}^a$. Let A_r be the set of number vertices in $\mathcal{K}_r^*$ that corresponds to $\mathtt{anc}(q) \cap \mathcal{W}_{\mathtt{anc}}(r)$, and let $g\colon A_r \to \mathtt{anc}(q)$ be the assignment that maps each vertex in A_r to the vertex in $\mathtt{anc}(q)$ it corresponds to. Let $R_k^r(q)$ be the set of $r' \in R_k(q)$ with $\mathcal{K}_r^* = \mathcal{K}_{r'}^*$. Let $m_r := |R_k^r(q)|$. We simultaneously create copies of the extended copies $\mathcal{K}_r^*$ for all $r \in R_k^r(q)$ and attach them to $\mathcal{K}_{\mathtt{anc}(q)}^a$. We create m_r copies $(\mathcal{K}_r^*)_i$, $i \in [m_r]$, of $\mathcal{K}_r^*$ by renumbering the number vertices in $\mathcal{K}_r^*$. To obtain $(\mathcal{K}_r^*)_i$, we renumber each vertex $c \in A_r$ to $h(g(c))$, and we renumber all vertices in $V_r^{\mathcal{K}} \setminus A_r$ so that they obtain numbers from $n + 1 + (i-1) \cdot |V_r^{\mathcal{K}} \setminus A_r|$ to $n + i \cdot |V_r^{\mathcal{K}} \setminus A_r|$ where n is the sum of $|\mathtt{anc}(q)|$ and of $|V_{r'} \setminus \mathtt{anc}(q)|$ for all $r' \in R_j(q)$ with $j > k$ and all $r' \in R_k(q)$ where $\mathcal{K}_{r'}^* <_k \mathcal{K}_r^*$. Now we can attach the copies $(\mathcal{K}_r^*)_i$ to $\mathcal{K}_{\mathtt{anc}(q)}^a$ by simply joining them. When we do that we remove the affiliation colors, and remove all integration colors (i,v) where $i \geq k$. As $[a_{k-1}, b_{k-1}]$ is the affiliation of r, we have $p_j(r) = p_j(q)$ for all $j \in J(r)$ with $j < k$ (Observation 271), and we can (and have to) copy the integration colors (j,v) where $j < k$.

After attaching all $\mathcal{K}_r^*$ for $r \in R(q)$ to $\mathcal{K}_{\mathtt{anc}(q)}^a$ we obtain $\mathcal{K}_q^a$. Let us look at the integration colors of the number vertices in $\mathcal{K}_q^a$. Let $j \in [0,d]$. Let us show that if $j \in J(q)$ and $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(q)$, then there is a number vertex in $\mathcal{K}_q^a$ with the integration color $(j, p_j(q))$. Let $j \in J(q)$ and $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(q)$. If $p_j(q) \in \mathtt{anc}(q)$, then there is a vertex in $\mathcal{K}_{\mathtt{anc}(q)}^a$, and therefore in $\mathcal{K}_q^a$, with the integration color $(j, p_j(q))$. Let $p_j(q) \in \mathcal{W}(q) \setminus \mathtt{anc}(q)$. Then there is a $k \in [d+1]$ and an $r \in R_k(q)$ such that $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(r)$ by Lemma 290. If $j < k$, we copied the integration color $(j, p_j(r)) = (j, p_j(q))$ (Observation 271). Let $j \geq k$. Then $p_j(q) \in \mathtt{V}(a_j, b_j) \subseteq \mathtt{V}(a_k, b_k)$ according to Observation 267. Let $r = [u,v]_{a_{k-1}, b_{k-1}}$ where $[u,v] \in \mathrm{C}_k^{\mathrm{con}}(q)$. Since $\mathtt{anc}(r) \setminus \{u,v\} \subseteq \mathtt{anc}(q)$ (Observation 266), we have $p_j(q) \in \mathcal{W}(r)$. Thus, $p_j(q) \in \mathcal{W}(r) \cap \mathtt{V}(a_k, b_k)$. As $j \geq k$ and $j \in J(q)$, we have $k \leq d$, and it follows from Observation 278 that $p_j(q) \in \mathtt{V}(u,v) \cap \mathtt{V}(a_k, b_k)$. Since $[u,v] \in \mathrm{C}_k^{\mathrm{con}}(q)$, the vertex $z([u,v],[a_k,b_k])$ is a or b. Consequently, $p_j(q) \in \mathtt{V}(u,v) \cap \mathtt{V}(a_k,b_k) \subseteq \mathtt{anc}(q)$ by Observation 250. Since $p_j(q) \in \mathcal{W}(q) \setminus \mathtt{anc}(q)$, the case where $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(r)$ for an $r \in R_k(q)$ and $j \geq k$ cannot occur.

Finally let us define the extended copy $\mathcal{K}_q^*$. To define the extended copy $\mathcal{K}_q^*$, we also construct $\mathcal{K}_q^b$. Note that the integration colors of $\mathcal{K}_q^a$ and $\mathcal{K}_q^b$ are the same, and for each $j \in J(q)$ there is at most one integration color with j as its first component. Since $\mathcal{K}_q^a$ and $\mathcal{K}_q^b$ do not have affiliation colors, we can linearly order them. First let us consider the case where $q = [f_1, f_2]_{[\,]}$. For $q = [f_1, f_2]_{[\,]}$, an extended copy $\mathcal{K}_q^*$ does not contain affiliation colors. Let $\mathcal{K}_q^c$ be the minimum of $\mathcal{K}_q^a$ and $\mathcal{K}_q^b$. Then we let $\mathcal{K}_q^*$ be $\mathcal{K}_q^c$. Now let us consider the case where $q \neq [f_1, f_2]_{[\,]}$. If $\mathcal{K}_q^a$ is smaller that $\mathcal{K}_q^b$, then we set the affiliation color of d to $\{a_{d-1}\}$ and the affiliation color of $2d+1$ to $\{b_{d-1}\}$. If $\mathcal{K}_q^b$ is smaller than $\mathcal{K}_q^a$, we set the affiliation color of d and $2d+1$ to $\{b_{d-1}\}$ and $\{a_{d-1}\}$ respectively. If $\mathcal{K}_q^b = \mathcal{K}_q^a$, then $\mathcal{H}_q^*$ is symmetric in a_{d-1} and b_{d-1}, and we set the affiliation color of d to $\{b_{d-1}, a_{d-1}\}$ and the one of $2d+1$ to $\{0\}$.

Trivial Affiliated Pairs

Now let $q \in V_{\mathcal{T}_{\text{gen}}}$ be a trivial affiliated subbundle pair. Note that this implies $q \neq [f_1, f_2]_{[\,]}$. Let $[a, b]$ be the underlying subbundle pair, d be the side depth and a_i, b_i for $i \in \{0, \ldots, d\}$ be the side ancestors of q.

First, let q be minimal consistent and $a = b$. According to Observation 275, we have $\mathcal{W}(q) = \{a\}$. It this case we define $\mathcal{K}_q^*$ analogous to the case where $q \in V_{\mathcal{T}_{\text{gen}}}$ is of minimal extended height and $a = b$.

In the following let $a \neq b$ or q be not minimal consistent. As $[a, b]$ is trivial, there exists an $f \in \mathtt{F}$ such that $a, b \in \mathtt{V}_f$. According to Observation 245, $V(a, b)$ is the vertex set of a directed path with ends a and b in $\mathtt{T}_f$. Without loss of generality let a be the first vertex of the path, then a_{d-1} is the parent of a in $\mathtt{T}_f$ (Observation 245). Let us assume $a \in \mathtt{O}$. Then $a = b$ and Observation 245 implies that a_{d-1} and b_{d-1} are the parents of a in the respective side trees. Thus, $\mathtt{sd}_{f_1}(a) = \mathtt{sd}_{f_2}(a) = d$. Hence, $[a, b]$ is consistent and $a = b$, a contradiction. Consequently, $a \in \mathtt{S}_f$. It follows that there exists a unique $f \in \mathtt{F}$ with $a, b \in \mathtt{V}_f$, and that we can identify the vertex $c \in \{a, b\}$ that is the first (last) vertex of the path induced by $V(a, b)$ in $\mathtt{T}_f$ by choosing the the vertex $c' \in \{a, b\}$ where $\mathtt{sd}_f(c')$ is minimal (maximal). In the following, let us suppose that $f = f_1$ and that a is the first vertex of the path induced by $V(a, b)$ in $\mathtt{T}_{f_1}$. Further, we suppose a_{d-1} is the parent of a in $\mathtt{T}_{f_1}$.

Let us illustrate the construction of $\mathcal{K}_q^*$ in the case that $a \neq b$ or q is not minimal consistent. The set $\mathcal{W}_{\mathtt{anc}}(q)$ is the union U of the set $\mathtt{anc}(q) \cup \mathtt{V}(a, b)$ and of all sets $\mathcal{W}_{\mathtt{anc}}(r)$ for $r \in R(q)$, and the pairwise intersection of these sets forming U is a subset of $\mathtt{anc}(q)$ according to Lemma 292. Furthermore, each edge of $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(q)]$ is an edge of $\mathtt{H}[\mathtt{anc}(q) \cup \mathtt{V}(a, b)]$ or of $\mathtt{H}[\mathcal{W}_{\mathtt{anc}}(r)]$ for an $r \in R(q)$ by Lemma 293. For all affiliated subbundle pairs $r \in R(q)$ there is a $j \in [d]$ and a $[u, v] \in \mathrm{C}_j^{\text{triv}}(q)$ such that $r = [u, v]_{a_{j-1}, b_{j-1}}$. Thus, we have $h_{\text{ex}}(r) < h_{\text{ex}}(q)$. Consequently, for these affiliated subbundle pairs r we already constructed the extended copies $\mathcal{K}_r^*$ and we can use them to construct $\mathcal{K}_q^*$.

Let $A_q := \mathtt{anc}(q) \cup \mathtt{V}(a, b)$. We first construct the part $\mathcal{K}_{A_q}^*$ of $\mathcal{K}_q^*$ that is based on the set of number vertices corresponding to $\mathtt{anc}(q) \cup \mathtt{V}(a, b)$. We assign numbers to all vertices in A_q. $V(a, b)$ is the vertex set of the directed path from a to b in $\mathtt{T}_{f_1}$. Let the length of the path be l and let this path from a to b be $a = a_d, a_{d+1}, \ldots, a_{d+l} = b$. We assign b_i to $h(b_i) = i + 1$ for all $i \in \{0, \ldots, d-1\}$ and a_i to $h(a_i) = d + 1 + i$ for all

$i \in \{0, \ldots, d+l\}$. The set $A_q = \mathtt{anc}(q) \cup \mathtt{V}(a,b)$ is a clique in $\mathtt{H}$: Since $V(a,b)$ induces a path in $\mathtt{T}_{f_1}$, Corollary 166 implies that $\mathtt{V}(a,b)$ is a clique. Inductively, it follows from Observation 267 and the fact that each $z \in \mathtt{V}(a_i, b_i)$ with $z \neq a_i$ and $z \neq b_i$ is adjacent to a_i and b_i for each $i \in [0,d]$ (Observation 190) that $\mathtt{anc}(q) \cup \mathtt{V}(a,b)$ is a clique. Thus, we let $1, \ldots, 2d+l+1$ be the vertices of $\mathcal{K}^*_{A_q}$, and we add all edges between these numbers to $\mathcal{K}^*_{A_q}$. As usual, we use h to define the parent relation and transfer all module, inclusion and integration colors according to the rules in (6.21). Remember that a_{d-1} is the parent of a in $\mathtt{T}_{f_1}$.[3] We let $\{b_{d-1}\}$ be the affiliation color of d and $\{a_{d-1}\}$ be the affiliation color of $2d$.

Let $k \in [d]$ and let $m \in [d, d+l]$. Note that $\mathtt{V}(a,b) = \{a_d, a_{d+1}, \ldots, a_{d+l}\}$. Let $R_{k,m}(q)$ be the set of all affiliated subbundle pairs $r \in R_k(q)$ where $r = [u,v]_{a_{k-1}, b_{k-1}}$ for $[u,v] \in \mathrm{C}^{\mathrm{triv}}_k(q)$ and $\mathtt{sd}_{f_1}(z([u,v],[a_k,b_k])) = m$. Then for all $r = [u,v]_{a_{k-1},b_{k-1}} \in R_{k,m}(q)$ we have $z([u,v],[a_k,b_k]) = a_m$. Thus, $R_k(q)$ is the union of all $R_{k,m}(q)$ where $m \in [d, d+l]$. We show that there is a linear order $\leq_{k,m}$ on the set of all extended copies $\mathcal{K}^*_r$ where $r \in R_{k,m}(q)$. In order to show this, we only need to show that there are linear orders $\leq_I$ and $\leq_A$, respectively, on the set of all integration colors and on the set of all affiliation colors occurring in any extended copy $\mathcal{K}^*_r$ for $r \in R_{k,m}(q)$.

First, let us show that there is a linear order $\leq_A$ on the set of all affiliation colors occurring in extended copies $\mathcal{K}^*_r$ where $r \in R_{k,m}(q)$. Each $r \in R_{k,m}(q)$ has the affiliation $[a_{k-1}, b_{k-1}]$. Therefore in each extended copy $\mathcal{K}^*_r$ with $r \in R_{k,m}(q)$ there can only occur the affiliation colors $\{\emptyset_{\mathrm{aff}}\}$, $\{a_{k-1}\}$, $\{b_{k-1}\}$ and $\{a_{k-1}, b_{k-1}\}$. We can identify the vertex $f \in \mathtt{F}$ with $a, b \in \mathtt{V}_f$, the first vertex of the path induced by $\mathtt{V}(a,b)$ in $\mathtt{T}_f$ and its parent in $\mathtt{T}_f$, which we supposed to be a_{d-1}. Thus, we let $\{\emptyset_{\mathrm{aff}}\} <_A \{a_{k-1}\} <_A \{b_{k-1}\} <_A \{a_{k-1}, b_{k-1}\}$. (If b_{d-1} is the parent of the first vertex of the path induced by $\mathtt{V}(a,b)$ in $\mathtt{T}_f$, we define $\{\emptyset_{\mathrm{aff}}\} <_A \{b_{k-1}\} <_A \{a_{k-1}\} <_A \{a_{k-1}, b_{k-1}\}$.)

Let us show that there is a linear order $\leq_I$ on the set of all integration colors occurring in extended copies $\mathcal{K}^*_r$ where $r \in R_{k,m}(q)$. As all affiliated subbundle pairs $r \in R_{k,m}(q)$ have the same affiliation $[a_{k-1}, b_{k-1}]$, Observation 271 implies that there is at most one integration color that contains l as its first component for every $l \leq k-1$. Clearly, for $l > k$ there are no integration colors with l as first component. For $l = k$ there might be multiple integration colors with l as first component. The set $R_{k,m}(q)$ contains only affiliated subbundle pairs r where $r = [u,v]_{[a_{k-1},b_{k-1}]}$ such that $[u,v] \in \mathrm{C}^{\mathrm{triv}}_k(q)$ and $\mathtt{sd}_{f_1}(z([u,v],[a_k,b_k])) = m$. Then $z([u,v],[a_k,b_k]) = a_m$. Thus, the only integration color with $l = k$ as first component is $\{(k, a_m)\}$. We obtain the linear order $<_I$ if we define $(i,v) <_I (i',v')$ whenever $i < i'$ for all integration colors $(i,v), (i',v') \in \mathbb{N} \times \mathtt{V}$ occurring in any extended copy $\mathcal{K}^*_r$ where $r \in R_{k,m}(q)$.

In the following we describe the construction of $\mathcal{K}^*_q$. We already assigned number vertices to the vertices in A_q and defined $\mathcal{K}^*_{A_q}$. Now we basically attach a copy of each extended copy $\mathcal{K}^*_r$ for $r \in R(q)$ to $\mathcal{K}^a_{A_q}$. We start with $k = d+1$ and $m = d+l$ and attach copies of the extended copies $\mathcal{K}^*_r$ for $r \in R_k(q)$ for each $k \in [d]$ and each $m \in [d, d+l]$ in the order given by the linear order $\leq_{k,m}$.

Let $k \in [d]$ and let $m \in [d, d+l]$. In order to attach a copy of the extended copy $\mathcal{K}^*_r$ for $r \in R_{k,m}(q)$ to $\mathcal{K}^*_{A_q}$, we first identify the number vertices of $\mathcal{K}^*_r$ that correspond to vertices in $\mathtt{anc}(q) \cup \mathtt{V}(a,b)$. Let $r \in R_{k,m}(q)$. Let $r = [u,v]_{a_{k-1},b_{k-1}}$ where

[3] This is important if $a = b$.

$[u,v] \in \mathrm{C}_k^{\mathrm{triv}}(q)$. Then $z([u,v],[a_k,b_k]) = a_m$. The set $\mathcal{W}_{\mathtt{anc}}(r)$ contains the set $\mathtt{anc}(r)$ and therefore the side ancestors $a_{k-1},\dots,a_0$ and $b_{k-1},\dots,b_0$. Since $a_m \in V(u,v)$, we have $a_m \in V(u,v) \subseteq W_k^r(u,v) \subseteq \mathcal{W}_k(r)$. It follows from Observation 277 that $\mathcal{W}_k(r)$, and therefore also $\mathcal{W}_{\mathtt{anc}}(r)$, contains the vertices $a_m,\dots,a_k$. Since $z([u,v],[a_k,b_k]) = a_m$, Observation 250 implies that the vertices $a_{d+l},\dots,a_{m+1}$ and $b_{d-1},\dots,b_k$ are not in $V(u,v)$. Since $a_{d+l},\dots,a_{m+1}$ and $b_{d-1},\dots,b_k$ are in $\mathtt{V}(a_k,b_k)$ (Observation 267), it follows that $a_{d+l},\dots,a_{m+1}$ and $b_{d-1},\dots,b_k$ are not in $\mathcal{W}(r) \subseteq \mathcal{W}_{\mathtt{anc}}(r)$ by Lemma 278. Hence, $\mathcal{W}_{\mathtt{anc}}(r) \cap (\mathtt{anc}(q) \cup \mathtt{V}(a,b)) = \{a_m,\dots,a_0,b_{k+1},\dots,b_0\}$. We can identify the number vertex corresponding to a_m in $\mathcal{K}_r^*$ with help of the integration color. The integration color of the vertex corresponding to a_m is (k,a_m). We can identify the number vertex corresponding to b_{k-1} with help of the affiliation color. Either there exists a number vertex with affiliation color $\{b_{k-1}\}$ in $\mathcal{K}_r^*$ or $\mathcal{H}_r^*$ is symmetric in a_{k-1} and b_{k-1}. Since there is (only) one vertex with integration color (k,a_m) in $\mathcal{K}_r^*$, which is a descendant of the number vertex corresponding to a_{k-1} but not of the number vertex corresponding to b_{k-1} ($a_m = p_k(r) \notin \mathtt{O}$, Observation 270), $\mathcal{H}_r^*$ cannot be symmetric in a_{k-1} and b_{k-1}. Thus, there exists a number vertex with affiliation color $\{b_{k-1}\}$ in $\mathcal{K}_r^*$. Using the parent relation we can determine all number vertices corresponding to ancestors of a_m and b_{k-1} in $\mathcal{K}_r^*$. Consequently, we can identify the number vertices corresponding to the vertices in $\mathtt{anc}(q) \cup \mathtt{V}(a,b)$ in $\mathcal{K}_r^*$.

Now that we can identify the number vertices of $\mathcal{K}_r^*$ that correspond to vertices in A_q, we can attach the extended copies $\mathcal{K}_r^*$ to $\mathcal{K}_{A_q}^*$. This is done analogously to the case where $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ is non-minimal consistent. When we attach the copies of $\mathcal{K}_r^*$ for $r \in R_{k,m}(q)$ to $\mathcal{K}_{A_q}^*$, we remove the affiliation colors, and remove all integration colors (i,v) where $i \geq k$. As $[a_{k-1},b_{k-1}]$ is the affiliation of r, we have $p_j(r) = p_j(q)$ for all $j \in J(r)$ where $j < k$ (Observation 271), and we can (and have to) copy the integration colors (j,v) where $j < k$.

After attaching all $\mathcal{K}_r^*$ for $r \in R(q)$ to $\mathcal{K}_{A_q}^*$ we obtain $\mathcal{K}_q^*$. Let us look at the integration colors of the number vertices in $\mathcal{K}_q^*$. Let $j \in [0,d]$. Let us show that if $j \in J(q)$ and $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(q)$, then there is a number vertex in $\mathcal{K}_q^*$ with the integration color $(j,p_j(q))$. Let $j \in J(q)$ and $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(q)$. If $p_j(q) \in A_q$, then there is a vertex in $\mathcal{K}_{A_q}^*$, and therefore in $\mathcal{K}_q^*$, with the integration color $(j,p_j(q))$. Let $p_j(q) \in \mathcal{W}(q) \setminus A_q$. Then there is a $k \in [d]$ and an $r \in R_k(q)$ such that $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(r)$ by Lemma 290. If $j < k$, we copied the integration color $(j,p_j(r)) = (j,p_j(q))$ (Observation 271). Let $j \geq k$. Then $p_j(q) \in \mathtt{V}(a_j,b_j) \subseteq \mathtt{V}(a_k,b_k)$ according to Observation 267. Let $r = [u,v]_{a_{k-1},b_{k-1}}$ where $[u,v] \in \mathrm{C}_k^{\mathrm{triv}}(q)$. Since $\mathtt{anc}(r) \setminus \{u,v\} \subseteq \mathtt{anc}(q)$ (Observation 266), we have $p_j(q) \in \mathcal{W}(r)$. Thus, $p_j(q) \in \mathcal{W}(r) \cap \mathtt{V}(a_k,b_k)$. It follows from Observation 278 that $p_j(q) \in \mathtt{V}(u,v) \cap \mathtt{V}(a_k,b_k)$. Since $[u,v] \in \mathrm{C}_k^{\mathrm{triv}}(q)$, the vertex $z := z([u,v],[a_k,b_k])$ is in $\mathtt{V}(a,b)$. By Observation 250, $z \notin \mathtt{O}$. Consequently, $p_j(q) \in \mathtt{V}(u,v) \cap \mathtt{V}(a_k,b_k) \subseteq \mathtt{anc}(q) \cup \mathtt{V}(a,b)$ by Observation 245. Since $p_j(q) \in \mathcal{W}(q) \setminus A_q$, the case where $p_j(q) \in \mathcal{W}_{\mathtt{anc}}(r)$ for an $r \in R_k(q)$ and $j \geq k$ cannot occur.

Defining $\mathcal{K}_q^*$ in FP+C

We can define $\mathcal{K}_q^*$ in simultaneous inflationary fixed-point logic.

Clearly, we can represent each affiliated subbundle pair $q \in V_{\mathcal{T}_{\mathrm{gen}}}$ by a 4-tuple of vertices in $\mathtt{V}$. Let $\check{q} \in \mathtt{V}^4$ denote the 4-tuple that represents q. (cf. Section 6.5.4, Defining the

Genealogical Decomposition Tree in FP+C). Since the genealogical decomposition tree is definable in FP+C, there exists an FP+C-formula which defines the total preorder on $V_{\mathcal{T}_{\text{gen}}}$ that is induced by the linear order on the extended heights.

Let $\mathcal{S} := \{V, E, T, L, Z, I, A\}$. For each relation $S \in \mathcal{S}$ we define $Q_S := \bigcup_{q \in V_{\mathcal{T}_{\text{gen}}}} \{\check{q}\} \times S_q^{\mathcal{K}}$. We can use a simultaneous inflationary fixed-point operator that recursively defines the relations Q_S for $S \in \mathcal{S}$ in relational variables X_S for $S \in \mathcal{S}$. For $S \in \mathcal{S}$ let X_S^i be the relation defined in round i of the recursion within the simultaneous inflationary fixed-point operator.

For each node $q \in V_{\mathcal{T}_{\text{gen}}}$ of minimal extended height, it should be easy to see that we can define the extended copy $\mathcal{K}_q^*$, that is, all relations $S_q^{\mathcal{K}}$ with $S \in \mathcal{S}$, in FP+C. We construct our simultaneous inflationary fixed-point formula such that initially for all q of minimal extended height and all $S \in \mathcal{S}$ every tuple in $\{\check{q}\} \times S_q^{\mathcal{K}}$ is added to the relation X_S^1. In round $i > 1$ of the recursion, we first check whether for all $r \in R(q)$ the extended copy $\mathcal{K}_r^*$ is already defined. We can do this for $r \in R(q)$, for example, by testing whether there exists a number $k \in N(\mathtt{U})$ such that $(\check{r}, k) \in X_V^{i-1}$. If for all $r \in R(q)$ the extended copy $\mathcal{K}_r^*$ is already defined, we use the extended copies $\mathcal{K}_r^*$ to define $\mathcal{K}_q^*$, that is, all relations $S_q^{\mathcal{K}}$ with $S \in \mathcal{S}$. From the description of the construction of $\mathcal{K}_q^*$, it should be verifiable that the relations $S_q^{\mathcal{K}}$ with $S \in \mathcal{S}$ can be defined in FP+C given X_S^{i-1} for all $S \in \mathcal{S}$.

Part II.

L-Recursion and a New Logic for Logarithmic Space

Introduction

We introduce a new logic for logarithmic space in this part. We define the logic LREC, an extension of first-order logic with counting, and show that it captures logarithmic space on directed trees. Afterwards, we present an extension $\mathsf{LREC}_=$ of LREC. The logic $\mathsf{LREC}_=$ captures logarithmic space on the class of interval graphs and the class of chordal claw-free graphs (and on the class of undirected trees). Except for the results regarding chordal claw-free graphs, the results of this part are joint work with Martin Grohe, André Hernich and Bastian Laubner, and have been published in [32] and [33].

The logic LREC is an extension of first-order logic with counting by a "limited recursion operator". The logic is more complicated than the transitive closure and fixed-point logics commonly studied in descriptive complexity, and it may look rather artificial at first sight. To explain the motivation for this logic, recall that fixed-point logics may be viewed as extensions of first-order logic by fixed-point operators that allow it to formalize recursive definitions in the logics. LREC is based on an analysis of the amount of recursion allowed in logarithmic space computations. The idea of the limited recursion operator is to control the depth of the recursion by a "resource term", thereby making sure that we can evaluate the recursive definition in logarithmic space. Another way to arrive at the logic is based on an analysis of the classes of Boolean circuits that can be evaluated in logarithmic space. We will take this route when we introduce the logic in Chapter 7.

LREC is easily seen to be (semantically) contained in FP+C. We show that LREC contains DTC+C, and as LREC captures LOGSPACE on directed trees, this containment is strict. Moreover, LREC is not contained in TC+C. Then we prove that undirected graph reachability is not definable in LREC. Hence, LREC does not contain transitive closure logic TC, not even in its symmetric variant STC, and therefore LREC is strictly contained in FP+C.

It can be argued that our proof of the inability of LREC to express graph reachability reveals a weakness in our definition of the logic rather than a weakness of the limited recursion operator underlying the logic: LREC is not closed under (first-order) logical reductions. To remedy this weakness, we introduce an extension $\mathsf{LREC}_=$ of LREC. It turns out that undirected graph reachability is definable in $\mathsf{LREC}_=$ (this is a convenient side effect of the definition and not a deep result). Thus $\mathsf{LREC}_=$ strictly contains symmetric transitive closure logic with counting. We prove that $\mathsf{LREC}_=$ captures LOGSPACE on the class of interval graphs. Afterwards, we show that the class of chordal claw-free graphs admits $\mathsf{LREC}_=$-definable canonization. On the one hand, this proves that canonization of chordal claw-free graphs is possible in logarithmic space. On the other hand, it shows that $\mathsf{LREC}_=$ captures LOGSPACE on the class of chordal claw-free graphs.

This part is organized as follows: We introduce the logic LREC in Chapter 7 and prove that its data complexity is in LOGSPACE. Then in Chapter 8, we prove that directed tree isomorphism and canonization are definable in LREC. As a consequence, LREC captures LOGSPACE on directed trees. In Chapter 9, we study the expressive power of LREC and prove that undirected graph reachability is not definable in LREC. The extension $\mathsf{LREC}_=$ is introduced in Chapter 10. Finally, our results on interval graphs and chordal claw-free graphs are presented in Chapter 11 and Chapter 12.

7. The Logic LREC

In this chapter, we introduce LREC as a first step towards the logic $\mathsf{LREC}_=$, to be introduced in Chapter 10. LREC is already expressive enough to capture LOGSPACE on directed trees, but still lacks several important properties. For example, it is unable to capture LOGSPACE on undirected trees and interval graphs (cf. Remark 350), and is not closed under first-order reductions (Chapter 10). On the other hand, although $\mathsf{LREC}_=$ could have been introduced without the detour via LREC, its definition is much easier to grasp by developing an understanding of LREC first.

Let us start our development of LREC by looking at how certain kinds of Boolean circuits can be evaluated in logarithmic space.

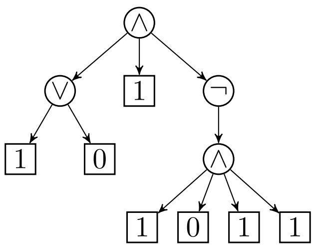


Figure 7.1.: Boolean formula

Figure 7.1 shows a *Boolean formula*, i.e., a Boolean circuit whose underlying graph is a tree. It is easy to evaluate such circuits in logspace: Start at the output node, determine the value of the first child recursively, then determine the value of the second child, and so on. For example, for a node that corresponds to an $\wedge$-gate, we determine the values of the children (one after the other) recursively until we reach a child with value 0 or the last child. Once we reached a child with value 0, we know the value of the node is 0 as well. If we reached the last child and its value is 1, we know the values of all children must have been one, and we know the value of the node is 1. To evaluate the formula we only have to store the current node and its value (if it has been determined already), since the parent node and the next child of the parent (if any) are uniquely determined by the current node. It is known that Boolean formula evaluation is complete for LOGSPACE under NC^1-reductions [1].[1] In contrast, Boolean *circuit* evaluation is PTIME-complete.

Let us now turn to formulas with *threshold gates*, which, in addition to Boolean gates, may contain gates of the form "$\geq i$" for a number i; such a gate outputs 1 if, and only if,

[1] Boolean formula evaluation is only complete for LOGSPACE if input formulas are represented as graphs (e.g., by the list of all edges plus gate types). It was however shown in [6] that the problem is complete for NC^1 under AC^0-reductions if input formulas are given by their natural string encoding.

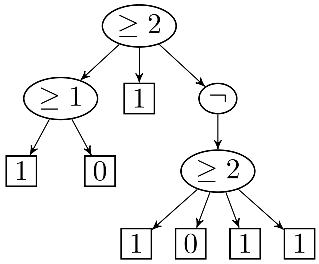


Figure 7.2.: Boolean formula with threshold gates

at least i input gates are set to 1. An example is shown in Figure 7.2. To evaluate such formulas in logarithmic space, we again start at the root and evaluate the values of the children recursively. For each node we count how many 1-values we have seen already. To this end, when evaluating the values of the children of a node v, we begin with the child with the largest subtree (i.e. the subtree with the largest number of nodes) and proceed to children with smaller subtrees. Let s be the size of the subtree of v. Then the first child of v has a subtree of size at most $s/1$, the second one has a subtree of size at most $s/2$ and so on. Thus, the ith child of v in this order has a subtree of size at most s/i. As $\log(s) = \log(s/i) + \log(i)$, we can use $\log(s/i)$ bits to determine the value of the ith child and $\log(i)$ bits to store a counter for the number of 1-values seen so far. It is easy to extend the algorithm to formulas with other *arithmetic gates* such as *modulo-gates*.

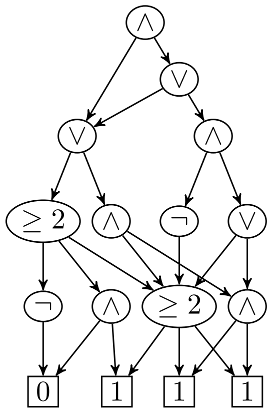


Figure 7.3.: Circuit with the 16-path property

As a more complicated example, let us consider the following type of circuit. A circuit C has the *m-path property* if for all paths P in C the product of the in-degrees of all but the first node on P is at most m. For example, formulas have the 1-path property, whereas the circuit in Figure 7.3 has the 16-path property. It is not hard to see that for every $k \geq 1$, circuits C having the $|C|^k$-path property can be evaluated in logspace. The idea here is very similar to the one for evaluating circuits with threshold gates. We start at the root node and evaluate the children recursively. After “entering” a node v from one of its parent nodes, say $p(v)$, we check whether v evaluates to 1 by counting the number of children that evaluate to one using the above-mentioned strategy, and return with this information to $p(v)$. In order to return to $p(v)$, we need to remember $p(v)$, which we do by storing the index of $p(v)$ among all the in-neighbors of v. This requires only $\log_2 d^-(v)$ bits of storage, where $d^-(v)$ denotes the in-degree of v. The space for writing down the index of the predecessor $p(v)$ for each vertex v on the path from the root to the currently visited vertex is thus bounded by the sum of the logarithms of the in-degrees of the vertices v on that path. Since C has the $|C|^k$-path property, this sum is bounded by $\log_2 |C|^k$, and thus logarithmic in the size of C. Another way of evaluating the circuit is to first "unravel" the circuit to a tree (i.e., a formula) which can be done in logarithmic space due to the $|C|^k$-path property, and then to evaluate the formula as above.

The logic LREC allows it to recursively define sets X of tuples based on graphs $\mathtt{G}$ that have the $|\mathtt{G}|^k$-path property for some $k \geq 1$.

We turn to the formal definition of the logic LREC. To define the syntax, let τ be a vocabulary. The set of all LREC[τ]-formulas is obtained by extending the formula formation rules of FO+C[τ] by the following rule: If $\bar{u}, \bar{v}, \bar{w}$ are compatible tuples of variables, $\bar{p}, \bar{r}$ are non-empty tuples of number variables, and $\varphi_{\mathtt{E}}$ and $\varphi_{\mathtt{C}}$ are LREC[τ]-formulas, then

$$\varphi := [\mathrm{lrec}_{\bar{u},\bar{v},\bar{p}}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{w}, \bar{r}) \tag{7.1}$$

is an LREC[τ]-formula, and we let $\mathrm{free}(\varphi) := \big(\mathrm{free}(\varphi_{\mathtt{E}}) \setminus (\tilde{u} \cup \tilde{v})\big) \cup \big(\mathrm{free}(\varphi_{\mathtt{C}}) \setminus (\tilde{u} \cup \tilde{p})\big) \cup \tilde{w} \cup \tilde{r}$.

To define the semantics of LREC[τ]-formulas, let A be a τ-structure and α an assignment in A. The semantics of LREC[τ]-formulas that are not of the form (7.1) is defined as usual.

Let φ be an LREC[τ]-formula of the form (7.1). We define a set $X \subseteq A^{\bar{u}} \times \mathbb{N}$ recursively as follows. We consider $\mathtt{E} := \varphi_{\mathtt{E}}[A, \alpha; \bar{u}, \bar{v}]$ as the edge relation of a directed graph $\mathtt{G}$ with vertex set $\mathtt{V} := A^{\bar{u}}$. Moreover, for each vertex $\bar{a} \in \mathtt{V}$ we think of the set $\mathtt{C}(\bar{a}) := \{\langle \bar{n} \rangle \mid \bar{n} \in \varphi_{\mathtt{C}}[A, \alpha[\bar{a}/\bar{u}]; \bar{p}]\}$ of integers as the label of $\bar{a}$. Let $\bar{a}\mathtt{E} := \{\bar{b} \in \mathtt{V} \mid \bar{a}\bar{b} \in \mathtt{E}\}$ and $\mathtt{E}\bar{b} := \{\bar{a} \in \mathtt{V} \mid \bar{a}\bar{b} \in \mathtt{E}\}$. Then, for all $\bar{a} \in \mathtt{V}$ and $\ell \in \mathbb{N}$,

$$(\bar{a}, \ell) \in X \ :\Longleftrightarrow\ \ell > 0 \ \text{ and } \ \left| \left\{ \bar{b} \in \bar{a}\mathtt{E} \ \middle|\ \left(\bar{b}, \left\lfloor \frac{\ell - 1}{|\mathtt{E}\bar{b}|} \right\rfloor \right) \in X \right\} \right| \in \mathtt{C}(\bar{a}). \tag{7.2}$$

Notice that X contains only elements $(\bar{a}, \ell)$ with $\ell > 0$. Hence, the recursion eventually stops at $\ell = 0$. We call X the *relation defined by φ in (A, α)*. Finally, we let

$$(A, \alpha) \models \varphi \ :\Longleftrightarrow\ (\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle) \in X.$$

Example 295 (Boolean circuit evaluation)**.** Let $\sigma := \{E, P_\wedge, P_\vee, P_\neg, P_0, P_1\}$. A Boolean circuit C may be viewed as a σ-structure, where $E(C)$ is the edge relation of C, and $P_\star(C)$ contains all $\star$-gates for $\star \in \{\wedge, \vee, \neg, 0, 1\}$. Suppose C has the $|C|$-path-property. Then,

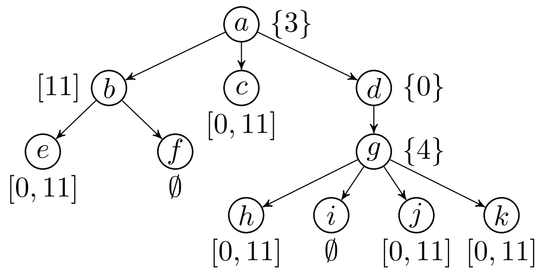
$$\varphi(z) := \exists r_1, r_2 \left([\mathrm{lrec}_{x,y,p}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](z, (r_1, r_2)) \wedge \forall r (r \leq r_1 \wedge r \leq r_2) \right)$$

with $\varphi_{\mathtt{E}}(x, y) := E(x, y)$ and

$$\varphi_{\mathtt{C}}(x, p) := (P_\wedge(x) \wedge \#y\, E(x, y) = p) \vee (P_\vee(x) \wedge p > 0) \vee (P_\neg(x) \wedge p = 0) \vee P_1(x)$$

states that gate z evaluates to 1.

For example, let C_1 be the circuit in Figure 7.1, and let α be the assignment in C_1 mapping z to the root of C_1, r_1 to 11, and r_2 to 11. Figure 7.4 shows the graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with $\mathtt{V} := C_1^x$, $\mathtt{E} := \varphi_{\mathtt{E}}[C_1, \alpha; x, y]$, and labels defined by $\varphi_{\mathtt{C}}$. The vertices a–k of $\mathtt{G}$ are



$C_1 = (V, E, P_\wedge, P_\vee, P_\neg, P_0, P_1)$
where
$P_\wedge = \{a, g\}$
$P_\vee = \{b\}$
$P_\neg = \{d\}$
$P_0 = \{f, i\}$
$P_1 = \{c, e, h, j, k\}$

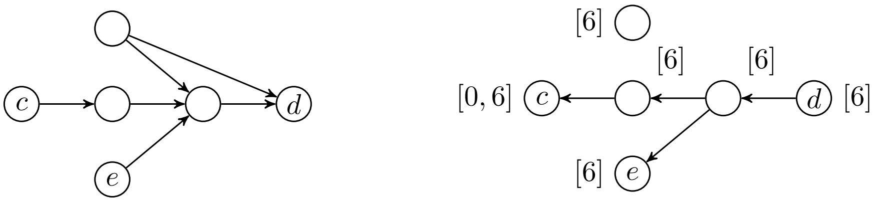
Figure 7.4.: The graph $\mathtt{G}$ for circuit C_1 from Example 295. Each vertex is labeled with a subset of $[0, 11]$.

precisely the vertices of C_1, and each vertex is labeled with a subset of $N(C_1) = [0, 11]$. Let X be the relation defined by $[\mathrm{lrec}_{x,y,p}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](z, (r_1, r_2))$ in (C_1, α). Let $\ell > 0$. For a leaf v of $\mathtt{G}$, we have $(v, \ell) \in X$ if and only if 0 occurs in the label of v. Hence, $(v, \ell) \in X$ for $v \in \{c, e, h, j, k\}$, but $(f, \ell) \notin X$ and $(i, \ell) \notin X$. Since $(e, \ell) \in X$, $(f, l) \notin X$ and 1 occurs in the label of b, we also have $(b, \ell + 1) \in X$. However, note that $(g, \ell + 1) \notin X$ because there are only three children v of g with $(v, \ell) \in X$, but 3 does not appear in the label of g. Consequently, $(d, \ell + 2) \in X$. Since we now have $(b, \ell + 2) \in X$, $(c, \ell + 2) \in X$, and $(d, \ell + 2) \in X$, we have $(a, \ell + 3) \in X$. Hence, $(a, \ell') \in X$ for all $\ell' \geq 4$, and in particular for $\ell' = \langle 11, 11 \rangle$. Therefore $(C_1, \alpha) \models \varphi$.

For the circuit C_1 above, we could have replaced the tuple (r_1, r_2) in the formula φ by a single number variable r. Due to the subtraction of 1 and the rounding when recursively evaluating whether $(v, \ell) \in X$ for $v \in \mathtt{V}$ and $\ell \in \mathbb{N}$ (see (7.2), the definition of X), a single number variable r does not suffice for circuits C that have the $|C|$-path property in general. However, as circuits are acyclic, each path has length at most $|C| - 1$ and a binary number (r_1, r_2) suffices to compensate the subtractions and roundings. ⌟

Example 296 (Deterministic transitive closure)**.** Let $G = (V, E)$ be a directed graph and $a, b \in V$. Then there is a *deterministic path* from a to b in G if there exists a path $v_1, \ldots, v_n$ from $a = v_1$ to $b = v_n$ in G such that for every $i \in [n-1]$, v_{i+1} is the unique out-neighbor of v_i. Figure 7.5a shows a directed graph with a deterministic path from c to d.

Let $\psi(\bar{u}, \bar{v})$ be an $\mathsf{LREC}[\tau]$-formula, and let $\bar{s}, \bar{t}$ be tuples of variables such that $\bar{u}, \bar{v}, \bar{s}, \bar{t}$ are pairwise compatible. We devise a formula $\varphi(\bar{s}, \bar{t})$ such that for any τ-structure A and assignment α in A, we have $(A, \alpha) \models \varphi(\bar{s}, \bar{t})$ if and only if in the graph $G = (V, E)$ defined by $V := A^{\bar{u}}$ and $E := \psi[A, \alpha; \bar{u}, \bar{v}]$ there is a deterministic path from $\alpha(\bar{s})$ to $\alpha(\bar{t})$. Note that there is such a path precisely if, in the graph obtained from G by reversing the edges, there is a path $v_n, \ldots, v_1$ from $\alpha(\bar{t})$ to $\alpha(\bar{s})$ such that for every $i \in [n-1]$, v_{i+1} is



(a) A graph with a deterministic path from c to d

(b) The associated labeled graph defined by $\varphi_{\mathtt{E}}$ and $\varphi_{\mathtt{C}}$

Figure 7.5.: Illustrations for Example 296

the unique in-neighbor of v_i. Therefore, we can choose φ like this:

$$\varphi \;:=\; \exists \bar{r}\,[\mathrm{lrec}_{\bar{v},\bar{u},\bar{p}}\ \varphi_{\mathtt{E}}(\bar{v},\bar{u}),\ \varphi_{\mathtt{C}}(\bar{v},\bar{p})](\bar{t},\bar{r}), \tag{7.3}$$

where $\bar{p}$ and $\bar{r}$ are $|\bar{u}|$-tuples of number variables, and

$$\varphi_{\mathtt{E}}(\bar{v},\bar{u}) \;:=\; \psi(\bar{u},\bar{v}) \wedge \forall \bar{v}'(\psi(\bar{u},\bar{v}') \to \bar{v}' = \bar{v}), \quad \varphi_{\mathtt{C}}(\bar{v},\bar{p}) \;:=\; \bar{v} = \bar{s} \vee (\bar{v} \neq \bar{s} \wedge \bar{p} \neq \bar{0}).$$

Informally, $\varphi_{\mathtt{E}}(\bar{v},\bar{u})$ removes all edges $\bar{a}\bar{b}$ of G, where $\bar{a}$ has more than one out-neighbor, and reverses the remaining edges. All that remains is to check whether there is a path from $\alpha(\bar{t})$ to $\alpha(\bar{s})$ in the graph defined by $\varphi_{\mathtt{E}}$. The node labeling formula $\varphi_{\mathtt{C}}$ is chosen in such a way that the latter is true if and only if $(\alpha(\bar{t}),\ell)$, for an $\ell \leq |V|$, appears in the relation X defined by $[\mathrm{lrec}_{\bar{v},\bar{u},\bar{p}}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{t},\bar{r})$ in (A,α). If, for example, G is the graph in Figure 7.5a, and if $\alpha(\bar{s}) = c$ and $\alpha(\bar{t}) = d$, then the labeled graph defined by $\varphi_{\mathtt{E}}$ and $\varphi_{\mathtt{C}}$ is as shown in Figure 7.5b, and it is easy to see that $(d,4) \in X$, while, for example, $(e,\ell) \notin X$ for all $\ell > 0$. ⌟

As from now, we use

$$[\mathrm{dtc}_{\,\bar{u},\bar{v}}\ \ \psi](\bar{s},\bar{t}) \tag{7.4}$$

as an abbreviation for the LREC-formula in (7.3). It follows that DTC $\leq$ LREC. Hence, we can do simple arithmetics in LREC, for example, addition and multiplication; we can also use the formula $\#\bar{u}\psi = \bar{p}$ (where $\bar{u}$ is a tuple of individual variables) to express that $|\{\bar{a} \in A^{\bar{u}} \mid (A,\alpha[\bar{a}/\bar{u}]) \models \psi\}| = \langle\alpha(\bar{p})\rangle_A$ in a structure A with an assignment α in A.

Remark 297. In the preceding example, the set X turned out to possess a certain monotonicity property: If $(\bar{a},\ell) \in X$ for some ℓ, then $(\bar{a},\ell') \in X$ for all $\ell' \geq \ell$. In general, however, the relation X defined by an lrec-operator does not possess this property. For instance, consider the circuit C_1 from Example 295, only now, let $i \in P_1$. Then $\mathtt{C}(i) = [0,11]$, and the relation X contains $(d,1)$ and $(d,2)$, but not $(d,3)$. ⌟

Example 298 (Reachability on graphs with maximum degree 2)**.** Let $\mathcal{H}$ be the class of all graphs H where each vertex has degree at most 2. By using the formula expressing deterministic reachability from Example 296, we show that there exists an LREC-formula ϕ expressing reachability on $\mathcal{H}$. Consider

$$\psi(\bar{x},\bar{y}) := E(x,x') \wedge E(y,y') \wedge x' = y \wedge x \neq y',$$

where $\bar{x} = (x, x')$ and $\bar{y} = (y, y')$. For each graph $H \in \mathcal{H}$, let $G' = (V', E')$ be the directed graph defined by $V' := H^{\bar{x}}$ and $E' := \psi[H; \bar{x}, \bar{y}]$. It is easy to see that $|\bar{a}E'| \leq 1$ for all $\bar{a} \in V'$. Further, let $a, b \in V(H)$ such that $a \neq b$, and a and b are not adjacent. Then a is connected to b in H, that is, there exists a path $a = a_0, a_1, \ldots, a_m = b$ of length $m \geq 2$ in H, precisely if there exists a directed path $(a, a') = (a_0, a_1), (a_1, a_2), \ldots, (a_{m-1}, a_m) = (b', b)$ of length $m - 1 \geq 1$ in G', that is, there exist $a', b' \in V(H)$ such that $a' \neq b$ and (a, a') is connected to (b', b) in G'. Thus,

$$\phi(s, t) := s = t \vee E(s, t) \vee \exists s' \exists t' (s' \neq t \wedge [\mathrm{dtc}_{\bar{x}, \bar{y}}\ \psi](s, s', t', t))$$

defines reachability on $\mathcal{H}$. ⌟

The following theorem shows that the data complexity of LREC is in LOGSPACE.

Theorem 299. *For every vocabulary τ, and every* LREC$[\tau]$*-formula φ there is a deterministic logspace Turing machine that, given a τ-structure A and an assignment α in A, decides whether $(A, \alpha) \models \varphi$.*

Proof. We proceed by induction on the structure of φ. The case where φ is *not* of the form (7.1) is easy. Let φ be of the form (7.1), i.e., let

$$\varphi \;=\; [\mathrm{lrec}_{\bar{u}, \bar{v}, \bar{p}}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{w}, \bar{r}).$$

Let $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ be the graph with vertex set $\mathtt{V} = A^{\bar{u}}$ and edge set $\mathtt{E} = \varphi_{\mathtt{E}}[A, \alpha; \bar{u}, \bar{v}]$. Further, let $\mathtt{C}(\bar{a}) := \{\langle \bar{n} \rangle \mid \bar{n} \in \varphi_{\mathtt{C}}[A, \alpha[\bar{u}/\bar{a}]; \bar{p}]\}$ for all $\bar{a} \in \mathtt{V}$, and let $X \subseteq \mathtt{V} \times \mathbb{N}$ be the relation defined by φ in (A, α). We construct a deterministic logspace Turing machine that decides whether $(\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle) \in X$.

The machine is constructed in two steps. The first step consists of constructing a deterministic logspace Turing machine M_1 that, given A and α as input, computes a labeled directed tree T that is obtained basically from "unraveling $\mathtt{G}$ starting at $\alpha(\bar{w})$ with "resource" $\langle \alpha(\bar{r}) \rangle$. The second step is to devise a deterministic logspace Turing machine M_2 that takes T as input and decides whether its root, $(\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle)$, belongs to X. The composition of M_1 and M_2 finally yields the desired machine.

We define a labeled directed tree T whose set W of vertices consists of all the sequences $((\bar{a}_0, \ell_0), \ldots, (\bar{a}_m, \ell_m))$ of pairs from $\mathtt{V} \times \mathbb{N}$ for some $m \in \mathbb{N}$ such that

1. $(\bar{a}_0, \ell_0) = (\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle)$,
2. $\bar{a}_{i+1} \in \bar{a}_i \mathtt{E}$ for all $i < m$, and
3. $\ell_{i+1} = \left\lfloor \frac{\ell_i - 1}{|\mathtt{E}\bar{a}_{i+1}|} \right\rfloor$ for all $i < m$.

There is an edge from $((\bar{a}_0, \ell_0), \ldots, (\bar{a}_m, \ell_m))$ to $((\bar{a}'_0, \ell'_0), \ldots, (\bar{a}'_{m'}, \ell'_{m'}))$ in T if $m' = m+1$, and $(\bar{a}'_i, \ell'_i) = (\bar{a}_i, \ell_i)$ for all $i \leq m$. We label each vertex $v = ((\bar{a}_0, \ell_0), \ldots, (\bar{a}_m, \ell_m)) \in W$ with the set $\mathtt{C}(v) := \mathtt{C}(\bar{a}_m)$, and with the number *fail*$(v) \in \{0, 1\}$ such that *fail*$(v) = 1$ iff $\ell_m = 0$. Note that *fail*$(v) = 1$ only if v is a leaf in T. Clearly, T is a labeled directed tree rooted at $(\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle)$.

Define $Y \subseteq W$ such that

$$v \in Y \iff |\{w \in Y \mid w \text{ is a child of } v\}| \in \mathtt{C}(v) \ \text{ and } \ \mathit{fail}(v) = 0 \quad (\text{for every } v \in W).$$

Claim 1. For every $v = ((\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m)) \in W$ we have $v \in Y$ if and only if $(\bar{a}_m, \ell_m) \in X$. In particular, $(\alpha(\bar{w}), \langle\alpha(\bar{r})\rangle) \in X$ if and only if $(\alpha(\bar{w}), \langle\alpha(\bar{r})\rangle) \in Y$.

Proof. The proof is by induction on the *rank* r_v of v in T: If v is a leaf in T, then $r_v = 0$; and if v is not a leaf in T, then r_v is one more than the maximum of the ranks of v's children. For every $v = ((\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m)) \in W$, let $\lambda(v) := (\bar{a}_m, \ell_m)$.

Suppose that $r_v = 0$, that is, v is a leaf in T. Consider $(\bar{a}, \ell) = \lambda(v)$. Then $\bar{a}\mathtt{E}$ is the empty set or $\ell = 0$. First consider the case that $\ell = 0$. In this case, $(\bar{a}, \ell) \notin X$ by the definition of X, and we have *fail*$(v) = 1$, which implies $v \notin Y$. Next consider the case that $\bar{a}\mathtt{E}$ is the empty set and $\ell > 0$. In this case,

$$v \in Y \iff 0 \in \mathtt{C}(v) = \mathtt{C}(\bar{a}) \iff (\bar{a}, \ell) \in X,$$

as desired.

Suppose now that $r_v = r + 1$, and that the claim is true for vertices w with $r_w \leq r$. In particular, since v is not a leaf we must have *fail*$(v) = 0$. This implies $\ell > 0$, and

$$\begin{aligned} v \in Y &\iff |\{w \in Y \mid w \text{ is a child of } v\}| \in \mathtt{C}(v) \\ &\iff |\{\lambda(w) \in X \mid w \text{ is a child of } v\}| \in \mathtt{C}(v) \quad \text{by the induction hypothesis.} \end{aligned} \tag{7.5}$$

Let W' be the set of all children w of v such that $\lambda(w) \in X$, and let $f \colon W' \to A^{\bar{u}}$ be such that for all $w \in W'$, $f(w)$ is the first component of $\lambda(w)$. Then f is a bijection from W' to the set of all tuples $\bar{b} \in \bar{a}\mathtt{E}$ with

$$\left(\bar{b}, \left\lfloor \frac{\ell - 1}{|\mathtt{E}\bar{b}|} \right\rfloor\right) \in X. \tag{7.6}$$

As a consequence, the number of all tuples $\bar{b} \in \bar{a}\mathtt{E}$ with (7.6) is precisely $|W'|$. Hence, by (7.5) and $\ell > 0$,

$$v \in Y \iff |W'| \in \mathtt{C}(v) = \mathtt{C}(\bar{a}) \iff \lambda(v) = (\bar{a}, \ell) \in X. \qquad \lrcorner$$

By Claim 1, it suffices to compute T, and use T to decide whether its root, $(\alpha(\bar{w}), \langle\alpha(\bar{r})\rangle)$, belongs to Y. This is precisely what the two machines M_1 and M_2 mentioned at the beginning of this proof do. We now prove the existence of such machines.

Claim 2. There is a deterministic logspace Turing machine that takes A and α as input and outputs T.

Proof. We first construct a deterministic logspace Turing machine M that takes A and α as input and outputs the vertices of T (represented as sequences $((\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m))$ as above). This machine makes use of a deterministic logspace Turing machine $M_{\mathtt{E}}$ that takes A, α and a pair $(\bar{a}, \bar{b}) \in \mathtt{V}^2$ as input and decides whether $\bar{a}\bar{b} \in \mathtt{E}$. Such a machine exists by the induction hypothesis. Once M is constructed, we can easily compute the edges and the labels of T, using a deterministic logspace Turing machine for computing the labels $\mathtt{C}(\bar{a})$ for each $\bar{a} \in \mathtt{V}$ as guaranteed by the induction hypothesis.

In what follows, we describe how M computes the vertices of T from A and α. We basically do a depth-first traversal (cf. Section 2.8.1) in $\mathtt{G}$ starting in $\alpha(\bar{w})$ with "resources" $\langle\alpha(\bar{r})\rangle$. Let k' be the length $|\bar{w}|$ of tuple $\bar{w}$ and k be the length $|\bar{r}|$ of tuple $\bar{r}$. In each step, we visit some vertex $\bar{a} \in \mathtt{V}$ with available "resources" $\ell < |N(A)|^k$. We also maintain the length m of the (non-simple) path $P = (\bar{a}_0, \dots, \bar{a}_m)$ on which $\bar{a} = \bar{a}_m$ was reached from $\alpha(\bar{w}) = \bar{a}_0$, and for each $i \in [m]$ a number $e_i \in [0, |\mathtt{E}\bar{a}_i| - 1]$ with the following property. For each $\bar{b} \in A^{\bar{u}}$ let $\bar{b}_0, \dots, \bar{b}_p$ be the elements of $\mathtt{E}\bar{b}$ ordered lexicographically according to their representation in the input string. Let $\mathrm{pre}(\bar{b}, i) := \bar{b}_i$. Then the number e_i will have the property that $\bar{a}_{i-1} = \mathrm{pre}(\bar{a}_i, e_i)$. As long as $l > 0$, i.e. our "resources" suffice, we move from $\bar{a}$ to some vertex $\bar{b} \in \bar{a}\mathtt{E}$, and we update ℓ to be

$$\mathrm{decr}(\ell, \bar{b}) \;:=\; \left\lfloor \frac{\ell - 1}{|\mathtt{E}\bar{b}|} \right\rfloor.$$

This ensures that the space needed to store the numbers $e_1, \dots, e_m$ is logarithmic in $|W|$ (which we shall prove later). Finally, upon visiting $\bar{a} = \bar{a}_m$, we write the sequence $(\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m)$ to the output tape, where the ℓ_i are the values for the available "resources" ℓ maintained along the (non-simple) path P.

More precisely, we proceed as follows. In each step, we store $\bar{a} \in \mathtt{V}$, $m \in \mathbb{N}$ and numbers $e_1, \dots, e_m$, additionally to the given values $\alpha(\bar{w})$ and $l_0 := \langle\alpha(\bar{r})\rangle$. (In the first step, we let $\bar{a} := \alpha(\bar{w})$ and $m := 0$.) Let $\bar{a}_0, \dots, \bar{a}_m$ be such that $\bar{a}_m = \bar{a}$, and $\bar{a}_{i-1} = \mathrm{pre}(\bar{a}_i, e_i)$ for each $i \in [m]$. (We will have $\bar{a}_0 = \alpha(\bar{w})$ in each step.) Further, let $\ell_0, \dots, \ell_m$ be such that $\ell_i = \mathrm{decr}(\ell_{i-1}, \bar{a}_i)$ for each $i \in [m]$. Let $\ell := \ell_m$. Notice that each of the $\bar{a}_i$ and ℓ_i can be computed in logarithmic space given $\bar{a}$, m, $e_1, \dots, e_m$, i and l_0 as input. For all $\bar{b} \in \mathtt{V}$ let $\preceq_{\bar{b}}$ be the linear order on $\bar{b}\mathtt{E}$ that corresponds to the lexicographic order on the representations of the vertices in $\bar{b}\mathtt{E}$ in the input string. We can perform the following moves:

- **down**: If $\bar{a}\mathtt{E} \neq \emptyset$ and $\ell > 0$, then we update $\bar{a}$ to be the first element in the linear order $\preceq_{\bar{a}}$ on $\bar{a}\mathtt{E}$, we increase m by one, and we let e_m be such that $\bar{a}_{m-1} = \mathrm{pre}(\bar{a}, e_m)$.
- **over**: If $m > 0$ and $\bar{a}$ is not the last element in $\preceq_{\bar{a}_{m-1}}$, then we update $\bar{a}$ to be the successor of $\bar{a}$ with respect to $\preceq_{\bar{a}_{m-1}}$, we keep m, and we update e_m such that $\bar{a}_{m-1} = \mathrm{pre}(\bar{a}, e_m)$.
- **up**: If $m > 0$, we update $\bar{a}$ to be $\bar{a}_{m-1}$, and decrease m by one.

The following procedure outputs all the vertices of T. Let $\bar{a} := \alpha(\bar{w})$ and $m := 0$. We repeat the following:

1. If the last move was **down**, **over** or there was no last move, this corresponds to a first visit of the vertex $\bar{a}$ with ℓ on the current path. Therefore, we write the sequence $(\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m)$ to the output tape. Then we perform the first move out of **down**, **over** or **up** that succeeds.
2. If the last move was **up**, this corresponds to a return from a child $\bar{b}$ of $\bar{a}$. Therefore, we do not write anything to the output tape. We perform the first move out of **over** or **up** that succeeds.

Maintaining the vertex $\bar{a} \in \mathtt{V}$ and the last move needs space $O(\log |U(A)|)$. Notice that

$$\ell_0 \;=\; \langle\alpha(\bar{r})\rangle \;\leq\; (|U(A)| + 1)^k - 1. \tag{7.7}$$

Since $\ell_i = \mathrm{decr}(\ell_{i-1}, \bar{a}_i) \leq \ell_{i-1} - 1$ for every $i \in [m]$, this implies

$$m \;\leq\; \ell_0 - \ell_m \;\leq\; \ell_0 \;<\; (|U(A)| + 1)^k.$$

Hence, m can be maintained in space $O(\log |U(A)|)$.

Now, $\ell_i = \left\lfloor \frac{\ell_{i-1}-1}{|\mathrm{E}\bar{a}_i|} \right\rfloor < \frac{\ell_{i-1}}{|\mathrm{E}\bar{a}_i|}$ yields $|\mathrm{E}\bar{a}_i| < \frac{\ell_{i-1}}{\ell_i}$ for $i < m$, as $\ell_i > 0$ for all $i < m$. Therefore,

$$\prod_{i=1}^{m} |\mathrm{E}\bar{a}_i| \;\leq\; \frac{\ell_0}{\ell_{m-1}} \cdot |\mathrm{E}\bar{a}_m| \;\leq\; \ell_0 \cdot |\mathrm{E}\bar{a}_m| \;\leq\; (|U(A)| + 1)^{k+k'}, \tag{7.8}$$

where the last inequality follows from (7.7). Each of the numbers e_i needs space $\eta_i := \lceil \log_2 |\mathrm{E}\bar{a}_i| \rceil$. Let $\mathcal{I}$ be the set of all $i \in [m]$ with $|\mathrm{E}\bar{a}_i| \geq 2$. As $2^{|\mathcal{I}|} \leq \prod_{i=1}^{m} |\mathrm{E}\bar{a}_i|$, we have $|\mathcal{I}| \leq \log_2(|U(A)| + 1)^{k+k'}$ by (7.8). Hence,

$$\sum_{i=1}^{m} \eta_i \;=\; \sum_{i \in \mathcal{I}} \lceil \log_2 |\mathrm{E}\bar{a}_i| \rceil \;\leq\; |\mathcal{I}| + \log_2 \prod_{i \in \mathcal{I}} |\mathrm{E}\bar{a}_i| \;\overset{(7.8)}{\leq}\; 2 \log_2(|U(A)| + 1)^{k+k'}.$$

As a consequence, we can store the numbers $e_1, \ldots, e_m$ as a single number e with $\eta := 2 \log_2(|U(A)| + 1)^{k+k'}$ bits, reserving η_i bits in e for the number e_i. We reserve the last η_m bits for e_m, of the remaining bits we reserve the last η_{m-1} bits for e_{m-1}, and so on. To extract e_i from e, we start by computing η_m from $\bar{a} = \bar{a}_m$. Let e_m be the number represented by the last η_m bits of e, and let $\bar{a}_{m-1} := \mathrm{pre}(\bar{a}_m, e_m)$. We then compute η_{m-1} from $\bar{a}_{m-1}$, let e_{m-1} be the number corresponding to bits $\eta - \eta_m - \eta_{m-1} + 1$ to $\eta - \eta_m$ of e, and let $\bar{a}_{m-2} := \mathrm{pre}(\bar{a}_{m-1}, e_{m-1})$. We continue this way until e_i is found. ⌟

Claim 3. There is a deterministic logspace Turing machine that takes T as input and decides whether the root $(\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle)$ of T belongs to Y.

Proof. Let $v_0 := (\alpha(\bar{w}), \langle \alpha(\bar{r}) \rangle)$. On input T, a deterministic logspace Turing machine can decide whether $v_0 \in Y$ as follows. The idea is to traverse the tree in a depth-first fashion (see Section 2.8.1), and count, for each node that is visited, the number of children that belong to Y. To implement this in logarithmic space, we proceed in steps as follows.

In each step, we are in a node v of T, which is v_0 in the first step. With each node v_i on the path $v_0, v_1, \ldots, v_m$ from v_0 to $v = v_m$ we associate $\ell_v(i)$ bits of memory for a counter $c(i)$ from 0 to $2^{\ell_v(i)} - 1$, where $\ell_v(i)$ will be specified below. The counter $c(i)$ counts the number of children of v_i that have already been processed and belong to Y (excluding the child of v_i in whose subtree we are currently in). We guarantee that the sum of the numbers $\ell_v(i)$ over $i \in [0, m]$ is bounded by $3 \cdot \log_2 |W|$. Moreover, it will be easy to determine $\ell_v(i)$ from v and i in logspace; so we can store the counters in a bit string of length at most $3 \cdot \log_2 |W|$, and identify the bits that belong to $c(i)$ from that bit string in logspace, given v and i. We ensure that there is always enough space to keep the counters in memory by visiting the children of each node in decreasing order of the number of nodes in the children's subtrees.

We need the following definitions and assumptions:

- The *size* $s(v)$ of a node $v \in W$ is the number of nodes in the subtree of T rooted at v. It is easy to compute this number in logarithmic space: all we need to do is to initialize a counter, iterate over all nodes of T, and for each such node move upwards and increment the counter by 1 if v is reached.

- Let $v \in W$, and let $w_1, \dots, w_p$ be the children of v such that $s(w_1) \geq s(w_2) \geq \cdots \geq s(w_p)$; children of the same size are ordered in lexicographic order based on their representation in the input string. For each node, we suppose that its children are ordered this way. Let $\text{child}(v, j) := w_j$ for every $j \in [p]$. The node $\text{child}(v, j)$ is easy to compute in logarithmic space, given v and j.
- Let $v \in W$, let $v_0, v_1, \dots, v_m$ be the path from v_0 to v, and let $i \in [0, m]$. Then

$$\ell_v(i) \; := \; \begin{cases} \lceil \log_2 j \rceil, & \text{if } i < m \text{ and } \text{child}(v_i, j) = v_{i+1}, \\ \lceil \log_2 |W| \rceil, & \text{if } i = m.^2 \end{cases}$$

  This number is easy to compute in logspace given v and i as input.

In the following we extend the depth-first tree traversal from Section 2.8.1 such that at each node v the counters $c(i)$, $i \in [0, m]$, of the ancestors of v are computed as well. At each step, we need to remember the current node v, m, the counters $c(0), \dots, c(m)$ and our last move. We extend the moves of the depth-first tree traversal from Section 2.8.1 as follows:

- **down**: If v has children, then we update v to be the first child of v, we increase m by one, and set $c(m) := 0$.
- **over**: If $m > 0$ and v is not the last child of its parent node, then we increase $c(m-1)$ by one if $c(m) \in \mathtt{C}(v)$ and $\mathit{fail}(v) = 0$ (we finished processing v), we update v to be its next sibling, and set $c(m) := 0$.
- **up**: If $m > 0$, we increase $c(m-1)$ by one if $c(m) \in \mathtt{C}(v)$ and $\mathit{fail}(v) = 0$ (we finished processing v), we update v to be its parent node, and decrease m by one.

Now, in the initial step, we set $v := v_0$, $m := 0$ and $c(0) := 0$. We repeat the following until no further move is possible, in which case we have reached the root of the tree. Then, we "accept" if and only if $c(0) \in \mathtt{C}(v_0)$ and $\mathit{fail}(v_0) = 0$.

1. If our last move was **down**, **over** or there was no last move, we perform the first move out of **down**, **over** or **up** that succeeds.
2. If our last move was **up**, then we are backtracking and we perform the first move out of **over** or **up** that succeeds.

It should be clear that this procedure correctly decides whether $v_0 \in Y$.

Concerning the space for the counters, let $j_0, j_1, \dots, j_{m-1}$ be such that $\text{child}(v_i, j_i) = v_{i+1}$ for every $i < m$. Then

$$\sum_{i<m} \ell_v(i) \; = \; \sum_{\substack{i<m \\ j_i \geq 2}} \lceil \log_2 j_i \rceil \; \leq \; \sum_{\substack{i<m \\ j_i \geq 2}} (1 + \log_2 j_i) \; = \; |\{i < m \mid j_i \geq 2\}| + \log_2 \prod_{i<m} j_i. \tag{7.9}$$

Now let us consider the size $s(v_{i+1})$ of the nodes v_{i+1} for every $i \in [0, m-1]$. We have

$$s(v_{i+1}) < \frac{s(v_i)}{j_i} \qquad \text{for every } i \in [0, m-1]. \tag{7.10}$$

To see this, let us consider $w_j := \text{child}(v_i, j)$ for every $j \leq j_i$. By the choice of $\text{child}(\cdot, \cdot)$, we have $s(w_1) \geq \cdots \geq s(w_{j_i})$. Consequently, if $s(w_{j_i}) = s(v_{i+1}) \geq s(v_i)/j_i$, then

[2] This ensures that $\ell_v(m)$ is easy to compute in logarithmic space as well.

$s(w_1) + \cdots + s(w_{j_i}) \geq s(v_i)$, which is impossible. As a consequence of (7.10), we have

$$|\{i < m \mid j_i \geq 2\}| < \log_2|W| \qquad \text{and} \qquad \prod_{i<m} j_i \overset{(7.10)}{<} \prod_{i<m} \frac{s(v_i)}{s(v_{i+1})} = \frac{s(v_0)}{s(v_m)} \leq |W|. \tag{7.11}$$

Altogether, this yields

$$\sum_{i \leq m} \ell_v(i) \overset{(7.9)}{\leq} |\{i < m \mid j_i \geq 2\}| + \log_2 \prod_{i<m} j_i + \log_2|W| + 1 \overset{(7.11)}{<} 3\log_2|W| + 1,$$

which implies $\sum_{i \leq m} \ell_v(i) \leq 3\log_2|W|$. ⌟

Altogether, this concludes the proof of Theorem 299. □

Remark 300. It follows from Example 296 that DTC+C $\leq$ LREC. This containment is strict as directed tree isomorphism is definable in LREC (we will show this in the next chapter), but not in DTC+C [18]. On the other hand, it is easy to see that the relation X defined by an LREC-formula of the form (7.1) in an interpretation (A, α) can be defined in fixed-point logic with counting. Hence, LREC $\leq$ FP+C, and this containment is strict since we show in Chapter 9 that undirected graph reachability is not LREC-definable. ⌟

8. Capturing LOGSPACE on Directed Trees

In this chapter we show that LREC captures LOGSPACE on the class of all directed trees. Our construction is based on Lindell's logspace tree canonization algorithm [53]. Note, however, that Lindell's algorithm makes essential use of a linear order on the tree's vertices that is given implicitly by the encoding of the tree. Here we do not have such a linear order, so we cannot directly translate Lindell's algorithm into an LREC-formula. We show that we can circumvent using the linear order if we have a formula for directed tree isomorphism. Hence, our first task is to construct such a formula.

8.1. Directed Tree Isomorphism

Let T be a directed tree. For every node $v \in V(T)$ let T_v be the subtree of T rooted at v, let $\text{size}(v) := |V(T_v)|$ be the *size* of v, and let $\#_s(v)$ be the number of children of v of size s. We construct an $\mathsf{LREC}[\{E\}]$-formula $\phi_{\cong}(x, y)$ that is true in a directed tree T with interpretations $v, w \in V(T)$ for x, y if and only if $T_v \cong T_w$. We assume that $|V(T)| \geq 4$, but it is easy to adapt the construction to directed trees with less than 4 vertices.

We implement the following recursive procedure to check whether $T_v \cong T_w$:

1. If $\text{size}(v) \neq \text{size}(w)$ or if $\#_s(v) \neq \#_s(w)$ for some $s \in [0, |V(T_v)| - 1]$, then return "$T_v \not\cong T_w$".
2. If for all children $\hat{v}$ of v there is a child $\hat{w}$ of w and a number k such that
   a) $T_{\hat{v}} \cong T_{\hat{w}}$,
   b) there are exactly k children $\mathring{w}$ of w with $T_{\hat{v}} \cong T_{\mathring{w}}$, and
   c) there are exactly k children $\mathring{v}$ of v with $T_{\mathring{v}} \cong T_{\hat{w}}$,

   then return "$T_v \cong T_w$".
3. Return "$T_v \not\cong T_w$".

Clearly, this procedure outputs "$T_v \cong T_w$" if and only if $T_v \cong T_w$.

To simplify the presentation we fix a directed tree T and an assignment α in T, but the construction will be uniform in T and α.

We construct a directed graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with labels $\mathtt{C}(v) \subseteq \mathbb{N}$ for each $v \in \mathtt{V}$ as follows. Let $\mathtt{V} := N(T) \times V(T)^4 \times N(T)$. The first component of each vertex is its *type*; the meaning of the other components will become clear soon. Although $\mathtt{G}$ will not be a tree, it is helpful to think of it as a *decision tree* for deciding $T_v \cong T_w$. For each pair $(v, w) \in V(T)^2$, we designate the vertex $\bar{a}_{v,w} = (0, v, w, v, w, 0)$ to stand for "$T_v \cong T_w$". Let us call (v, w) *easy* if v, w satisfy the condition in line 1 of the procedure (i.e., $\text{size}(v) \neq \text{size}(w)$, or $\#_s(v) \neq \#_s(w)$ for some $s \in [0, |V(T_v)| - 1]$). Note that the set of all such easy pairs is LREC-definable.[1] If (v, w) is easy, then $\bar{a}_{v,w}$ has no outgoing edges and $\mathtt{C}(\bar{a}_{v,w}) = \emptyset$.

[1] Using the dtc-operator (7.4) from Example 296 we can construct an $\mathsf{LREC}[\{E\}]$-formula defining the descendant relation between vertices in a directed tree, and using this formula it is easy to determine the size and the number of children of size s of a vertex.

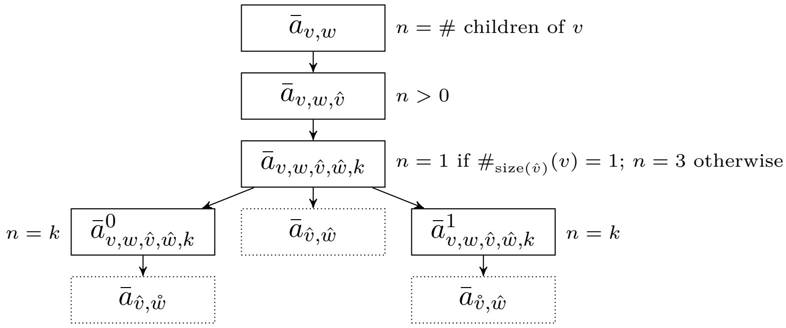


Figure 8.1.: Sketch of "decision tree" for deciding $T_v \cong T_w$. Here, $\hat{v}, \mathring{v}$ range over the children of v; $\hat{w}, \mathring{w}$ range over the children of w; and $k \in [\#_{\text{size}(\hat{v})}(v)]$. Moreover, $\hat{v}, \mathring{v}, \hat{w}, \mathring{w}$ all have the same size. Labels indicate which integers n belong to the set $\mathtt{C}(\bar{a})$ labeling each vertex $\bar{a}$. If $\hat{v}$ is the only child of v of size size$(\hat{v})$, then $\bar{a}_{\hat{v},\hat{w}}$ is the only child of $\bar{a}_{v,w,\hat{v},\hat{w},k}$.

On the other hand, if (v, w) is not easy, then $\mathtt{G}$ contains the following edges and labels (see Figure 8.1 for an illustration):

- The vertex $\bar{a}_{v,w}$ has an outgoing edge to $\bar{a}_{v,w,\hat{v}} := (1, v, w, \hat{v}, w, 0)$, for each child $\hat{v}$ of v. Furthermore, $\mathtt{C}(\bar{a}_{v,w}) = \{n\}$ where n is the number of children of v. This corresponds to "for all children $\hat{v}$ of v…" in the above procedure's step 2.
- The vertex $\bar{a}_{v,w,\hat{v}}$ has an outgoing edge to $\bar{a}_{v,w,\hat{v},\hat{w},k} := (2, v, w, \hat{v}, \hat{w}, k)$, for each child $\hat{w}$ of w with size$(\hat{w})$ = size$(\hat{v})$ and each $k \in [\#_{\text{size}(\hat{v})}(v)]$. Furthermore, $\mathtt{C}(\bar{a}_{v,w,\hat{v}}) = N(T) \setminus \{0\}$. This branching corresponds to "…there is a child $\hat{w}$ of w and a number k such that…".
- The vertex $\bar{a}_{v,w,\hat{v},\hat{w},k}$ has an outgoing edge to $\bar{a}_{\hat{v},\hat{w}}$. If $\hat{v}$ is the only child of v of size size$(\hat{v})$, then this is the only outgoing edge, and we let $\mathtt{C}(\bar{a}_{v,w,\hat{v},\hat{w},k}) = \{1\}$. Otherwise, there are additional outgoing edges to $\bar{a}^i_{v,w,\hat{v},\hat{w},k} = (3 + i, v, w, \hat{v}, \hat{w}, k)$ for $i \in \{0, 1\}$, and we let $\mathtt{C}(\bar{a}_{v,w,\hat{v},\hat{w},k}) = \{3\}$. This corresponds to conditions 2a–2c.
- The vertex $\bar{a}^0_{v,w,\hat{v},\hat{w},k}$ has outgoing edges to $\bar{a}_{\hat{v},\mathring{w}}$ for each child $\mathring{w}$ of w of size size$(\hat{v})$, and $\bar{a}^1_{v,w,\hat{v},\hat{w},k}$ has outgoing edges to $\bar{a}_{\mathring{v},\hat{w}}$ for each child $\mathring{v}$ of v of size size$(\hat{w})$ = size$(\hat{v})$. Furthermore, $\mathtt{C}(\bar{a}^i_{v,w,\hat{v},\hat{w},k}) = \{k\}$. The vertex $\bar{a}^i_{v,w,\hat{v},\hat{w},k}$ corresponds to condition 2b for $i = 0$, and to 2c for $i = 1$.

From the above description it should be easy to construct $\mathsf{LREC}[\{E\}]$-formulas $\varphi_{\mathtt{E}}(\bar{u}, \bar{u}')$ and $\varphi_{\mathtt{C}}(\bar{u}, p)$ with variable tuples $\bar{u} = (q_t, x, y, \hat{x}, \hat{y}, q_k)$ and $\bar{u}' = (q'_t, x', y', \hat{x}', \hat{y}', q'_k)$, such that $\varphi_{\mathtt{E}}[T; \bar{u}, \bar{u}'] = \mathtt{E}$, and $\varphi_{\mathtt{C}}[T, \bar{a}; p] = \mathtt{C}(\bar{a})$ for each $\bar{a} \in \mathtt{V}$.

Let

$$\phi_{\cong}(x, y) := \exists \bar{r}\, (\text{“}\forall \bar{r}'\ \bar{r} \geq \bar{r}'\text{”} \wedge [\text{lrec}_{\bar{u},\bar{u}',p}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}]((0, x, y, x, y, 0), \bar{r})),$$

where $\bar{r}$ is a 4-tuple of number variables. Let X be the relation defined by $\phi_{\cong}$ in (T, α). Then:

Lemma 301. *For all $v, w \in V(T)$ and all $\ell \geq \text{size}(v)^4$,*

$$(\bar{a}_{v,w}, \ell) \in X \quad \Longleftrightarrow \quad T_v \cong T_w.$$

Proof. The proof is by induction on size(v). Let size$(v) = 1$. If $(\bar{a}_{v,w}, \ell) \in X$, then (v, w) is not easy, which implies size$(w) = 1$ and hence $T_v \cong T_w$. Now suppose that $T_v \cong T_w$.

Then $\text{size}(w) = 1$ which implies that (v, w) is not easy. Furthermore, as v has no children in T, we know that $\bar{a}_{v,w}$ has no children in $\mathtt{G}$ and $\mathtt{C}(\bar{a}_{v,w}) = \{0\}$. Hence, $(\bar{a}_{v,w}, \ell) \in X$ for all $\ell \geq 1 = \text{size}(v)^4$.

In order to show Lemma 301 for vertices $v \in V(T)$ with $\text{size}(v) > 1$, let us observe the following. All vertices in Figure 8.1 except the type 0-vertices have exactly one incoming edge. For the in-degree of a type 0-vertex $\bar{a}_{v',w'}$ we show the following claim.

Claim 302. *Let (v, w) be not easy. Let v' be a child of v of size s', and let w' be a child of w of size s' in T. Further, let $\bar{a}_{v',w'}$ be a type 0-vertex.*

a) *If $\#_{s'}(v) = 1$, the in-degree $d_{v',w'}$ of $\bar{a}_{v',w'}$ is 1.*
b) *If $\#_{s'}(v) > 1$, the in-degree $d_{v',w'}$ of $\bar{a}_{v',w'}$ is at most $d := 3 \cdot \#_{s'}(v)^2$.*

Proof. First note that vertex $\bar{a}_{v',w'}$ can only have incoming edges from

1. vertices $\bar{a}_{v,w,v',w',k}$, where v and w are the (unique) parents of v' and w', respectively, and $k \in [\#_{s'}(v)]$;
2. vertices $\bar{a}^0_{v,w,v',\hat{w},k}$, where v, w, k are as above and $\hat{w}$ is a child of w of size s'; and
3. vertices $\bar{a}^1_{v,w,\hat{v},w',k}$, where v, w, k are as above and $\hat{v}$ is a child of v of size s'.

Let us deal with the case that $\#_{s'}(v) > 1$ first. Then clearly each of the three cases above yields at most $\#_{s'}(v)^2$ incoming edges. Hence, the in-degree of $\bar{a}_{v',w'}$ is at most d.

Now let $\#_{s'}(v) = 1$. Then case 1 yields one incoming edge for vertex $\bar{a}_{v',w'}$. Next, let us consider case 3. Since v' is the only child of v of size s', vertex $\bar{a}^1_{v,w,v',w',1}$ is the only candidate for an incoming edge. But as $\#_{s'}(v) = 1$, vertex $\bar{a}_{v',w'}$ is the only child of $\bar{a}_{v,w,v',w',1}$. Hence, there are no edges adjacent to vertex $\bar{a}^1_{v,w,v',w',1}$. Similarly, we can show for case 2 that there cannot exist an incoming edge from a vertex $\bar{a}^0_{v,w,v',w'',1}$ where w'' is a child of w of size s'. As a consequence, $\bar{a}_{v',w'}$ has exactly 1 incoming edge. ⌟

We use Claim 302 to prove the following claim.

Claim 303. *Let (v, w) be not easy. Let v' be a child of v of size s', and let w' be a child of w of size s' in T. Further, let $\bar{a}_{v',w'}$ be a type 0-vertex, and let $d_{v',w'}$ be the in-degree of vertex $\bar{a}_{v',w'}$. If $\ell \geq \text{size}(v)^4$, then $\ell' \geq (s')^4$, where*

a) $\ell' := \lfloor (\ell - 3)/d_{v',w'} \rfloor$ *if $\#_{s'}(v) = 1$, and*
b) $\ell' := \lfloor (\ell - 4)/d_{v',w'} \rfloor$ *if $\#_{s'}(v) > 1$.*

Proof. Let $s := \text{size}(v) - 1$. We have $s \geq s' \geq 1$.

a) If $\#_{s'}(v) = 1$, then $d_{v',w'} = 1$ by Claim 302, and we have

$$\ell' = \ell - 3 \geq (s + 1)^4 - 3 \geq s^4 \geq (s')^4.$$

b) If $\#_{s'}(v) > 1$, then $d_{v',w'} \leq d$ where $d = 3 \cdot \#_{s'}(v)^2$ by Claim 302, and we have

$$\begin{aligned}\ell' &\geq \frac{\ell-4}{d_{v',w'}} - 1 \geq \frac{\ell-4}{d} - 1 \overset{*}{\geq} \frac{\ell}{d} - \frac{4}{3\cdot 2^2} - 1 \geq \frac{(s+1)^4}{d} - 2 \overset{\circledast}{\geq} \frac{s^4 + 4\cdot s^3}{d} - 2 \\ &\overset{\star}{\geq} \frac{\#_{s'}(v)^4\cdot(s')^4 + 4\cdot\#_{s'}(v)^3\cdot(s')^3}{3\cdot\#_{s'}(v)^2} - 2 \geq \frac{\#_{s'}(v)^2}{3}(s')^4 + \frac{4\cdot\#_{s'}(v)}{3}(s')^3 - 2 \\ &\overset{*}{\geq} \frac{2^2}{3}(s')^4 + \frac{4\cdot 2}{3}(1)^3 - 2 \geq (s')^4.\end{aligned}$$

For the inequalities marked by the symbol "$*$" we use $\#_{s'}(v) \geq 2$, the inequality marked by "$\circledast$" is a consequence of the binomial theorem, and for the inequality marked by "$\star$" we use $s \geq \#_{s'}(v)\cdot s'$. ⌟

Now let $\mathrm{size}(v) > 1$. Let $\ell \geq \mathrm{size}(v)^4$. We show that $(\bar{a}_{v,w}, \ell) \in X$ if and only if $T_v \cong T_w$, for all $v, w \in V(T)$ where (v,w) is not easy. Since $(\bar{a}_{v,w}, \ell) \in X$ as well as $T_v \cong T_w$ implies that (v,w) is not easy, we obtain that the statements "$(\bar{a}_{v,w}, \ell) \in X$" and "$T_v \cong T_w$" are equivalent for all $v, w \in V(T)$.

Let (v,w) be not easy. This implies that $\mathrm{size}(w) = \mathrm{size}(v)$ and $\#_t(v) = \#_t(w)$ for all $t \in \mathbb{N}$. It is easy to see that $(\bar{a}_{v,w}, \ell) \in X$ if and only if $(\bar{a}_{v,w,\hat{v}}, \ell - 1) \in X$ for all children $\hat{v}$ of v in T. This again is precisely the case if for all children $\hat{v}$ of v in T there is a child $\hat{w}$ of w of size $\hat{s} := \mathrm{size}(\hat{v})$ in T and a number $k \in [\#_{\hat{s}}(v)]$ such that $(\bar{a}_{v,w,\hat{v},\hat{w},k}, \ell - 2) \in X$. If we continue this and go through the complete "decision tree" from Figure 8.1, we obtain that $(\bar{a}_{v,w}, \ell) \in X$ if and only if for all children $\hat{v}$ of v in T there is a child $\hat{w}$ of w of size $\hat{s} := \mathrm{size}(\hat{v})$ in T and a number $k \in [\#_{\hat{s}}(v)]$ such that

- $(\bar{a}_{\hat{v},\hat{w}}, \lfloor(\ell-3)/d_{\hat{v},\hat{w}}\rfloor) \in X$,
- there are exactly k children $\mathring{w}$ of w of size $\hat{s}$ such that $(\bar{a}_{\hat{v},\mathring{w}}, \lfloor(\ell-4)/d_{\hat{v},\mathring{w}}\rfloor) \in X$ if $\#_{\hat{s}}(v) > 1$, and
- there are exactly k children $\mathring{v}$ of v of size $\hat{s}$ such that $(\bar{a}_{\mathring{v},\hat{w}}, \lfloor(\ell-4)/d_{\mathring{v},\hat{w}}\rfloor) \in X$ if $\#_{\hat{s}}(v) > 1$,

where $d_{v',w'}$ denotes the in-degree of vertex $\bar{a}_{v',w'}$ for all $v', w' \in V(T)$. By Claim 303 and the induction hypothesis, this is equivalent to: For all children $\hat{v}$ of v in T there is a child $\hat{w}$ of w of size $\hat{s} := \mathrm{size}(\hat{v})$ in T and a number $k \in [\#_{\hat{s}}(v)]$ such that

- $T_{\hat{v}} \cong T_{\hat{w}}$,
- there are exactly k children $\mathring{w}$ of w of size $\hat{s}$ such that $T_{\hat{v}} \cong T_{\mathring{w}}$ if $\#_{\hat{s}}(v) > 1$, and
- there are exactly k children $\mathring{v}$ of v of size $\hat{s}$ such that $T_{\mathring{v}} \cong T_{\hat{w}}$ if $\#_{\hat{s}}(v) > 1$.

As $T_{v'} \cong T_{w'}$ implies $\mathrm{size}(v') = \mathrm{size}(w')$ for all $v', w' \in V(T)$ and the last two properties hold for $k = 1$ if $\#_{\hat{s}}(v) = 1$, this corresponds to Step 2 of the procedure given at the beginning of Section 8.1, which is equivalent to $T_v \cong T_w$ for not easy pairs (v,w). □

Corollary 304. *Let $v, w \in V(T)^2$. Then, $T \models \phi_{\cong}[v,w]$ if and only if $T_v \cong T_w$.*

Proof. Let $v, w \in V(T)^2$. Then $T \models \phi_{\cong}[v,w]$ holds precisely if $(\bar{a}_{v,w}, |N(T)|^{|\bar{r}|} - 1) \in X$. Furthermore, $|N(T)|^{|\bar{r}|} - 1 \geq |V(T)|^4 \geq \mathrm{size}(v)^4$. Therefore, by the preceding lemma, $(\bar{a}_{v,w}, |N(T)|^{|\bar{r}|} - 1) \in X$ is equivalent to $T_v \cong T_w$, and the corollary follows. □

8.2. Defining an Order on Directed Trees

Lindell's tree canonization algorithm is based on a logspace-computable linear order on isomorphism classes of directed trees. We show that a slightly refined version of this order is LREC-definable.

Let T be a directed tree. For each $v \in V(T)$ let $\pi(v) := \big(\text{size}(v), \#_1(v), \dots, \#_{\text{size}(v)-1}(v)\big)$ be the *profile* of v.[2] Let $\preceq$ be the total preorder on $V(T)$, where $v \prec w$ whenever $\pi(v) < \pi(w)$ lexicographically, or $\pi(v) = \pi(w)$ and the following is true:

(A) Let $v_1, \dots, v_m$ and $w_1, \dots, w_m$ be the children of v and w, respectively, ordered such that $v_1 \preceq \dots \preceq v_m$ and $w_1 \preceq \dots \preceq w_m$. Then there is an $i \in [m]$ with $v_i \prec w_i$, and for all $j < i$ we have $v_j \preceq w_j$ and $w_j \preceq v_j$.

Note that $v \preceq w$ and $w \preceq v$ if, and only if, $T_v \cong T_w$. We denote $v \preceq w$ and $w \preceq v$ by $v \simeq w$. We show that $\preceq$ is LREC-definable.

In order to do this, we first present a statement (B) that is equivalent to (A). For all $t, u \in V(T)$ let $\theta_u(t)$ be the number of children u' of u with $u' \simeq t$. We call a child $\hat{v}$ of v *good* if $\theta_v(\hat{v}) > \theta_w(\hat{v})$ and for all children v' of v with $\text{size}(v') < \text{size}(\hat{v})$ we have $\theta_v(v') = \theta_w(v')$.

(B) There is a good child $\hat{v}$ of v, a child $\hat{w}$ of w of size $s := \text{size}(\hat{v})$ and a $k \in [0, \#_s(v) - 1]$ such that:
   1. there are exactly k children $\mathring{w}$ of w of size s with $\mathring{w} \prec \hat{v}$;

   and if $k > 0$, then

   2. for all k children w' of w of size s with $w' \prec \hat{v}$ we have $\theta_v(w') = \theta_w(w')$;
   3. $\hat{w} \prec \hat{v}$;
   4. there are exactly $k - \theta_v(\hat{w})$ children $\mathring{v}$ of v of size s with $\mathring{v} \prec \hat{w}$; and
   5. for all $k - \theta_v(\hat{w})$ children v' of v of size s with $v' \prec \hat{w}$ we have $\theta_v(v') = \theta_w(v')$.

Lemma 305. *Let v, w be nodes of a directed tree T, and $\pi(v) = \pi(w)$. Then* (A) *and* (B) *are equivalent.*

Proof. Let v, w be nodes of T, and $\pi(v) = \pi(w)$. First, we show that (A) implies (B). Let $v_1, \dots, v_m$, $w_1, \dots, w_m$ and $i \in [m]$ be as in (A). Notice that for all nodes v', w' of T, $\text{size}(v') < \text{size}(w')$ implies $v' \prec w'$. We let $\hat{v} := v_i$. Clearly, v_i is a good child. Let $s := \text{size}(\hat{v})$. Let $[l, l'] \subseteq [m]$ be the set of indices j where $\text{size}(v_j) = s$ (which is precisely the case if $\text{size}(w_j) = s$). If there does not exists a $j \in [l, l']$ such that $w_j \prec v_i$, we let $k := 0$ and $\hat{w}$ be an arbitrary child of w of size s. Then all conditions of (B) are satisfied. Now suppose there is a $j \in [l, l']$ such that $w_j \prec v_i$ and let j be maximal. Then, we let $\hat{w} := w_j$ and $k := j - l + 1$. Again, it is not hard to verify that that all conditions of (B) are satisfied.

Next, we prove that (B) implies (A). Let $\hat{v}$, $\hat{w}$, s and k be as in (B), and let $v_1, \dots, v_m$ and $w_1, \dots, w_m$ be the children of v and w, respectively, such that $v_1 \preceq \dots \preceq v_m$ and $w_1 \preceq \dots \preceq w_m$. Let $[l, l'] \subseteq [m]$ be again the set of indices j where $\text{size}(v_j) = s$. As $\hat{v}$ is a good child and $\pi(v) = \pi(w)$, we have $v_j \simeq w_j$ for all $j < l$. Thus, without loss of generality assume $l = 1$. If $k = 0$, then $\hat{v} \preceq w_1$ (condition 1), and therefore $v_1 \preceq w_1$.

[2] Lindell's order can be obtained by replacing $\pi(v)$ with $\pi'(v) := \big(\text{size}(v), \#\text{children of } v\big)$.

If $v_1 \prec w_1$, then statement (B) is satisfied with $i := 1$. If $v_1 \simeq w_1$, then $\hat{v} \simeq w_1$, and (B) is satisfied for $i := \theta_w(w_1) + 1$ as $\hat{v}$ is a good child. Now let $k > 0$. By condition 1 we have $w_k \prec \hat{v}$ and $\hat{v} \preceq w_{k+1}$. Condition 2 implies that there are indices $i_1, \ldots, i_k$ with $i_1 < \cdots < i_k$ such that $v_{i_j} \simeq w_j$ for all $j \in [k]$. Let k' be maximal such that $v_{k'} \simeq w_k$. Then $k' = i_k$. Condition 2 also yields that we have $\theta_v(v_j) = \theta_w(v_j)$ only for those $j \in [k']$ where $j = i_r$ for an $r \in [k]$. Since $\hat{w} \prec \hat{v}$ (condition 3), there exists a $j \in [k]$ such that $\hat{w} = w_j$ and $\theta_v(\hat{w}) = \theta_w(\hat{w})$. It follows from condition 4 and 5 that $k' = k$ and $v_k \simeq \hat{w}$. Thus, $v_j \simeq w_j$ for all $j \in [k]$. Now we can argue analogous to the case where $k = 0$: As $\hat{v} \preceq w_{k+1}$, we have $v_{k+1} \preceq w_{k+1}$. If $v_{k+1} \prec w_{k+1}$, then statement (B) is satisfied with $i := k + 1$. If $v_{k+1} \simeq w_{k+1}$, then $\hat{v} \simeq w_{k+1}$, and (B) is satisfied for $i := k + \theta_w(w_{k+1}) + 1$ as $\hat{v}$ is a good child. □

To simplify the presentation, we again fix a directed tree T and an assignment α, and we assume that $|V(T)| \geq 5$.

We apply the lrec-operator to the following graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with labels $\mathtt{C}(v) \subseteq \mathbb{N}$ for each $v \in \mathtt{V}$. Let $\mathtt{V} := N(T) \times V(T)^4 \times N(T)$. For each $(v, w) \in V(T)^2$, the vertex $\bar{a}_{v,w} = (0, v, w, v, w, 0)$ represents "$v \prec w$". If $\pi(v) < \pi(w)$, then $\bar{a}_{v,w}$ has no outgoing edges and $\mathtt{C}(\bar{a}_{v,w}) = \{0\}$. If $\pi(v) > \pi(w)$, then $\bar{a}_{v,w}$ has no outgoing edges and $\mathtt{C}(\bar{a}_{v,w}) = \emptyset$. Note that the relation "$\pi(v) \leq \pi(w)$" is LREC-definable. Suppose that $\pi(v) = \pi(w)$. Then $\mathtt{G}$ contains the following edges and labels, and the "decision tree" in Figure 8.2 checks precisely the conditions in (B).

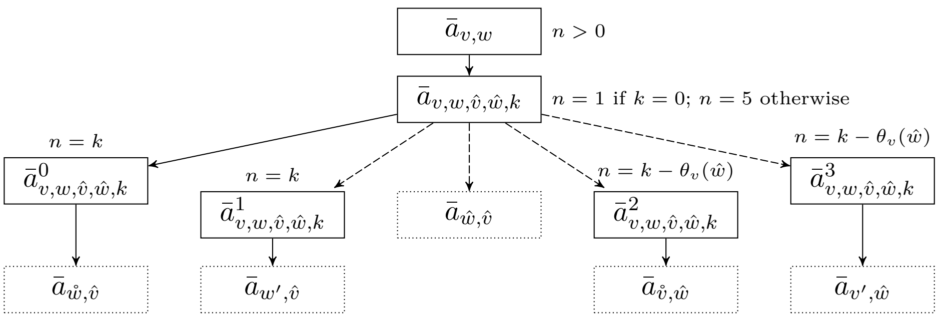


Figure 8.2.: Gadget for deciding $v \prec w$ when $\pi(v) = \pi(w)$. Here, $\hat{v}$ ranges over good children of v, nodes $\hat{w}, \mathring{w}, w'$ range over children of w, nodes $\mathring{v}, v'$ over children of v, and $k \in [0, \#_{\text{size}(\hat{v})}(v) - 1]$. Moreover, $\hat{w}, \mathring{w}, w', \mathring{v}, v'$ all have size $s := \text{size}(\hat{v})$, and $\theta_v(w') = \theta_w(w')$ and $\theta_v(v') = \theta_w(v')$. The dashed edges from $\bar{a}_{v,w,\hat{v},\hat{w},k}$ to $\bar{a}_{\hat{w},\hat{v}}$ and to $\bar{a}^i_{v,w,\hat{v},\hat{w},k}$ for $i \in \{1, 2, 3\}$, exist only if $k > 0$. Labels indicate which integers n belong to the set $\mathtt{C}(\bar{a})$ labeling each vertex $\bar{a}$.

- The vertex $\bar{a}_{v,w}$ has an outgoing edge to $\bar{a}_{v,w,\hat{v},\hat{w},k} := (1, v, w, \hat{v}, \hat{w}, k)$, for each good child $\hat{v}$ of v, each child $\hat{w}$ of w of size $s := \text{size}(\hat{v})$ and each $k \in [0, \#_s(v) - 1]$. Furthermore, $\mathtt{C}(\bar{a}_{v,w}) = N(T) \setminus \{0\}$. This branching corresponds to "There is a good child $\hat{v}$ of v, a child $\hat{w}$ of w of size $s := \text{size}(\hat{v})$ and a $k \in [0, \#_s(v) - 1]$ such that…"

- The vertex $\bar{a}_{v,w,\hat{v},\hat{w},k}$ has an outgoing edge to $\bar{a}^0_{v,w,\hat{v},\hat{w},k} := (2, v, w, \hat{v}, \hat{w}, k)$. If $k = 0$, this is the only outgoing edge, and we let $\mathtt{C}(\bar{a}_{v,w,\hat{v},\hat{w},k}) = \{1\}$. Otherwise, there are additional outgoing edges to $\bar{a}_{\hat{w},\hat{v}}$ and to $\bar{a}^i_{v,w,\hat{v},\hat{w},k} = (2 + i, v, w, \hat{v}, \hat{w}, k)$ for all $i \in \{1, 2, 3\}$, and we let $\mathtt{C}(\bar{a}_{v,w,\hat{v},\hat{w},k}) = \{5\}$. This corresponds to conditions 1–5 in (B).
- The vertex $\bar{a}^0_{v,w,\hat{v},\hat{w},k}$ has outgoing edges to $\bar{a}_{\mathring{w},\hat{v}}$ for each child $\mathring{w}$ of w of size $\text{size}(\hat{v})$, and $\bar{a}^1_{v,w,\hat{v},\hat{w},k}$ has outgoing edges to $\bar{a}_{w',\hat{v}}$ for each child w' of w of size $\text{size}(w') = \text{size}(\hat{v})$ with $\theta_v(w') = \theta_w(w')$. Note that $\theta_v(w') = \theta_w(w')$ is LREC-definable. Furthermore, $\mathtt{C}(\bar{a}^i_{v,w,\hat{v},\hat{w},k}) = \{k\}$ for $i \in \{0, 1\}$. The vertex $\bar{a}^i_{v,w,\hat{v},\hat{w},k}$ corresponds to condition 1 for $i = 0$, and to 2 for $i = 1$.
- The vertex $\bar{a}^2_{v,w,\hat{v},\hat{w},k}$ has outgoing edges to $\bar{a}_{\mathring{v},\hat{w}}$ for each child $\mathring{v}$ of v of size $\text{size}(\hat{w})$, and $\bar{a}^3_{v,w,\hat{v},\hat{w},k}$ has outgoing edges to $\bar{a}_{v',\hat{w}}$ for each child v' of v of size $\text{size}(v') = \text{size}(\hat{w})$ with $\theta_v(v') = \theta_w(v')$. Moreover, $\mathtt{C}(\bar{a}^i_{v,w,\hat{v},\hat{w},k}) = \{k - \theta_v(\hat{w})\}$ for $i \in \{2, 3\}$. The vertex $\bar{a}^i_{v,w,\hat{v},\hat{w},k}$ corresponds to condition 4 for $i = 2$, and to 5 for $i = 3$.

Using the formula $\phi_{\cong}$ from the previous section it is now straightforward to construct $\mathsf{LREC}[\{E\}]$-formulas $\varphi_{\mathtt{E}}(\bar{u}, \bar{u}')$ and $\varphi_{\mathtt{C}}(\bar{u}, p)$ that define the edge relation $\mathtt{E}$ of $\mathtt{G}$ and the sets $\mathtt{C}(\bar{a})$ for each $\bar{a} \in \mathtt{V}$, where $\bar{u}$ and $\bar{u}'$ are as in the definition of $\phi_{\cong}$. Let

$$\phi_{\prec}(x, y) := \exists \bar{r}\, (\text{“}\forall \bar{r}'\ \bar{r} \geq \bar{r}'\text{”} \wedge [\text{lrec}_{\bar{u},\bar{u}',p}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}]((0, x, y, x, y, 0), \bar{r}),$$

where $\bar{r}$ is a 5-tuple of number variables. Let X be the relation defined by $\phi_{\prec}$ in (T, α). We then have:

Lemma 306. *For all $v, w \in V(T)$ and all $l \geq \text{size}(v)^5$,*

$$(\bar{a}_{v,w}, \ell) \in X \quad \Longleftrightarrow \quad v \prec w.$$

Proof. The proof is similar to the proof of Lemma 301.

We prove this lemma by induction on $\text{size}(v)$. Suppose $\text{size}(v) = 1$. If $v \prec w$, then $\pi(v) < \pi(w)$, and by the construction of $\mathtt{G}$ this immediately implies $(\bar{a}_{v,w}, \ell) \in X$ for all $\ell \geq 1 = \text{size}(v)^5$. Now let $(\bar{a}_{v,w}, \ell) \in X$. Then $\pi(v) \leq \pi(w)$. We cannot have $\pi(v) = \pi(w)$, because then $0 \notin \mathtt{C}(\bar{a}_{v,w})$ (see Figure 8.2), so that X would contain at least one tuple of the form $\big((1, v, w, \hat{v}, \cdot, \cdot), \ell - 1\big)$ with $\hat{v}$ a child of v. But such a tuple does not exist, since v has no children. It follows that $\pi(v) < \pi(w)$ which implies $v \prec w$.

In order to show Lemma 306 for vertices $v \in V(T)$ with $\text{size}(v) > 1$, let us consider the in-degrees of the vertices of $\mathtt{G}$ first. All vertices in Figure 8.2 except the type 0-vertices have exactly one incoming edge. For the in-degree of a type 0-vertex $\bar{a}_{v',w'}$ we show the following claim.

Claim 307. *Let $\pi(v) = \pi(w)$. Let v' be a child of v of size s', and let w' be a child of w of size s' in T. Further, let $\bar{a}_{v',w'}$ be a type 0-vertex.*

a) *If $\#_{s'}(v) = 1$, the in-degree $d_{v',w'}$ of $\bar{a}_{v',w'}$ is at most 1.*
b) *If $\#_{s'}(v) > 1$, the in-degree $d_{v',w'}$ of $\bar{a}_{v',w'}$ is at most $d := 5 \cdot \#_{s'}(v)^2$.*

Proof. Vertex $\bar{a}_{v',w'}$ can only have incoming edges from

1. vertices $\bar{a}_{w,v,w',v',k}$, where v and w are the (unique) parents of v' and w', respectively, and $k+1 \in [\#_{s'}(v)]$ (note that $\#_{s'}(v) = \#_{s'}(w)$ since $\pi(v) = \pi(w)$);
2. vertices $\bar{a}^0_{w,v,w',\hat{v},k}$, where v, w, k are as above and $\hat{v}$ is a child of v of size s';
3. vertices $\bar{a}^1_{w,v,w',\hat{v},k}$, where v, w, k are as above and $\hat{v}$ is a child of v of size s';
4. vertices $\bar{a}^2_{v,w,\hat{v},w',k}$, where v, w, k are as above and $\hat{v}$ is a good child of v of size s';
5. vertices $\bar{a}^3_{v,w,\hat{v},w',k}$, where v, w, k are as above and $\hat{v}$ is a good child of v of size s'.

Let us deal with the case that $\#_{s'}(v) > 1$ first. Clearly, each of the five cases above yields at most $\#_{s'}(v)^2$ incoming edges in this case. Hence, the in-degree of $\bar{a}_{v',w'}$ is at most d.

Now let $\#_{s'}(v) = 1$. Then $k = 0$ for all possible in-neighbors of $\bar{a}_{v',w'}$. If $k = 0$, then vertices of type 1 only have edges to vertices of type 2. Thus, vertex $\bar{a}_{v',w'}$ can only have in-neighbors of type 2. Since $\#_{s'}(v) = 1$ there exists only one child $\hat{v}$ of v of size s'. Hence, $\bar{a}_{v',w'}$ can at most have an incoming edge from the vertex $\bar{a}^0_{w,v,w',\hat{v},0}$. (This edge only exists if w' is a good child of w of size s.) ⌟

We use Claim 307 to prove the following claim.

Claim 308. *Let $\pi(v) = \pi(w)$. Let v' be a child of v of size s', and let w' be a child of w of size s' in T. Further, let $\bar{a}_{v',w'}$ be a type 0-vertex, and let $d_{v',w'} > 0$ be the in-degree of vertex $\bar{a}_{v',w'}$. If $\ell \geq \mathrm{size}(v)^5$, then $\ell' \geq (s')^5$, where $\ell' := \lfloor (\ell - 3)/d_{v',w'} \rfloor$.*

Proof. Let $s := \mathrm{size}(v) - 1$. We have $s \geq s' \geq 1$.

a) If $\#_{s'}(v) = 1$, then $d_{v',w'} = 1$ by Claim 307, and we have

$$\ell' = \ell - 3 \geq (s+1)^5 - 3 \geq s^5 \geq (s')^5.$$

b) If $\#_{s'}(v) > 1$, then $d_{v',w'} \leq d$ where $d = 5 \cdot \#_{s'}(v)^2$ by Claim 307, and we have

$$\begin{aligned}
\ell' &\geq \frac{\ell - 3}{d_{v',w'}} - 1 \geq \frac{\ell - 3}{d} - 1 \overset{*}{\geq} \frac{\ell}{d} - \frac{3}{5 \cdot 2^2} - 1 \geq \frac{(s+1)^5}{d} - 2 \geq \frac{s^5 + 5 \cdot s^4}{d} - 2 \\
&\overset{\star}{\geq} \frac{\#_{s'}(v)^5 \cdot (s')^5 + 5 \cdot \#_{s'}(v)^4 \cdot (s')^4}{5 \cdot \#_{s'}(v)^2} - 2 \geq \frac{\#_{s'}(v)^3}{5}(s')^5 + \frac{5 \cdot \#_{s'}(v)^2}{5}(s')^4 - 2 \\
&\overset{*}{\geq} \frac{2^3}{5}(s')^5 + \frac{5 \cdot 2^2}{5}(1)^4 - 2 \geq (s')^5.
\end{aligned}$$

For the inequalities marked by the symbol "$*$" we use $\#_{s'}(v) \geq 2$, the inequality marked by "⊛" is a consequence of the binomial theorem, and for the inequality marked by "$\star$" we use $s \geq \#_{s'}(v) \cdot s'$. ⌟

Now let $\mathrm{size}(v) = s + 1$ for some $s \geq 1$. Let $\ell \geq \mathrm{size}(v)^5$.

First, suppose that $\pi(v) \neq \pi(w)$. If $(\bar{a}_{v,w}, \ell) \in X$, then $\pi(v) < \pi(w)$, which implies $v \prec w$. If $v \prec w$, then $\pi(v) < \pi(w)$, and it follows from the construction of G that $(\bar{a}_{v,w}, \ell) \in X$ for all $\ell \geq 1$, in particular, for all $\ell \geq \mathrm{size}(v)^5$.

Let $\pi(v) = \pi(w)$. By going through the complete "decision tree" from Figure 8.2, we obtain that $(\bar{a}_{v,w}, \ell) \in X$ if, and only if, there exists a good child $\hat{v}$ of v, a child $\hat{w}$ of w of size $\hat{s} := \mathrm{size}(\hat{v})$ and a $k \in [0, \#_s(v) - 1]$ such that

- there are exactly k children $\mathring{w}$ of w of size $\hat{s}$ where $(\bar{a}_{\mathring{w},\hat{v}}, \lfloor(\ell-3)/d_{\mathring{w},\hat{v}}\rfloor) \in X$,

and if $k > 0$, then

- there are exactly k children w' of w of size $\hat{s}$ with $\theta_v(w') = \theta_w(w')$ such that $(\bar{a}_{w',\hat{v}}, \lfloor(\ell-3)/d_{w',\hat{v}}\rfloor) \in X$,
- $(\bar{a}_{\hat{w},\hat{v}}, \lfloor(\ell-2)/d_{\hat{w},\hat{v}}\rfloor) \in X$,
- there are exactly $k - \theta_v(\hat{w})$ children $\mathring{v}$ of v of size $\hat{s}$ with $(\bar{a}_{\mathring{v},\hat{w}}, \lfloor(\ell-3)/d_{\mathring{v},\hat{w}}\rfloor) \in X$,
- there are exactly $k - \theta_v(\hat{w})$ children v' of v of size $\hat{s}$ with $\theta_v(v') = \theta_w(v')$ such that $(\bar{a}_{v',\hat{w}}, \lfloor(\ell-3)/d_{v',\hat{w}}\rfloor) \in X$.

As usual, $d_{v'',w''}$ denotes the in-degree of vertex $\bar{a}_{v'',w''}$ for all $v'', w'' \in V(T)$. By Claim 308 and the induction hypothesis, this is equivalent to: There exists a good child $\hat{v}$ of v, a child $\hat{w}$ of w of size $\hat{s} := \text{size}(\hat{v})$ and a $k \in [0, \#_s(v) - 1]$ such that

- there are exactly k children $\mathring{w}$ of w of size $\hat{s}$ where $\mathring{w} \prec \hat{v}$;

and if $k > 0$, then

- there are exactly k children w' of w of size $\hat{s}$ with $\theta_v(w') = \theta_w(w')$ such that $w' \prec \hat{v}$,
- $\hat{w} \prec \hat{v}$,
- there are exactly $k - \theta_v(\hat{w})$ children $\mathring{v}$ of v of size $\hat{s}$ with $\mathring{v} \prec \hat{w}$, and
- there are exactly $k - \theta_v(\hat{w})$ children v' of v of size $\hat{s}$ with $\theta_v(v') = \theta_w(v')$ such that $v' \prec \hat{w}$.

It follows from Lemma 305 that the above is equivalent to $v \prec w$. ☐

Corollary 309. *Let $v, w \in V(T)$. Then, $T \models \phi_\prec[v, w]$ if and only if $v \prec w$.*

8.3. Canonizing Directed Trees

We now construct an LREC-formula $\gamma(p, p')$ such that for every directed tree T we have $T \cong ([|V(T)|], \gamma[T; p, p'])$. As a linear order is available on the number sort, this yields an LREC-canonization of the class of directed trees. Since DTC captures LOGSPACE on ordered structures [41], we immediately obtain:

Theorem 310. LREC *captures* LOGSPACE *on the class of directed trees.*

Directed tree isomorphism is in LOGSPACE by Lindell's tree canonization algorithm, but not TC+C-definable [18]. Thus, we obtain:

Corollary 311. LREC $\not\leq$ TC+C *on the class of all directed trees.*

We use l-recursion to define a set $X \subseteq V(T) \times N(T)^2$ (for the sake of simplicity, we omit the "resources" in this description) such that for every node $v \in V(T)$ the set $X_v := \{(m, n) \in N(T)^2 \mid (v, m, n) \in X\}$ is the edge relation of an isomorphic copy $([|V(T_v)|], X_v)$ of T_v. Each node of T is numbered by its position in the preorder traversal sequence, e.g., the root is numbered 1, its first child v_1 is numbered 2, its second child v_2 is numbered $2 + \text{size}(v_1)$, and so on.

To apply the lrec operator, we define a graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with labels $\mathtt{C}(v) \subseteq \mathbb{N}$ for each $v \in \mathtt{V}$ as follows. Let $\mathtt{V} := V(T) \times N(T)^2$, where $(v, m, n) \in \mathtt{V}$ stands for "$(m, n) \in X_v$?".

If v is a leaf, then X_v should be empty, so for all $m, n \in N(T)$ we let (v, m, n) have no outgoing edges and define $\mathtt{C}((v, m, n)) := \emptyset$. Suppose that v is not a leaf and w is a child of v. Let D_w be the set of all children w' of v with $w' \prec w$, and let e_w be the number of children w' of v with $T_w \cong T_{w'}$. For each $i \in [0, e_w - 1]$, the set X_v will contain an edge from 1 to $p_{w,i} := 2 + \sum_{w' \in D_w} \mathrm{size}(w') + i \cdot \mathrm{size}(w)$, and the edges in $\{(p_{w,i} - 1 + m, p_{w,i} - 1 + n) \mid (m, n) \in X_w\}$. Hence, we let $(v, 1, p_{w,i})$ have no outgoing edges and define $\mathtt{C}((v, 1, p_{w,i})) := \{0\}$. Furthermore, for all $m, n \in N(T)$ and all $i < e_w$, we let $\bar{a} := (v, p_{w,i} - 1 + m, p_{w,i} - 1 + n)$ have an edge to (w, m, n) and define $\mathtt{C}(\bar{a}) := \{e_w\}$.

It is now easy to construct LREC-formulas $\varphi_{\mathtt{E}}(x_1, p_1, p_1', x_2, p_2, p_2')$ and $\varphi_{\mathtt{C}}(x_1, p_1, p_1', q)$ that define the graph $\mathtt{G}$ and the labels $\mathtt{C}(\cdot)$. Let

$$\gamma(p, p') := \exists x \exists r \Big(\text{“}x \text{ is the root”} \wedge \forall r'\, r' \leq r \wedge [\mathrm{lrec}_{(x_1, p_1, p_1'), (x_2, p_2, p_2'), q}\ \varphi_{\mathtt{E}}, \varphi_{\mathtt{C}}]\big((x, p, p'), r\big)\Big).$$

Noting that the in-degree of each vertex (v, m, n) is at most e_v, it is straightforward to show that γ defines an isomorphic copy of a directed tree:

Lemma 312. *Let X be the relation defined by formula γ in T, let $v \in V(T)$ and let $X_v := \{(m, n) \mid ((v, m, n), \ell) \in X$ for some $\ell \geq \mathrm{size}(v)\}$. Then $T_v \cong ([|V(T_v)|], X_v)$.*

Proof. The proof is by induction on $\mathrm{size}(v)$. Clearly, the lemma is true if $\mathrm{size}(v) = 1$. Suppose that $\mathrm{size}(v) = s + 1$. By the induction hypothesis, for each child w of v we have $T_w \cong ([|V(T_w)|], X_w)$.

Let $\ell \geq \mathrm{size}(v)$. For all children w of v and all $m, n \in N(T)$, the in-degree of (w, m, n) in $\mathtt{G}$ is at most e_w and $e_w \cdot \mathrm{size}(w) < \mathrm{size}(v)$. Thus, $\lfloor (\ell - 1)/e_w \rfloor \geq \lfloor (\mathrm{size}(v) - 1)/e_w \rfloor \geq \mathrm{size}(w)$. As a consequence, we have $\{(p_{w,i} - 1 + m, p_{w,i} - 1 + m) \mid (m, n) \in X_w\} \subseteq X_v$ for each child w of v and $i < e_w$. Furthermore, by construction, we have $(1, p_{w,i}) \in X_v$ for each child w of v and $i < e_w$, and there are no more edges. It is easy to see that $T_v \cong ([|V(T_v)|], X_v)$. □

8.4. Colored Directed Trees

The results of the previous sections extend to colored directed trees with a linear order on the color set and, in particular, to LO-colored directed trees (see Section 2.3.4 for exact definitions). In the following we only consider colored directed trees that have a linear order on their colors. It will be straightforward how to extend the results to LO-colored directed trees.

For each colored directed tree S, the linear order on the colors of S induces a total preorder $\trianglelefteq$ on the set $V(S)$ of nodes of S. Let $\preceq$ be as in Section 8.2. We define a refinement $\preceq'$ of $\preceq$ by letting $v \prec' w$ whenever $v \prec w$, or: $v \preceq w$ and $w \preceq v$ and $v \triangleleft w$. It should be obvious how to modify $\phi_{\prec}(x, y)$ to an LREC$[\{E, \leqslant\}]$-formula $\phi_{\prec'}(x, y)$ defining $\prec'$. Thus, there is an LREC-definable total preorder $\prec'$ on $V(S)$ which induces a linear order on the isomorphism classes of the colored subtrees of S.

We use the total preorder $\preceq'$ for canonization of colored directed trees S. We let the universe $U(S)$ of the colored directed tree S be the union of its set $V(S)$ of nodes and set $C(S)$ of colors. For a colored directed tree S, let $R(S)$ be the set of all tuples $(v, c, w, d) \in (V(S) \times C(S))^2$ where v is colored with c, the color of w is d and there is an edge from v to w in S. We call R the *colored edge relation* of colored directed tree S. For canonization we extend the vertex set $X \subseteq V(T) \times N(T)^2$ from Section 8.3 to a set $X' \subseteq V(S) \times (N(S) \times C(S))^2 \subseteq U(S) \times (N(S) \times U(S))^2$ such that for every $v \in V(S)$ the set $X'_v := \{(m, c, n, d) \in (N(S) \times C(S))^2 \mid (v, m, c, n, d) \in X'\}$ is the colored edge relation of an isomorphic copy of S_v.

To apply the lrec operator, we define a graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with labels $\mathtt{C}(v) \subseteq \mathbb{N}$ similar to the one in Section 8.3. We let $\mathtt{V} := U(S) \times (N(S) \times U(S))^2$. If v is a color or a leaf, then X'_v should be empty. Thus, we let (v, m, c, n, d) have no outgoing edges and define $\mathtt{C}((v, m, c, n, d)) := \emptyset$ for all $m, n \in N(S)$, $c, d \in U(S)$. In the following let us suppose $v \in V(S)$ is not a leaf and w is a child of v. Again, let D_w be the set of all children w' of v with $w' \prec w$, and let e_w be the number of children w' of v with $S_w \cong S_{w'}$. Further, let c_v and c_w be the respective colors of v and w in the colored directed tree S. Now, for each $i \in [0, e_w - 1]$, we let X'_v contain an edge from 1 colored by c_v to $p_{w,i} := 2 + \sum_{w' \in D_w} \text{size}(w') + i \cdot \text{size}(w)$ colored by c_w, and the "colored edges" in $\{(p_{w,i} - 1 + m, c, p_{w,i} - 1 + n, d) \mid (m, c, n, d) \in X_w\}$. Therefore, we let $(v, 1, c_v, p_{w,i}, c_w)$ have no outgoing edges and define $\mathtt{C}((v, 1, c_v, p_{w,i}, c_w)) := \{0\}$. Moreover, for all $m, n \in N(T), c, d \in U(T)$ and all $i < e_w$, we let $\bar{a} := (v, p_{w,i} - 1 + m, c, p_{w,i} - 1 + n, d)$ have an edge to (w, m, c, n, d) and define $\mathtt{C}(\bar{a}) := \{e_w\}$.

Now it is not hard to construct the formulas necessary to define the graph $\mathtt{G}$ and the labels $\mathtt{C}(\cdot)$. We can use these formulas to obtain an LREC-formula that defines the colored edge relation of an isomorphic copy of a colored directed tree with the set $[|V(S)|]$ of nodes and the set $C(S)$ of colors. Note that there is a linear order on the set $C(S)$ of colors, and of course, the linear order on the number sort is a linear order on the set of nodes. Thus, we can use the colored edge relation of the isomorphic copy to define a canon of S.

We obtain that the class of colored directed trees with a linear order on its color set admits LREC-definable canonization. Analogously we can show that there also is an LREC-canonization of LO-colored directed trees. As a consequence, we obtain the following corollary.

Corollary 313. LREC *captures* LOGSPACE *on the class of colored directed trees with a linear order on its color set and on the class of* LO*-colored directed trees.*

9. Inexpressibility of Reachability in Undirected Graphs

While LREC captures LOGSPACE on directed trees, its expressive power still lacks the ability to define certain important problems on undirected graphs that can be defined easily in other logics such as STC with logspace data complexity. As an example, we show in this chapter that LREC cannot define reachability in undirected graphs:

Theorem 314. *There is no* LREC$[\{E\}]$*-formula* $\varphi(x, y)$ *such that for all undirected graphs* G *and all* $v, w \in V(G)$, $G \models \varphi[v, w]$ *iff there is a path from* v *to* w *in* G.

As an immediate corollary we obtain:

Corollary 315. STC $\not\leq$ LREC

To prove Theorem 314, we show that reachability is not LREC-definable on a certain class of undirected graphs. This class, called $\mathcal{C}$ throughout this chapter, is defined in terms of the following family of graphs G_n, for $n \geq 1$. Here, each graph G_n consists of $2 \cdot n^2$ vertices, which are partitioned into *layers* $V_1^1, \ldots, V_n^1, V_1^2, \ldots, V_n^2$ with $|V_i^j| = n$. Any two vertices in consecutive layers V_i^j and V_{i+1}^j are connected by an edge, that is, the set $E(G_n)$ of edges of G_n is the set of all binary subsets $\{v, w\}$ with $v \in V_i^j$ and $w \in V_{i+1}^j$ where $i \in [n-1]$ and $j \in [2]$. For example, the graph G_3 is shown in Figure 9.1.

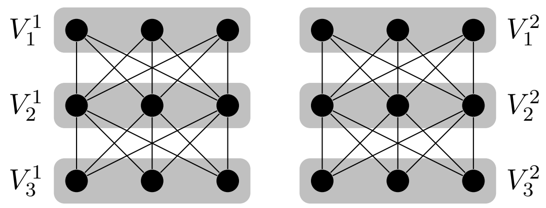


Figure 9.1.: The graph G_3. The gray areas highlight the different layers of G_3.

Now, the class $\mathcal{C}$ is defined as follows. We also define the subclass $\mathcal{C}_{\geq d}$ of $\mathcal{C}$ for $d \geq 1$, which we need in order to prove Theorem 314.

$$\mathcal{C} := \{G \mid G \text{ is a graph such that } G \cong G_n \text{ for some } n \geq 1\}.$$
$$\mathcal{C}_{\geq d} := \{G \mid G \text{ is a graph such that } G \cong G_n \text{ for some } n \geq d\}.$$

The key property of the graphs in $\mathcal{C}$ that enables us to show that reachability on $\mathcal{C}$ is not LREC-definable is that they are rich in a certain kind of automorphisms. Indeed, let v and w be nodes occurring in the same layer of G_n. Then there is an automorphism of G_n

swapping v and w, and fixing the remaining vertices point-wise. To see why this could be useful at all, consider an LREC-formula φ of the form $[\mathrm{lrec}_{\bar{u}_1,\bar{u}_2,\bar{p}}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{w},\bar{r})$, where (for the sake of a nicer illustration of the idea) all free structure variables of $\varphi_{\mathtt{E}}$ are among $\tilde{u}_1 \cup \tilde{u}_2$. Suppose we want to decide membership of a tuple $(\bar{a}_0, \ell_0)$ in the relation X defined by φ in (G_n, α), for an assignment α. First, we would compute the graph $\mathtt{G}$ with vertex set $G_n^{\bar{u}_1}$ and edge set $\mathtt{E}$ defined by $\varphi_{\mathtt{E}}$, and then we would recurse to decide which of the tuples $(\bar{a}_1, \ell_1)$, for successor nodes $\bar{a}_1$ of $\bar{a}_0$ in $\mathtt{G}$ and $\ell_1 = \lfloor(\ell_0 - 1)/|\mathtt{E}\bar{a}_1|\rfloor$, belong to X. To decide membership of each of the tuples $(\bar{a}_1, \ell_1)$ in X, we again have to recurse to decide which of the tuples $(\bar{a}_2, \ell_2)$, for successor nodes $\bar{a}_2$ of $\bar{a}_1$ in $\mathtt{G}$ and $\ell_2 = \lfloor(\ell_1 - 1)/|\mathtt{E}\bar{a}_2|\rfloor$, belong to X, and so on. Exploiting the above-mentioned automorphisms enables us to show that along each branch $(\bar{a}_0, \ell_0), (\bar{a}_1, \ell_1), (\bar{a}_2, \ell_2), \dots$ of the "recursion tree", we see only a constant number of tuples $(\bar{a}_{i+1}, \ell_{i+1})$, where $\bar{a}_{i+1}$ does not contain all the vertices of G_n that occur in $\bar{a}_i$, or vice versa. Thus, on this branch we are left with finitely many subpaths "in between" those tuples on which all tuples contain the same vertices of G_n. If all those subpaths had constant length, then the whole "recursion tree" would have constant depth, so that we could easily find an FO+C-formula that is equivalent to φ on $\mathcal{C}$ (provided $\varphi_{\mathtt{E}}$ and $\varphi_{\mathtt{C}}$ are equivalent to FO+C-formulas). Since reachability is not FO+C definable on $\mathcal{C}$, this would immediately imply Theorem 314. In general, the subpaths do not have constant length (due to number variables that may occur in $\bar{u}_1$ and $\bar{u}_2$), so that we move to a logic that is more expressive than FO+C, but still lacks the ability to define reachability on $\mathcal{C}$.

More precisely, we show that every LREC$[\{E\}]$-formula φ is equivalent to a formula in the *infinitary counting logic* $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ on the subclass $\mathcal{C}_{\geq d}$ of C where d is a constant that depends only on φ. The infinitary counting logic $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ was introduced in [51] (see also [52, Section 8.2]). The fact that $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-formulas without free number variables are Gaifman-local [51] then yields that reachability is not $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-definable on $\mathcal{C}_{\geq d}$, and hence not LREC-definable.

9.1. The Logic $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$

Before delving into the details of translating LREC-formulas into $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-formulas, we give here a brief review of the logic $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$. For a detailed account, we refer the reader to [52, Section 8.2].

$\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ on the one hand extends FO+C by allowing for infinite disjunctions and conjunctions, and on the other hand imposes restrictions so as to make the resulting logic not too powerful. While in the context of FO+C, we equipped structures A with a counting sort $N(A) = [0, |U(A)|]$, in the context of $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ we extend this counting sort to the set of all natural numbers. Furthermore, $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-formulas may use any natural number $n \in \mathbb{N}$ as a constant, which is always interpreted as n.

$\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ is a restriction of the extremely powerful logic $\mathcal{L}_{\infty\omega}(\mathbf{C})$, which is defined as follows. A *term* t is a structure variable, a number variable or a non-negative integer. If t is a structure variable, we call t *structure term*, and otherwise *number term*. The atomic formulas of $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$ have the form

1. $R(x_1, \dots, x_r)$, where $R \in \tau$ is of arity r, and $x_1, \dots, x_r$ are structure variables; or
2. $t = u$, where t and u are either structure terms or number terms.

The set of all $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$-formulas is the smallest set that contains all atomic formulas, and is closed under the following formula formation rules:

3. If $\varphi \in \mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$, then $\neg\varphi \in \mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$.
4. If $\Phi \subseteq \mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$, then $\bigvee \Phi$ and $\bigwedge \Phi$ belong to $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$.
5. If $\varphi \in \mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$ and u is a variable, then $\exists u\varphi$ and $\forall u\varphi$ belong to $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$.
6. If $\varphi \in \mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$, x is a structure variable and p a number variable, then $\exists^{\geq p} x\,\varphi$ belongs to $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$.
7. If $\varphi \in \mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$, $\bar{x}$ is a tuple of free structure variables of φ, and p is a number variable, then $\#\bar{x}\,\varphi = p$ belongs to $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$.[1]

The semantics of $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$-formulas constructed as in 1, 2, 3, 5 and 7 is as usual. The semantics of formulas of the form $\bigvee \Phi$ or $\bigwedge \Phi$ is "at least one $\varphi \in \Phi$ is satisfied" and "all $\varphi \in \Phi$ are satisfied", respectively. Formulas of the form $\exists^{\geq p} x\varphi$ have the meaning "there are at least p assignments to x for which φ is satisfied".

$\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formulas are those $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$-formulas whose rank is bounded. Here, the *rank* $\mathrm{rk}(\varphi)$ of an $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$-formula φ is defined as follows. We have

- $\mathrm{rk}(\varphi) = 0$ for atomic formulas φ,
- $\mathrm{rk}(\neg\varphi) = \mathrm{rk}(\varphi)$,
- $\mathrm{rk}(\bigvee \Phi) = \mathrm{rk}(\bigwedge \Phi) = \sup_{\varphi\in\Phi} \mathrm{rk}(\varphi)$,
- $\mathrm{rk}(\exists u\varphi) = \mathrm{rk}(\forall u\varphi) = \mathrm{rk}(\varphi)$ if u is a number variable,
- $\mathrm{rk}(\exists u\varphi) = \mathrm{rk}(\forall u\varphi) = \mathrm{rk}(\exists^{\geq p} u\varphi) = 1 + \mathrm{rk}(\varphi)$ if u is a structure variable, and
- $\mathrm{rk}(\#\bar{x}\,\varphi = p) = |\bar{x}| + \mathrm{rk}(\varphi)$.

Now, an $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$-formula φ belongs to $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$ if there is a number $n \in \mathbb{N}$ with $\mathrm{rk}(\varphi) \leq n$.

Formula (8.5) in [51] shows that every predicate on $\mathbb{N}$ is definable by an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula of rank 0. Thus, we can assume that $+, -, \cdot, \leq$, and every further predicate on natural numbers is available. Further, we can extend the tuple of variables $\bar{x}$ in formulas of form 7 to arbitrary individual variables:

Observation 316. *Let φ be an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula, p be a number variable and $\bar{u}$ be a non-empty tuple of individual variables with exactly k occurrences of structure variables. Then there exists an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula ψ of rank at most $k + \mathrm{rk}(\varphi)$ such that*

$$(A, \alpha) \models \psi \iff \alpha(p) = |\{\bar{a} \in A^{\bar{u}} \mid (A, \alpha[\bar{a}/\bar{u}]) \models \varphi\}| < \infty.$$

Observation 316 is proved in Section A.2 in the Appendix. We use

$$\#\bar{u}\varphi = p$$

as an abbreviation for the $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula ψ in Observation 316, and note that

- $\mathrm{rk}(\#\bar{u}\,\varphi = p) \leq k + \mathrm{rk}(\varphi)$ where k is the number of occurrences of structure variables in $\bar{u}$.

[1] Originally, in [51] and [52, Section 8.2], $\#\bar{x}\,\varphi$ is defined as a number term of logic $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$ that is interpreted as the value $|\{\bar{a} \in A^{\bar{u}} \mid (A, \alpha[\bar{a}/\bar{u}]) \models \varphi\}|$. Clearly, every formula as defined above is an $\mathcal{L}_{\infty\omega}(\mathbf{C})[\tau]$-formula regarding the original definition. Moreover, Proposition 8.8 in [52] shows that the two versions of this logic, and also the resulting versions of logic $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$, are equivalent.

As shown in [51], every $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$ formula without free number variables is Gaifman local. To make this precise, we need some more notation. Given an undirected graph G and vertices $v, w \in V(G)$, let $\mathrm{dist}^G(v, w)$ denote the length of a shortest path from v to w in G, or ∞ if there is no such path. For all $k \geq 1$, all tuples $\bar{v} = (v_1, \ldots, v_k) \in V(G)^k$ and all $r \in \mathbb{N}$, let $B_r^G(\bar{v}) := \{w \in V(G) \mid \exists i \in [k]\colon \mathrm{dist}^G(v_i, w) \leq r\}$, and define $N_r^G(\bar{v})$ to be the subgraph of G induced by $B_r^G(\bar{v})$. The following theorem is stated in [51] for arbitrary vocabularies:

Theorem 317 ([51], restricted form of Theorem 3.8)**.** *For every $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\varphi(\bar{x})$ without free number variables, there is an $r \in \mathbb{N}$ such that for all graphs G and all $\bar{a}, \bar{b} \in V(G)^{|\bar{x}|}$ with $(N_r^G(\bar{a}), \bar{a}) \cong (N_r^G(\bar{b}), \bar{b})$ we have: $G \models \varphi[\bar{a}] \iff G \models \varphi[\bar{b}]$.*

Using Theorem 317, it is straightforward to show that:

Corollary 318. *Let $d \geq 1$. There does not exist an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\varphi(x, y)$ such that for all $G_n \in \mathcal{C}_{\geq d}$ and all $v, w \in V(G)$ we have $G_n \models \varphi[v, w]$ iff there is a path from v to w in G_n.*

Proof. For a contradiction, suppose that $\varphi(x, y)$ is an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula such that for all $G_n \in \mathcal{C}_{\geq d}$ and all $v, w \in V(G_n)$ we have $G_n \models \varphi[v, w]$ iff there is a path from v to w in G_n. Let $r \in \mathbb{N}$ be as guaranteed by Theorem 317. Let $d' := \max\{2r + 3, d\}$. We can now pick vertices $v, w_1, w_2 \in G_{d'}$ with $(N_r^{G_{d'}}(v, w_1), v, w_1) \cong (N_r^{G_{d'}}(v, w_2), v, w_2)$ such that w_1 is reachable from v, but w_2 is not reachable from v. Since $G_{d'} \models \varphi[v, w_1]$, we then have $G_{d'} \models \varphi[v, w_2]$, a contradiction. ☐

9.2. Translation of LREC-Formulas into $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-Formulas

We now describe the translation of an LREC-formula φ into an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})$-formula $\tilde{\varphi}$ that is equivalent to φ on $\mathcal{C}_{\geq d}$ for a constant d that depends only on φ. The translation proceeds by induction on the structure of φ, where the only interesting case is that of LREC-formulas φ of the form

$$[\mathrm{lrec}_{\bar{u}_1, \bar{u}_2, \bar{p}}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{w}, \bar{r}).$$

To decide whether φ holds in a given graph G_n under an assignment α, $\tilde{\varphi}$ needs to check whether the tuple $(\bar{a}_0, \ell_0)$, for $\bar{a}_0 := \alpha(\bar{w})$ and $\ell_0 := \langle \alpha(\bar{r}) \rangle$, belongs to the relation X defined by φ in (G_n, α). To this end, it looks at the directed graph $\mathtt{G}$ with vertex set $G_n^{\bar{u}_1}$ and edge set $\varphi_{\mathtt{E}}[G_n, \alpha; \bar{u}_1, \bar{u}_2]$, or rather at its ℓ_0-*unraveling* $\mathtt{G}^{(\bar{a}_0, \ell_0)}$ at $\bar{a}_0$:

Definition 319. The ℓ-*unraveling* of a directed graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ at a vertex $v \in \mathtt{V}$ is the directed tree $\mathtt{G}^{(v,\ell)}$ defined as follows:

1. The nodes of $\mathtt{G}^{(v,\ell)}$ are all finite sequences $((v_0, \ell_0), \ldots, (v_n, \ell_n)) \in (\mathtt{V} \times \mathbb{N})^{n+1}$, where $(v_0, \ell_0) = (v, \ell)$, $(v_0, \ldots, v_n)$ is a (non-simple) path in $\mathtt{G}$, and $\ell_i = \lfloor (\ell_{i-1} - 1)/|\mathtt{E}v_i| \rfloor$ for every $i \in [n]$.
2. There is an edge from a node $((v_0, \ell_0), \ldots, (v_m, \ell_m))$ to a node $((v'_0, \ell'_0), \ldots, (v'_n, \ell'_n))$ whenever $n = m + 1$, and $(v'_i, \ell'_i) = (v_i, \ell_i)$ for every $i \leq m$.
3. Each node $((v_0, \ell_0), \ldots, (v_m, \ell_m))$ is labeled with (v_m, ℓ_m).

For each node of $\mathtt{G}^{(\bar{a}_0,\ell_0)}$, $\tilde{\varphi}$ checks whether its label belongs to X. Clearly, this suffices to decide whether $(\bar{a}_0, \ell_0) \in X$.

Our construction is based on the following property of $\mathtt{G}^{(\bar{a}_0,\ell_0)}$:

Lemma 320. *Let $\varphi_{\mathtt{E}}(\bar{u}_1, \bar{u}_2, \bar{u}_{\mathtt{E}})$ be a formula, where the variable tuples $\bar{u}_1$ and $\bar{u}_2$ are compatible. Let $n \geq |\bar{u}_1|+|\bar{u}_{\mathtt{E}}|+2$ and let α be an assignment for $\varphi_{\mathtt{E}}$ in G_n. Let $\bar{a}_{\mathtt{E}} := \alpha(\bar{u}_{\mathtt{E}})$. Further, let $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ be the graph with $\mathtt{V} := G_n^{\bar{u}_1}$ and $\mathtt{E} := \varphi_{\mathtt{E}}[G_n, \alpha; \bar{u}_1, \bar{u}_2]$. Consider a node $((\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m))$ in $\mathtt{G}^{(\bar{a},\ell)}$, where $\ell \leq |N(G_n)|^r - 1$. Then, the size of*

$$\mathcal{I} := \{i \in [m] \mid (\tilde{a}_{i-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) \neq (\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)\}$$

is bounded by a constant that depends only on $|\bar{u}_1|$, $|\bar{u}_{\mathtt{E}}|$ and r.

Proof. We first show that the size of

$$\begin{aligned}\mathcal{K} &:= \{i \in \mathcal{I} \mid \tilde{a}_{i-1} \cap V(G_n) \not\subseteq (\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)\} \\ &= \{i \in \mathcal{I} \mid (\tilde{a}_{i-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) \not\subseteq (\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)\}\end{aligned}$$

is bounded by a constant that only depends on $|\bar{u}_1|$, $|\bar{u}_{\mathtt{E}}|$ and r. To this end, consider an $i \in \mathcal{K}$ and a $b \in \tilde{a}_{i-1} \cap V(G_n)$ such that $b \notin \tilde{a}_i \cup \tilde{a}_{\mathtt{E}}$. Let us call an element $b' \in V(G_n)$ a *sibling* of b if b and b' belong to the same layer in G_n. A sibling of b is *good* if it does not occur in $\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}$. For instance, b is a good sibling of b. There exist at least

$$n - |\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}| \geq n - (|\bar{u}_1| + |\bar{u}_{\mathtt{E}}|)$$

good siblings of b in the graph G_n. Each good sibling b' gives rise to an *automorphism* $f_{b'} : V(G_n) \to V(G_n)$ of G_n that fixes all the vertices in $V(G_n) \setminus \{b, b'\}$ point-wise, maps b to b', and maps b' to b. As a consequence, for each good sibling b' we have

$$\begin{aligned}G_n \models \varphi_{\mathtt{E}}[\bar{a}_{i-1}, \bar{a}_i, \tilde{a}_{\mathtt{E}}] &\iff G_n \models \varphi_{\mathtt{E}}[f_{b'}(\bar{a}_{i-1}, \bar{a}_i, \tilde{a}_{\mathtt{E}})] \\ &\iff G_n \models \varphi_{\mathtt{E}}[f_{b'}(\bar{a}_{i-1}), \bar{a}_i, \tilde{a}_{\mathtt{E}}],\end{aligned}$$

where $f_{b'}(\bar{a})$ is the tuple obtained from a tuple $\bar{a}$ by replacing each element b'' in $\bar{a}$ that belongs to $V(G_n)$ with $f_{b'}(b'')$. Thus, $f_{b'}(\bar{a}_{i-1})\bar{a}_i \in \mathtt{E}$ for each good sibling b' of b. This implies

$$|\mathtt{E}\bar{a}_i| \geq n - d_1,$$

where $d_1 := |\bar{u}_1| + |\bar{u}_{\mathtt{E}}|$.

Observe that, by the definition of $\mathtt{G}^{(\bar{a},\ell)}$, we have $\ell_0 = \ell < |N(G_n)|^r \leq (2n^2+1)^r \leq (2n)^{2r}$. Further, we have $|\mathtt{E}\bar{a}_m| \leq (2n^2)^{|\bar{u}_1|}$ and $\ell_0 \cdot |\mathtt{E}\bar{a}_m| \geq \prod_{i=1}^m |\mathtt{E}\bar{a}_i|$ (cf. inequality (7.8)). Hence,

$$\begin{aligned}(2n)^{2(r+|\bar{u}_1|)} \;&\geq\; (2n)^{2r} \cdot (2n)^{2|\bar{u}_1|} \;\geq\; \ell_0 \cdot |\mathtt{E}\bar{a}_m| \;\geq\; \prod_{i=1}^{m} |\mathtt{E}\bar{a}_i| \\ &\geq\; \prod_{i \in \mathcal{K}} |\mathtt{E}\bar{a}_i| \;\geq\; \prod_{i \in \mathcal{K}} (n - d_1) \;=\; (n-d_1)^{|\mathcal{K}|}.\end{aligned}$$

For $n \geq d_1 + 2$ this implies $|\mathcal{K}| \leq \log_{n-d_1}(2n)^{2(r+|\bar{u}_1|)} \leq 2(r + |\bar{u}_1|)(1 + \log_{n-d_1} n)$, which is bounded by a constant d_2 that only depends on $|\bar{u}_1|$, $|\bar{u}_{\mathtt{E}}|$ and r.

To conclude the proof, consider a maximal set $\mathcal{I}' \subseteq \mathcal{I}$ such that there are no $i, i' \in \mathcal{I}'$ and $k \in \mathcal{K}$ with $i \leq k \leq i'$. We show that $|\mathcal{I}'|$ is bounded by a constant d_3 that depends only on $|\bar{u}_1|$, $|\bar{u}_{\mathtt{E}}|$ and r. This then implies the lemma as

$$|\mathcal{I}| \leq (|\mathcal{K}| + 1) \cdot (d_3 + 1) \leq (d_2 + 1) \cdot (d_3 + 1).$$

Let $i_{\min} := \min \mathcal{I}'$ and $i_{\max} := \max \mathcal{I}'$, and notice that

$$(\tilde{a}_{i_{\min}-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) \ \subseteq\ (\tilde{a}_{i_{\min}} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) \ \subseteq\ \cdots \ \subseteq\ (\tilde{a}_{i_{\max}} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n).$$

Since $(\tilde{a}_{i_{\max}} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)$ contains at most $d_3 := |\bar{u}_1|$ elements that do not belong to the set $(\tilde{a}_{i_{\min}-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)$, there are at most d_3 indices $i \in [i_{\min}, i_{\max}]$ with $(\tilde{a}_{i-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) \subsetneq (\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)$. Hence, $|\mathcal{I}'| \leq d_3$, as desired. □

We are now ready to show that each $\mathsf{LREC}[\{E\}]$-formula φ is equivalent to an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula on a subclass $\mathcal{C}_{\geq d}$ of $\mathcal{C}$, where d depends on formula φ.

Lemma 321. *For every $\mathsf{LREC}[\{E\}]$-formula $\varphi(\bar{u})$, there is an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\tilde{\varphi}(\bar{u})$ and a constant d_φ such that for all $G_n \in \mathcal{C}_{\geq d_\varphi}$ and all $\bar{a} \in G_n^{\bar{u}}$, we have:*

$$G_n \models \varphi[\bar{a}] \iff G_n \models \tilde{\varphi}[\bar{a}].$$

Proof. As mentioned above, we proceed by induction on the structure of φ. The only interesting case is that of an $\mathsf{LREC}[\{E\}]$-formula of the form

$$\varphi = [\mathrm{lrec}_{\bar{u}_1, \bar{u}_2, \bar{p}}\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{w}, \bar{r}).$$

Let $\bar{u}_{\mathtt{E}}$ be an enumeration of all variables in $\mathrm{free}(\varphi_{\mathtt{E}})$ that are not listed in $\bar{u}_1\bar{u}_2$, and let $\bar{u}_{\mathtt{C}}$ be an enumeration of all variables in $\mathrm{free}(\varphi_{\mathtt{C}})$ that are not listed in $\bar{u}_1\bar{p}$. Further, let $d_\varphi := \max\{d_{\varphi_{\mathtt{E}}}, d_{\varphi_{\mathtt{C}}}, |\bar{u}_1| + |\bar{u}_{\mathtt{E}}| + 2\}$.

We aim to construct, for all integers $n \geq d_\varphi$ and $\ell \leq |N(G_n)|^{|\bar{r}|} - 1$, an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\psi_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ such that for all assignments α in G_n, and all $\bar{a} \in G_n^{\bar{u}_1}$,

$$G_n \models \psi_{n,\ell}[\bar{a}, \alpha(\bar{u}_{\mathtt{E}}), \alpha(\bar{u}_{\mathtt{C}})] \iff (\bar{a}, \ell) \in X,$$

where X is the relation defined by φ in (G_n, α). Furthermore, the rank of each $\psi_{n,\ell}$ will be bounded by a constant that depends only on φ, so that

$$\tilde{\varphi} := \bigvee_{\substack{n \geq d_\varphi \\ \ell < (2n^2+1)^{|\bar{r}|}}} \left(\text{“the universe has size } 2n^2\text{”} \wedge \text{“}\bar{r} \text{ represents the number } \ell\text{”} \wedge \psi_{n,\ell}(\bar{w}, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})\right)$$

is an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula that is equivalent to φ on $\mathcal{C}_{\geq d_\varphi}$.

Construction of $\psi_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$: Fix $n \geq d_\varphi$ and $\ell \leq |N(G_n)|^{|\bar{r}|} - 1$. To simplify the presentation, we also fix an assignment α in G_n, and the graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with $\mathtt{V} := G_n^{\bar{u}_1}$ and $\mathtt{E} := \varphi_{\mathtt{E}}[G_n, \alpha; \bar{u}_1, \bar{u}_2]$; the formula $\psi_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ we are going to construct will however not depend on α. Let $\bar{a}_{\mathtt{E}} := \alpha(\bar{u}_{\mathtt{E}})$ and $\bar{a}_{\mathtt{C}} := \alpha(\bar{u}_{\mathtt{C}})$. For every $\bar{a} \in \mathtt{V}$, let

$$t_{n,\ell}(\bar{a}) := \max \Big\{ t \in \mathbb{N} \ \Big|\ \text{there is a node } (\bar{a}_0, \ell_0), \ldots, (\bar{a}_m, \ell_m) \text{ in } \mathtt{G}^{(\bar{a},\ell)} \text{ such that } t \text{ equals } \big|\{ i \in [m] \mid (\tilde{a}_{i-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) \neq (\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)\}\big| \Big\}.$$

As $n \geq |\bar{u}_1| + |\bar{u}_{\mathtt{E}}| + 2$, Lemma 320 implies that there is a constant t^* that only depends on φ such that

$$t_{n,\ell}(\bar{a}) < t^* \quad \text{for all } \bar{a} \in \mathtt{V}.$$

In what follows, we construct, for all $t \leq t^*$, an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\psi^t_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ such that for all $\bar{a} \in \mathtt{V}$ with $t_{n,\ell}(\bar{a}) < t$, we have:

$$G_n \models \psi^t_{n,\ell}[\bar{a}, \bar{a}_{\mathtt{E}}, \bar{a}_{\mathtt{C}}] \iff (\bar{a}, \ell) \in X,$$

where X is the relation defined by φ in (G_n, α). Furthermore, the rank of $\psi^t_{n,\ell}$ will not depend on n or ℓ. The desired formula $\psi_{n,\ell}$ can then be defined as:

$$\psi_{n,\ell} := \psi^{t^*}_{n,\ell}.$$

Construction of $\psi^t_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$*:* We construct the formulas $\psi^t_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ by induction on t. For $t = 0$, we define $\psi^0_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ to be an arbitrary unsatisfiable formula with bounded rank. The idea for the construction of $\psi^{t+1}_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ for $t \geq 0$ is as follows. Let $\bar{a} \in \mathtt{V}$, and

$$\begin{aligned}\mathcal{Q}(\bar{a}) := \{(\bar{a}_m, \ell_m) \mid ((\bar{a}_0, \ell_0), \dots, (\bar{a}_m, \ell_m)) \in V(\mathtt{G}^{(\bar{a},\ell)}),\ \text{and for all } i \in [m] \text{ we have:}\\ (\tilde{a}_{i-1} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) = (\tilde{a}_i \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)\}.\end{aligned}$$

To check whether $(\bar{a}, \ell) \in X$, we "guess" the set $\hat{X} = \mathcal{Q}(\bar{a}) \cap X$, and then simply check whether $(\bar{a}, \ell) \in \hat{X}$. To guess $\hat{X}$, we can use an infinite disjunction over all subsets R of $\mathcal{Q}(\bar{a})$. Then we only need to verify for each R whether R indeed corresponds to $\hat{X}$. For the latter, we count, for each pair $(\bar{a}', \ell') \in \mathcal{Q}(\bar{a})$, the number of pairs $(\bar{a}'', \ell'')$ such that $\bar{a}'\bar{a}'' \in \mathtt{E}$, $\ell'' = \lfloor (\ell' - 1)/|\mathtt{E}\bar{a}''| \rfloor$ and $(\bar{a}'', \ell'') \in X$, and check that $(\bar{a}', \ell') \in R$ whenever this number belongs to the label of $\bar{a}'$ defined by $\varphi_{\mathtt{C}}$. How do we check whether $(\bar{a}'', \ell'') \in X$? If $(\tilde{a}' \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) = (\tilde{a}'' \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)$, that is, if $(\bar{a}'', \ell'') \in \mathcal{Q}(\bar{a})$, then we simply check whether $(\bar{a}'', \ell'') \in R$. Otherwise, we use the formula $\psi^t_{n,\ell''}$.

Let $\varphi'_{\mathtt{E}}$ and $\varphi'_{\mathtt{C}}$ be $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formulas that are equivalent to $\varphi_{\mathtt{E}}$ and $\varphi_{\mathtt{C}}$ on $\mathcal{C}_{\geq d_\varphi}$, respectively. Such formulas exist by the induction hypothesis. Using $\varphi'_{\mathtt{E}}$ we construct, for each $\ell' \in [0, \ell]$, an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\chi_{\ell'}(\bar{u}_1, \bar{u}'_1, \bar{u}_{\mathtt{E}})$ such that for all $\bar{a}, \bar{a}' \in G_n^{\bar{u}_1}$,

$$G_n \models \chi_{\ell'}[\bar{a}, \bar{a}', \bar{a}_{\mathtt{E}}] \iff (\bar{a}', \ell') \in \mathcal{Q}(\bar{a}).$$

Here, $\bar{u}'_1$ is a tuple of distinct variables that is compatible with, but disjoint from $\bar{u}_1$. We let $M(\bar{u}_1, \bar{u}_E)$ be the set of all tuples $\bar{u}$ where $\bar{u}$ is obtained from $\bar{u}'_1$ by replacing each structure variable with a structure variable from $\tilde{u}_1 \cup \tilde{u}_E$ and each number variable with an integer from $N(G_n) = [0, 2n^2]$. Hence, all tuples $\bar{u} \in M(\bar{u}_1, \bar{u}_E)$ are compatible to $\bar{u}_1$ and all structure variables of $\bar{u}$ also occur in $\bar{u}_1$ or $\bar{u}_E$. Note that for all $\bar{a}, \bar{a}' \in \mathtt{V}$ with $(\tilde{a} \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n) = (\tilde{a}' \cup \tilde{a}_{\mathtt{E}}) \cap V(G_n)$ there is a $\bar{u} \in M(\bar{u}_1, \bar{u}_E)$ such that $\alpha[\bar{a}/\bar{u}_1](\bar{u}) = \bar{a}'$. We let

$$\begin{aligned}\chi_{\ell'} := \bigvee_{m \in \mathbb{N}} \bigvee_{\substack{(\bar{w}_0, \dots, \bar{w}_m)\\ \in M(\bar{u}_1, \bar{u}_E)^m}} \Bigg(& \bigwedge_{i \in [m]} \text{“}\big(\alpha(\tilde{w}_{i-1}) \cup \alpha(\tilde{u}_{\mathtt{E}})\big) \cap V(G_n) = \big(\alpha(\tilde{w}_i) \cup \alpha(\tilde{u}_{\mathtt{E}})\big) \cap V(G_n)\text{”}\\ & \wedge\ \bar{w}_0 = \bar{u}_1\ \wedge \bigwedge_{i \in [m]} \varphi'_{\mathtt{E}}(\bar{w}_{i-1}, \bar{w}_i, \bar{u}_{\mathtt{E}})\ \wedge\ \bar{w}_m = \bar{u}'_1\\ & \wedge\ \exists l_0 \dots \exists l_m \Big(l_0 = l\ \wedge \bigwedge_{i \in [m]} \text{“}\ell_i = \Big\lfloor \frac{\ell_{i-1} - 1}{|\mathtt{E}\,\alpha(\bar{w}_i)|} \Big\rfloor\text{”}\ \wedge\ l_m = l' \Big) \Bigg)\end{aligned}$$

Let $\mathcal{Q}'$ be the set of all pairs $(\bar{u}', \ell')$, where $\ell' \in [0, \ell]$ and $\bar{u}' \in M(\bar{u}_1, \bar{u}_E)$. Intuitively, each $R' \subseteq \mathcal{Q}'$ corresponds to a guess of $\mathcal{Q}(\bar{a}) \cap X$ as described above. For each $R' \subseteq \mathcal{Q}'$, let $\psi^{t+1}_{n,\ell,R'}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$ be

$$\begin{aligned}
&\bigwedge_{(\bar{u}',\ell')\in R'} \Big(\chi_{\ell'}(\bar{u}_1, \bar{u}', \bar{u}_{\mathtt{E}}) \to \exists \bar{p}\, \exists p \Big(\#\bar{u}''(\varphi'_{\mathtt{E}}(\bar{u}', \bar{u}'', \bar{u}_{\mathtt{E}}) \wedge \vartheta_{R',\bar{u}',\ell'}(\bar{u}'')) = p \\
&\qquad\qquad \wedge \text{“}\alpha(\bar{p}) \text{ represents } \alpha(p)\text{”} \wedge \varphi'_{\mathtt{C}}(\bar{u}', \bar{p}, \bar{u}_{\mathtt{C}})\Big)\Big) \\
\wedge &\bigwedge_{\substack{(\bar{u}',\ell')\in \mathcal{Q}' \\ (\bar{u}',\ell')\notin R'}} \Big(\chi_{\ell'}(\bar{u}_1, \bar{u}', \bar{u}_{\mathtt{E}}) \to \exists \bar{p}\, \exists p \Big(\#\bar{u}''(\varphi'_{\mathtt{E}}(\bar{u}', \bar{u}'', \bar{u}_{\mathtt{E}}) \wedge \vartheta_{R',\bar{u}',\ell'}(\bar{u}'')) = p \\
&\qquad\qquad \wedge \text{“}\alpha(\bar{p}) \text{ represents } \alpha(p)\text{”} \wedge \neg\varphi'_{\mathtt{C}}(\bar{u}', \bar{p}, \bar{u}_{\mathtt{C}})\Big)\Big)
\end{aligned}$$

where

$$\begin{aligned}
\vartheta_{R',\bar{u}',\ell'}(\bar{u}'') := \bigvee_{\ell''\in[0,\ell']} \Big(&\text{“}\ell'' = \Big\lfloor \frac{\ell'-1}{|\mathrm{E}\alpha(\bar{u}'')|} \Big\rfloor\text{”} \wedge \Big((\chi_{\ell''}(\bar{u}', \bar{u}'', \bar{u}_{\mathtt{E}}) \wedge \text{“}(\bar{u}'', \ell'') \in R'\text{”}) \\
&\vee (\neg\chi_{\ell''}(\bar{u}', \bar{u}'', \bar{u}_{\mathtt{E}}) \wedge \psi^{t}_{n,\ell''}(\bar{u}'', \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}}))\Big)\Big)
\end{aligned}$$

and “$(\bar{u}'', \ell'') \in R'$” stands for $\bigvee_{(\bar{u}^\star,\ell^\star)\in R'} (\bar{u}^\star = \bar{u}'' \wedge \ell^\star = \ell'')$. Then it is not hard to see that the formula

$$\psi^{t+1}_{n,\ell}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}}) := \bigvee_{\substack{R'\subseteq \mathcal{Q}' \\ (\bar{u}_1,\ell)\in R'}} \psi^{t+1}_{n,\ell,R'}(\bar{u}_1, \bar{u}_{\mathtt{E}}, \bar{u}_{\mathtt{C}})$$

is as desired. Clearly, there is a constant c that does not depend on n or ℓ such that $\mathrm{rk}(\psi^{t+1}_{n,\ell}) \leq c + \mathrm{rk}(\psi^{t}_{n,\ell})$ forall $t \geq 0$. As t^* does not depend on n or ℓ, the rank of $\psi^{t^*}_{n,\ell}$ does not depend on n or ℓ. □

To conclude this chapter, we proof Theorem 314 which follows from Lemma 321 and Corollary 318.

Proof (Theorem 314). Let us assume there is an LREC[$\{E\}$]-formula $\varphi(x, y)$ that defines reachability on the class of all undirected graphs. By Lemma 321 there is a constant d_φ and an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\{E\}]$-formula $\tilde{\varphi}(x, y)$ that defines reachability on $\mathcal{C}_{\geq d_\varphi}$. This is a contradiction to Corollary 318. □

10. $\mathsf{LREC}_=$ - An Extension of LREC

The proof of the previous chapter's Theorem 314 indicates that LREC is not closed under logical reductions, not even under very simple first-order reductions.[1] Indeed, there is a first-order reduction that maps a graph G_n, for $n \geq 4$, as defined in Chapter 9 to a disjoint union $\hat{G}_n$ of two paths on n vertices each, by identifying vertices in the same layer (see Example 322). Reachability on the class of all graphs isomorphic to $\hat{G}_n$ for an $n \geq 4$ is LREC-definable (see Example 298). Hence, if LREC was closed under first-order reductions, then reachability on the class of all graphs isomorphic to G_n for some n would be LREC-definable, contradicting the previous chapter's results.

Example 322. Consider the $\mathsf{FO}[\{E\},\{E\}]$-transduction $\Theta = (\theta_U(x), \theta_\approx(x,y), \theta_E(x,y))$ with $\theta_U(x) := \top$, $\theta_\approx(x,y) := \forall z\big(E(x,z) \leftrightarrow E(y,z)\big)$ and $\theta_E(x,y) := E(x,y)$. Recall the definition of the graphs G_n from Chapter 9. For $n \geq 4$, the equivalence relation $\approx$ generated by $\theta_\approx[G_n; x, y]$ is $\theta_\approx[G_n; x, y]$ itself. It relates any two vertices that occur in the same layer of G_n. Hence, for $n \geq 4$, $\Theta[G_n]$ is the disjoint union of two paths of length n. ⌟

In this chapter, we introduce an extension $\mathsf{LREC}_=$ of LREC whose data complexity is still in LOGSPACE, and thus captures LOGSPACE on directed trees, while being closed under logical reductions. The idea is to admit a third formula $\varphi_=$ in the lrec-operator that generates an equivalence relation on the vertices of the graph defined by $\varphi_{\mathtt{E}}$.

Let τ be a vocabulary. The set of all $\mathsf{LREC}_=[\tau]$-formulas is obtained from $\mathsf{LREC}[\tau]$ by replacing the rule for the lrec-operator from Chapter 7 as follows: If $\bar{u}, \bar{v}, \bar{w}$ are compatible tuples of variables, $\bar{p}, \bar{r}$ are non-empty tuples of number variables, and $\varphi_=$, $\varphi_{\mathtt{E}}$ and $\varphi_{\mathtt{C}}$ are $\mathsf{LREC}_=$-formulas, then the following is an $\mathsf{LREC}_=[\tau]$-formula:

$$\varphi := [\mathrm{lrec}_{\bar{u},\bar{v},\bar{p}}\ \varphi_=,\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{w}, \bar{r}). \tag{10.1}$$

We let $\mathrm{free}(\varphi) := \big(\mathrm{free}(\varphi_=) \setminus (\tilde{u} \cup \tilde{v})\big) \cup \big(\mathrm{free}(\varphi_{\mathtt{E}}) \setminus (\tilde{u} \cup \tilde{v})\big) \cup \big(\mathrm{free}(\varphi_{\mathtt{C}}) \setminus (\tilde{u} \cup \tilde{p})\big) \cup \tilde{w} \cup \tilde{r}$.

To define the semantics of $\mathsf{LREC}_=[\tau]$-formulas φ of the form (10.1), let A be a τ-structure and α an assignment in A. Let $\mathtt{V}_0 := A^{\bar{u}}$ and $\mathtt{E}_0 := \varphi_{\mathtt{E}}[A, \alpha; \bar{u}, \bar{v}]$. We define $\sim$ to be the reflexive, symmetric, transitive closure of the binary relation $\varphi_=[A, \alpha; \bar{u}, \bar{v}]$ over $\mathtt{V}_0$. Now consider the graph $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ with

$$\mathtt{V} := \mathtt{V}_0/_\sim \quad \text{and} \quad \mathtt{E} := \{(\bar{a}/_\sim, \bar{b}/_\sim) \in \mathtt{V}^2 \mid \bar{a}\bar{b} \in \mathtt{E}_0\}.$$

To every $\bar{a}/_\sim \in \mathtt{V}$ we assign the set

$$\mathtt{C}(\bar{a}/_\sim) := \{\langle \bar{n} \rangle \mid \text{there is an } \bar{a}' \in \bar{a}/_\sim \text{ with } \bar{n} \in \varphi_{\mathtt{C}}[A, \alpha[\bar{a}'/\bar{u}]; \bar{p}]\}$$

of labels. Then the definition of X can be taken verbatim from Chapter 7. We let $(A, \alpha) \models \varphi$ if and only if $\big(\alpha(\bar{w})/_\sim, \langle \alpha(\bar{r}) \rangle\big) \in X$. As for LREC, we have:

[1] First-order transductions are defined analogous to L-tranductions where $\mathsf{L} \geq \mathsf{STC}$ (see Definition 1), only that FO-transductions require the formula $\theta_\approx$ to not just generate but to define an equivalence relation.

Theorem 323. *For every vocabulary τ, and every* LREC$_=[\tau]$*-formula φ there is a deterministic logspace Turing machine that, given a τ-structure A and an assignment α in A, decides whether $(A, \alpha) \models \varphi$.*

Sketch. The proof is a straightforward modification of the proof of Theorem 299. The only difference is that, when we deal with LREC$_=$-formulas of form (10.1), we use the vertex set $\mathtt{V}$, the edge set $\mathtt{E}$, and the labels $\mathtt{C}(\cdot)$ as defined above to compute the set X. It is easy to compute these sets by first computing the relation $\sim$ from $\varphi_=[A, \alpha; \bar{u}, \bar{v}]$ using Reingold's logspace algorithm for undirected reachability [62]. Note that once $\sim$ has been obtained, the equivalence class of every element $\bar{a} \in A^{\bar{u}}$ can be determined. $\square$

The following example shows that undirected graph reachability is definable in LREC$_=$. This does not involve an implementation of Reingold's algorithm in our logic, but just uses the observation that the computation of the equivalence relation $\sim$ boils down to the computation of undirected reachability.

Example 324 (Undirected reachability)**.** The following LREC$_=$-formula defines undirected graph reachability:

$$\varphi(s,t) \;:=\; [\mathrm{lrec}_{x,y,p}\ \varphi_=(x,y),\ \varphi_{\mathtt{E}}(x,y),\ \varphi_{\mathtt{C}}(x,p)](s,1),$$

where $\varphi_=(x,y) := E(x,y)$, $\varphi_{\mathtt{E}}(x,y) := x \neq x$ and $\varphi_{\mathtt{C}}(x,p) := x = t$. To see this, let G be an undirected graph and α an assignment in G. Define $\sim$, $\mathtt{V}$, $\mathtt{E}$, $\mathtt{C}$ and the set X as above. Clearly, the set $\mathtt{V}$ consists of the connected components of G. Furthermore, the set $\mathtt{E}$ is empty since φ_E is unsatisfiable. Therefore, for all $v \in V(G)$ we have $(v/_\sim, 1) \in X$ iff $0 \in \mathtt{C}(v/_\sim)$. The latter is true precisely if $\alpha(t) \in v/_\sim$, i.e., if v and $\alpha(t)$ are in the same connected component of G. It follows that for all $v, w \in V(G)$ we have $G \models \varphi[v, w]$ if and only if v and w are in the same connected component of G, that is, if there is a path between v an w in G. ⌟

Remark 325. It follows immediately from the previous example that STC+C $\leq$ LREC$_=$. Actually, the containment is strict, because LREC $\not\leq$ STC+C by Corollary 311. Note also that LREC$_=$ $\leq$ FP+C. ⌟

The following proposition shows that we can pull back arbitrary LREC$_=$-formulas under parameterized LREC$_=$-transductions. Hence, LREC$_=$ is closed under LREC$_=$-reductions.

Proposition 326 (Transduction Lemma)**.** *Let τ_1, τ_2 be vocabularies. Let $\Theta(\bar{X})$ be a parameterized* LREC$_=[\tau_1, \tau_2]$*-transduction, where ℓ-tuple $\bar{u}$ is the tuple of domain variables.*[2] *Further, let $\psi(x_1, \ldots, x_\kappa, p_1, \ldots, p_\lambda)$ be an* LREC$_=[\tau_2]$*-formula where $x_1, \ldots, x_\kappa$ are structure variables and $p_1, \ldots, p_\lambda$ are number variables. Then there exists an* LREC$_=[\tau_1]$*-formula $\psi^{-\Theta}(\bar{X}, \bar{u}_1, \ldots, \bar{u}_\kappa, \bar{q}_1, \ldots, \bar{q}_\lambda)$, where $\bar{u}_1, \ldots, \bar{u}_\kappa$ are compatible with $\bar{u}$ and $\bar{q}_1, \ldots, \bar{q}_\lambda$ are ℓ-tuples of number variables, such that for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$, all $\bar{a}_1, \ldots, \bar{a}_\kappa \in A^{\bar{u}}$ and all $\bar{n}_1, \ldots, \bar{n}_\lambda \in N(A)^\ell$,*

[2] Note that $\bar{X}$ denotes a tuple of individual variables, and does not contain any relational variables. We stick to the capitalized letter "X" just out of habit.

$$
\begin{aligned}
A \models \psi^{-\Theta}[\bar{P}, \bar{a}_1, \dots, \bar{a}_\kappa, \bar{n}_1, \dots, \bar{n}_\lambda] \iff & \ \bar{a}_1/_{\approx}, \dots, \bar{a}_\kappa/_{\approx} \in U(\Theta[A, \bar{P}]), \\
& \ \langle \bar{n}_1 \rangle_A, \dots, \langle \bar{n}_\lambda \rangle_A \in N(\Theta[A, \bar{P}]) \ \textit{and} \\
& \ \Theta[A, \bar{P}] \models \psi[\bar{a}_1/_{\approx}, \dots, \bar{a}_\kappa/_{\approx}, \langle \bar{n}_1 \rangle_A, \dots, \langle \bar{n}_\lambda \rangle_A],
\end{aligned}
$$

where $\approx$ *is the equivalence relation of* $(A, \bar{P})$ *under* Θ*.*

The proof of Proposition 326 can be found in Section A.1.1 in the Appendix.

Remark 327. Since in STC+C (and actually in STC) it is possible to transform trees into directed trees (see Example 8), the results from Chapter 8 and the fact that $\mathsf{LREC}_=$ is closed under logical reductions imply that $\mathsf{LREC}_=$ captures LOGSPACE on the class of all trees, directed as well as undirected. ⌟

11. Capturing LOGSPACE on Interval Graphs

With the added expressive power of $\mathsf{LREC}_=$, it is not only possible to capture $\mathsf{LOGSPACE}$ on the class of all trees, but also on the class of all interval graphs, as we shall show in this chapter. Basically, interval graphs are graphs whose vertices are closed intervals, and whose edges join any two distinct intervals with a non-empty intersection. They form a well-established and widely investigated class of graphs, and it was recently shown [47] (see also [50]) that interval graph canonization is in $\mathsf{LOGSPACE}$.

To prove that $\mathsf{LREC}_=$ captures $\mathsf{LOGSPACE}$ on interval graphs, we proceed as in the case of directed trees. First, we describe an $\mathsf{LREC}_=$-definable canonization procedure for interval graphs, and then we use the fact that DTC (and hence $\mathsf{LREC}_=$) captures $\mathsf{LOGSPACE}$ on ordered structures. Our canonization procedure combines algorithmic techniques from [47] with the logical definability framework in [49]. Parts of this chapter can be found in more detail in [50].

11.1. Background on Interval Graphs

In this section, we define interval graphs and state some basic properties. For a more detailed exposition, we refer the reader to [50].

Definition 328 (Interval graph, interval representation)**.** Given a finite collection $\mathcal{I}$ of closed intervals $I_i = [a_i, b_i] \subset \mathbb{N}$, let $G_{\mathcal{I}} = (V, E)$ be the graph with vertex set $V = \mathcal{I}$, joining two distinct intervals $I_i, I_j \in V$ by an edge whenever $I_i \cap I_j \neq \emptyset$. We call $\mathcal{I}$ an *interval representation* of a graph G if $G \cong G_{\mathcal{I}}$. A graph G is an *interval graph* if there is an interval representation of G.

Figure 11.1 shows an interval graph G together with an interval representation of G.

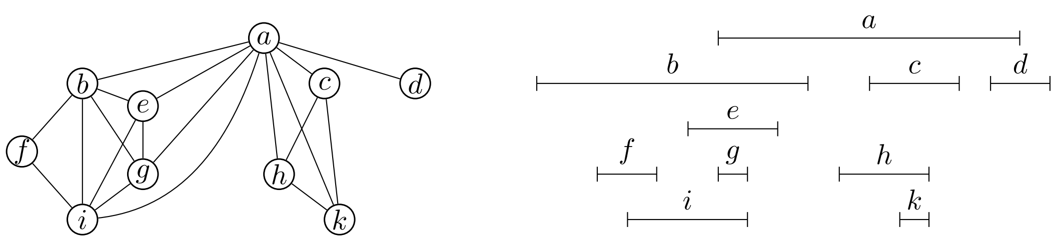


Figure 11.1.: An interval graph G and an interval representation of G.

An interval representation $\mathcal{I}$ of a graph G is called *minimal* if the set $\bigcup\mathcal{I} \subset \mathbb{N}$ is of minimum size among all interval representations of G. Clearly, for any interval representation $\mathcal{I}$ there exists a minimal interval representation $\mathcal{I}_{\min}$ such that $G_{\mathcal{I}} \cong G_{\mathcal{I}_{\min}}$.

We denote the set of all max cliques of G by $\mathcal{M}_G$. Let $\mathcal{I}$ be a minimal interval representation of G and I_v denote the interval in $\mathcal{I}$ that corresponds to vertex $v \in V$.

Lemma 329 ([50], Lemma 4.3.1)**.** *For each $k \in \bigcup \mathcal{I}$, the set $M(k) = \{v \mid k \in I_v\}$ is a max clique of G. Furthermore, for any max clique M of G, we have $\bigcap_{v \in M} I_v = \{k\}$ for some $k \in \bigcup \mathcal{I}$.* □

Thus, any minimal interval representation of G induces a linear order on $\mathcal{M}_G$ which has the property that each vertex is contained in *consecutive* max cliques. It is known [23, 57] that a graph G is an interval graph if and only if its max cliques can be brought into a linear order, so that each vertex of G is contained in consecutive max cliques.

Thus, max cliques play an important role for the structure of interval graphs. Our canonization procedure essentially relies on bringing the max cliques of an interval graph into a suitable order.

The maximal cliques of an interval graph $G = (V, E)$ can be handled rather easily in our logic:

Vertices $w, w' \in V$ *span* a max clique A if A is the only max clique containing w and w'.

Lemma 330. *Each max clique of an interval graph can be spanned by two vertices.*

Proof. Let $\mathcal{I}$ be a minimal interval representation of G, and let M be a max clique of G. By Lemma 329, there is a $k \in \mathbb{N}$ such that $\bigcap_{v \in M} I_v = \{k\}$. Thus, there are vertices $w, w' \in M$ with $\min I_w = k$ and with $\max I_{w'} = k$. Suppose there is a max clique M' of G with $w, w' \in M'$. Then $\bigcap_{v \in M'} I_v = \{k'\}$ for some $k' \in \mathbb{N}$. As $\min I_w = k$ and $\max I_{w'} = k$ we have $k = k'$, and therefore, $M = M'$. □

As a consequence (see Section 2.8.2), the max cliques of G as well as the equivalence relation on vertex pairs defining the same max clique are first-order definable.

11.2. Twinless Modular Decompositions

Our canonization procedure relies on a specific decomposition of graphs, which we call *twinless modular decomposition*. It is similar to the modular decomposition introduced by Gallai (see Section 3.2). The basic building blocks are also modules.

As in Gallai's modular decomposition, the twinless modular decomposition decomposes graph G into its connected components $W_1, \ldots, W_k$ if G is not connected, and into the connected components $W_1, \ldots, W_k$ of the complement graph $\overline{G}$ of G if the complement graph $\overline{G}$ is not connected. For graphs G with more than one vertex where both G and $\overline{G}$ are connected, the set of maximal proper modules of G is a partition of G's vertex set, which is used to decompose the graph G in Gallai's modular decomposition. For the twinless modular decomposition, we use a slightly different partition into modules $W_1, \ldots, W_k$ in this case. We define it in Section 11.4. The main difference between our decomposition and Gallai's is that we do not bother to create extra modules for sets of pairwise connected twins[1] since we can handle them perfectly well with our methods.

Let $\mathcal{W}_G$ be the set of modules $W_1, \dots, W_k$ if G has more than one vertex, and $\mathcal{W}_G := \{V\}$ if $|V| = 1$. Let $\sim_G$ be the equivalence relation on V corresponding to the partition $\mathcal{W}_G$ (i.e., $v \sim_G w$ whenever $v, w \in W_i$ for some $i \in [k]$). Let us consider the graph

$$L_G := (V/_{\sim_G}, E_{L_G}), \quad \text{where} \quad E_{L_G} := \{(u/_{\sim_G}, v/_{\sim_G}) \mid (u,v) \in E\}.$$

Intuitively, L_G is the graph obtained from G by collapsing all the modules in $\mathcal{W}_G$ into single vertices. Since each pair of modules $W_i, W_j \in \mathcal{W}_G$, $i \neq j$, is either completely connected or completely disconnected, G is completely determined by L_G and the graphs $G[W_i]$, for $i \in [k]$, where $G[W_i]$ denotes the subgraph of G induced by the vertices in W_i. By decomposing the $G[W_i]$, $i \in [k]$, inductively until we arrive at singleton sets everywhere, we obtain G's twinless modular decomposition.

We define the *twinless modular decomposition tree* $T(G)$ of a graph G recursively. If $|V| = 1$, then $T(G)$ is the rooted tree that consists of only one node, node V, which is the root of $T(G)$. Let $|V| > 1$. Then, the twinless modular decomposition tree $T(G)$ is a rooted tree which consists of a node V, which is the root of $T(G)$, and of subtrees $T(G[W])$ for all $W \in \mathcal{W}_G$. We obtain $T(G)$ by adding an edge from V to the root of $T(G[W])$ for all $W \in \mathcal{W}_G$. This twinless modular decomposition tree is uniquely determined for every graph G.

Notice that for an interval graph G where $\overline{G}$ is not connected, all except one connected component of $\overline{G}$ must contain only a single vertex. Each of these single vertices is adjacent to all other vertices in G. We call a vertex with that property an *apex*. Thus, if G is an interval graph with $\overline{G}$ disconnected, then $\mathcal{W}_G = \bigcup_{a \in A}\{\{a\}\} \cup \{V \setminus A\}$ where A is the set of apices, and the graph L_G is isomorphic to a complete graph. Also, if G contains an apex, then either $|V| = 1$ or $\overline{G}$ is not connected.

The following three sections are about defining and canonizing the graph L_G for an interval graph G. This is easy for unconnected graphs G or graphs that have at least one apex. Thus, we will consider connected graphs without any apices.

11.3. Extracting Information About the Order of Maximal Cliques

Throughout this section let G be a connected interval graph without any apices.

We call a max clique C a possible *end* of G if there is a minimal interval representation $\mathcal{I}$ of G so that C is minimal with respect to the order induced by $\mathcal{I}$.

Now we pick a max clique M of G. We assume it to be a possible end of G, and present a recursive procedure that turns out to recover all the information about the order of the max cliques induced by choosing M as an end of G.

Let $M \in \mathcal{M}_G$. The binary relation $\prec_M$ is defined recursively on the elements of $\mathcal{M}_G$ as

[1] We call two vertices v and w *twins* if $N(v) \cup \{v\} = N(w) \cup \{w\}$.

follows:

$$\text{Initialisation:}\quad M \prec_M C \text{ for all } C \in \mathcal{M}_G \setminus \{M\}$$

$$C \prec_M D \quad \text{if} \begin{cases} \exists E \in \mathcal{M}_G \text{ with } E \prec_M D \text{ and } (E \cap C) \setminus D \neq \emptyset \quad \text{or} \\ \exists E \in \mathcal{M}_G \text{ with } C \prec_M E \text{ and } (E \cap D) \setminus C \neq \emptyset. \end{cases} \qquad (\bigstar)$$

By exploiting the definition's symmetry, $\prec_M$ can be defined through a reachability query in the undirected graph O_M, which has pairs of max cliques from $\mathcal{M}_G$ as its vertices, and in which two vertices (A, B) and (C, D) are connected by an edge whenever $A \prec_M B$ implies $C \prec_M D$ with one application of ($\bigstar$). Hence:

Lemma 331. *There exists an* STC*-formula that for any interval graph G and for any max clique M of G defines the relation $\prec_M$.*

We now state a few important properties of $\prec_M$. Recall that a binary relation R on a set A is *asymmetric* if $ab \in R$ implies $ba \notin R$ for all $a, b \in A$. In particular, asymmetric relations are irreflexive.

Lemma 332 ([49], Lemma IV.3, Corollary IV.6, Lemma IV.7)**.** *Let M be a max clique of an interval graph G. Then the following properties are equivalent:*

- *$\prec_M$ is asymmetric,*
- *$\prec_M$ is a strict weak order (that is, $\prec_M$ is irreflexive, transitive, and incomparability is an equivalence relation),*
- *M is a possible end of G.* □

Since $\prec_M$ is STC-definable and asymmetry of $\prec_M$ is FO-definable, the preceding lemma gives us a way to define possible ends of interval graphs in STC+C.

If M is a possible end of an interval graph G, we have the following connection between $\prec_M$ and a minimal interval representation of G, which has M as its first clique.

Lemma 333 ([49], Lemma IV.7)**.** *Let M be a possible end of an interval graph G. Let $\mathcal{I}$ be a minimal interval representation of G, which has M as its first clique, and let $\lhd_{\mathcal{I}}$ be the linear order $\mathcal{I}$ induces on the max cliques of G. Then $\prec_M \subseteq \lhd_{\mathcal{I}}$.* □

Lemma 334. *Let $\mathcal{C} \subset \mathcal{M}_G$ be a set of max cliques with $M \notin \mathcal{C}$. Suppose that for all $B \in \mathcal{M}_G \setminus \mathcal{C}$ and any $C, C' \in \mathcal{C}$ it holds that $B \cap C = B \cap C'$. Then the max cliques in $\mathcal{C}$ are mutually incomparable with respect to $\prec_M$.*

Proof. By a derivation chain of length k we mean a finite sequence $X_0 \prec_M Y_0$, $X_1 \prec_M Y_1$, …, $X_k \prec_M Y_k$ such that $X_0 = M$ and for each $i \in [k]$, the relation $X_i \prec_M Y_i$ follows from $X_{i-1} \prec_M Y_{i-1}$ by one application of ($\bigstar$). Clearly, whenever it holds that $X \prec_M Y$ there is a derivation chain that has $X \prec_M Y$ as its last element.

Suppose for contradiction that there are $C, C' \in \mathcal{C}$ with $C \prec_M C'$. Let $M \prec_M Y_0$, $X_1 \prec_M Y_1$, …, $X_k \prec_M Y_k$ be a derivation chain for $C \prec_M C'$. Since $X_k = C$, $Y_k = C'$, and $M \notin \mathcal{C}$, there is a largest index i so that either X_i or Y_i is not contained in $\mathcal{C}$.

If $X_i \notin \mathcal{C}$, then $X_{i+1} \in \mathcal{C}$ and $Y_i = Y_{i+1} \in \mathcal{C}$ and it holds that $(X_i \cap X_{i+1}) \setminus Y_{i+1} \neq \emptyset$. Hence, $X_i \cap X_{i+1} \neq X_i \cap Y_{i+1}$, contradicting the assumption of the lemma. Similarly, if

$Y_i \notin \mathcal{C}$, then $Y_{i+1} \in \mathcal{C}$ and $X_i = X_{i+1} \in \mathcal{C}$ and it holds that $(Y_i \cap Y_{i+1}) \setminus X_{i+1} \neq \emptyset$. Thus, $Y_i \cap Y_{i+1} \neq Y_i \cap X_{i+1}$, again a contradiction. □

The *width* of a vertex $v \in V$ in G, denoted $\text{width}(v)$, is the number of max cliques of G that v is contained in. Recall from Section 11.1 that the equivalence relation on vertex pairs defining the same max clique is first-order definable. Note that, since equivalence classes can be counted in STC+C [49, Lemma II.7], the width of a vertex is STC+C-definable on the class of all interval graphs.

Lemma 335 ([49], Lemma IV.4, Corollary IV.5)**.** *Suppose M is a possible end of G and $\mathcal{C}$ is a maximal set of $\prec_M$-incomparable max cliques. Then*

- $B \cap C = B \cap C'$ *for all* $C, C' \in \mathcal{C}$, $B \in \mathcal{M}_G \setminus \mathcal{C}$,
- $S_{\mathcal{C}} := \bigcup_{C \in \mathcal{C}} C \setminus \bigcup_{B \in \mathcal{M}_G \setminus \mathcal{C}} B$ *is a module of* G*, and*
- $S_{\mathcal{C}} = \{v \in \bigcup \mathcal{C} \mid \text{width}(v) \leq |\mathcal{C}|\}$. □

Finally, let $\sim_M^G$ be the equivalence relation on V for which $x \sim_M^G y$ if and only if $x = y$, or there exists a maximal set $\mathcal{C}$ of incomparable max cliques (with respect to $\prec_M$) with $|\mathcal{C}| > 1$ so that $x, y \in S_{\mathcal{C}}$. Let $G_M = G/_{\sim_M^G} := (V/_{\sim_M^G}, E_M)$, where $E_M := \{(u/_{\sim_M^G}, v/_{\sim_M^G}) \mid (u, v) \in E\}$. It is easy to check that $\sim_M^G$ and the graph G_M are STC+C-definable.

If $\mathcal{C}$ is a maximal set of $\prec_M$-incomparables in G with $|\mathcal{C}| > 1$, then there is precisely one max clique $M_{\mathcal{C}}$ in G_M which contains all the equivalence classes associated with $\mathcal{C}$, i.e., $M_{\mathcal{C}} = \{v/_{\sim_M^G} \mid v \in \bigcup \mathcal{C}\}$. We conclude:

Lemma 336. *$\prec_M$ induces a strict linear order on G_M's max cliques. In particular, G_M is an interval graph.* □

11.4. Modules $\mathcal{W}_G$ and the Graph L_G

We are now ready to give the definition of the set $\mathcal{W}_G$, which we mentioned in Section 11.2, for connected interval graphs G without an apex. Furthermore, we take a look at the graph L_G from Section 11.2 and its properties. In particular, we prove that L_G and an isomorphic copy of L_G on the number sort are STC+C-definable.

Let $G = (V, E)$ be a connected interval graph without an apex. Then G contains more than one max clique. Let $\mathfrak{P}_G$ be the set of all maximal proper subsets $\mathcal{C}$ of $\mathcal{M}_G$ with the property that for any $B \in \mathcal{M}_G \setminus \mathcal{C}$ we have $B \cap C = B \cap C'$ for all $C, C' \in \mathcal{C}$. We must have $|\mathfrak{P}_G| \geq 3$ since G is connected and no vertex may be included in all max cliques of G. Furthermore, if $\mathcal{C}, \mathcal{C}' \in \mathfrak{P}_G$ and $\mathcal{C} \neq \mathcal{C}'$, then $\mathcal{C} \cap \mathcal{C}' = \emptyset$. To see this, suppose that $D \in \mathcal{C} \cap \mathcal{C}'$. Then $B \cap A = B \cap D = B \cap C$ for all $A, C \in \mathcal{C} \cup \mathcal{C}'$ and $B \notin \mathcal{C} \cup \mathcal{C}'$. As $\mathcal{M}_G \setminus (\mathcal{C} \cup \mathcal{C}')$ is not empty ($|\mathfrak{P}_G| \geq 3$), $\mathcal{C} \cup \mathcal{C}'$ is a proper subset of $\mathcal{M}_G$ satisfying the above property, which contradicts the maximality of $\mathcal{C}$ and $\mathcal{C}'$. We conclude that $\mathfrak{P}_G$ is a partition of $\mathcal{M}_G$.

For each $\mathcal{C} \in \mathfrak{P}_G$ with $|\mathcal{C}| \geq 2$ we define $S_{\mathcal{C}} = \bigcup \mathcal{C} \setminus \bigcup(\mathcal{M}_G \setminus \mathcal{C})$. The correspondence in names to the modules $S_{\mathcal{C}}$ as defined in Lemma 335 is intended, of course, and makes sense since the sets $\mathcal{C} \in \mathfrak{P}_G$ enjoy the same interaction properties with the rest of the graph as maximal sets of $\prec_M$-incomparable max cliques (cf. Lemma 335).

We can now define the modules $\mathcal{W}_G$ mentioned in Section 11.2 for connected interval graphs G without an apex. We let $\mathcal{S} := \{S_\mathcal{C} \mid \mathcal{C} \in \mathfrak{P}_G \text{ with } |\mathcal{C}| \geq 2\}$, and define

$$\mathcal{W}_G := \mathcal{S} \cup \bigcup_{v \in V \setminus \bigcup \mathcal{S}} \{\{v\}\}.$$

From the fact that $\mathfrak{P}_G$ is a partition of $\mathcal{M}_G$, we conclude that $\mathcal{W}_G$ forms a partition of V, whereby inducing the equivalence relation $\sim_G$ on V. In the following, we call this equivalence relation alternatively $\sim_{\mathfrak{P}_G}$.

In the following we construct graphs isomorphic to L_G that will help us to define L_G in STC+C. Let Z_M be the max clique which is $\prec_M$-maximal in G_M. Now we forget about $\prec_M$ and consider $\prec_{Z_M}$ on G_M. We write

$$L_M := G_M/_{\sim^{G_M}_{Z_M}} = (V(G_M)/_{\sim^{G_M}_{Z_M}}, E(G_M)/_{\sim^{G_M}_{Z_M}})$$

with $E(G_M)/_{\sim^{G_M}_{Z_M}} = \{(u/_{\sim^{G_M}_{Z_M}}, v/_{\sim^{G_M}_{Z_M}}) \mid (u,v) \in E(G_M)\}$. Lemma 336 implies again that $\prec_{Z_M}$ induces a strict linear order on the max cliques of L_M.

Lemma 337. *Let G be a connected interval graph that does not have an apex, and let $M_1, \ldots, M_k$ be its possible ends. Then all of the graphs L_{M_l}, $l \in [k]$, are isomorphic to L_G.*

Proof. Equivalence relation $\sim_{\mathfrak{P}_G}$ does the same as $\sim^G_M$, only that it is based on $\mathfrak{P}_G$ instead of the (finer) partition of max cliques induced by a strict weak ordering $\prec_M$.

Our goal is to show that each L_M with $M \in \{M_1, \ldots, M_k\}$ is isomorphic to $G/_{\sim_{\mathfrak{P}_G}}$. For this it is enough to show that the concatenation of equivalence relation $\sim^G_M$ with $\sim^{G_M}_{Z_M}$ is equal to $\sim_{\mathfrak{P}_G}$. Whenever $\mathcal{C} \in \mathfrak{P}_G$ and $M \notin \mathcal{C}$, Lemma 334 implies that the max cliques in $\mathcal{C}$ are $\prec_M$-incomparable. As the sets in $\mathfrak{P}_G$ were chosen to be maximal, $\mathcal{C}$ is also a maximal set of $\prec_M$-incomparables (Lemma 335). It follows that $\sim_{\mathfrak{P}_G}$ is equal to $\sim^G_M$ on $\bigcup_{M \notin \mathcal{C} \in \mathfrak{P}_G} \mathcal{C}$.

When forming $G_M = G/_{\sim^G_M}$, each maximal set of $\prec_M$-incomparable max cliques $\mathcal{C}$ is replaced by the max clique $M_\mathcal{C} = \{v/_{\sim^G_M} \mid v \in \bigcup \mathcal{C}\}$. Note that this is also true when $\mathcal{C}$ consists of just one max clique. As a result, $\mathfrak{P}_G$ induces a partition $\mathfrak{P}_M$ of the max cliques of G_M. Also, if $\mathcal{C}_M$ is the equivalence class of $\mathfrak{P}_M$ which contains M, then $\mathcal{C}_M$ is the only equivalence class of $\mathfrak{P}_M$ which is possibly not a singleton. As $|\mathfrak{P}_M| \geq 3$, we have $Z_M \notin \mathcal{C}_M$.

The final step is to show that $\sim_{\mathfrak{P}_M}$ equals $\sim^{G_M}_{Z_M}$ on G_M. If $v/_{\sim^G_M}$ is a vertex of G_M and $v/_{\sim^G_M}$ is an equivalence class of $\sim^G_M$ with $|v/_{\sim^G_M}| > 1$, then $v/_{\sim^G_M}$ is only contained in one max clique of G_M. Hence, $\mathfrak{P}_M$ inherits from $\mathfrak{P}_G$ the property that it partitions the set of max cliques $\mathcal{M}_{G_M}$ of G_M into maximal sets $\mathcal{C}$ so that for any $B \in \mathcal{M}_{G_M} \setminus \mathcal{C}$ we have $B \cap C = B \cap C'$ for all $C, C' \in \mathcal{C}$. Arguing analogously as above, it follows that $\sim_{\mathfrak{P}_M}$ equals $\sim^{G_M}_{Z_M}$. Therefore, $\sim_{\mathfrak{P}_G}$ is equal to the concatenation of $\sim^G_M$ with $\sim^{G_M}_{Z_M}$ and L_M is isomorphic to L_G. □

The following two corollaries are consequences of Lemma 337.

Corollary 338. *The graph L_G is an interval graph, for all interval graphs G.*

Proof. Clearly, L_G is an interval graph if G is not connected or has an apex. For connected graphs G without apices, it follows from Lemma 336 and the previous lemma that L_G is an interval graph. ☐

Corollary 339. *The graph L_G is* STC+C*-definable for all interval graphs G, that is, there are* STC+C*-formulas $\varphi_\sim$ and φ_L such that $\varphi_\sim$ defines the equivalence relation $\sim_G$, and φ_L the edge relation of the graph L_G.*[2]

Proof. If G is not connected or G contains an apex, then $\sim_G$ is STC-definable. If G is connected and does not contain an apex, then for each possible end M of G the concatenation of equivalence relation $\sim_M^G$ with $\sim_{Z_M}^{G_M}$ is equal to $\sim_G$, which was shown in the proof of Lemma 337. The STC+C-definability of equivalence relation $\sim_G$ is a direct consequence of the STC+C-definability of the possible ends M and the equivalence relation $\sim_M^G$, Lemma 336, which allows us to define max clique Z_M, and the STC+C-definability of $\sim_{Z_M}^{G_M}$.

Since all equivalence classes of $\sim_G$ are modules of G, the edge relation of L_G can be defined as the set of all edges of G between vertices in different equivalence classes. ☐

Notice that, if A is a max clique of G, then

$$A_{L_G} := \{v/_{\sim_G} \mid v \in A\}$$

is a max clique of L_G, and that all max cliques of L_G are of this form. In particular, for each possible end M of G the max clique M_{L_G} is a possible end of L_G, and for each possible end N of L_G there exists a possible end M of G such that $N = M_{L_G}$.

Lemma 340. *Let G be a connected graph without any apices.*

1. *For each possible end N of L_G the relation $\prec_N$ is a strict linear order on the max cliques of L_G.*
2. *The graph L_G has exactly two ends $N_1, N_2 \in \mathcal{M}_{L_G}$, and the strict linear order $\prec_{N_i}$ is the reverse of $\prec_{N_{3-i}}$.*

Proof.

1. Let is assume $\prec_N$ is not a strict linear order. Then $\prec_N$ is a strict weak order (Lemma 332) and there exists a set $\mathcal{C}$ of $\prec_N$-incomparable max cliques of L_G. By Lemma 334 we have $B \cap C = B \cap C'$ for all $C, C' \in \mathcal{C}$ and all $B \in \mathcal{M}_{L_G} \setminus \mathcal{C}$, which is a contradicting to the construction of L_G.

[2] We do not define the graph L_G explicitly, but rather implicitly within G. That is, we do not single out a representative of each equivalence class $v/_{\sim_G}$ of $\sim_G$, but treat all vertices in $v/_{\sim_G}$ as representatives of $v/_{\sim_G}$.

2. Let G be a connected graph without any apices. Then $|\mathcal{M}_{L_G}| > 1$. Thus, there exist at least two possible ends of L_G. Let us assume there exists two possible ends N and N' of L_G such that N is neither the $\prec_{N'}$-least nor the $\prec_{N'}$-most max clique. There are minimal interval representations $\mathcal{I}$ and $\mathcal{I}'$, which have N and N' as first clique, respectively. As $\prec_N$ and $\prec_{N'}$ are strict linear orders, they are the strict linear orders that $\mathcal{I}$ and $\mathcal{I}'$ induce on the max cliques of G by Lemma 333. Lemma 333 also implies that N' is not the $\prec_N$-largest max clique. Clearly, $N \prec_N N'$ and $N' \prec_{N'} N$. Let $\mathcal{A}$ be the set of max cliques A of L_G with $N' \prec_N A$, and $\mathcal{A}'$ be the set of max cliques A of L_G such that $N \prec_{N'} A$. Further, let $\mathcal{B}$ be the set of max cliques B of L_G where $N \prec_N B \prec_N N'$, and let $\mathcal{B}'$ be the set of max cliques B of L_G with $N' \prec_{N'} B \prec_{N'} N$.
First, assume there exists a max clique $A \in \mathcal{A}$ such that $N \prec_{N'} A$, and let A be $\prec_N$-least with that property. Let A_1 be the $\prec_N$-predecessor of A. Thus, $A_1 = N'$ or $A_1 \in \mathcal{A}$. As A is $\prec_N$-least with that property, we have $A_1 \prec_{N'} N$ in both cases. Let $\mathcal{C} := \{N'' \in \mathcal{M}_{L_G} \mid N'' \prec_N A\}$. Then $|\mathcal{C}| \geq 2$ and $\mathcal{M}_{L_G} \setminus \mathcal{C} \neq \emptyset$. We show that for any $B \in \mathcal{M}_{L_G} \setminus \mathcal{C}$ we have $B \cap C = B \cap C'$ for all $C, C' \in \mathcal{C}$. Let $C \in \mathcal{C}$. Then we have $C \cap A \subseteq A_1 \cap A$ as $C \preceq_N A_1 \prec_N A$. Further, we have $A_1 \cap A \subseteq N \cap A$ since $A_1 \prec_{N'} N \prec_{N'} A$. Finally, $N \cap A \subseteq C \cap A$ follows from $N \preceq_N C \prec_N A$. It follows that $C \cap A = N \cap A$ for all $C \in \mathcal{C}$. Thus, each vertex that is contained in $A \cap C$ for any max clique $C \in \mathcal{C}$ is contained in all max cliques in $\mathcal{C}$. Now, each vertex that is in the intersection of $B \in \mathcal{M}_{L_G} \setminus \mathcal{C}$ and any max clique $C \in \mathcal{C}$, is also in $A \cap C$. Therefore, each vertex in $B \cap C$ for any max clique $C \in \mathcal{C}$ is contained in all max cliques in $\mathcal{C}$. As a consequence, we obtain $C \cap B = C' \cap B$ for all $C, C' \in \mathcal{C}$ where $|\mathcal{C}| \geq 2$, which is a contradiction to the construction of L_G.
Next, let us assume there does not exist a max clique $A \in \mathcal{A}$ such that $N \prec_{N'} A$. Thus, $\mathcal{A} \subseteq \mathcal{B}'$ and $\mathcal{A}' \subseteq \mathcal{B}$. Let A be the $\prec_N$-predecessor of N'. If $A \in \mathcal{A}'$, let $\mathcal{C} := \{N'' \in \mathcal{M}_{L_G} \mid A \prec_N N''\}$. As above we can show that $C \cap A = N' \cap A$ for all $C \in \mathcal{C}$, which analogously leads to a contradiction. If $A \notin \mathcal{A}'$, then let A' be the $\prec_N$-successor of the $\prec_N$-largest max clique of all max cliques in $\mathcal{A}'$. Then $A' \prec_N N'$, and $A'' \prec_N A'$ for all $A'' \in \mathcal{A}'$. Further, let $\mathcal{C} := \{N'' \in \mathcal{M}_{L_G} \mid N'' \prec_N A'\}$. Again, it is possible to show that $C \cap A' = N \cap A'$ for all $C \in \mathcal{C}$ and derive a contradiction. Thus, there are precisely two ends $N_1, N_2 \in \mathcal{M}_{L_G}$. It follows from Lemma 333 that the strict linear order $\prec_{N_i}$ is the reverse of $\prec_{N_{3-i}}$. ☐

Lemma 341. *There exists an* STC+C*-formula* ρ *that defines all pairs* $(u, v), (u', v') \in V^2$ *such that* (u, v) *and* (u', v') *span max cliques* M *and* A *of* G*, where* M_{L_G} *is an end of* L_G*, and* A_{L_G} *appears within the first* $\lfloor \frac{m}{2} \rfloor$ *max cliques of* L_G *with respect to* $\prec_{M_{L_G}}$*.*

Proof. According to Lemma 340 there are exactly two strict linear orderings of L_G's max cliques, each the reverse of the other. By Corollary 339 we can define L_G in STC+C. Further, we can define the ends of L_G (Lemma 332), and for an end N of L_G we can define the strict linear order $\prec_N$ (Lemma 331) in STC+C. Hence, given max cliques M and A, we can check whether M_{L_G} is an end of L_G and whether A_{L_G} appears within the first $\lfloor \frac{m}{2} \rfloor$ max cliques of L_G regarding $\prec_{M_{L_G}}$ in STC+C. ☐

Lemma 342. *There is an* STC+C*-formula that is satisfied for all pairs* $(u, v) \in V^2$ *and numbers* $p \in N(V)$ *where* (u, v) *spans a max clique* A *of* G *and there is a strict linear order* $\prec$ *on the max cliques of* L_G *such that* A_{L_G} *appears at position* p *regarding* $\prec$*.*

Proof. By Corollary 339 we can define L_G in STC+C. Further, we can define the ends of L_G (Lemma 332), and for an end N of L_G we can define the strict linear order $\prec_N$ (Lemma 331) in STC+C. Hence, given max clique A, we can check whether A_{L_G} appears at position p regarding $\prec_N$ for some end N of L_G in STC+C. □

Lemma 343. *There is an* STC+C*-formula that defines an isomorphic copy of L_G on the number sort for all interval graphs G. We denote this isomorphic copy of L_G on the number sort by $K(L_G)$.*

Proof. Lemma 343 is easy to see for graphs that are not connected or contain an apex. For connected interval graphs G that do not have any apices, Lemma 343 follows directly from the definability of L_G (Corollary 339) and Section IV.B (and Remark IV.2) in [49]. In Section IV.B in [49] Laubner shows that there is an STC+C-formula ε that defines for each interval graph an ordered copy on the number sort if there is an STC+C-definable strict linear order on the graph's max cliques. More precisely, let $\prec$ be an STC+C-definable strict linear order on the graph's max cliques. Let A_v be the $\prec$-least max clique of G containing v. Then then the binary relation $<^G$, were

$$v <^G w :\Longleftrightarrow \begin{cases} A_v \prec A_w, \text{ or} \\ A_v = A_w \text{ and } \operatorname{width}(v) < \operatorname{width}(w) \end{cases}$$

is a strict weak order on the vertex set where two vertices are incomparable iff they are connected twins. If $[v]$ denotes the equivalence class of vertices incomparable to v, then $[v]$ is represented by the numbers from the interval $[a+1, a+|[v]|]$, where a is the number of vertices which are strictly $<^G$-smaller than v. This way we obtain an isomorphic copy of a graph on the number sort. It is not hard to see that this isomorphic copy is STC+C-definable. Let us consider the strict linear orders $\prec_N$ of L_G for ends N of L_G. The graph L_G, the ends N and these strict linear orders are STC+C-definable (Corollary 339, Lemma 332, Lemma 331). For each possible end N, formula ε can use $\prec_N$ to define an ordered copy of L_G. We choose the ordered copy that is lexicographically minimal as canon $K(L_G)$ of L_G. Clearly, $K(L_G)$ is STC+C definable. □

Let G be a connected interval graph. We call L_G *symmetric* if $(L_G, \prec_{N_1})$ is isomorphic to $(L_G, \prec_{N_2})$ where $\{N_1, N_2\}$ is the set of ends of L_G.[3] Note that $|\mathcal{M}_{L_G}| = 1$ iff G has an apex. Thus, L_G is symmetric for interval graphs G with apices.

Corollary 344. *Let G be connected. If $|\mathcal{M}_{L_G}| = 1$ or L_G is not symmetric, then there is an* STC+C*-formula that defines the distinguished strict linear order $\prec_{L_G}$ on L_G's max cliques such that $(L_G, \prec_{L_G})$ is isomorphic to $(K(L_G), \prec_{K(L_G)})$ where $\prec_{K(L_G)}$ is the lexicographically minimal strict linear order on the max cliques of the ordered graph $K(L_G)$.*[4]

Proof. Let G be connected. If $|\mathcal{M}_{L_G}| = 1$, then L_G is a complete graph, and the unique strict linear order on the max cliques of L_G is STC+C-definable. Now let L_G be not

[3] $(L_G, \prec_{N_2})$ is a structure, if $\prec_{N_i}$ is understood as a binary relation on pairs of spanning vertices of max cliques of L_G instead of pairs of max cliques.

[4] Again, we understand a strict linear order on the max cliques of a graph as a binary relation on pairs of spanning vertices of max cliques.

symmetric. Then G is a connected graph without any apices. Similar to formula ε from the proof above, we can define a formula γ that defines the strict linear order $\prec_{N'}$ for the possible end $N' \in \mathcal{M}_{L_G}$ for which formula ε defines the canon of L_G. As L_G is not symmetric, and there exist only two ends N that lead to strict linear orders $\prec_N$ which are the reverse of each other (Lemma 340), γ defines a distinguished strict linear order $\prec_{N'}$. Further, $[|N'|]$ is the $\prec_{[|N'|]}$-least max clique of $K(L_G)$. Thus $(L_G, \prec_{N'}) \cong (K(L_G), \prec_{[|N'|]})$, and $\prec_{[|N'|]}$ is the strict linear order on the max cliques of $K(L_G)$ that is lexicographically minimal. □

11.5. The Colored Twinless Modular Decomposition Tree

To obtain a complete invariant of an interval graph $G = (V, E)$, we construct a refinement of the twinless modular decomposition tree, the colored twinless modular decomposition tree, in this section.

Let us consider the twinless modular decomposition tree $T(G)$ of an interval graph G. We call a module $W \in V(T(G))$ a *decomposition module* if $W = V$, or $|W| > 1$ and $G[W^*]$ is a connected graph, where W^* is the parent of W in $T(G)$. All modules W where $G[W^*]$ is not connected are called *component modules*. We let $\mathcal{W}_G^{dec}$ be the set of all decomposition modules and $\mathcal{W}_G^{con}$ be the set of all component modules occurring in the twinless modular decomposition tree of G. Note that all nodes of the twinless modular decomposition tree are either decomposition modules, component modules or singleton sets with a parent W^* where $G[W^*]$ is connected. Further, note that component modules cannot be adjacent. Therefore, each component module is a connected component of a decomposition module.

Let $P' := \{(M, n) \mid M \in \mathcal{M}_G, n \in [|V|]\}$. Recall the definition of the *width* of a vertex from Section 11.3, and that it is STC+C-definable. For each $(M, n) \in P'$, define $V_{M,n}$ as the connected component of $G[\{v \in V \mid \text{width}(v) \leq n\}]$ which intersects with M if non-empty or the empty set otherwise, and let $G_{M,n} := G[V_{M,n}]$. Now let P be the set of those $(M, n) \in P'$ for which the following properties are satisfied:

1. The number n is maximal among those n' with the property that $V_{M,n'} = V_{M,n}$.
2. For all $m' > n$ where $V_{M,m'}$ is a module, $V_{M,n}$ is a subset of a non-singleton equivalence class of $\sim_{G_{M,m'}}$, or there exists a vertex $a \in V_{M,m'} \setminus V_{M,n}$ that is an apex of $G_{M,m'}$.

As max cliques, the width of vertices and connectivity are STC+C-definable, $V_{M,n}$ is STC+C-definable as well. Further, with regard to Corollary 339 it is not hard to see that the above two properties are STC+C-definable. Therefore, we can make the following observation.

Observation 345. *There is an* STC+C*-formula* $\varphi_P(x, y, p)$ *such that for all interval graphs* $G = (V, E)$*, all* $v, w \in V$*, and all* $n \in [|V|]$*, we have* $G \models \varphi[v, w, n]$ *iff vertices* v *and* w *span a max clique* M *of* G *and* $(M, n) \in P$*.*

In the following we show that the sets $V_{M,n}$ with $(M, n) \in P$ are precisely the connected components of the graphs induced by the decomposition modules.

Lemma 346. *The set D is a connected component of $G[W]$ for a decomposition module W if, and only if, there exists a pair $(M, n) \in P$ such that $D = V_{M,n}$.*

Proof. Notice that for all modules W of G and all max cliques C of G with $C \cap W \neq \emptyset$ the set $W \cap C$ is a max clique of $G[W]$, and every max clique of $G[W]$ is of that form. Further, an easy induction shows that for all modules $W \in \mathcal{W}_G^{dec} \cup \mathcal{W}_G^{con}$ the following properties are satisfied:

(A) Let $C, C' \in \mathcal{M}_G$ be max cliques of G with $C \neq C'$ where $C \cap W \neq \emptyset$ and $C' \cap W \neq \emptyset$. Then for max cliques $C \cap W$, $C' \cap W$ of $G[W]$ we have $C \cap W \neq C' \cap W$.
(B) Let $\mathcal{C} := \{C \in \mathcal{M}_G \mid C \cap W \neq \emptyset\}$. Then for all $B \in \mathcal{M}_G \setminus \mathcal{C}$ and all $C, C' \in \mathcal{C}$ we have $B \cap C = B \cap C'$.
(C) Let $\mathcal{C}$ be the set from (B). Then $W = \bigcup_{C \in \mathcal{C}} V_{C,c}$ where $c := |\mathcal{C}|$ if $G[W]$ has an apex and $c := |\mathcal{C}| - 1$ if $G[W]$ has no apices, and for each $C \in \mathcal{C}$ the set $V_{C,c}$ is a connected component of W.

In order to show Lemma 346, we also need the following properties:

Claim 1. If $V_{M,k}$ is a connected component of $G[W]$ for a decomposition module W of G, and $V_{M,k} \subsetneq V_{M,l}$ for an $l > k$, then $W \subsetneq V_{M,l}$.

Proof. Let c be defined as in Property (C) for decomposition module W. Then we have $V_{M,k} = V_{M,c}$ by Property (C). Let k' be the maximum width of a vertex in W. Clearly, $k' \leq c$ and $V_{M,c} = V_{M,k'}$. Thus, $V_{M,k} = V_{M,k'}$, and we can assume that $k \geq k'$. Further, for each $l > k$ with $V_{M,k} \subsetneq V_{M,l}$ we have $V_{M,l} \not\subseteq W$ as $V_{M,l} \subseteq W$ leads to a contradiction, because $G[V_{M,l}]$ is connected and $V_{M,k} \subsetneq V_{M,l}$ is a connected component of $G[W]$. Thus, $V_{M,l} \setminus W$ is non-empty. As the connected component $V_{M,k}$ is a subset of $V_{M,l} \cap W$, the set $V_{M,l} \cap W$ is also non-empty. Now $G[V_{M,l}]$ being connected implies that there exists a vertex $v \in V_{M,l} \setminus W$ that is adjacent to a vertex in $V_{M,l} \cap W$. Since W is a module, v is adjacent to all vertices in W. Further, width$(w) \leq k' \leq l$ for all vertices $w \in W$. It follows that $W \subseteq V_{M,l}$. ⌟

Claim 2. Let $(M, d) \in P'$ and $V_{M,d}$ be a module in $\mathcal{W}_G^{dec} \cup \mathcal{W}_G^{con}$. If $V_{M,d}$ is a clique, then there exists only one max clique $C \in \mathcal{M}_G$ with $C \cap V_{M,d} \neq \emptyset$.

Proof. Since $V_{M,d}$ is a clique, there must exist a max clique $B \in \mathcal{M}_G$ with $V_{M,d} \subseteq B$. Let us assume, there exists a max clique $B' \in \mathcal{M}_G$ different from B with $B' \cap V_{M,d} \neq \emptyset$. According to Property A we have $B \cap V_{M,d} \neq B' \cap V_{M,d}$ and therefore $B' \cap V_{M,d} \subsetneq V_{M,d}$. Since $V_{M,d}$ is a module, $B' \cup V_{M,d}$ is a clique, a contradiction to B' being a max clique. ⌟

Claim 3. Let $(M, d) \in P'$ and $V_{M,d}$ be a module in $\mathcal{W}_G^{dec} \cup \mathcal{W}_G^{con}$. Further, let $0 < d' < d$ be such that $V_{M,d'} \subsetneq V_{M,d}$, and let $A \neq \emptyset$ be the set of apices of $G_{M,d}$. Then we have $V_{M,d'} \subseteq V_{M,d} \setminus A$.

Proof. Let $V_{M,d} \in \mathcal{W}_G^{dec} \cup \mathcal{W}_G^{con}$, and let $0 < d' < d$ be such that $V_{M,d'} \subsetneq V_{M,d}$. Thus, $V_{M,d} \neq \emptyset$. Further, let $\mathcal{C}$ be the set of max cliques $C \in \mathcal{M}_G$ with $C \cap V_{M,d} \neq \emptyset$ and $c := |\mathcal{C}|$. Clearly, $c \geq 1$ and for all vertices $v \in V_{M,d}$ we have width$(v) \leq c$. In the following we show that width$(a) = c$ for each apex a of $G_{M,d}$: Let a be an apex of $G_{M,d}$, and let us assume that there exists a max clique $C \in \mathcal{M}_G$ with $C \cap V_{M,d} \neq \emptyset$ and $a \notin C$.

Apex a is adjacent to the vertices in $C \cap V_{M,d}$, and since $V_{M,d}$ is a module and each vertex in $C \cap V_{M,d}$ is adjacent to the vertices in $C \setminus V_{M,d}$, a is also adjacent to the vertices in $C \setminus V_{M,d}$. Therefore, $C \cup \{a\}$ is a clique, which is a contradiction to C being a maximal clique of G.

As the set A of apices of $G_{M,d}$ is non-empty, there exists a vertex $a \in V_{M,d}$ with $\text{width}(a) = c$. Thus, we have $c \leq d$. It follows that $V_{M,d} = V_{M,c}$, because $\text{width}(v) \leq c$ for all vertices $v \in V_{M,d}$. Now $V_{M,d'} \subsetneq V_{M,d}$ implies $d' < c$. Since $\text{width}(v) = c$ for all apices $a \in A$, we conclude $V_{M,d'} \subseteq V_{M,d} \setminus A$. ⌟

To proceed with the proof of Lemma 346, we first show that if D is a connected component of $G[W]$ for a decomposition module $W \in \mathcal{W}_G^{dec}$, and $M \in \mathcal{M}_G$ with $M \cap D \neq \emptyset$, then there is an $n \in \mathbb{N}$ such that $(M, n) \in P$ and $V_{M,n} = D$.

We proof this by induction on the depth of the twinless modular decomposition tree: Clearly, if D is a connected component of $G[V]$ for the decomposition module V (i.e., a connected component of G), then $D = V_{M,|V|}$ for a max clique M with $M \cap D \neq \emptyset$, and $(M, |V|) \in P$.

Now, let D be a component of $G[W]$ for a module $W \in \mathcal{W}_G^{dec}$ with $W \neq V$. Let c be the number c' of max cliques of G intersecting with W if $G[W]$ has an apex and $c' - 1$ if $G[W]$ has no apices. According to Property C, $V_{M,c} = D$. Let n be maximal with $V_{M,n} = V_{M,c}$. Then $(M, n) \in P'$ and $D = V_{M,n}$. Choosing (M, n) like that ensures that Property 1 is satisfied for (M, n).

It remains to show Property 2. Let $m' > n$ and let $V_{M,m'}$ be a module. According to Property 1 we have $V_{M,n} \subsetneq V_{M,m'}$. Thus, Claim 1 implies $W \subsetneq V_{M,m'}$.

First, let us assume there exists an apex a of $G_{M,m'}$. If there exists an apex of $G_{M,m'}$ in $V_{M,m'} \setminus W$, we have shown Property 2. Thus, let us suppose all apices of $G_{M,m'}$, in particular a, are in W. Since W is a module and $a \in W$, the vertex sets $V_{M,m'} \setminus W$ and W must be completely connected. If W contains two vertices w, w' that are not adjacent, then in the minimal interval representation the interval of each vertex in $V_{M,m'} \setminus W$ has to intersect with the intervals of both w and w'. Thus, the intervals of all vertices in $V_{M,m'} \setminus W$ intersect with each other and each vertex in $V_{M,m'} \setminus W$ is an apex, a contradiction. Next, let us assume W is a clique. Let W^* be the parent module of W in the twinless modular decomposition tree of G. Since W is a decomposition module, $|W| > 1$ and $G[W^*]$ contains either an apex, or is connected and contains no apices. $G[W^*]$ cannot contain an apex, because then all vertices in W^* form a clique and W is not contained in the set $\mathcal{W}_{G[W^*]}$ of modules. If $G[W^*]$ is connected and contains no apices, then $W = S_{\mathcal{C}}$ for $\mathcal{C} \in \mathfrak{P}_{G[W^*]}$ where $\mathcal{C}$ is a set of max cliques of $G[W^*]$ with $|\mathcal{C}| \geq 2$ (see Section 11.4). As $G[W]$ is connected, $W = V_{M,n}$. According to Claim 2 there exists only one max clique C of G with $C \cap W \neq \emptyset$. Consequently, $C' := C \cap W^*$ is the only max clique in $G[W^*]$ with $C' \cap W \neq \emptyset$, a contradiction. Hence, W cannot be a clique, and we have shown that there are no apices of $G_{M,m'}$ in W.

Now let us assume that there does not exist an apex of $G_{M,m'}$. Thus, $\sim_{G_{M,m'}}$ is constructed as described in Section 11.4. Let W' be the parent module W^* of W in the twinless modular decomposition tree of G if W^* is a decomposition module, or if W^* is a component module, let W' be the parent of module W^*. Then W', like W, is a decomposition module. Further, let D' be the connected component of $G[W']$ that contains W. Notice that no matter what set we chose for W', we have $D' = W^*$. According to Property C, there exists an $n' \in [|V|]$ such that $D' = V_{M,n'}$. Let n' be maximal with that property. Therefore, $W^* = V_{M,n'}$ and W^* is a component of a graph that is induced by a decomposition module. By inductive assumption we have $(M, n') \in P$. If $V_{M,n'} = V_{M,m'}$, then $V_{M,n}$ is a subset of the non-singleton equivalence class W of $\sim_{G_{M,m'}}$, and we are done. Therefore, let us assume $V_{M,n'} \neq V_{M,m'}$.

If $n' < m'$, then $V_{M,n} \subseteq W \subsetneq W^* = V_{M,n'} \subsetneq V_{M,m'}$. As (M, n') satisfies Property 2 and there does not exist an apex in $G_{M,m'}$, the set $V_{M,n'}$, and therefore also the set $V_{M,n} \subsetneq V_{M,n'}$, is a subset of a non-singleton equivalence class of $\sim_{G_{M,m'}}$.

It remains to consider the case $m' < n'$. Then $V_{M,n} \subseteq W \subsetneq V_{M,m'} \subsetneq V_{M,n'} = W^*$. If $G[W^*] = G[V_{M,n'}]$ contains an apex, then $W = V_{M,n'} \setminus A$ where A is the set of apices of $G_{M,n'}$. According to Claim 3, $V_{M,m'} \subseteq V_{M,n'} \setminus A$. But this implies $V_{M,m'} \subseteq W$, a contradiction. Therefore, let us assume $W^* = V_{M,n'}$ is connected and does not contain an apex. Then $W = S_{\mathcal{C}}$ for a $\mathcal{C} \in \mathfrak{P}_{G[W^*]}$ with $|\mathcal{C}| \geq 2$ where $\mathfrak{P}_{G[W^*]}$ is the set of all maximal proper subsets $\mathcal{C}'$ of $\mathcal{M}_{G[W^*]}$, the set of max cliques of $G[W^*]$, with the property that for any $B \in \mathcal{M}_{G[W^*]} \setminus \mathcal{C}'$ we have $C \cap B = C' \cap B$ for all $C, C' \in \mathcal{C}'$. For all $C \in \mathcal{M}_{G[W^*]}$ with $C \cap V_{M,m'} \neq \emptyset$, let $f(C)$ be the set $C \cap V_{M,m'}$. As $V_{M,m'}$ is a module, the set $\{f(C) \mid C \in \mathcal{M}_{G[W^*]}, C \cap V_{M,m'} \neq \emptyset\}$ is the set $\mathcal{M}_{G_{M,m'}}$ of max cliques of $G_{M,m'}$. Let $f(\mathcal{C})$ be the set $\{f(C) \mid C \in \mathcal{C}\}$. Then $f(\mathcal{C})$ is exactly the set of max cliques of $G_{M,m'}$ that have a non-empty intersection with W. Let $f(C), f(C') \in f(\mathcal{C})$ and $f(B) \in \mathcal{M}_{G_{M,m'}} \setminus f(\mathcal{C})$. Then $f(C) \cap f(B) = f(C') \cap f(B)$, because $C \cap B = C' \cap B$ and therefore $(C \cap V_{M,m'}) \cap (B \cap V_{M,m'}) = (C' \cap V_{M,m'}) \cap (B \cap V_{M,m'})$. Further, $|f(\mathcal{C})| > 1$, since $|\mathcal{C}| > 1$ and for $C, C' \in \mathcal{C} \subseteq \mathcal{M}_{G[W^*]}$ with $C \neq C'$ we have $C \cap W \neq C' \cap W$ according to Property A. Consequently, $(C \cap V_{M,m'}) \cap W \neq (C' \cap V_{M,m'}) \cap W$ and $f(C) \neq f(C')$ for max cliques $f(C), f(C') \in f(\mathcal{C})$. We obtain that there exists a subset $f(\mathcal{C}')$ of $\mathcal{M}_{G_{M,m'}}$ with $f(\mathcal{C}) \subseteq f(\mathcal{C}')$ such that $f(\mathcal{C}') \in \mathfrak{P}_{G_{M,m'}}$. As there exists no max clique $f(B) \in \mathcal{M}_{G_{M,m'}} \setminus f(\mathcal{C}')$ with $f(B) \cap W \neq \emptyset$, it holds that $W \subseteq S_{f(\mathcal{C}')}$ and we have shown that $V_{M,n}$ is a subset of the non-singleton equivalence class $S_{f(\mathcal{C}')}$ of $\sim_{G_{M,m'}}$.

For the other direction, let $(M, n) \in P$, we need to show that $V_{M,n}$ is a connected component of a graph induced by a decomposition module. We prove this by induction on n. Clearly, this holds for $n = |V(G)|$, so let $n < |V(G)|$. Let p be minimal such that $p > n$ and $(M, p) \in P$. Since $(M, |V|) \in P$ such a number exists. By inductive assumption we know that $V_{M,p}$ is a connected component of a graph induced by a decomposition module. Thus, $V_{M,p}$ is a module occurring in $V(T(G))$, the nodes of the twinless modular decomposition tree of G.

Since $(M, n) \in P$, (M, n) satisfies Property 2. Thus, $V_{M,n}$ is a subset of a non-singleton equivalence class of $\sim_{G_{M,p}}$, or there exists an apex of $G_{M,p}$ in $V_{M,p} \setminus V_{M,n}$.

Let $V_{M,n}$ be a subset of a non-singleton equivalence class W of $\sim_{G_{M,p}}$. As $G[V_{M,p}]$ is connected, the equivalence class W is a decomposition module. Let D be the connected component of $G[W]$ that contains $V_{M,n}$. If $V_{M,n} = D$, then $V_{M,n}$ is a connected component of the graph $G[W]$ induced by decomposition module W, and we are done. If $V_{M,n}$ is a proper subset of D, we obtain a contradiction to the choice of p, since we have already shown that for the connected component D of $G[W]$ there must exist an $m \in [|V|]$ such that $(M, m) \in P$ and $V_{M,m} = D$, and $n < m < p$.

Now let there be a vertex $a \in V_{M,p} \setminus V_{M,n}$ that is an apex of $G_{M,p}$. Let A be the set of apices of $G_{M,p}$. According to Claim 3 we have $V_{M,n} \subseteq V_{M,p} \setminus A$. Further, $|V_{M,p} \setminus A| = 1$ implies that $v \in V_{M,p} \setminus A$ is also an apex. Consequently, $|V_{M,p} \setminus A| > 1$. Therefore, we have either shown that $V_{M,n}$ is a connected component of the non-singleton class $V_{M,p} \setminus A$ of $\sim_{G_{M,p}}$ or obtain a contradiction to the choice of p. $\square$

We are now ready to define the colored twinless modular decomposition tree. An illustration of the tree can be found in Figure 11.2.

Formally, the *colored twinless modular decomposition tree* of G is defined as $\mathcal{T} = \mathcal{T}_G = (V_\mathcal{T}, E_\mathcal{T}, f_\mathcal{T})$, where $V_\mathcal{T}$ is the set of nodes, $E_\mathcal{T}$ is the set of edges and $f_\mathcal{T} \colon V_\mathcal{T} \to C_\mathcal{T}$ is the coloring of $\mathcal{T}$. Thus, $C_\mathcal{T}$ is the set of colors. In the following we define the set $V_\mathcal{T}$ and $E_\mathcal{T}$ of nodes and edges, respectively, and the coloring $f_\mathcal{T}$ with the set $C_\mathcal{T}$ of colors. The set $V_\mathcal{T}$ is the union of the following sets:

- the set $\mathcal{V}$ of *component nodes* $v_{V_{M,n}}$, one for each set $V_{M,n}$ with $(M,n) \in P$,
- the set $\mathcal{A}$ of *arrangement nodes* $a_{Q,V_{M,n}}$, one for each set $V_{M,n}$ with $(M,n) \in P$ and each non-empty subset $Q \subseteq Q(L_{G_{M,n}})$, where $Q(L_{G_{M,n}})$ is a set of strict linear orders on the max cliques of $L_{G_{M,n}}$. We let
  - $Q(L_{G_{M,n}}) := \{\prec_{L_{G_{M,n}}}\}$ where $\prec_{L_{G_{M,n}}}$ is the distinguished strict linear order from Corollary 344, if $|\mathcal{M}_{L_{G_{M,n}}}| = 1$ or $L_{G_{M,n}}$ is not symmetric, and
  - $Q(L_{G_{M,n}}) := \{\prec_{N_1}, \prec_{N_2}\}$ where N_1, N_2 are the two ends of $L_{G_{M,n}}$, otherwise.
- the set $\mathcal{S}$ of *module nodes* $s_{W,V_{M,n}}$, one for each set $V_{M,n}$ with $(M,n) \in P$ and each non-singleton equivalence class W of $\sim_{G_{M,n}}$, and
- $\{s_V\}$, where s_V is a special node acting as the root of $\mathcal{T}$.

We color the nodes in $\mathcal{V}$ by assigning to each $v_{V_{M,n}} \in \mathcal{V}$ the ordered graph $K(L_{G_{M,n}})$. The nodes in $\mathcal{A}$ remain uncolored, and may therefore be exchanged by an automorphism of $\mathcal{T}$ whenever their subtrees are isomorphic. Each $s_{W,V_{M,n}} \in \mathcal{S}$ is colored with the size of W and the set $c(s_{W,V_{M,n}})$ of integers corresponding to the positions that the max clique A of $L_{G_{M,n}}$ which contains W takes in the orders $Q(L_{G_{M,n}})$ of $L_{G_{M,n}}$.

The edge relation $E_\mathcal{T}$ of $\mathcal{T}$ is now defined with all edges directed away from the root s_V.

- The root s_V is connected to all $v_{V_{M,n}} \in \mathcal{V}$ with $n = |V|$.
- Each $v_{V_{M,n}} \in \mathcal{V}$ is connected to all nodes in $\mathcal{A}$ of the form $a_{Q,V_{M,n}}$. Therefore, $v_{V_{M,n}}$ is connected to one or three nodes.
- Each $a_{Q,V_{M,n}} \in \mathcal{A}$ is connected to all those $s_{W,V_{M,n}} \in \mathcal{S}$ so that Q is the set of orders of $L_{G_{M,n}}$ under which the max clique A of $L_{G_{M,n}}$ which contains $W \in V(L_{G_{M,n}})$ attains its minimal position.
- Every $s_{W,V_{M,n}} \in \mathcal{S}$ is connected to those nodes $v_{V_{M',n'}} \in \mathcal{V}$ for which $V_{M',n'}$ is a connected component of the graph induced by module W, that is, for each max clique B of G with $B \cap W \neq \emptyset$ the node $s_{W,V_{M,n}} \in \mathcal{S}$ is connected to $v_{V_{B,n'}} \in \mathcal{V}$ with $n' = \max\{m < n \mid (V_{B,m}) \in P\}$.

The point of the arrangement nodes $\mathcal{A}$ is to ensure that the order of submodules is properly accounted for. If our tree did not have such a safeguard, exchanging modules in symmetric positions might give rise to a non-isomorphic graph, but it would not change the tree, so $\mathcal{T}$ would be useless for the task of distinguishing between these two graphs.

Lemma 347 below shows that our colored twinless modular decomposition trees are a

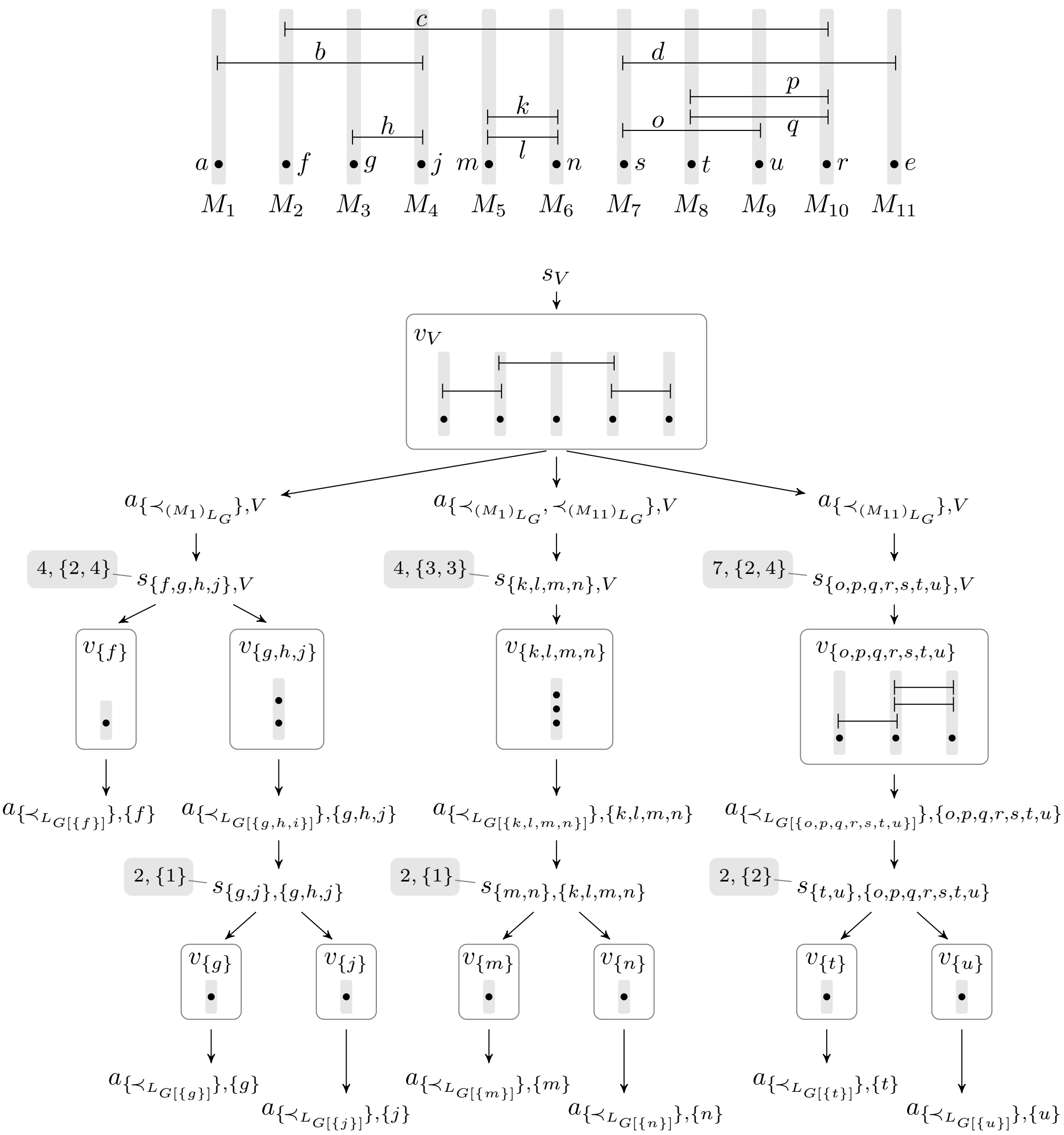


Figure 11.2.: An interval graph and its colored twinless modular decomposition tree. Each component vertex v_U is represented together with the interval graph $L_{G[U]}$ whose canon is the color of v_U. The colors of module vertices are indicated in the gray fields next to them.

complete invariant of interval graphs, so colored twinless modular decomposition trees can be used to tell whether two interval graphs are isomorphic.

Lemma 347 ([47],[50][5])**.** *Let G and H be interval graphs. If their colored twinless modular decomposition trees are isomorphic, then so are G and H.* □

We will later need STC+C-definability of this colored tree. We define the colored twinless modular decomposition tree $\mathcal{T}_G$ as an LO-colored directed tree (defined in Section 2.3.4). More precisely, we understand $\mathcal{T}_G$ as a tuple $(V \dot\cup [n], E, \leq_{[n]}, L)$ where (V, E) is a directed tree, $\leq_{[n]}$ is the natural order on $[n]$ with $n \in \mathbb{N}$, and $L \subseteq V \times [n]^2$ is a relation that assigns to each vertex $b \in V$ a color $L_b := \{(m, m') \mid (b, m, m') \in L\}$. It is easy to bring the colored twinless modular decomposition tree into this form. For example, if b is a component node, say $v_{V_{M,n}}$, then L_b is the representation of the canon of $L_{V_{M,n}}$ (see Section 4.2), that is, L_b consists of all pairs (m, m') where $\{m, m'\}$ is an edge of $K(L_{V_{M,n}})$ and the pair (n_L, n_L) where n_L is the number of vertices of $K(L_{V_{M,n}})$. Furthermore, if b is a module node, say $s_{W,V_{M,n}}$, then L_b consists of the pair $(0, m)$ where m is the size of the equivalence class W, and of all pairs $(1, m')$ where $m' \in c(b)$.

In order to define the colored twinless modular decomposition tree $\mathcal{T}_G$ we show that there are STC+C-formulas $\theta_V(\bar{u})$, $\theta_\approx(\bar{u}, \bar{u}')$, $\theta_E(\bar{u}, \bar{u}')$ and $\theta_L(\bar{u}, r, r')$, where $\bar{u}, \bar{u}'$ are compatible tuples and r, r' are number variables, such that for all interval graphs G,

- $\theta_\approx[G; \bar{u}, \bar{u}']$ generates an equivalence relation $\approx$,
- $V_\mathcal{T} := \theta_V[G; \bar{u}]/_\approx$ is the set of nodes of $\mathcal{T}_G$,
- $E_\mathcal{T} := \{(\bar{a}/_\approx, \bar{b}/_\approx) \in V_\mathcal{T}^2 \mid (\bar{a}, \bar{b}) \in \theta_E[G; \bar{u}, \bar{u}']\}$ is the edge relation of $\mathcal{T}_G$, and
- $L_\mathcal{T} := \{(\bar{a}/_\approx, n, n') \in V_\mathcal{T} \times N(G)^2 \mid (\bar{a}, r, r') \in \theta_L[G; \bar{u}, r, r']\}$ is the color relation of $\mathcal{T}_G$.

We let $\bar{u}$ be the tuple $(p, \bar{u}_{V_{M,n}}, \bar{u}_Q, \bar{u}_W)$ where

$$\begin{aligned} \bar{u}_{V_{M,n}} &:= (x_M, y_M, p_n) \\ \bar{u}_Q &:= (x_{N_1}, y_{N_1}, x_{N_2}, y_{N_2}) \\ \bar{u}_W &:= (x_A, y_A). \end{aligned}$$

The tuple $\bar{u}'$ is defined analogously.

Remember that the sets $V_{M,n}$, and therefore the induced subgraphs $G_{M,n}$, are STC+C definable, and that STC+C is closed under logical reductions.

First, let us sketch how formula θ_V is defined. Variable p identifies the type of the node. For all α, we let $\theta_V[G, \alpha; p]$ be a subset of $\{0, 1, 2, 3\}$. Type 0 corresponds to the root node s_V, type 1 to component nodes $v_{V_{M,n}} \in \mathcal{V}$, type 2 to arrangement nodes $a_{Q,V_{M,n}} \in \mathcal{A}$ and type 3 to module nodes $s_{W,V_{M,n}} \in \mathcal{S}$.

[5] The graphs $L_{G_{M,n}}$ resemble the concept of *overlap components* used in [47] for the definition of a similar kind of modular decomposition tree. Overlap components are connected components of the subgraph of G in which only those edges are present for which the closed neighborhood of neither endpoint is strictly contained in the closed neighborhood of the other (intuitively, their intervals *overlap* or are equal). It can be checked that overlap components and graphs $L_{G_{M,n}}$ only differ in the way they treat vertices that are contained in just one max clique: overlap components treat them as further modules (which they trivially are), the $L_{G_{M,n}}$ graphs directly put them into their unambiguous places. In [47] the authors show Lemma 347 for this similar kind of modular decomposition tree. A detailed proof of Lemma 347 can be found in [50].

- For $\alpha(p) \in \{1,2,3\}$, we let $\theta_V[G,\alpha;x_M,y_M,p_n]$ be the set of triples (v,w,n) where v,w span a max clique M of G and $(M,n) \in P$.
- For $\alpha(p) = 2$ and $\alpha(x_M,y_M,p_n) = (v,w,n)$ where v,w span max clique M of G, we let $\theta_V[G,\alpha;x_{N_1},y_{N_1},x_{N_2},y_{N_2}]$ be the set of tuples $(v_{N_1},w_{N_1},v_{N_2},w_{N_2})$ where for all $i \in [2]$,
  - v_{N_i},w_{N_i} span a max clique M_i of G such that $N_i := (M_i)_{L_{G_{M,n}}}$ is an end of $L_{G_{M,n}}$ if $L_{G_{M,n}}$ is symmetric, and
  - v_{N_i},w_{N_i} span a max clique M' of G such that $N' := M'_{L_{G_{M,n}}}$ is the end of $L_{G_{M,n}}$ for which $\prec_{N'}$ is the distinguished order on the max cliques of $L_{G_{M,n}}$ (cf. Corollary 344) otherwise.
- For $\alpha(p) = 3$ and $\alpha(x_M,y_M,p_n) = (v,w,n)$ where v,w span max clique M of G, we let $\theta_V[G,\alpha;x_A,y_A]$ be the set of pairs (v_A,w_A) where v_A,w_A span a max clique A of G and there exists a non-singleton equivalence class W of $\sim_{G_{M,n}}$ such that $A \cap W \neq \emptyset$.

We define the equivalence relation $\approx$ such that $\theta_V[G,\alpha;\bar{u}]/_{\approx}$ is the set of nodes of $\mathcal{T}_G$. We let tuples $\bar{a},\bar{a}' \in \mathcal{T}_G^{\bar{u}}$ be equivalent if they have the same type, and satisfy all of the following properties for the respective type.

- Type 1, 2 or 3: Vertices v,w and v',w' span the same max clique and $n = n'$.
- Type 2: Vertices v_{N_i},w_{N_i} and v'_{N_i},w'_{N_i} span M_i and M'_i such that $N_i = N'_i$ for $N_i := (M_i)_{L_{G_{M,n}}}$ and $N'_i := (M_i)'_{L_{G_{M,n}}}$ for all $i \in [2]$, and $\{N_1,N_2\} = \{N'_1,N'_2\}$. Note that this way we obtain 3 equivalence classes if the graph $L_{G_{M,n}}$ is symmetric.
- Type 3: Vertices v_A,w_A and v'_A,w'_A span max clique A and A', and $A_{L_{G_{M,n}}} = A'_{L_{G_{M,n}}}$. Thus, there exists a non-singleton equivalence class W of equivalence relation $\sim_{G_{M,n}}$ such that $A \cap W \neq \emptyset$ and $A' \cap W \neq \emptyset$.

Now let us consider the edge relation. We let formula $\theta_E^{i,i'}(\bar{u},\bar{u}')$ define the edges connecting vertices of type i with vertices of type i'. We let

$$\begin{aligned}
\theta_E^{1,2}(\bar{u},\bar{u}') &:= \bar{u}_{V_{M,n}} = \bar{u}'_{V_{M,n}},\\
\theta_E^{2,3}(\bar{u},\bar{u}') &:= \bar{u}_{V_{M,n}} = \bar{u}'_{V_{M,n}} \wedge \rho(x_{N_1},y_{N_1},x_A,y_A) \wedge \rho(x_{N_2},y_{N_2},x_A,y_A), \quad \text{and}\\
\theta_E^{2,3}(\bar{u},\bar{u}') &:= (x_A,y_A) = (x'_M,y'_M) \wedge \text{"}q' \text{ is maximal with } \varphi_P(x'_M,y'_M,q') \wedge q' < q\text{“.}
\end{aligned}$$

where formula ρ is defined in Lemma 341 and formula φ_P in Observation 345. Given the above formulas, it should be clear how to define θ_E.

It remains to define θ_L. We can define the coloring of all vertices of type 1 in STC+C according to Lemma 343. The coloring of all vertices of type 3 is STC+C-definable by Lemma 342 and Corollary 339, which says that the equivalence relation $\sim_{G_{M,n}}$ and therefore the sizes of the respective equivalence classes of $\sim_{G_{M,n}}$ are STC+C-definable.

In Section 8.4 we described how to define a total preorder $\preceq'$ on the nodes of an LO-colored directed tree, that is, a strict linear order on the isomorphism classes of the colored subtrees identified by its root nodes. We use this total preorder $\preceq'$ in the next section for canonization.

11.6. Canonization

This section deals with the canonization of interval graphs, that is, how to construct an $\mathsf{LREC}_=$-formula $\kappa'(p,q)$ such that $G \cong ([|V(G)|], \kappa'[G;p,q])$ for each interval graph G. As a result we obtain the following:

Theorem 348. $\mathsf{LREC}_=$ *captures* LOGSPACE *on the class of all interval graphs.*

We use the colored twinless modular decomposition tree and the total preorder $\preceq'$ (Section 8.4) on its nodes for canonization. We apply l-recursion on the colored twinless modular decomposition tree, and as we have done for canonizing trees we build the canon from the leaves to the root of the tree. Recursively, we construct the canon by first building the disjoint union of the canons of the components of submodules, then use the arrangement nodes to insert all submodules at the correct side and build the canon of the corresponding component of a module.

In the following we explain the canonization procedure in more detail. The following lemma shows that it suffices to give an $\mathsf{LREC}_=$-formula $\kappa(p,q)$ such that for every interval graph G we have $G \cong ([|V(G)|], \kappa[\mathcal{T}_G;p,q])$. It follows from the Transduction Lemma (Lemma 326) and the fact that the colored twinless modular decomposition tree of an interval graph is STC+C-definable.

Lemma 349. *If there exists an* $\mathsf{LREC}_=$*-formula* $\kappa(p,q)$ *such that for all interval graphs* G *we have* $G \cong ([|V(G)|], \kappa[\mathcal{T}_G;p,q])$ *and* $\kappa[\mathcal{T}_G;p,q] \subseteq [|V(G)|]^2$*, then there also exists an* $\mathsf{LREC}_=$*-formula* $\kappa'(p',q')$ *such that for all interval graphs* G*,* $G \cong ([|V(G)|], \kappa'[G;p',q'])$*.*

Proof. As showed at the end of Section 11.5, the nodes, edges and colors of the colored twinless modular decomposition tree of an interval graph G are definable by STC+C-formulas $\theta_V(\bar{u})$, $\theta_\approx(\bar{u},\bar{v})$, $\theta_E(\bar{u},\bar{v})$ and $\theta_L(\bar{u},r,r')$, where $\bar{u},\bar{v}$ are compatible tuples and r,r' are number variables. We can use these formulas to define the STC+C-counting transduction $\Theta^\# = (\theta_V(\bar{u}), \theta_\approx(\bar{u},\bar{v}), \theta_E(\bar{u},\bar{v}), \theta_L(\bar{u},r,r'))$. The definition and more information on counting transductions can be found in Section 2.5.4. Note that $\Theta^\#[G] \cong \mathcal{T}_G$. According to Proposition 14, there exists an STC+C-transduction Θ with $\Theta[G] \cong \Theta^\#[G]$. Thus, there exists an STC+C-transduction Θ with $\Theta[G] \cong \mathcal{T}_G$.

We now apply the Transduction Lemma (Lemma 326) with the transduction Θ to obtain an $\mathsf{LREC}_=$-formula $\kappa^{-\Theta}(\bar{p}',\bar{q}')$ such that for all $\bar{m},\bar{n} \in N(G)^{|\bar{u}|}$, $G \models \kappa^{-\Theta}[\bar{m},\bar{n}]$ iff $\langle\bar{m}\rangle_G, \langle\bar{n}\rangle_G \in N(\Theta[G])$ and $\Theta[G] \models \kappa[\langle\bar{m}\rangle_G, \langle\bar{n}\rangle_G]$. As $\kappa[\mathcal{T}_G;p,q] \subseteq [|V(G)|]^2$, the condition $\langle\bar{m}\rangle_G, \langle\bar{n}\rangle_G \in N(\Theta[G])$ can be replaced by $\langle\bar{m}\rangle_G, \langle\bar{n}\rangle_G \in N(G)$. Hence, the tuples $\bar{p}',\bar{q}'$ of number variables in $\kappa^{-\Theta}$ can be identified with single number variables p',q', which yields the desired formula $\kappa'(p',q')$. □

In general, the canonization procedure is similar to the one of directed trees. To apply l-recursion we use a graph $\mathtt{G} = (\mathtt{V},\mathtt{E})$ with labels $\mathtt{C}(v) \subseteq \mathbb{N}$ for all $v \in \mathtt{V}$. We let $\mathtt{V} := V(\mathcal{T}_G) \times N(\mathcal{T}_G)^2$ be the vertices of $\mathtt{G}$ and for all component nodes $v_{V_{M,n}} \in \mathcal{V}$, $(v_{V_{M,n}},p,q) \in X$ stands for "$(p,q) \in E_{v_{V_{M,n}}}$?", where $E_{v_{V_{M,n}}}$ is the edge relation of an isomorphic copy $([|V_{M,n}|], E_{v_{V_{M,n}}})$ of $G_{M,n}$.

In the following we explain the edge relation $\mathtt{E}$ and labels $\mathtt{C}$ of graph $\mathtt{G}$.

Edges introduced by module nodes.
In $\mathcal{T}_G$, each node $s_{W,V_{M,n}} \in \mathcal{S}$ is connected to those $v_{V_{M',n'}} \in \mathcal{V}$ for which $V_{M',n'}$ is a connected component of the graph induced by module W.[6] We can use the available total preorder $\prec'$ on the children of $s_{W,V_{M,n}}$ to construct the edge relation of the canon of the disjoint union of the children's canons from the edge relation of the canons of the children. For a node $s \in \mathcal{S}$ and a child $v := v_{V_{M',n'}} \in \mathcal{V}$ of s, let D_v be the set of all children v' of s with $v' \prec' v$, and e_v be the number of children v' of s defining modules $V_{M'',n''}$ such that $V_{M'',n''}$ and $V_{M',n'}$ induce subgraphs that are isomorphic (i.e., $v' \preceq' v$ and $v \preceq' v'$). For all $p, q \in N(\mathcal{T}_G)^2$ and all $i \in [0, e_v - 1]$, we let $\bar{s} := (s, p_{v,i} + p, p_{v,i} + q)$ have an edge to (v, p, q) where $p_{v,i} := \sum_{v_{V_{M',n'}} \in D_v} |V_{M',n'}| + i \cdot |V_{M,n}|$ and define $\mathtt{C}(\bar{s}) = \{e_v\}$. Notice that here we can have an in-degree greater than 1.

Edges introduced by arrangement nodes.
Let us consider a node $a_{Q,V_{M,n}} \in \mathcal{A}$. Its children in $\mathcal{T}_G$ are nodes $s_{W,V_{M,n}}$ for submodules W of the module $V_{M,n}$, and we need to integrate the canons of them into the canon $K(L_{G_{M,n}})$ of $L_{G_{M,n}}$. In order to do this, we need

- the edge relation of $K(L_{G_{M,n}})$, which is encoded in the color of the parent node of $a_{Q,V_{M,n}}$,
- the size of the vertex set of $K(L_{G_{M,n}})$, which is also encoded in the color of the parent node of $a_{Q,V_{M,n}}$,
- the size of each submodule W corresponding to a child $s_{W,V_{M,n}}$ of $a_{Q,V_{M,n}}$, which is encoded in the color of the respective child, and
- for each submodule W corresponding to a child $s_{W,V_{M,n}}$, the set $c(s_{W,V_{M,n}})$ of possible positions of the max clique of $L_{G_{M,n}}$ that contains the submodule W regarding the strict linear orders in $Q(L_{G_{M,n}})$. This set of positions is also encoded in the color of the respective child $s_{W,V_{M,n}}$.

We assume that the canon $K(L_{G_{M,n}})$ is assigned to the first part $[1, |V(G_{V_{M,n}})|]$ of the number sort. Notice that on the number sort we have a distinguished order $\prec_{K(L_{G_{M,n}})}$ on the max cliques of $K(L_{G_{M,n}})$ (see Corollary 344).

For the children nodes $s_{W,V_{M,n}}$ of $a_{Q,V_{M,n}}$ we already know that $(s_{W,V_{M,n}}, p, q) \in X$ corresponds to "$(p,q) \in E_W$?", where E_W is the edge relation of an isomorphic copy $([|W|], E_W)$ of $G[W]$. Now we create the necessary edges in $\mathtt{G}$ such that $(a_{Q,V_{M,n}}, p, q) \in X$ corresponds to "$(p,q) \in E'_{V_{M,n}}$?" where $E'_{V_{M,n}}$ contains the following edges: For each child $s_{W,V_{M,n}}$ of $a_{Q,V_{M,n}}$, we take the edges of the canon of $G[W]$ and shift them into another range of the number sort, that is, we add to the number vertices of every edge of the canon of $G[W]$ the size of the vertex set of $K(L_{G_{M,n}})$ and the sizes of all modules W' corresponding to certain other children $s_{W',V_{M,n}}$ of a or of siblings of a. We will specify the children later. Further, $E'_{V_{M,n}}$ contains for each child $s_{W,V_{M,n}}$ of $a_{Q,V_{M,n}}$ edges between certain vertices of the canon $K(L_{G_{M,n}})$ and the vertices of the shifted canon of $G[W]$. More precisely, we use the total preorder $\preceq'$ on the nodes of $\mathcal{T}_G$ and the set $c(s_{W,V_{M,n}})$ of positions to determine a subset $\mathrm{pos}(s_{W,V_{M,n}}) \subseteq c(s_{W,V_{M,n}})$ of positions that indicate where to integrate the canon of $G[W]$ into $K(L_{G_{M,n}})$ in order to obtain a canon of $G_{M,n}$. For each position $r \in \mathrm{pos}(s_{W,V_{M,n}})$ we determine the max clique of $K(L_{G_{M,n}})$ at position r regarding the order $\prec_{K(L_{G_{M,n}})}$ and find the minimal vertex m in this max

[6] Here, we can include node s_V, where we let W correspond to V.

clique that can represent module W (a vertex that is in no further max clique). We let $E'_{V_{M,n}}$ contain the edges between each neighbor of m in $K(L_{G_{M,n}})$, and every vertex of the shifted canon of $G[W]$. Notice that W is a module, and therefore, these are the edges connecting the shifted canon of $G[W]$ with the rest of the graph.

We obtain the edge relation of the canon of $G_{M,n}$ in the subsequent step, when considering the edges introduced by component nodes. Then, we add the edges of the canon $K(L_{G_{M,n}})$ to the set $E'_{V_{M,n}}$ of edges, and afterwards remove all the vertices m representing non-singleton equivalence classes W and close the gaps such that the vertex set of the canon is $[|V_{M,n}|]$.

Now let us define the edges of $\mathtt{G}$ introduced by arrangement nodes.

First, we define the set of positions $\mathrm{pos}(s)$ for every module node $s := s_{W,V_{M,n}} \in \mathcal{S}$. If $c(s)$ is a singleton set, then the position is already determined and we let $\mathrm{pos}(s) := c(s)$. In the following, let $c(s)$ be not a singleton set. Let a be the parent of s. Notice that in this case, a has two siblings and the graph $L_{G_{M,n}}$ is symmetric. If there does not exist a sibling a' of a with $|c(s')| = 2$ for a child s' of a', then we let $\mathrm{pos}(s) := \{\max(c(s))\}$. Now, let there be a sibling a' of a with $|c(s')| = 2$ for a child s' of a'. Note that there can exist only one sibling a' with this property.

- If $a \prec' a'$, then we let $\mathrm{pos}(s) := \{\min(c(s))\}$.
- If $a' \prec' a$, then we let $\mathrm{pos}(s) := \{\max(c(s))\}$.
- If neither $a \prec' a'$ nor $a' \prec' a$, then we let $\mathrm{pos}(s) := c(s)$.

If neither $a \prec' a'$ nor $a' \prec' a$, the subtrees of $\mathcal{T}_G$ rooted at a and a' are isomorphic. As a consequence, the non-singleton equivalence classes W of $\sim_{G_{M,n}}$ occur in symmetric positions regarding each strict linear order $\prec \in Q(L_{G_{M,n}})$, and each pair W, W' of non-singleton equivalence classes in symmetric positions induces isomorphic subgraphs.

Let $a := a_{Q,V_{M,n}} \in \mathcal{A}$, and let D_a be the set of all nodes s that are a child of a or a child of a sibling of a. Let $\mathrm{Pos}(a) := \bigcup_{s \in D(a)} \mathrm{pos}(s)$. The set $\mathrm{Pos}(a)$ is the set of all positions r of a max cliques A regarding $\prec_{K(L_{G_{M,n}})}$ where a vertex of A has to be replaced by the canon of $G(W)$ for a non-singleton equivalence class W. For all $r \in \mathrm{Pos}(a)$, let $P(r) := \{r' \in \mathrm{Pos}(a) \mid r' < r\}$ and $\mathrm{size}(r) := |W|$ where $r \in \mathrm{pos}(s_{W,V_{M,n}})$. The position r specifies a max clique A of $K(L_{G_{M,n}})$ regarding $\prec_{K(L_{G_{M,n}})}$. The value $\mathrm{size}(r)$, for $r \in \mathrm{Pos}(a)$, is the size of the module whose induced subgraph has to replace a vertex of A. Note that $\mathrm{size}(r)$ is well-defined: Only if a and a sibling a' of a have each a child s such that $|c(s)| = 2$, and neither $a \prec' a'$ nor $a' \prec' a$, there can be (exactly) two nodes $s_{W,V_{M,n}}$ with $r \in \mathrm{pos}(s_{W,V_{M,n}})$. One is a child of a and the other one a child of a'. As neither $a \prec' a'$ nor $a' \prec' a$, the colored subtrees rooted at a and a' are isomorphic. The size of W is encoded in the color of each node $s_{W,V_{M,n}}$. Thus, the two nodes $s_{W,V_{M,n}}$ with $r \in \mathrm{pos}(s_{W,V_{M,n}})$ have to correspond to modules W of equal size. Hence, $\mathrm{size}(r)$ is well-defined.

Now let s be a child of a. Let $d_r := |V(L_{G_{M,n}})| + \sum_{r' \in P(r)} \mathrm{size}(r')$. For all $r \in \mathrm{pos}(s)$ and $p, q \in N(\mathcal{T}_G)^2$ where $d_r + p, d_r + q \in N(\mathcal{T}_G)^2$, we let $\bar{a} := (a, d_r + p, d_r + q)$ have an edge to (s, p, q) and define $\mathtt{C}(\bar{a}) = \{|\mathrm{pos}(s)|\}$. Further, let $|W|$ be the size of the module corresponding to s. Let m_r be the minimal vertex in the max clique of $K(L_{G_{M,n}})$ at the position $r \in \mathrm{pos}(s)$ that is in no further max clique of $K(L_{G_{M,n}})$, and let $N(m_r)$ be the set of neighbors of m_r in $K(L_{G_{M,n}})$. For all $r \in \mathrm{pos}(s)$, $p \in N(m_r)$ and $q \in [d_r + 1, d_r + |W|]$, we define $\mathtt{C}(a,p,q) = \mathtt{C}(a,q,p) = N(T_G)$. For all $p,q \in N(\mathcal{T}_G)$ where we have not yet assigned a color to vertex (a,p,q) of $\mathtt{G}$ we let $\mathtt{C}(a,p,q) = \emptyset$. Notice that $\mathtt{C}(a,p,q) = \emptyset$ and there are no outgoing edges of (a,p,q) for all $p,q \in N(\mathcal{T}_G)$ if a does not have any children.

Note, that we only obtain in-degrees larger than 1, that is, in-degrees of 2, if the graph $L_{G_{M,n}}$ is symmetric and we insert pairs of isomorphic graphs $G[W]$ for non-singleton equivalence classes W of $\sim_{G_{M,n}}$ in symmetric positions at both sides.

Edges introduced by component nodes.
Let $v = v_{V_{M,n}} \in \mathcal{V}$. We now define the edge relation of $\mathtt{G}$ such that $(v_{V_{M,n}}, p, q) \in X$ stands for "$(p,q) \in E_{v_{V_{M,n}}}$?", where $E_{v_{V_{M,n}}}$ is the edge relation of an isomorphic copy of $G_{M,n}$ on the number sort. Thus, we add the edges of the canon $K(L_{G_{M,n}})$ to the set of edges $E'_{V_{M,n}}$, which is the edge relation defined through X in the previous step, remove all vertices m representing non-singleton equivalence classes W and close the gaps such that the vertex set of the isomorphic copy is $[|V_{M,n}|]$.

Let $R := \{m_r \mid r \in Pos(a)\}$ be the subset of vertices of $K(L_{V_{M,n}})$ that represent non-singleton equivalence classes and have to be removed. Let n_L be the size of the vertex set of $K(L_{V_{M,n}})$, and let $\leq' := \leq_{N(\mathcal{T}_G)\setminus R}$ be the natural linear order on the the numbers in $N(\mathcal{T}_G) \setminus R$. For all p in $[|N(\mathcal{T}_G)| - |R|]$, we let $f(p) = q$ if q is at position p in $\leq'$. For all $p,q \in [1, n_L - |R|]$, we let $\mathtt{C}(v,p,q) = N(\mathcal{T}_G)$ if $\{f(p), f(q)\}$ is an edge in $K(L_{V_{M,n}})$ and $\mathtt{C}(v,p,q) = \emptyset$ otherwise. Also we let $\mathtt{C}(v,p,q) = \emptyset$ whenever p or q is contained in $[|N(\mathcal{T}_G)| - |R| + 1, |N(\mathcal{T}_G)|]$. We let $\bar{v} := (v,p,q)$ have an edge to $(a, f(p), f(q))$ and define $\mathtt{C}(\bar{v}) = \{1\}$ for all pairs $(p,q),(q,p) \in [|N(\mathcal{T}_G)| - |R|] \times [n_L - |R| + 1, |N(\mathcal{T}_G)| - |R|]$.

Finishing the construction.
In order to actually perform l-recursion we need sufficient "resources". Taking a look at the in-degrees, we notice that they are only larger than one when we treat isomorphic connected components while building the disjoint union, or when the graph $L_{G_{M,n}}$ is symmetric and we insert the induced subgraphs of non-singleton modules twice in symmetric positions at both sides. Either way, an in-degree of d means that we insert at least d disjoint isomorphic copies into the graph on the number sort. Hence, it suffices to use a binary resource term.

Remark 350. It is possible to show that there is no LREC+TC$[\{E\}]$-sentence φ such that for all connected interval graphs G_1, G_2 we have $G_1 \dot\cup G_2 \models \varphi$ if and only if $G_1 \cong G_2$. The proof is based on similar ideas as the proof of Theorem 314. ⌟

12. Capturing LOGSPACE on Chordal Claw-Free Graphs

We show that the class of chordal claw-free graphs admits $\mathsf{LREC}_=$-definable canonization in this chapter. As a result, $\mathsf{LREC}_=$ captures logarithmic space on the class of chordal claw-free graphs.

A graph is *chordal* if all its cycles of length at least 4 have a chord, which is an edge that connects two non-consecutive vertices of the cycle. Chordal graphs can be characterized in various interesting ways (see [2]). In particular, chordal graphs are the intersection graphs of subtrees of a tree [5, 22, 67]. A tree representation, called a clique tree, can be computed in linear time [38]. In a clique tree each node of the tree corresponds to a max clique of a chordal graph. Clique trees of chordal graphs are properly defined in Section 12.1.

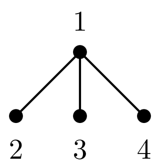


Figure 12.1.: The claw $K_{1,3} = ([4], \{\{1,2\},\{1,3\},\{1,4\}\})$

A *claw-free* graph is a graph that does not have a claw as an induced subgraph. A *claw* is a graph that is isomorphic to the complete bipartite graph $K_{1,3}$. Figure 12.1 shows a picture of a claw. There exists a variety of types of claw-free graphs. For instance, the graph of the icosahedron, complements of triangle-free graphs, the Schläfli graph and proper circular arc graphs are claw-free graphs [9]. Further, claw-free graphs are generalizations of line graphs. The *line graph* G' of a graph $G = (V, E)$ is the graph with vertex set E where there is an edge between two vertices of G' whenever the corresponding edges are adjacent in G. For a given graph G the line graph G' is illustrated in Figure 12.2. It is not hard to see that line graphs are claw-free.

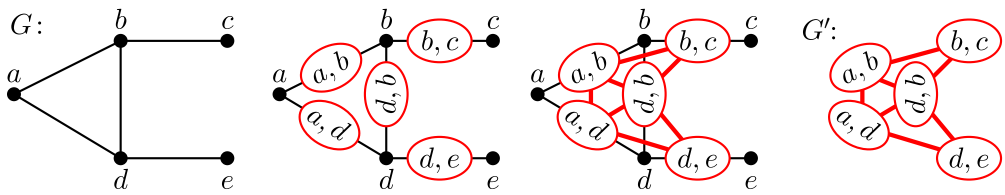


Figure 12.2.: The line graph G' of a graph G

In 2010 Grohe showed that FP+C captures polynomial time on chordal line graphs, that is he showed that there exists an FP+C-canonization of chordal line graphs [29]. At the same time he conjectured that his result can be generalized to chordal claw-free graphs. Our result confirms his conjecture and improves it by showing that chordal claw-free graphs admit $\mathsf{LREC}_=$-definable canonization. Hence, $\mathsf{LREC}_=$ captures LOGSPACE on the class of chordal claw-free graphs.

In order to prove that chordal claw-free graphs admit $\mathsf{LREC}_=$-definable canonization, the strategy is the following: According to Proposition 19, it is sufficient to show that there exists an $\mathsf{LREC}_=$-definable (parameterized) canonization of connected chordal claw-free graphs. To define such a canonization we use the clique tree of these graphs. We show that for connected chordal claw-free graphs the clique tree is unique. Further, the clique tree has a special structure, and the max cliques of it intersect in specific ways. We color every max clique with information about its intersection with other max cliques. We know that there is an $\mathsf{LREC}_=$-canonization (for several types) of colored trees. However, before we can canonize the colored tree, we have to define it in our logic. In order to do this, we first show that the clique tree can be defined in first order logic. Then we transform the clique tree into a directed tree, and color each max clique by using an LO-coloring. We obtain what we call the supplemented clique tree, and show that it is definable in STC+C. Next we apply the $\mathsf{LREC}_=$-canonization of LO-colored trees to the supplemented clique tree and obtain the canon of this colored directed tree. Due to the LO-coloring the information about the max cliques is also contained in the coloring of the canon of the supplemented clique tree. This information and the ordering of the nodes of the canon allow us to determine the max cliques of the canon of the graph by reconstructing the clique tree. Having the max cliques of the canon we can easily construct the canon.

12.1. Introduction of Clique Trees

In this section we introduce clique trees of chordal graphs and show some basic properties of clique trees of chordal claw-free graphs.

Clique Trees of Chordal Graphs

Chordal graphs are the intersection graphs of subtrees of a tree [5, 22, 67]. A clique tree of a chordal graph specifies such a tree representation.

Let G be a chordal graph, and let $\mathcal{M}$ be the set of max cliques of G. Further, let $\mathcal{M}_v$ be the set of all max cliques that contain a vertex v of G. A *clique tree* of G is a tree $T = (\mathcal{M}, \mathcal{E})$ whose vertex set is the set $\mathcal{M}$ of all max cliques, where for all $v \in V$ the induced subgraph $T[\mathcal{M}_v]$ is connected. Hence, for each $v \in V$ the induced subgraph $T[\mathcal{M}_v]$ is a subtree of T. The subtrees $T[\mathcal{M}_v]$ of T for all $v \in V$ are a tree representation of G, which shows that G is an intersection graph of subtrees of a tree:

$$\begin{array}{rl} & \text{There is an edge, between vertices } v \text{ and } w \text{ is } G. \\ \iff & \text{There is a max clique } M \text{ that contains vertices } v \text{ and } w. \\ \iff & \text{There is a max clique } M \text{ that is a node of } T[\mathcal{M}_v] \text{ and } T[\mathcal{M}_w]. \\ \iff & \text{The subtrees } T[\mathcal{M}_v] \text{ and } T[\mathcal{M}_w] \text{ of } T \text{ intersect.} \end{array}$$

Clique trees were introduced independently by Buneman [5], Gavril [22] and Walter [67]. An example of a clique tree of a chordal graph is shown in Figure 12.3. A detailed introduction of chordal graphs and their clique trees can be found in [2].

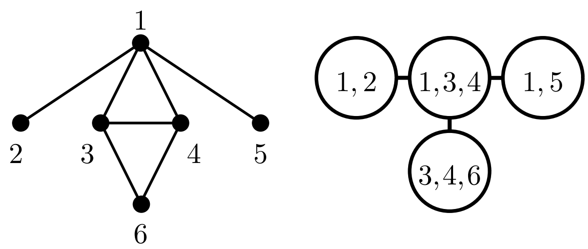


Figure 12.3.: A graph and a clique tree of it

Let $T = (\mathcal{M}, \mathcal{E})$ be a clique tree of a chordal graph G. It is easy to see that clique tree T satisfies the *clique-intersection property*:

> Let $M_1, M_2, M_3 \in \mathcal{M}$ be nodes of the tree T. If M_2 is on the path from M_1 to M_3, then $M_1 \cap M_3 \subseteq M_2$.

Clique Trees of Chordal Claw-free Graphs

In the following we consider chordal claw-free graphs G. For each vertex v, we prove that the set of max cliques $\mathcal{M}_v$ induces a path in each clique tree. We use this property and further ones to show that the clique tree of a connected chordal claw-free graph is unique in the following section.

Lemma 351. *Let T be a clique tree of a chordal claw-free graph $G = (V, E)$. Then for all $v \in V$ the induced subtree $T[\mathcal{M}_v]$ is a path in T.*

Proof. Let us assume that there exists a vertex $v \in V$ such that the graph $T[\mathcal{M}_v]$ is not a path in T. As $T[\mathcal{M}_v]$ is a subtree of T, there exists a max clique $B \in \mathcal{M}_v$ such that B has a degree of at least 3. Let $A_1, A_2, A_3 \in \mathcal{M}_v$ be three distinct neighbors of B in $T[\mathcal{M}_v]$. Since A_i and B are distinct max cliques, we have $A_i \not\subseteq B$. Thus, for all $i \in [3]$ there exists a vertex $a_i \in A_i \setminus B$, and for each $i \in [3]$, we have $A_i \in \mathcal{M}_{a_i}$, $B \notin \mathcal{M}_{a_i}$ and $\mathcal{M}_{a_i}$ is connected in T. As T is a tree, A_1, A_2, and A_3 are all in different connected components of $T[\mathcal{M} \setminus \{B\}]$. Therefore, $\mathcal{M}_{a_i} \cap \mathcal{M}_{a_{i'}} = \emptyset$ for all $i, i' \in [3]$ with $i \neq i'$. Now, we can show that $\{v, a_1, a_2, a_3\}$ induces a claw in G, which contradicts G being claw-free. For all $i \in [3]$, there is an edge between v and a_i, because $v, a_i \in A_i$. To show that vertices a_i and $a_{i'}$ are not adjacent for $i \neq i'$, let us assume the opposite. If there exists an edge between a_i and $a_{i'}$, then there must be a max clique M containing a_i and $a_{i'}$. As a consequence, M is in $\mathcal{M}_{a_i} \cap \mathcal{M}_{a_{i'}}$, which is a contradiction as $\mathcal{M}_{a_i} \cap \mathcal{M}_{a_{i'}}$ must be empty. □

Lemma 352. *Let T be a clique tree of chordal claw-free graph $G = (V, E)$. Further, let $v \in V$, and let A_1, A_2, A_3 be distinct max cliques in $\mathcal{M}_v$. Then A_2 lies between A_1 and A_3 on the path $T[\mathcal{M}_v]$ if and only if $A_2 \subseteq A_1 \cup A_3$.*

Proof. First, let us assume $A_2 \subseteq A_1 \cup A_3$ and let us prove that max clique A_2 must be between max cliques A_1 and A_3 on the path $T[\mathcal{M}_v]$. For a contradiction let us, without loss of generality, suppose A_1 is situated between A_2 and A_3. Then the clique intersection property implies that $A_2 \cap A_3 \subseteq A_1$. Further, it follows from $A_2 \subseteq A_1 \cup A_3$ that $A_2 \setminus A_3 \subseteq A_1$. Consequently, we have $A_2 \subseteq A_1$, which is a contradiction to A_1 and A_2 being distinct max cliques.

Now let max clique A_2 lie between max cliques A_1 and A_3 on the path $T[\mathcal{M}_v]$. In order to prove that $A_2 \subseteq A_1 \cup A_3$, let us assume there exists a vertex $a_2 \in A_2 \setminus (A_1 \cup A_3)$. For the following part of the proof an illustration can be found in Figure 12.4. Let $P = B_1, \ldots, B_l$ be the path $T[\mathcal{M}_v]$ (Lemma 351). Without loss of generality, let $A_i = B_{j_i}$ for all $i \in [3]$ where $j_1, j_2, j_3 \in [l]$ with $j_1 < j_2 < j_3$. Further, let $A'_1 := B_{j_1+1}$ and $A'_3 := B_{j_3-1}$, and let $a_1 \in A_1 \setminus A'_1$ and $a_3 \in A_3 \setminus A'_3$. Since A_1 and A'_1, and A_3 and A'_3 are distinct max cliques, such vertices exist. We show that the set $\{v, a_1, a_2, a_3\}$ induces a claw in G, which is a contradiction as G is claw-free. Clearly, for all $i \in [3]$ vertices v and a_i are adjacent. It remains to show that there is no edge between a_i and $a_{i'}$ for all $i, i' \in [3]$. We proceed analog to the previous proof. Let T' be the subgraph of T after removing the edge between A_1 and A'_1, and the edge between A_3 and A'_3. Then T' consists of three connected components, containing the sets $\mathcal{M}_{a_1}$, $\mathcal{M}_{a_2}$ and $\mathcal{M}_{a_3}$, respectively. Again an edge between vertices a_i and $a_{i'}$ for $i \neq i'$ implies that there exists a max clique M containing a_i and $a_{i'}$, which means $M \in \mathcal{M}_{a_i} \cap \mathcal{M}_{a_{i'}}$, a contradiction. □

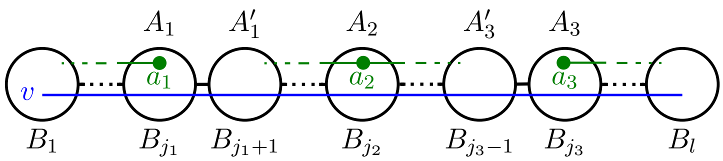
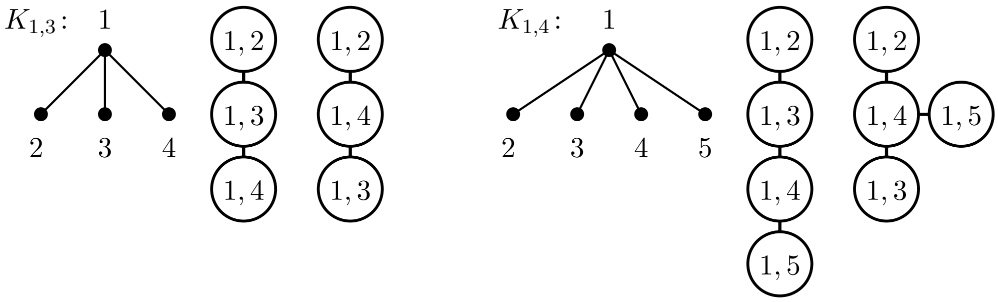


Figure 12.4.: Illustration for the proof of Lemma 352

12.2. Uniqueness of the Clique Tree for Connected Graphs

In the following we show that the clique tree of a connected chordal claw-free graph is unique. Notice, that this is a property that does not hold for unconnected graphs. Given an unconnected chordal (claw-free) graph, each subgraph induced by a connected component has a clique tree. We can connect the clique trees for the connected components in an arbitrary way to obtain a clique tree of the entire graph. Further, connected chordal graphs in general also do not have a unique clique tree. For example, the claw is a connected chordal graph having multiple clique trees (see Figure 12.5a). We can also consider star graphs in general, that is, graphs $K_{1,n}$ for $n > 0$. For $n > 3$ the clique trees of star graph $K_{1,n}$ are not even necessarily isomorphic as shown in Figure 12.5b for the $K_{1,4}$.



(a) The $K_{1,3}$ and two clique trees (b) The $K_{1,4}$ and two non-isomorphic clique trees

Figure 12.5.: Examples for connected chordal graphs where the clique tree is not unique

Lemma 353. *Let $T_1 = (\mathcal{M}, \mathcal{E}_1)$ and $T_2 = (\mathcal{M}, \mathcal{E}_2)$ be clique trees of a chordal claw-free graph $G = (V, E)$. Then for every $v \in V$ we have $T_1[\mathcal{M}_v] = T_2[\mathcal{M}_v]$.*

Proof. According to Lemma 351, $T_1[\mathcal{M}_v]$ and $T_2[\mathcal{M}_v]$ are paths in T_1 and T_2, respectively. Clearly, the paths have the same vertex set $\mathcal{M}_v$. Let us assume there exist distinct max cliques $A, B \in \mathcal{M}_v$ such that, without loss of generality, A, B are adjacent in $T_1[\mathcal{M}_v]$ but not adjacent in $T_2[\mathcal{M}_v]$. As A, B are not adjacent in $T_2[\mathcal{M}_v]$, there exists a max clique $C \in \mathcal{M}_v$ which lies between A and B on the path $T_2[\mathcal{M}_v]$. Thus, $A \cap B \subseteq C$ according to the clique-intersection property. In $T_1[\mathcal{M}_v]$ max cliques A and B are adjacent. Therefore, either A is between B and C, or B lies between A and C. Without loss of generality, let A lie between B and C on the path induced by $\mathcal{M}_v$ in T_1. Then $A \subseteq B \cup C$ by Lemma 352. Thus, we have $A \setminus C \subseteq B$. In combination with $A \cap B \subseteq C$, this yields that $A \setminus C \subseteq C$. Consequently, $A \setminus C$ must be empty, which is a contradiction to A and C being distinct max cliques. □

Lemma 354. *Let T be a clique tree of a connected chordal claw-free graph G. Then*

$$T = \bigcup_{v \in V} T[\mathcal{M}_v].$$

Proof. Clearly, the graphs T and $T' := \bigcup_{v \in V} T[\mathcal{M}_v]$ have the same vertex set, and T' is a subgraph of T. Let us assume there exist max cliques $A, B \in \mathcal{M}$ such that there is an edge between A and B in T but A and B are not adjacent in T'. First of all, we show that $A \cap B = \emptyset$. Suppose there exists a vertex $v \in A \cap B$. Then A and B are vertices on the path induced by $\mathcal{M}_v$. As there is an edge between A and B in T, max cliques A and B must be adjacent in $T[\mathcal{M}_v]$ and therefore in T', a contradiction. Thus, we have $A \cap B = \emptyset$.

Next, we show that this implies that G is not connected. Let $\mathcal{C}_A$ and $\mathcal{C}_B$ be the connected components of T that we obtain after removing the edge between A and B. Further, let $W_A := \bigcup \mathcal{C}_A$ and $W_B := \bigcup \mathcal{C}_B$. We prove that W_A and W_B form a partition of the vertex set V such that no pair of vertices $a \in W_A$, $b \in W_B$ is connected in G. Then G cannot be connected and we have a contradiction. First we show that $\{W_A, W_B\}$ is a partition of V. Clearly, $W_A \cup W_B = V$, and since $A \subseteq W_A$ and $B \subseteq W_B$, W_A and W_B are non-empty. It remains to show that W_A and W_B are disjoint. Thus, let us suppose there exists a vertex $w \in W_A \cap W_B$. Then there exist max cliques $M_A \in \mathcal{C}_A$ and $M_B \in \mathcal{C}_B$, with $w \in M_A$ and $w \in M_B$. The path $T[\mathcal{M}_w]$ must contain the edge between A and B. Thus, $w \in A \cap B$, a contradiction. Hence, $W_A \cap W_B = \emptyset$. Now let us assume there exist vertices $a \in W_A$, $b \in W_B$ which are connected in G. Since W_A and W_B form a partition of V, there must exist vertices $a' \in W_A$, $b' \in W_B$ which are adjacent in G. Thus, there exists a max clique M with $a', b' \in M$. As either $M \in \mathcal{C}_A$ or $M \in \mathcal{C}_B$, we obtain that $a', b' \in W_A$ or $a', b' \in W_B$, a contradiction. □

As a direct consequence of Lemma 353 and Lemma 354 we obtain the following corollary, which is the main result of this section.

Corollary 355. *Let T_1 and T_2 be clique trees of a connected chordal claw-free graph G. Then $T_1 = T_2$.*

12.3. Structure of the Clique Tree

The clique tree plays an important role in the subsequent canonization of connected chordal claw-free graphs. Thus, we analyze the structure of clique trees of connected chordal claw-free graphs in this section.

In the previous section we showed that a connected chordal claw-free graph G has a unique clique tree. In the following let G be a connected chordal claw-free graph G and let T be its clique tree.

Lemma 356. *Let $v \in V$. Then for all $w \in V \setminus \{v\}$ the induced subgraph $T[\mathcal{M}_v \setminus \mathcal{M}_w]$ is connected.*[1]

Proof. Let $P = A_1, \dots, A_l$ be the path $T[\mathcal{M}_v]$, and let us assume $T[\mathcal{M}_v \setminus \mathcal{M}_w]$ is not connected. Then there exist $i, j, k \in [l]$ with $i < j < k$ such that $A_i, A_k \in \mathcal{M}_v \setminus \mathcal{M}_w$ and $A_j \in \mathcal{M}_w$. By Lemma 352 we have $A_j \subseteq A_i \cup A_k$. Thus, vertex $w \in A_j$ is also contained in A_i or A_k, a contradiction. □

[1] We define the empty set as connected.

Let P and Q be two paths in T. We call $(A', A, \{A_P, A_Q\}) \in V^2 \times \binom{V}{2}$ a *fork* of P and Q, if $P[\{A', A, A_P\}]$ and $Q[\{A', A, A_Q\}]$ are induced subpaths of length 3 of P and Q, respectively, and neither A_P occurs in Q nor A_Q occurs in P. Figure 12.6 shows a fork of paths P and Q. We say P and Q *fork (in B)* if there exists a fork $(A', A, \{A_P, A_Q\})$ of P and Q (with $A = B$).

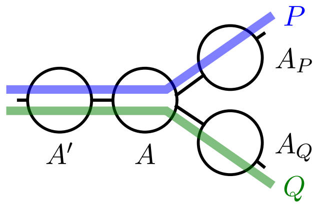


Figure 12.6.: A fork $(A', A, \{A_P, A_Q\})$ of paths P and Q

Lemma 357. *Let $v, w \in V$. If the paths $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ fork, then $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ are paths of length* 3.

Proof. Clearly, if $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ fork, then they must be paths of length at least 3. It remains to prove that their length is at most 3. For a contradiction, let us assume the length of $T[\mathcal{M}_v]$ is at least 4. Let $(A_1, B, \{A_2, A_2'\})$ be a fork of $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ where $A_2 \in \mathcal{M}_v \setminus \mathcal{M}_w$ and $A_2' \in \mathcal{M}_w \setminus \mathcal{M}_v$.

First let us assume there exists a max clique $A_0 \in \mathcal{M}_v$ such that $P = A_0, A_1, B, A_2$ is a subpath of $T[\mathcal{M}_v]$ of length 4. According to Lemma 356, $\mathcal{M}_w$ must not be a separator of $T[\mathcal{M}_v]$. Thus, we have $A_0 \in \mathcal{M}_w$ (see Figure 12.7a). Now A_0 and A_1 are distinct max cliques. Therefore, there exists a vertex $u \in A_1 \setminus A_0$. As P is a subpath of $T[\mathcal{M}_v]$ and $P' = A_0, A_1, B, A_2'$ is a subpath of $T[\mathcal{M}_w]$, vertex u is not only contained in A_1 but also in B, A_2 and A_2' by Lemma 352 (see Figure 12.7b). As a consequence, $T[\mathcal{M}_u] \supseteq T[\{A_1, B, A_2, A_2'\}]$ is not a path, a contradiction to Lemma 351.

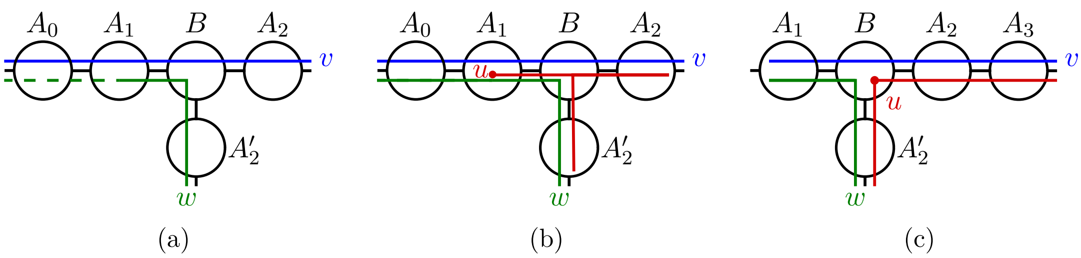


Figure 12.7.: Illustrations for the proof of Lemma 357

Next, let us assume there exists a max clique $A_3 \in \mathcal{M}_v$ such that $P = A_1, B, A_2, A_3$ is a subpath of $T[\mathcal{M}_v]$ of length 4. Further, $P' = A_1, B, A_2'$ is a subpath of $T[\mathcal{M}_w]$. As A_1

and B are max cliques, there exists a vertex $u \in B \setminus A_1$. By Lemma 352, vertex u is also contained in A_2, A_3 and A'_2 as shown in Figure 12.7c. Now let us consider the paths $T[\mathcal{M}_v]$ and $T[\mathcal{M}_u]$. $Q = A_3, A_2, B, A_1$ is a subpath of $T[\mathcal{M}_v]$, and $Q' = A_3, A_2, B, A'_2$ is a subpath of $T[\mathcal{M}_u]$. Clearly, $(A_2, B, \{A_1, A'_2\})$ is a fork of $T[\mathcal{M}_v]$ and $T[\mathcal{M}_u]$. According to the previous part of this proof, we obtain a contradiction. □

Let B be a max clique and let there be vertices $u, v, w \in V$ such that the paths $T[\mathcal{M}_u]$, $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ pairwise fork in B and $\mathcal{M}_u \cap \mathcal{M}_v \cap \mathcal{M}_w = \{B\}$. Then we call max clique B a *fork clique*.

Observation 358. *Let $u, v, w \in V$ such that $T[\mathcal{M}_u]$, $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ pairwise fork in B and $\mathcal{M}_u \cap \mathcal{M}_v \cap \mathcal{M}_w = \{B\}$. Then there exist distinct max cliques $A_1, A_2, A_3 \in \mathcal{M}$ in the neighborhood of B such that*

$$\begin{aligned}\mathcal{M}_u &= \{A_1, B, A_2\},\\ \mathcal{M}_v &= \{A_2, B, A_3\},\\ \mathcal{M}_w &= \{A_3, B, A_1\}.\end{aligned}$$

Proof. By Lemma 357 the paths $T[\mathcal{M}_u]$, $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ are of length 3. Thus, $|\mathcal{M}_u| = |\mathcal{M}_v| = |\mathcal{M}_w| = 3$. As $T[\mathcal{M}_u]$ and $T[\mathcal{M}_v]$ fork in B, there exist distinct max cliques $A_1, A_2, A_3 \in \mathcal{M}$ in the neighborhood of B such that

$$\begin{aligned}\mathcal{M}_u &= \{A_2, B, A_1\} \text{ and}\\ \mathcal{M}_v &= \{A_2, B, A_3\}.\end{aligned}$$

Since $T[\mathcal{M}_v]$ and $T[\mathcal{M}_w]$ fork in B, there exists a max clique $A'_3 \neq A_3$ or a max clique $A'_2 \neq A_2$ such that

$$\begin{aligned}\mathcal{M}_w &= \{A_2, B, A'_3\} \text{ or, respectively,}\\ \mathcal{M}_w &= \{A'_2, B, A_3\},\end{aligned}$$

In the first case, the intersection of $\mathcal{M}_u$, $\mathcal{M}_v$ and $\mathcal{M}_w$ does not only contain B but also A_2. Hence, the first case cannot occur and we must have the second case.

Lastly, we know that $T[\mathcal{M}_w]$ and $T[\mathcal{M}_u]$ fork in B. Since A_3 is distinct from A_1 and A_2, and $A'_2 \neq A_2$, it follows that $A'_2 = A_1$. □

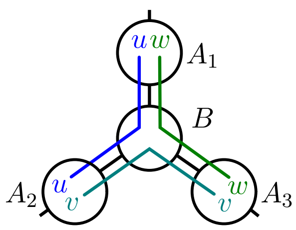


Figure 12.8.: A fork triangle around fork clique B

We say that the max cliques A_1, A_2, A_3 form a *fork triangle* around B. Figure 12.8 depicts such a fork triangle around a fork clique B. Clearly, a fork clique B is a node of degree at least 3.

Lemma 359. *Let $v, w \in V$, and let $B \in \mathcal{M}$ be a max clique. If $T[\mathcal{M}_u]$ and $T[\mathcal{M}_v]$ fork in B, then B is a fork clique.*

Proof. Let $T[\mathcal{M}_u]$ and $T[\mathcal{M}_v]$ fork in B. Then $T[\mathcal{M}_u]$ and $T[\mathcal{M}_v]$ are paths of length 3 by Lemma 357. Let $\mathcal{M}_u = \{A_2, B, A_1\}$ and $\mathcal{M}_v = \{A_2, B, A_3\}$ with $A_1 \neq A_3$ (see Figure 12.9). Since B and A_2 are max cliques, there exists a vertex $w \in B \setminus A_2$. Now, we can apply Lemma 352 to the paths $T[\mathcal{M}_u]$ and $T[\mathcal{M}_v]$, and obtain that $w \in A_1$ and $w \in A_3$. As $T[\mathcal{M}_w]$ and $T[\mathcal{M}_u]$ fork, the path $T[\mathcal{M}_w]$ must be of length 3 by Lemma 357, and cannot contain any further max cliques. Hence, B is a fork clique and A_1, A_2 and A_3 form a fork triangle. □

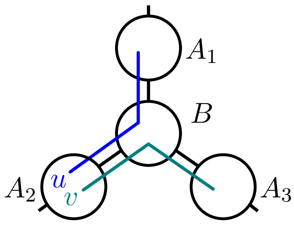
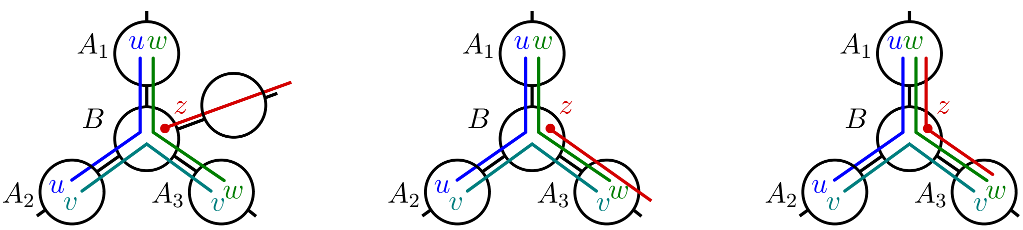


Figure 12.9.: Illustration for the proof of Lemma 359

Lemma 360. *Let $z \in V$. If the path $T[\mathcal{M}_z]$ contains a fork clique, then $|\mathcal{M}_z| = 3$ and the fork clique is in the middle of path $T[\mathcal{M}_z]$.*

Proof. Let B be a fork clique on $T[\mathcal{M}_z]$. Consequently, there exist $u, v, w \in V$ and neighbor max cliques A_1, A_2, A_3 of B such that $\mathcal{M}_u = \{A_1, B, A_2\}$, $\mathcal{M}_v = \{A_2, B, A_3\}$ and $\mathcal{M}_w = \{A_3, B, A_1\}$. Let $\mathcal{W}$ be the set $\{A_1, A_2, A_3\}$ of max cliques that form a fork triangle around B. Let us consider $|\mathcal{M}_z \cap \mathcal{W}|$. If $|\mathcal{M}_z \cap \mathcal{W}| \leq 1$, then $\mathcal{M}_z$ is a separator for at least one of the paths $T[\mathcal{M}_u]$, $T[\mathcal{M}_v]$ or $T[\mathcal{M}_w]$ as shown in Figure 12.10a and 12.10b,



(a) $|\mathcal{M}_z \cap \mathcal{W}| = 0$ (b) $|\mathcal{M}_z \cap \mathcal{W}| = 1$ (c) $|\mathcal{M}_z \cap \mathcal{W}| = 2$

Figure 12.10.: Illustrations for the proof of Lemma 360

and we have a contradiction to Lemma 356. Clearly, we cannot have $|\mathcal{M}_z \cap \mathcal{W}| = 3$, since $T[\mathcal{M}_z]$ must be a path. It remains to consider $|\mathcal{M}_z \cap \mathcal{W}| = 2$, which is illustrated in Figure 12.10c. In this case, $\mathcal{M}_z$ forks with one of the paths $T[\mathcal{M}_u]$, $T[\mathcal{M}_v]$ or $T[\mathcal{M}_w]$ in B, and must be of length 3 according to Lemma 357. Obviously, fork clique B is in the middle of the path $T[\mathcal{M}_z]$. □

Let B be a max clique. If for all $v \in B$ max clique B is an end of path $T[\mathcal{M}_v]$, we call B a *star clique*. Thus, B is a star clique if, and only if, every vertex in B is contained in at most one neighbor max clique of B. A picture of a star clique can be found in Figure 12.11. Clearly, every max clique that is a leaf of clique tree T is a star clique.

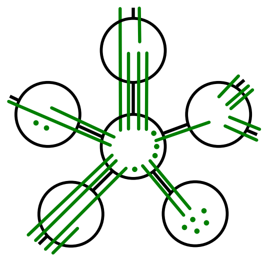


Figure 12.11.: A star clique of degree 5

Lemma 361. *Let B be a max clique. If B is of degree at least 3, then B is a star clique or a fork clique.*

Proof. Let us assume B is of degree at least 3 and not a star clique. Thus, there exists a vertex $u \in B$ and two neighbor max cliques A_1, A_2 of B in T which also contain vertex u. Let C be a neighbor of B with $C \neq A_1$ and $C \neq A_2$. Since $\{B, C\}$ is an edge of the clique tree of G, there must be a vertex $w \in V$ such that $B, C \in \mathcal{M}_w$ according to Lemma 354. See Figure 12.12 for an illustration. By Lemma 356 the induced subgraph $T[\mathcal{M}_u \setminus \mathcal{M}_w]$ must be connected. Thus, we have $A_1 \in \mathcal{M}_w$ or $A_2 \in \mathcal{M}_w$. Hence, $T[\mathcal{M}_u]$ and $T[\mathcal{M}_w]$ fork in B, and Lemma 359 implies that B is a fork clique. □

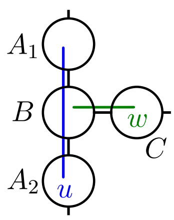


Figure 12.12.: Illustration for the proof of Lemma 361

Now let us consider nodes of T of degree at least 4.

Lemma 362. *Let B be a max clique. If B is of degree at least 4, then B is a star clique.*

Proof. Let us assume max clique B is of degree at least 4, but B is not a star clique. According to the previous lemma, B must be a fork clique. Thus, there exists vertices $u, v, w \in V$ and max cliques A_1, A_2, A_3 such that $\mathcal{M}_u = \{A_1, B, A_2\}$, $\mathcal{M}_v = \{A_2, B, A_3\}$ and $\mathcal{M}_w = \{A_3, B, A_1\}$. Now let C be a neighbor of B in T that is distinct from A_1, A_2 and A_3. According to Lemma 354 there must be a vertex $z \in V$ such that $B, C \in \mathcal{M}_z$ (see Figure 12.13a). Since $T[\mathcal{M}_w \setminus \mathcal{M}_z]$ must be connected (Lemma 356), we have $A_1 \in \mathcal{M}_z$ or $A_3 \in \mathcal{M}_z$. Without loss of generality, let $A_1 \in \mathcal{M}_z$ (see Figure 12.13b). Furthermore, $T[\mathcal{M}_v \setminus \mathcal{M}_z]$ has to be connected. Thus, max clique A_2 or max clique A_3 is in $\mathcal{M}_z$, but then $T[\mathcal{M}_z]$ is not a path. Hence, we obtain a contradiction to Lemma 351 and B is a star clique. □

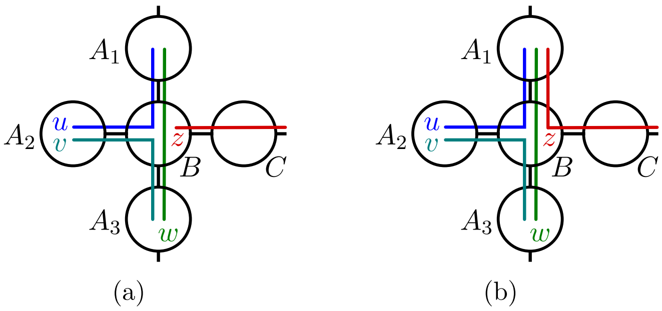


Figure 12.13.: Illustrations for the proof of Lemma 362

Corollary 363. *If B is a fork clique, then the degree of B is 3, and the neighbors of B form a fork triangle.*

Thus, a fork clique has exactly 3 neighbors forming a fork triangle. Further, it only consists of vertices that are contained in exactly two of its neighbor max cliques by Lemma 360. Figure 12.14 shows a sketch of a fork clique and its fork triangle.

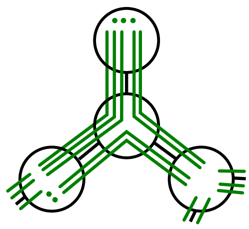


Figure 12.14.: A fork clique

Lemma 364. *Let B be a fork clique. Then every neighbor of B in clique tree T is a star clique.*

Proof. Let us assume max clique A is a neighbor of fork clique B, and A is not a star clique. Then there exists a vertex $z \in A$ and two neighbor max cliques C_1 and C_2 of A with $C_1 \neq C_2$, $z \in C_1$ and $z \in C_2$. First let us suppose $C_1 \neq B$ and $C_2 \neq B$ as depicted in Figure 12.15a. Since B is a fork clique and the degree of B is 3 (Corollary 363), there exists a vertex $u \in B$ such that $u \in A$ and A is an end of $T[\mathcal{M}_u]$. Clearly, $\mathcal{M}_u$ is a separator of $T[\mathcal{M}_z]$ which is a contradiction to Lemma 356. Next let us assume that, without loss of generality, $C_1 = B$. This case is illustrated in Figure 12.15b. Then the path $T[M_z]$ contains fork clique B. By Lemma 360 we have $|\mathcal{M}_z| = 3$ and fork clique B is in the middle of path $T[\mathcal{M}_z]$. Hence, we have a contradiction to $C_2 \in \mathcal{M}_z$. □

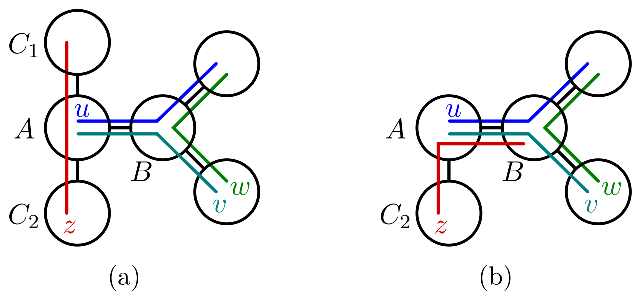


Figure 12.15.: Illustrations for the proof of Lemma 364

12.4. Defining the Clique Tree in FO

In this section we use the property that each max clique of a chordal claw-free graph can be spanned by three vertices to define the clique tree of a connected chordal claw-free graph in first order logic.

Remember, vertices $b_1, b_2, b_3 \in V$ *span* a max clique $A \in \mathcal{M}$ if A is the only max clique that contains the vertices b_1, b_2, b_3. Thus, vertices $b_1, b_2, b_3 \in V$ span max clique $A \in \mathcal{M}$ if and only if $\mathcal{M}_{b_1} \cap \mathcal{M}_{b_2} \cap \mathcal{M}_{b_3} = \{A\}$.

Lemma 365. *Every max clique of a chordal claw-free graph G is spanned by three vertices.*

Proof. Let T be a clique tree of G. Let us consider an arbitrary max clique B, and let $v \in B$ be an arbitrary vertex that is contained in max clique B. By Lemma 351, the induced subgraph $T[\mathcal{M}_v]$ is a path $P = B_1, \ldots, B_l$. Let $B = B_i$. Now let us pick two more vertices. If $i > 1$, let u be a vertex in $B \setminus B_{i-1}$, and let w be a vertex in $B \setminus B_{i+1}$ if $i < l$. We let $u = v$ if $i = 1$, and we let $w = v$ if $i = l$. Then u, v, w span max clique B. Clearly, B contains all of the three vertices. It remains to show, that there does not exist a max clique $A \in \mathcal{M}$ with $A \neq B$ and $u, v, w \in A$. Let us suppose such a max clique A exists. Since $v \in A$, max clique A is a node on path P. Without loss of generality let $A = B_j$ for $j < i$. As $u \notin B_{i-1}$, we have $j < i - 1$. Since T is a tree, $\{B_{i-1}\}$ separates $A = B_j$ and $B = B_i$ in T. Thus, every path connecting A and B in T must contain B_{i-1}. As $u \in A$ and $u \in B$ the path $T[\mathcal{M}_u]$ contains A and B, and therefore connects the two max cliques. Hence, it must also contain max clique B_{i-1}. We obtain a contradiction, because $u \notin B_{i-1}$. □

As a direct consequence of Lemma 365, there exists an at most cubic number of max cliques in a chordal claw-free graph.

Now we use the property that for chordal claw-free graphs all max cliques are spanned by three vertices to define the clique tree of a connected chordal claw-free graph. We showed in Section 2.8.2 that if all max cliques of a graph can be spanned by three vertices, then there exists an FO-formula (presented in (2.2)) which decides for three vertices whether the vertices are spanning vertices of a max clique. Moreover, we showed in (2.4) that the equivalence classes of spanning vertices that are spanning the same max cliques are FO-definable. We use these equivalence classes to represent the max cliques, and obtain the following corollary.

Corollary 366. *The max cliques of a chordal claw-free graph are* FO*-definable.*

In fact, given a tuple of spanning vertices we can also decide which vertices belong to the max clique that is spanned: There further exists an FO-formula (see (2.3), Section 2.8.2) that is satisfied for vertices v_1, v_2, v_3, w in a chordal claw-free graph G if and only if the vertices v_1, v_2, v_3 span a max clique A and $w \in A$. We need this property to define the edge relation of the clique tree of a connected chordal claw-free graph G.

Lemma 367. *There exists an* FO*-formula that defines the edge relation of the clique tree of a connected chordal claw-free graph G.*

Proof. We have to show that there exists an FO-formula that decides whether two max cliques are adjacent. By Lemma 354 there is an edge between two max cliques in the clique tree if, and only if, there is an edge between the max cliques on a path $T[\mathcal{M}_v]$ for a vertex v of G. Further, we know for max cliques $A, B \in \mathcal{M}_v$ that they are adjacent precisely when there does not exist a max clique $C \in \mathcal{M}_v$ with $C \subseteq A \cup B$ (Lemma 352). Thus, there is an edge between max cliques A and B if, and only if, there exists a vertex v such that $v \in A$ and $v \in B$ and for all max cliques C with $v \in C$ we have $C \nsubseteq A \cup B$. Clearly, this can be put in form of a first order formula. □

Corollary 368. *There exists an* FO*-transduction that defines the clique tree of G for every connected chordal claw-free graph G.*

12.5. Directed Clique Trees

Now we transfer the clique tree into a rooted tree. We show that there exists a parameterized STC-transduction, which reduces each connected chordal claw-free graph to one of its rooted clique trees. In this transduction, the root of the clique tree represents the parameter. Further, we summarize the basic structure of rooted clique trees.

Let R be a leaf of the clique tree T of a connected chordal claw-free graph G. We call such a max clique R a *leaf max clique*. We transform clique tree T into a directed tree by rooting T at max clique R. We let $\mathcal{E}_R$ be the edge relation of this directed tree, and we denote this directed clique tree rooted at R by $T^R = (\mathcal{M}, \mathcal{E}_R)$.

Each tree can be transformed into a directed tree in STC when additionally given a node, the root node of the directed tree (see Example 8 in Section 2.5.2). Thus, there is an STC-formula which given the spanning vertices of R and spanning vertices of max cliques A and B decides whether there is an arc from A to B in T^R. It is not hard to see that we can use this formula to construct a parameterized transduction which given a connected chordal claw-free graph defines the directed clique tree T^R of G. The spanning vertices of R are the parameter of this transduction.

Let us briefly consider the directed clique tree T^R and summarize its basic structure. Since the root of T^R is a leaf of T, each node of T^R with more than one child is of degree at least 3 in T. Hence, the following corollary is an immediate consequence of Lemma 361.

Corollary 369. *If max clique A is a node in the directed clique tree T^R with at least two children, then A is a star clique or a fork clique.*

If max clique A is not a fork clique, then max clique A is a star clique or A has at most one child. Therefore, the vertices occurring in A are contained in at most one child of A. If max clique A is a fork clique, it has exactly two children (Corollary 363), and the vertices in A can be contained in both of its children. Moreover, we know that there actually exist a vertex in A that occurs in both of its children. We can use this property to identify fork cliques. Further, the three neighbors of fork cliques are star cliques (Lemma 364). Hence, if we have a fork clique, we know the vertices in that fork clique only occur in its three neighbors and in no further descendants or ancestors.

12.6. The Supplemented Clique Tree

We want to use the directed clique tree and its properties to construct the canon of a connected chordal claw-free graph G. Therefore, we include some of the structural information about each max clique into the directed clique tree by means of an LO-coloring. (LO-colored graphs were introduced in Section 2.3.4.) We present this LO-colored directed clique tree, called supplemented clique tree, in this section. Further, we show that every connected chordal claw-free graph can be reduced to such a supplemented clique tree by means of a parameterized STC+C-transduction.

Let G be a connected chordal claw-free graph, and T^R be the directed clique tree of G rooted at a leaf max clique R. We color every max clique A in T^R with a binary numeric relation, which encodes certain properties of A in T^R. For example, if n is the number of vertices in A that are not contained in any child max clique of A, we add $(0, n)$ to the color of A. We also add a pair to the color of a max clique A to encode the number of vertices that are in A and its parent max clique if it exists. Further, we need to know the number of vertices in A that are contained in at least two children max cliques of A. If this number is different from 0, then clearly A is a fork clique and we also know how many vertices in A are contained in both children of A. We construct the color of A such that it includes all these numbers.

We call such a colored directed clique tree of a connected chordal claw-free graph G a *supplemented clique tree* of G. Hence, a supplemented clique tree of a connected chordal claw-free graph G is a 3-tuple $S^R = (\mathcal{M}, \mathcal{E}_R, P)$ where

- R is a leaf of the clique tree of G,
- $T^R := (\mathcal{M}, \mathcal{E}_R)$ is the directed clique tree of G with root R, and
- $P \subseteq \mathcal{M} \times [|\mathcal{M}|]^2$ is a ternary relation where
  - $(A, 0, n) \in P$ iff n is the number of elements in A that do not occur in any child of A in T^R,
  - $(A, 1, n) \in P$ iff n is the number of elements contained in the intersection of A and the parent of A in T^R if $A \neq R$, and $n = 0$ if $A = R$,
  - $(A, 2, n) \in P$ iff n is the number of elements in A that occur in at least two child max cliques of A in T^R.

We understand the supplemented clique tree S^R as an LO-colored graph. Thus, in its structural representation the supplemented clique tree S^R corresponds to the 6-tuple $(\mathcal{M} \,\dot\cup\, [0, |V|], \mathcal{M}, \mathcal{E}_R, [0, |V|], \leq_{[0,|V|]}, P)$, where $[0, |V|]$ is the set of basic color elements and $\leq_{[0,|V|]}$ is the natural linear order on $[0, |V|]$. In order to be able to consider supplemented clique trees instead of connected chordal claw-free graphs, we need a (parameterized) transduction defining us (an isomorphic copy of) the supplemented clique tree S^R for every connected chordal claw-free graph. So far, we have shown in Section 12.4 that there exists a parameterized STC-transduction which reduces every connected chordal claw-free graph G to the directed clique tree $T^R = (\mathcal{M}, \mathcal{E}_R)$ where root max clique R is determined by the parameters of the transduction. We can extend this parameterized transduction by STC+C-formulas that define the set of basic color elements $[0, |V|]$, the linear order $\leq_{[0,|V|]}$ on the basic color elements and the color relation P. (By using the formulas for the max cliques and edge relation from Section 12.4, the properties encoded in the color of a max clique are easily expressible in STC+C.) We obtain a parameterized counting transduction[2] that defines us the supplemented clique tree $S^R = (\mathcal{M} \,\dot\cup\, [0, |V|], \mathcal{M}, \mathcal{E}_R, [0, |V|], \leq_{[0,|V|]}, P)$ of each connected chordal claw-free graph G. Now we apply Proposition 14. As a result, we obtain a parameterized STC+C-transduction that defines an LO-colored graph isomorphic to S^R for every connected chordal claw-free graph G.

12.7. Canonization

In the following we describe how we obtain an $\mathsf{LREC}_=$-canonization of the class of connected chordal claw-free graphs.

From the previous section we know that there exists a parameterized STC+C- and therefore $\mathsf{LREC}_=$-transduction that gives us (an isomorphic copy of) the supplemented clique tree in form of an LO-colored directed tree. Further, in Section 8.4 we showed how

[2] Counting transductions are introduced in Section 2.5.1.

to obtain an LREC-canonization of LO-colored directed trees. As a consequence, there exists a parameterized $\mathsf{LREC}_=$-transduction for the composition of both transductions by Proposition 12. Hence, there is a parameterized $\mathsf{LREC}_=[\{E\}, \{V, E, M, \trianglelefteq, L, \leq\}]$-transduction $\Theta(\bar{x})$ that maps every connected chordal claw-free graph G to the canon $K(S^R)$ of the supplemented clique tree S^R, where the parameters are the spanning vertices of the root max clique R (of the underlying directed tree) of the supplemented clique tree.

In order to prove that there exists a parameterized $\mathsf{LREC}_=$-canonization of connected chordal claw-free graphs, we show that there is an $\mathsf{LREC}_=[\{V, E, M, \trianglelefteq, L, \leq\}, \{E, \leq\}]$-transduction Θ', which maps the canon of the supplemented clique tree to a canon $(K, \leq_K)$ of the original graph G. Then we can again apply Proposition 12 and obtain an $\mathsf{LREC}_=$-transduction for the composition of $\Theta(\bar{x})$ and Θ', which is a parameterized $\mathsf{LREC}_=$-canonization of the class of connected chordal claw-free graphs.

We let the canon $(K, \leq_K)$ of graph G defined by transduction Θ' consist of an isomorphic copy K of G on the number sort. We let $[|V|]$ be the vertex set of K. Further, we let $\leq_K$ be the natural linear order on $[|V|]$. As $[|V|]$ is the set of basic color elements of the supplemented clique tree S^R, the set of basic color elements $M(K(S^R))$ of the canon $K(S^R)$ of the supplemented clique tree S^R also contains exactly $|V|$ elements. Hence, we can easily define the vertex set of the canon by counting the number of basic color elements. We let $\Theta' = (\theta_U(p), \theta_E(p, p'), \theta_\leq(p, p'))$ be the $\mathsf{LREC}_=[\{V, E, M, \trianglelefteq, L, \leq\}, \{E, \leq\}]$-transduction, where

$$\begin{aligned} \varphi_U(p) &:= \exists q \left(p \leq q \wedge p \neq 0 \wedge \#x\, M(x) = q \right), \quad \text{and} \\ \theta_\leq(p, p') &:= p \leq p'. \end{aligned}$$

In order to show the existence of $\theta_E(p, p')$, which, given the canon of a supplemented clique tree of G, defines the edge relation of isomorphic copy K of G on the number sort, we exploit the property that $\mathsf{LREC}_=$ captures logarithmic space on ordered structures. Hence, it suffices to present a logspace algorithm that, given the canon of a supplemented clique tree of G and two numbers, decides whether the pair of numbers belongs to the edge relation of the isomorphic copy K of graph G on the number sort.

In the next section we present a logspace algorithm that outputs the max cliques of the canon. As every edge is a subset of a max clique and every two vertices in a max clique form an edge, we make the following observation.

Observation 370. *Let $\mathcal{M}_H$ be the set of all max cliques of a graph H. Then the set $\bigcup_{M \in \mathcal{M}_H} \binom{M}{2}$ of all binary subsets of max cliques of $\mathcal{M}_H$ is the edge relation of H.*

We can extend each logspace algorithm that outputs the max cliques of the canon to a logspace algorithm that decides whether a pair of numbers is an edge of the canon. We simply conduct a logspace computation that goes through all pairs of distinct vertices of each max clique and compares them to the given pair of numbers.

12.8. Algorithm for Computing the Max Cliques of the Canon

In this section we present an algorithm such that for every connected chordal claw-free graph G, given the canon $K(S^R)$ of a supplemented clique tree S^R of G, the algorithm computes the max cliques of an isomorphic copy K of G on the number sort with vertex set $[|V|]$.

The algorithm performs a post-order depth-first traversal on the underlying tree of the canon $K(S^R)$ of the supplemented clique tree S^R. Each node of the canon $K(S^R)$ corresponds to a max clique in the supplemented clique tree S^R, and for each node in $K(S^R)$ the algorithm constructs a copy of the corresponding max clique on the number sort. In order to construct the max cliques the algorithm uses the information we have on the structure of the clique tree and the information on the structure of the max cliques that is contained in the coloring of the nodes.

In the following we describe the algorithm. We start with its structure, then focus on its basic idea and necessary observations, until we finally present it. Afterwards, we prove its correctness and show that its data complexity is in LOGSPACE.

For the canon $K(S^R) = (U_K, V_K, E_K, M_K, \trianglelefteq_K, L_K, \leq_K)$ of the supplemented clique tree, which is an ordered LO-colored graph, we can easily compute the natural colors in logspace. Hence, for simplicity we assume the canon $K(S^R)$ of the supplemented clique tree is colored with the natural colors. Therefore, $M_K = [0, |V|]$ and $\trianglelefteq_K = \leq_{[0,|V|]}$.

Post-Order Depth-First Tree Traversal

Let G be a connected chordal claw-free graph, and $K(S^R)$ be the canon of the supplemented clique tree S^R of G. The algorithm uses post-order traversal (see e.g. [63]) on the underlying directed tree of $K(S^R)$ to construct the max cliques of the canon K of G. Like pre-order and in-order traversal, post-order traversal is a type of depth-first tree traversal, that specifies a linear order on the nodes of a tree. Depth-first tree traversal was introduced in Section 2.8.1.

Keep in mind that the universe, and therefore the nodes, of the canon of the supplemented clique tree are linearly ordered. Thus, we have a linear order on the children of a node, and we assume the children of a node to be given in that order.

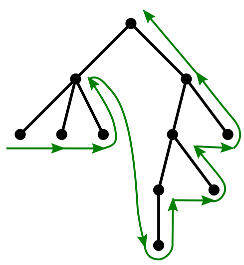
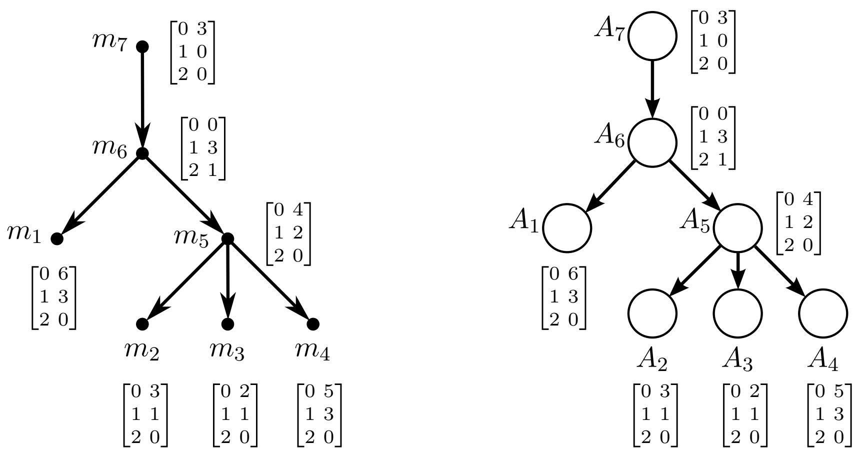


Figure 12.16.: Post-order traversal

The *post-order traversal sequence* consists of every node we visit during the depth-first traversal in order of its last visit. During depth-first traversal we visit each node with children twice: After we visit it for the first time, we go down to its first child, and after we finished traversing its children we visit it for the second time. Nodes without children are visited only once. It follows that we obtain the post-order traversal sequence by successively adding all nodes visited during depth-first traversal that are not followed by the move **down**. The post-order traversal is depicted in Figure 12.16 (cf. Figure 2.1, which illustrates the depth-first traversal).

Let $m_1, \dots, m_{|\mathcal{M}|}$ be the post-order traversal sequence for the canon $K(S^R)$ of the supplemented clique tree S^R. Then, $m_1, \dots, m_{|\mathcal{M}|}$ is a permutation of the nodes of the underlying directed tree of $K(S^R)$. We know there exists an isomorphism I between $K(S^R)$ and S^R. For all $k \in [|\mathcal{M}|]$ the isomorphism I maps the node m_k of the underlying directed tree of $K(S^R)$ to a max clique $A_k := I(m_k)$ in the supplemented clique tree S^R. Notice that the nodes m_k and A_k contain the same information in their color. We call $A_1, \dots, A_{|\mathcal{M}|}$ the traversal sequence *transferred* by isomorphism I. The isomorphism I also transfers the ordering of the children of a node. Figure 12.17 shows an example of a canon $K(S^R)$ and its post-order traversal sequence $m_1, \dots, m_{|\mathcal{M}|}$, and the corresponding supplemented clique tree S^R and its transferred traversal sequence $A_1, \dots, A_{|\mathcal{M}|}$.



(a) Canon $K(S^R)$ and its post-order traversal sequence

(b) The supplemented clique tree S^R and its transferred traversal sequence

Figure 12.17.

Clearly, in the post-order traversal sequence of a tree, a proper descendant of a node occurs before the node. Regarding the supplemented clique tree S^R, this means:

Observation 371. *Let $A_1, \dots, A_{|\mathcal{M}|}$ be the transferred post-order traversal sequence on S^R, and let $i, i' \in [|\mathcal{M}|]$. If max clique A_i is a proper descendant of max clique $A_{i'}$ in T^R, then $i < i'$.*

Intersections of Max Cliques with preceding Max Cliques in the transferred post-order traversal sequence

We traverse the underlying directed tree of $K(S^R)$ in post-order, and we construct the max cliques of an isomorphic copy of G on the number sort during this post-order traversal. So for each node m_k of the directed tree we construct a max clique B_{m_k} of numbers, which is the max clique of the canon of G corresponding to max clique A_k of graph G.

In order to construct these max cliques B_{m_k} during the traversal of the underlying directed tree of $K(S^R)$, we have to decide on numbers for all vertices that are supposed to be in such a max clique. The numbering happens according to the post-order traversal sequence. The hard part will be to detect which vertices have already occurred in a max clique corresponding to a node we have visited before reaching m_k, and to determine the numbers they were assigned to. Then we can correctly create new numbers for newly occurring vertices and reuse the ones that have already occurred. Thus, in the following we take the transferred post-order traversal sequence $A_1, \dots, A_{|\mathcal{M}|}$ and study the intersection of a max clique A_k with max cliques that precede A_k in the transferred traversal sequence.

An important observation in this respect is that if A_k is a fork clique, then the vertices in A_k only occur in the two children and the parent max clique of fork clique A_k (Corollary 363). Thus, apart from the two children of A_k the vertices in A_k are not contained in any other max clique previously visited in the transferred traversal sequence. Further, each vertex in A_k occurs in at least one child max clique of A_k. Hence, each vertex in A_k is contained in a max clique that was visited before.

If max clique A_k is not a fork clique, we so far only know that the vertices in A_k occur in no more than one child max clique of A_k, because if A_k is not a fork clique, then it has only one child or is a star clique (Lemma 361). In the following we show that each vertex in non-fork clique A_k that occurs in a max clique that is visited before A_k is either contained in exactly one child of A_k or in the first child of a fork clique A_l if A_k is the second child of A_l.

Lemma 372. *Let $A_1, \dots, A_{|\mathcal{M}|}$ be the transferred post-order traversal sequence on S^R. Let $k \in [|\mathcal{M}|]$ and let $v \in A_k$. If there exists a $j < k$ such that $v \in A_j$ and A_j is not a descendant of A_k in the underlying directed tree T^R of S^R, then A_j is the first and A_k the second child of a fork clique.*

Proof. Let $j < k$ and $v \in A_j \cap A_k$. Further, let A_j be not a descendant of A_k. As $j < k$, max clique A_j also cannot be a proper ancestor of A_k by Observation 371. Consequently, the smallest common ancestor A_l of A_j and A_k must be a proper ancestor of A_j and A_k. Thus, A_l has at least two children. Since max clique A_l cannot be the root of the directed tree T^R (the root is a leaf of the clique tree), A_l is a node of degree at least 3 in T. Lemma 361 yields that A_l is either a star or a fork clique. According to the clique intersection property vertex v is contained in A_l and every max clique on the path between A_j and A_k. Thus, A_l must be a fork clique, and $T[\mathcal{M}_v]$ must be a path of length 3 (Lemma 357). Therefore, A_j and A_k are the children of fork clique A_l. Since $j < k$, we have that A_j is the first and A_k the second child of A_l. □

Lemma 373. *Let $A_1, \ldots, A_{|\mathcal{M}|}$ be the transferred post-order traversal sequence on S^R. Let $k \in [|\mathcal{M}|]$ and $v \in A_k$. If there exists a $j < k$ such that $v \in A_j$ and A_k is not a fork clique, then there exists exactly one $i \in [|\mathcal{M}|]$ such that $v \in A_i$ and either*

1. *A_i is a child of A_k or*
2. *A_i is the first child of a fork clique and A_k the second one.*

Proof. Let $j < k$ and $v \in A_j \cap A_k$, and let A_k be not a fork clique. If A_j is a descendant of A_k, then there exists an $i \in [|\mathcal{M}|]$ such that $v \in A_i$ and A_i is a child of A_k by the clique intersection property. If A_j is not a descendant of A_k, then by the previous lemma there exists an $i \in [|\mathcal{M}|]$, that is, $i = j$, such that A_i is the first and A_k the second child of a fork clique. Thus, there exists an $i \in [|\mathcal{M}|]$ such that $v \in A_i$ and property 1 or 2 is satisfied. Now, let us assume there exist $i_1, i_2 \in [|\mathcal{M}|]$ with $i_1 \neq i_2$ such that $v \in A_{i_m}$ and property 1 or 2 is satisfied regarding i_m for $m \in [2]$. Clearly, property 2 cannot be satisfied regarding both, i_1 and i_2. Let us consider the case where property 1 is satisfied for i_1 and property 2 is satisfied for i_2. Then A_{i_2} is the first child of a fork clique A_l and A_k is the second one. According to Lemma 364 max clique A_k is a star clique. If v is contained in A_{i_2} and in a child max clique A_{i_1} of A_k, then by the clique intersection property, we also have $v \in A_k$ and $v \in A_l$, which is a contradiction to A_k being a star clique. Next let us consider the case, where property 1 is satisfied regarding i_1 and i_2. Then A_k has at least two children, and since it cannot be the root of T^R, it must be a node of degree at least 3. As max clique A_k is not a fork clique, it must be a star clique by Lemma 361. Now, the set $\mathcal{M}_v$ contains two children of A_k. Therefore, the path $T[\mathcal{M}_v]$ also contains A_k, but not as an end, a contradiction to A_k being a star clique. □

Now, if A_k is a fork clique, then we know the vertices of A_k all occur in its two children, which occur before A within the transferred traversal sequence. If A_k is not a fork clique, then by Lemma 373 the vertices in A_k that occur in max cliques before A_k within the transferred traversal sequence are precisely the vertices in the pairwise intersection of A_k with its children, and the intersection of A_k with its sibling if A_k is the second child of a fork clique. Further, Lemma 373 yields that these intersections are disjoint sets of vertices.

Algorithm to Construct the Max Cliques B_m

We now include the new knowledge about the intersection of max cliques with preceding max cliques into our construction of the sets B_m. For the numbers in B_m we will maintain the property that if a number $l \in B_m$ is contained in more ancestors of B_m than a number $l' \in B_m$, then $l > l'$. Thus, for each child max clique B_k of max clique B_m the intersection $B_k \cap B_m$ contains precisely the $|B_k \cap B_m|$ largest numbers of B_k. In the following we present an algorithm that computes the sets B_m.

During the algorithm, we need to remember or compute a couple of values: At each step of our traversal, we let `count` be the total number of vertices we have created so far. We update this number after each visit of a node in the post-order traversal sequence of the directed tree. Sometimes we need to recompute this number for another node m'. We denote this recomputed value by `count`(m'). Note, that this is the updated value of `count` *after* visiting m'. Further, we exploit the information contained in the color of a node m. We let

- `in0children`(m) be the number of vertices that are contained in the max clique represented by m but do not occur in a max clique corresponding to any child of m,
- `inparent`(m) be the number of vertices that are contained in the max clique represented by m and the max clique represented by its parent (if m is the root of the tree, then `inparent`(m) will be 0), and
- `in2children`(m) be the number of vertices that are contained in the max clique corresponding to m and in at least two max cliques represented by children of m.

We also need the boolean values

- `isforkclique`(m) which indicates whether m corresponds to a fork clique, and
- `isforkchild2`(m) which indicates whether m is the second child of a node corresponding to a fork clique.

With help of the above values, we can complete the algorithm. Thus, let us describe the algorithm at a node m during the post-order traversal. The algorithm distinguishes between the following cases. For each case we list the numbers belonging to max clique B_m, and indicate the values used to determine the numbers in B_m.

1. **Node m corresponds to a fork clique (`isforkclique`$(m) = \text{true}$).**
   Let m' be the first child of m, and m'' the second one. We determine `count`(m'), and we know `count`$(m'') =$ `count`. Further, we need `inparent`(m') and `inparent`(m''), and `in2children`(m). We let B_m be the set of numbers in

   $$[\texttt{count}(m') - \texttt{inparent}(m') + 1,\ \texttt{count}(m')] \quad \text{and}$$
   $$[\texttt{count}(m'') - \texttt{inparent}(m'') + \texttt{in2children}(m) + 1,\ \texttt{count}(m'')].$$

   We do not increase `count`.

2. **Node m does not correspond to a fork clique (`isforkclique`$(m) = \text{false}$).**
   Let $m_1, \ldots, m_k$ be the children of m where $k \geq 0$. Now for all $j \in [k]$ we determine `isforkclique`(m_j), and distinguish between the following two cases.

   a) `isforkclique`$(m_j) = \text{false}$:
   We determine `count`(m_j) and `inparent`(m_j) and we add to B_m the numbers in

   $$[\texttt{count}(m_j) - \texttt{inparent}(m_j) + 1,\ \texttt{count}(m_j)]$$

   b) `isforkclique`$(m_j) = \text{true}$:
   Let m'_j and m''_j be the children of m_j. We add to B_m the numbers in

   $$[\texttt{count}(m'_j) - \texttt{inparent}(m'_j) + \texttt{in2children}(m_j) + 1,\ \texttt{count}(m'_j)] \quad \text{and}$$
   $$[\texttt{count}(m''_j) - \texttt{inparent}(m''_j) + \texttt{in2children}(m_j) + 1,\ \texttt{count}(m''_j)].$$

   Further, we determine `isforkchild2`(m) and depending on the value, we do the following.

c) `isforkchild2`(m) = false:
We increase `count` by `in0children`(m), and add to B_m the vertices in

$$[\texttt{count} - \texttt{in0children}(m) + 1,\ \texttt{count}].$$

d) `isforkchild2`(m) = true:
Let p be the parent of m, and let m' be the first sibling of m. We increase `count` by `in0children`(m) − `in2children`(p). We add to B_m the vertices in the intervals

$$[\texttt{count}(m')-\texttt{inparent}(m')+1,\ \texttt{count}(m')-\texttt{inparent}(m')+\texttt{in2children}(p)],$$
$$[\texttt{count} - \texttt{in0children}(m) + \texttt{in2children}(p) + 1,\ \texttt{count}].$$

In the following we illustrate the algorithm with an example.

i	m_i	Case	B_{m_i}	`count`
				0
1	m_1	2 c)	$[1, 6]$	6
2	m_2	2 c)	$[7, 9]$	9
3	m_3	2 c)	$[10, 11]$	11
4	m_4	2 c)	$[12, 16]$	16
5	m_5	2 a) for m_2	$[9, 9]$	
		2 a) for m_3	$[11, 11]$	
		2 a) for m_4	$[14, 16]$	
		2 d)	$[4, 4] \cup [17, 19]$	19
6	m_6	1	$[4, 6] \cup [19, 19]$	19
7	m_7	2 b) for m_6	$[5, 6] \cup [19, 19]$	
		2 c)	$[20, 22]$	22

Table 12.1.: The algorithm applied to the example in Figure 12.17a

Example 374. The algorithm can be applied to the canon $K(S^R)$ depicted in Figure 12.17a. Table 12.1 shows the computed values at each step of the algorithm. It contains for each node m_i of the post-order traversal sequence the cases that need to be considered, for each case the obtained partial intervals forming B_{m_i}, and the value of `count`. Figure 12.18 shows the max cliques B_{m_i} for all i. ⌟

Correctness of the Algorithm

Now we show that the presented algorithm returns the max cliques of a graph K with vertex set $[|V|]$ that is isomorphic to G.

We prove that there is a bijection h between V and $[|V|]$, so that for all $k \in [|\mathcal{M}|]$ we have $h(A_k) = B_{m_k}$. Then h is a graph isomorphism between G and graph K on the

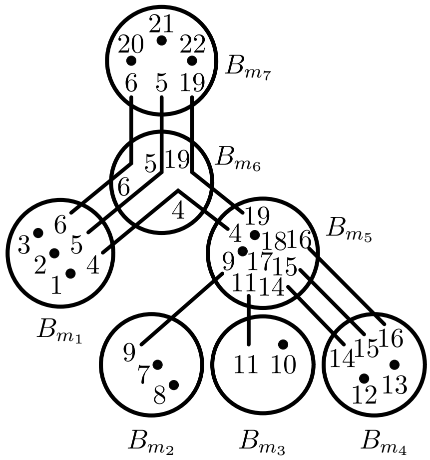


Figure 12.18.: The clique tree with the max cliques B_{m_i}, $i \in [7]$

number sort, because for all $v, v' \in V$:

$$\begin{aligned} & \text{There is an edge between } v \text{ and } v' \text{ in } G. \\ \overset{\text{Obs. 370}}{\iff} \; & \text{There exists a } k \in [|\mathcal{M}|] \text{ with } v, v' \in A_k. \\ \iff \; & \text{There exists a } k \in [|\mathcal{M}|] \text{ with } h(v), h(v') \in B_{m_k}. \\ \overset{\text{Obs. 370}}{\iff} \; & \text{There is an edge between } h(v) \text{ and } h(v') \text{ in } K. \end{aligned}$$

We prove the existence of bijection h by showing the claim below by induction along the post-order traversal sequence. First, we introduce the necessary definitions used in the claim.

Let T^R be the underlying directed clique tree of the supplemented clique tree S^R. For all max cliques $A \in \mathcal{M}$ and for all $v \in A$ we let $\#\text{anc}_A(v)$ be the number of max cliques in T^R that contain vertex v and are an ancestor of A. Clearly, for every vertex $v \in A$ the number $\#\text{anc}_A(v)$ is at least 1. Let $A_1, \dots, A_{|\mathcal{M}|}$ be the transferred traversal sequence. For $i \in [|\mathcal{M}|]$ and $c \in [2]$ let S_i^c be the set of vertices v of max clique A_i, where $\#\text{anc}_{A_i}(v) > c$. Thus, if max clique A_i has a parent max clique P_i in T^R, then S_i^1 is the set of vertices in $A_i \cap P_i$. Hence, $\texttt{inparent}(m_i) = |S_i^1|$. If again P_i has a parent in T^R, then S_i^2 is the subset of vertices of A_i which are contained in P_i and the parent of P_i. For example, if A_l is a fork clique with children A_i and A_j, then A_l is the disjoint union of S_i^1 and S_j^2. Further, if $A_{l'}$ is the parent max clique of fork clique A_l, then $A_{l'}$ is the disjoint union of S_i^2 and S_j^2.

Claim 375. *For all $l \in [|\mathcal{M}|]$ there exists a bijection h_l between $A_1 \cup \dots \cup A_l$ and $[\texttt{count}(m_l)]$, such that for all $i \in [l]$ we have*

1. $h_l(A_i) = B_{m_i}$,
2. $\#\text{anc}_{A_i}(v) \le \#\text{anc}_{A_i}(v')$ *for all vertices $v, v' \in A_i$ with $h_l(v) \le h_l(v')$ if A_i is neither a fork clique nor the second child of a fork clique,*
3. $h_l(S_i^1) = [\texttt{count}(m_i) - \texttt{inparent}(m_i) + 1,\ \texttt{count}(m_i)]$ *if A_i is neither a fork clique nor the second child of a fork clique, and*
4. $h_l(S_i^2) = [\texttt{count}(m_i) - \texttt{inparent}(m_i) + \texttt{in2children}(p_i) + 1,\ \texttt{count}(m_i)]$ *if A_i is the second child of a fork clique. (We let p_i be the parent of m_i.)*

Proof (Claim 375). We prove Claim 375 by induction on $l \in [0, |\mathcal{M}|]$. Notice that $l = 0$ is not included in the claim, but we extend it to $l = 0$. Although there does not actually exist a node m_0, we let $\texttt{count}(m_0)$ be 0. This makes sense, since 0 is the initial value of count. We let $h_0 \colon \emptyset \to \emptyset$ be the empty mapping. Clearly, h_0 meets all the requirements. Now let $l > 0$ and let there be a bijection h_{l-1} with properties 1 to 4 for all $i \in [l-1]$. We show the existence of bijection h_l.

First, let us consider the case where m_l corresponds to a fork clique. Clearly, A_l is a subset of the set of vertices occurring in A_l's children max cliques and $\texttt{count}(m_l) = \texttt{count}(m_{l-1})$. Thus, we let $h_l := h_{l-1}$, and we know by inductive assumption that it is a bijection. By inductive assumption we also know that h_l satisfies properties 1 to 4 for all $i < l$. Therefore, it remains to show these properties for $i = l$. As A_l is a fork clique, and cannot be the second child of a fork clique, properties 2, 3 and 4 trivially hold for $i = l$. Thus, we only have to show that h_l satisfies property 1 for $i = l$, that is, $h_l(A_l) = B_{m_l}$.

So let us prove that $h_l(A_l) = B_{m_l}$. Let m_i and m_j with $i < j < l$ be the first and second child of m_l. Since m_i cannot correspond to a fork clique or the second child of a fork clique, we have $h_l(S_i^1) = [\texttt{count}(m_i) - \texttt{inparent}(m_i) + 1,\ \texttt{count}(m_i)]$ by inductive assumption. Analogously, we know $h_l(S_j^2) = [\texttt{count}(m_j) - \texttt{inparent}(m_j) + \texttt{in2children}(m_l) + 1,\ \texttt{count}(m_j)]$ because m_j corresponds to the second child of a fork clique. We obtain that $B_{m_l} = h_l(S_i^1) \cup h_l(S_j^2)$. As A_l is a fork clique, A_l is the disjoint union of S_i^1 and S_j^2. Hence, we have $B_{m_l} = h_l(A_l)$.

Next, let m_l be a node that does not correspond to a fork clique. By Lemma 373 we know that there are $\texttt{in0children}(m_l)$ vertices in $A_l' := A_l \setminus \bigcup_{i<l} A_i$ if A_l is not the second child of a fork clique, and $\texttt{in0children}(m_l) - \texttt{in2children}(m_{l+1})$ vertices in A_l' if A_l is the second child of a fork clique (then m_{l+1} is the parent of m_l). Thus, A_l' and the set B_{m_l}' of newly occurring numbers in B_{m_l} have the same cardinality. We let h_l be an extension of h_{l-1} that bijectively maps the vertices in A_l' to the numbers in B_{m_l}' such that $h_l(v) \leq h_l(v')$ implies $\#\mathrm{anc}_{A_l}(v) \leq \#\mathrm{anc}_{A_l}(v')$ for all $v, v' \in A_l'$. Then h_l is a bijection between $A_1 \cup \cdots \cup A_l$ and $[\texttt{count}(m_l)]$. By inductive assumption we already know that h_l satisfies properties 1 to 4 for all $i < l$. Thus, we only need to show them for $i = l$.

Let us show property 1: Let $m_{i_1}, \ldots, m_{i_k}$ with $i_1 < \cdots < i_k < l$ be the children of m_l. Further, if m_l corresponds to the second child of a fork clique, then let m_{i_0} be its sibling. Clearly, $i_0 < i_1$. According to Lemma 373 max clique A_l is the disjoint union of A_l' and the sets $A_l \cap A_{i_j}$ for $j \in [k]$ if A_l is not the second child of a fork clique, and for $j \in [0, k]$ otherwise. Consequently, $h_l(A_l)$ is the disjoint union of $h_l(A_l')$ and $h_l(A_l \cap A_{i_j})$ for all feasible $j \leq k$. First, let us consider the children of m_l, that is, all m_{i_j} with $j \in [k]$. For each child m_{i_j} of m_l we have $A_l \cap A_{i_j} = S_{i_j}^1$. Now suppose for the child m_{i_j} we have $\texttt{isforkclique}(m_{i_j}) = \text{false}$. Then A_{i_j} is neither a fork clique nor the second child of a fork clique. Therefore, we have $h_l(A_l \cap A_{i_j}) = h_l(S_{i_j}^1) = h_{l-1}(S_{i_j}^1) = [\texttt{count}(m_{i_j}) - \texttt{inparent}(m_{i_j}) + 1,\ \texttt{count}(m_{i_j})]$ by inductive assumption. Next, let us assume $\texttt{isforkclique}(m_{i_j}) = \text{true}$. Then m_{i_j} corresponds to a fork clique. Let m_i and $m_{i'}$ be the children of m_{i_j}. Since $m_{i'}$ is the second child of a fork clique we know be inductive assumption $h_l(S_{i'}^2) = h_{l-1}(S_{i'}^2) = [\texttt{count}(m_{i'}) - \texttt{inparent}(m_{i'}) + \texttt{in2children}(m_{i_j}) + 1,\ \texttt{count}(m_{i'})]$. Further, m_i is neither a fork clique nor the second child of a fork clique. Thus, $h_l(S_i^1) = h_{l-1}(S_i^1) = [\texttt{count}(m_i) - \texttt{inparent}(m_i) + 1,\ \texttt{count}(m_i)]$. The set S_i^2 contains exactly the vertices $v \in S_i^1$ with $\#\mathrm{anc}_{A_i}(v) \neq 2$. Consequently, property 2 yields

$h_l(S_i^2) = [\texttt{count}(m_i) - \texttt{inparent}(m_i) + \texttt{in2children}(m_{i_j}) + 1,\ \texttt{count}(m_i)]$. Clearly, since A_{i_j} is a fork clique, $h_l(A_l \cap A_{i_j}) = h_l(S_{i_j}^1)$ is the disjoint union of $h_l(S_i^2)$ and $h_l(S_{i'}^2)$. Now let m_l correspond to the second child of a fork clique, and let us consider m_{i_0}, the sibling of m_l. The node m_{i_0} is neither a fork clique nor the second child of a fork clique. Thus, we have $h_l(S_{i_0}^1) = h_{l-1}(S_{i_0}^1) = [\texttt{count}(m_{i_0}) - \texttt{inparent}(m_{i_0}) + 1,\ \texttt{count}(m_{i_0})]$. Since the set $A_l \cap A_{i_0}$ contains precisely the vertices $v \in S_{i_0}^1$ with $\#\text{anc}_{A_{i_0}}(v) = 2$, that is, the vertices in the parent A_{l+1} of A_{i_0} and A_l that are contained in both of A_{l+1}'s children max cliques, property 2 implies $h_l(A_l \cap A_{i_0}) = [\texttt{count}(m_{i_0}) - \texttt{inparent}(m_{i_0}) + 1,\ \texttt{count}(m_{i_0}) - \texttt{inparent}(m_{i_0}) + \texttt{in2children}(m_{l+1})]$, where m_{l+1} is the parent node of m_l and m_{i_0}. Finally, by definition of h_l we know $h_l(A_l') = [\texttt{count}(m_l) - \texttt{in0children}(m_l) + 1,\ \texttt{count}(m_l)]$ if m_l does not correspond to the second child of a fork clique, and $h_l(A_l') = [\texttt{count}(m_l) - \texttt{in0children}(m_l) + \texttt{in2children}(m_{l+1}) + 1,\ \texttt{count}(m_l)]$ otherwise. Thus, we have shown that the disjoint union of $h_l(A_l')$ and the sets $h_l(A_l \cap A_{i_j})$ for all feasible $j \leq k$ is exactly the set B_{m_l}. Hence $h_l(A_l) = B_{m_l}$.

We prove the remaining properties separately for star cliques and for max cliques that are neither star nor fork cliques. We first consider the case where A_l is a star clique. Let us show property 2. We have to prove that $\#\text{anc}_{A_l}(v) \leq \#\text{anc}_{A_l}(v')$ for vertices $v, v' \in A_l$ with $h_l(v) \leq h_l(v')$ if A_l is neither a fork clique nor the second child of a fork clique. Thus, let A_l be a star clique that is not the second child of a fork clique. Let $A_{i_1}, \dots, A_{i_k}$ with $i_1 < \dots, i_k$ be the children of A_l. As shown above A_l is the disjoint union of of A_l' and $A_l \cap A_{i_j}$ for all $j \in [k]$. As A_l is a star clique we know $\#\text{anc}_{A_l}(v) = 1$ for all $v \in A_l \cap A_{i_j}$ for $j \in [k]$. Now let us consider $v, v' \in A_l$ with $h_l(v) \leq h_l(v')$. If $v \in A_l \setminus A_l'$ and $v' \in A_l$, we have $\#\text{anc}_{A_l}(v) = 1$ and therefore $\#\text{anc}_{A_l}(v) \leq \#\text{anc}_{A_l}(v')$. It remains to consider the case where $v \in A_l'$. Since $h_l(v) \leq h_l(v')$ and each number in $h(A_l')$ is greater than every number in $h(A_l \setminus A_l')$, we also have $v' \in A_l'$. Then $\#\text{anc}_{A_l}(v) \leq \#\text{anc}_{A_l}(v')$ follows directly from the construction of h_l. To show property 3 we let A_l again be a star clique that is not the second child of a fork clique. We have already seen that $\#\text{anc}_{A_l}(v) = 1$ for all $v \in A_l \setminus A_l'$. Therefore, we have $S_l^1 \subseteq A_l'$. Now $h_l(S_l^1) = [\texttt{count}(m_l) - \texttt{inparent}(m_l) + 1,\ \texttt{count}(m_l)]$ follows directly from property 2. It remains to show property 4. This time, let A_l be a star clique that is the second child of a fork clique A_{l+1}. According to Lemma 373, all vertices in A_l are either contained in a child max clique of A_l, in its sibling max clique, or in A_l'. We know $\#\text{anc}_{A_l}(v) = 1$ for all $v \in A_l$ that are also contained in a child of A_l, and $\#\text{anc}_{A_l}(v) = 2$ for $v \in A_l$ if and only if v is also contained in the sibling max clique of A_l. Consequently, S_l^2 must be a subset of A_l', and property 2 yields $h_l(S_l^2) = [\texttt{count}(m_l) - \texttt{inparent}(m_l) + \texttt{in2children}(m_{l+1}) + 1,\ \texttt{count}(m_l)]$.

Now let us consider max cliques A_l that are neither fork nor star cliques. Then A_l cannot be the parent or a child of a fork clique as children of fork cliques are star cliques according to Lemma 364. Further, A_l must have precisely one child and a parent, since A_l has at most one child by Corollary 369 and max cliques of degree 1 are trivially star cliques. To show property 2 let us consider $v, v' \in A_l$ with $h_l(v) \leq h_l(v')$. The child A_{l-1} of A_l is neither a fork clique nor the second child of a fork clique. Thus, according to the inductive assumption we have $\#\text{anc}_{A_{l-1}}(v) \leq \#\text{anc}_{A_{l-1}}(v')$ for $v, v' \in A_{l-1}$. Further, if $v, v' \in A_l' = A_l \setminus A_{l-1}$, then $\#\text{anc}_{A_l}(v) \leq \#\text{anc}_{A_l}(v')$ follows directly from the construction of h_l. Since every number in $h(A_l')$ is greater than each number in $h(A_l \setminus A_l')$, it remains to consider v, v' with $v \in A_l \setminus A_l'$ and $v' \in A_l'$. Let us assume $\#\text{anc}(v)_{A_l} > \#\text{anc}_{A_l}(v')$ for such v and v'. Then $\mathcal{M}_{v'}$ is a separator of the path induced by $\mathcal{M}_v$ in the clique tree of G, which is a contradiction to Lemma 356. Thus, $\#\mathrm{anc}(v)_{A_l} \leq \#\mathrm{anc}_{A_l}(v')$ for all $v, v' \in A_l$ with $h_l(v) \leq h_l(v')$. Next, let us show property 3. We know that $S^1_{l-1} = A_l \cap A_{l-1}$. As A_{l-1} is neither a fork clique nor the second child of a fork clique, we have $h_l(S^1_{l-1}) = [\texttt{count}(m_{l-1}) - \texttt{inparent}(m_{l-1}) + 1,\ \texttt{count}(m_{l-1})]$ by inductive assumption. Further, the set $h_l(A'_l)$ is precisely the interval $[\texttt{count}(m_{l-1})+1,\ \texttt{count}(m_l)]$. Hence, $h_l(A_l)$ is the interval $[\texttt{count}(m_{l-1}) - \texttt{inparent}(m_{l-1}) + 1,\ \texttt{count}(m_l)]$, and property 3 follows directly from property 2. Finally, property 4 holds trivially since A_l cannot be the second child of a fork clique. □

For $l = |\mathcal{M}|$ bijection h_l is a bijection between V and $|V|$ such that $h(A_i) = B_{m_i}$ for all $i \in [|\mathcal{M}|]$. Thus, the above claim proves the existence of a graph isomorphism h between G and graph K.

Corollary 376. *Graph K is an isomorphic copy of G on the number sort.*

The Algorithm needs Logarithmic Space

It remains to analyze the data complexity of the algorithm. During the depth-first traversal, we need to remember the current node, the last move and `count`. As we want to visit the vertices in post-order, we also compute the next move at each node. If it is not **down**, then we visit the current node for the last time and it belongs to the post-order traversal sequence. Clearly, post-order depth-first traversal is possible in logspace.

At each node m, we use the values of $\texttt{in0children}(m')$, $\texttt{inparent}(m')$, $\texttt{in2children}(m')$, $\texttt{isforkclique}(m')$, $\texttt{isforkchild2}(m')$ and $\texttt{count}(m')$ for the necessary nodes m' to distinguish between the different cases and to compute the partial intervals that form B_m.

We can easily determine the values $\texttt{in0children}(m')$, $\texttt{inparent}(m')$ and $\texttt{in2children}(m')$ for nodes m' of the tree in logspace. We obtain these values from the color of m'. Further, we can use the value of $\texttt{in2children}(m')$ to determine in logspace whether a node m' corresponds to a fork clique, that is, whether $\texttt{isforkclique}(m') = \text{true}$, because only fork cliques A have the property that there exists vertices in A that occur in (at least) two child max cliques of A. Additionally, we have to compute $\texttt{isforkchild2}(m')$. Clearly, this is possible in logarithmic space as well by deciding whether m' is the second child of a node corresponding to a fork clique. We do not need to remember any of the above values. We can recompute them whenever we need them.

To compute the partial intervals that form B_m, we may also have to recompute $\texttt{count}(m')$ for a certain nodes m' of the tree. The recomputation is possible in logarithmic space: After visiting a node m'', `count` stays the same, if $\texttt{isforkclique}(m'')$ is true. If $\texttt{isforkclique}(m'')$ is false, then depending on the value of $\texttt{isforkchild2}(m'')$, `count` is increased by $\texttt{in0children}(m'')$ or by $\texttt{in0children}(m'') - \texttt{in2children}(p'')$ where p is the parent node of m''. Thus, we can easily recompute $\texttt{count}(m')$ by a new post-order traversal.

13. Conclusion

In this thesis, we have proved various new capturing results for the complexity classes PTIME and LOGSPACE. We conclude this thesis with this work's implication for future research and open problems raised by our results.

In the first part of the thesis, we showed that FP+C captures PTIME on the class of permutation graphs as well as on the class of chordal comparability graphs. To this end, we stated and proved the Modular Decomposition Theorem. Of course, the question arises whether this theorem can be use to prove capturing results on further classes of graphs. It is also an open question whether a tool similar to the Modular Decomposition Theorem, can be obtained for generalizations of the modular decomposition, like the split decomposition (also called join decomposition). The class of circle graphs and the class of parity graphs (which both contain the class of distance-hereditary graphs), are both classes of graphs that are well-structured with respect to split decompositions. It would by interesting to find out whether a "Split Decomposition Theorem" can be used to prove that FP+C captures PTIME on any of these two graph classes.

In order to prove the Modular Decomposition Theorem, we showed that the modular decomposition of a graph is definable in STC. As a side result, we proved that there exists a logarithmic-space algorithm that computes the modular decomposition tree of a graph. This is of particular interest for the design of logarithmic-space algorithms in the context of algorithmic graph theory, where modular decompositions found a variety of applications.

In the second part of this thesis, we introduce the new logics LREC and $\mathsf{LREC}_=$ which extend first-order logic with counting by a recursion operator that can be evaluated in logarithmic space. By capturing LOGSPACE on trees, interval graphs and chordal claw-free graphs, we obtain the first logical characterizations of LOGSPACE on non-trivial natural classes of unordered structures. It would be interesting to extend our results to further classes of structures such as the class of planar graphs or classes of graphs of bounded tree width. The author conjectures that $\mathsf{LREC}_=$ captures LOGSPACE on the class of all planar graphs equipped with an embedding.

The expressive power of $\mathsf{LREC}_=$ is not yet well-understood. For example, it is an open question whether directed graph reachability is expressible in $\mathsf{LREC}_=$, and even whether $\mathsf{LREC}_=$ has the same expressive power as FP+C. The fact that directed graph reachability is complete for NL indicates that the answer to both questions is negative.

It is obvious that our capturing results can be transferred to non-deterministic logarithmic space (NL) by adding a transitive closure operator to $\mathsf{LREC}_=$. However, a natural "non-deterministic" variant of our limited recursion operator that allows directed graph reachability to be expressed would yield a logic that contains TC and contributes more to the question of what constitutes NL. We leave it as an open problem to find such an operator.

A. Appendix

A.1. Proofs of Properties of Transductions

This section contains the proof of the Transduction Lemma for fixed-point logic with counting (Proposition 11) and for the logic $\mathsf{LREC}_=$ (Proposition 326). Further it contains the proof of Proposition 12, which states that the composition of two transductions is again a transduction.

A.1.1. Proof of the Transduction Lemma

In the following we prove the Transduction Lemma introduced in Section 2.5 for fixed-point logic with counting and for the logic $\mathsf{LREC}_=$ introduced in Chapter 10.

Transduction Lemma for FP+C

Before we prove the Transduction Lemma for fixed-point logic with counting, we repeat it and use this repetition to described the variable tuples and the assigned values in more detail.

Proposition 11 [Transduction Lemma] (repeated). *Let τ_1, τ_2 be vocabularies, let*

$$\Theta(\bar{X}) = \Big(\theta_{dom}(\bar{X}), \theta_U(\bar{X}, \bar{u}), \theta_\approx(\bar{X}, \bar{u}, \bar{u}'), \big(\theta_R(\bar{X}, \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)})\big)_{R\in\tau_2}\Big)$$

be a parametrized $\mathsf{FP{+}C}[\tau_1, \tau_2]$-transduction, and let $\phi(x_1, \ldots, x_\kappa, p_1, \ldots, p_\lambda, Y_1, \ldots, Y_\mu)$ be an $\mathsf{FP{+}C}[\tau_2]$-formula where $x_1, \ldots, x_\kappa$ are structure variables, $p_1, \ldots, p_\lambda$ number variables and $Y_1, \ldots, Y_\mu$ relational variables. Let ℓ be the length of $\bar{u}$. Then there exists an $\mathsf{FP{+}C}[\tau_1]$-formula $\phi^{-\Theta}(\bar{X}, \bar{u}_1, \ldots, \bar{u}_\kappa, \bar{q}_1, \ldots, \bar{q}_\lambda, Z_1, \ldots, Z_\mu)$, where $\bar{u}_1, \ldots, \bar{u}_\kappa$ are compatible with $\bar{u}$, $\bar{q}_1, \ldots, \bar{q}_\lambda$ are ℓ-tuples of number variables and for each $m \in [\mu]$ variable Z_m is a relational variable of type $\mathrm{t}(Y_m^{\ltimes\bar{u}})$, such that for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$, all $\bar{a}_1, \ldots, \bar{a}_\kappa \in A^{\bar{u}}$, all $\bar{n}_1, \ldots, \bar{n}_\lambda \in N(A)^\ell$ and all $S_m \in A^{Z_m}$ with $m \in [\mu]$,

$$\begin{aligned}
A \models \phi^{-\Theta}&[\bar{P}, \bar{a}_1, \ldots, \bar{a}_\kappa, \bar{n}_1, \ldots, \bar{n}_\lambda, S_1, \ldots, S_\mu]\\
\iff\ & \bar{a}_1/_\approx, \ldots, \bar{a}_\kappa/_\approx \in U(\Theta[A, \bar{P}]),\\
& \langle\bar{n}_1\rangle_A, \ldots, \langle\bar{n}_\lambda\rangle_A \in N(\Theta[A, \bar{P}]),\\
& \langle S_m\rangle_{A,\approx}^{Y_m} \in \Theta[A, \bar{P}]^{Y_m} \text{ for all } m \in [\mu] \text{ and}\\
& \Theta[A, \bar{P}] \models \phi\Big[\bar{a}_1/_\approx, \ldots, \bar{a}_\kappa/_\approx, \langle\bar{n}_1\rangle_A, \ldots, \langle\bar{n}_\lambda\rangle_A, \langle S_1\rangle_{A,\approx}^{Y_1}, \ldots, \langle S_\mu\rangle_{A,\approx}^{Y_\mu}\Big],
\end{aligned}$$

where $\approx$ is the equivalence relation as defined in Definition 7.

Proof. The proof uses induction on the structure of ϕ. We assume that all structure variables, number variables and relational variables occurring in ϕ are among $x_1, \dots, x_\kappa$, $p_1, \dots, p_\lambda$ and $Y_1, \dots, Y_\mu$, respectively. To simplify the presentation, we consider a fixed pair $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$ and let $\approx$ be the equivalence relation as defined in Definition 7. We also consider fixed $\bar{a}_1, \dots, \bar{a}_\kappa \in A^{\bar{u}}$, $\bar{n}_1, \dots, \bar{n}_\lambda \in N(A)^\ell$ and $S_m \in A^{Z_m}$ for $m \in [\mu]$. The reader should consider $(A, \bar{P})$ and these tuples to be universally quantified in the statements where they occur.

Using $\theta_\approx(\bar{X}, \bar{u}, \bar{u}')$, it is easy to construct an IFP+C-formula $\theta'_\approx(\bar{X}, \bar{u}, \bar{u}')$ such that $\theta'_\approx[A, \bar{P}; \bar{u}, \bar{u}']$ is the equivalence relation generated by relation $\theta_\approx[A, \bar{P}; \bar{u}, \bar{u}']$. Let $\chi_\mathrm{s}(\bar{X}, \bar{u}) := \exists \bar{u}'(\theta_U(\bar{X}, \bar{u}') \wedge \theta'_\approx(\bar{X}, \bar{u}', \bar{u}))$. Then for all $\bar{a} \in A^{\bar{u}}$,

$$A \models \chi_\mathrm{s}[\bar{P}, \bar{a}] \iff \bar{a}/_\approx \in U(\Theta[A, \bar{P}]).$$

The construction from the proof of Lemma 2.4.3 in [50] shows how to count definable equivalence classes in DTC+C. Using this construction, we can construct an IFP+C-formula $\delta_U^\#(\bar{X}, \bar{q})$ such that for all $\bar{n} \in N(A)^\ell$ we have $A \models \delta_U^\#[\bar{P}, \bar{n}]$ whenever $\langle \bar{n} \rangle_A = |U(\Theta[A, \bar{P}])|$. We let $\chi_\mathrm{n}(\bar{X}, \bar{q}) := \exists \bar{q}' \big(\delta_U^\#(\bar{X}, \bar{q}') \wedge \text{“}\bar{q} \leq \bar{q}'\text{”}\big)$. Then for all $\bar{n} \in N(A)^\ell$, we have

$$A \models \chi_\mathrm{n}[\bar{P}, \bar{n}] \iff \langle \bar{n} \rangle_A \in N(\Theta[A, \bar{P}]).$$

For $m \in [\mu]$ let r_m be the arity of relational variable Y_m, and let $\mathrm{t}(Y_m) = (t_1, \dots, t_{r_m})$ be its type. Let $\mathrm{s}(Y_m) := \{i \in [r_m] \mid t_i = \mathrm{s}\}$ and $\mathrm{n}(Y_m) := \{i \in [r_m] \mid t_i = \mathrm{n}\}$. Further, let $(\bar{v}_1, \dots, \bar{v}_{r_m})$ be a tuple of variables that has the same type as Z_m. We let

$$\chi_m(\bar{X}, Z_m) := \forall \bar{v}_1 \dots \forall \bar{v}_{r_m} \left(Z_m(\bar{v}_1, \dots, \bar{v}_{r_m}) \rightarrow \left(\bigwedge_{i \in \mathrm{s}(Y_m)} \chi_\mathrm{s}(\bar{X}, \bar{v}_i) \ \wedge \bigwedge_{i \in \mathrm{n}(Y_m)} \chi_\mathrm{n}(\bar{X}, \bar{v}_i) \right) \right).$$

Then for all $S_m \in A^{Z_m}$,

$$A \models \chi_m[\bar{P}, S_m] \iff \langle S_m \rangle_{A,\approx}^{Y_m} \in \Theta[A, \bar{P}]^{Y_m}.$$

Finally, let

$$\chi := \bigwedge_{k \in [\kappa]} \chi_\mathrm{s}(\bar{X}, \bar{u}_k) \ \wedge \bigwedge_{l \in [\lambda]} \chi_\mathrm{n}(\bar{X}, \bar{q}_l) \ \wedge \bigwedge_{m \in [\mu]} \chi_m(\bar{X}, Z_m).$$

Then,

$$\begin{aligned} A \models \chi[\bar{P}, \bar{a}_1, \dots, \bar{a}_\kappa, \bar{n}_1, \dots, \bar{n}_\lambda, S_1, \dots, S_\mu] \iff\ & \bar{a}_1/_\approx, \dots, \bar{a}_\kappa/_\approx \in U(\Theta[A, \bar{P}]), \\ & \langle \bar{n}_1 \rangle_A, \dots, \langle \bar{n}_\lambda \rangle_A \in N(\Theta[A, \bar{P}]) \text{ and} \\ & \langle S_m \rangle_{A,\approx}^{Y_m} \in \Theta[A, \bar{P}]^{Y_m} \text{ for all } m \in [\mu]. \end{aligned}$$

Given τ_2-formula $\phi(x_1, \dots, x_\kappa, p_1, \dots, p_\lambda, Y_1, \dots, Y_\mu)$ we now construct a τ_1-formula $\phi^{-\Theta}(\bar{X}, \bar{u}_1, \dots, \bar{u}_\kappa, \bar{q}_1, \dots, \bar{q}_\lambda, Z_1, \dots, Z_\mu)$ inductively as follows:

1. Suppose that $\phi = R(x_{k_1}, \dots, x_{k_i})$, where $k_1, \dots, k_i \in [\kappa]$. Let $I := \{k_1, \dots, k_i\}$. Then,

$$\phi^{-\Theta} := \chi \wedge (\exists \bar{u}'_k)_{k \in I} \left(\bigwedge_{k \in I} \Big(\theta'_\approx(\bar{X}, \bar{u}_k, \bar{u}'_k) \wedge \theta_U(\bar{X}, \bar{u}'_k) \Big) \wedge \theta_R(\bar{X}, \bar{u}'_{k_1}, \dots, \bar{u}'_{k_i}) \right).$$

2. Assume $\phi = Y_m(x_{k_1}, \dots, x_{k_i}, p_{l_1}, \dots, p_{l_j})$ with $k_1, \dots, k_i \in [\kappa]$, $l_1, \dots, l_j \in [\lambda]$ and $m \in [\mu]$. Let $I := \{k_1, \dots, k_i\}$. Then,

$$\phi^{-\Theta} := \chi \wedge (\exists \bar{u}'_k)_{k \in I} \left(\bigwedge_{k \in I} \theta'_{\approx}(\bar{X}, \bar{u}_k, \bar{u}'_k) \wedge Z_m(\bar{u}'_{k_1}, \dots, \bar{u}'_{k_i}, \bar{q}_{l_1}, \dots, \bar{q}_{l_j}) \right).$$

3. If $\phi = x_k = x_{k'}$, where $k, k' \in [\kappa]$, then

$$\phi^{-\Theta} := \chi \wedge \theta'_{\approx}(\bar{X}, \bar{u}_k, \bar{u}_{k'}).$$

4. If $\phi = p_l \star p_{l'}$, where $\star \in \{=, \leq\}$ and $l, l' \in [\lambda]$, then

$$\phi^{-\Theta} := \chi \wedge \text{“}\bar{q}_l \star \bar{q}_{l'}\text{”}.$$

5. If $\phi = \neg\psi$, then

$$\phi^{-\Theta} := \chi \wedge \neg\psi^{-\Theta}.$$

6. If $\phi = \psi_1 \star \psi_2$, where $\star \in \{\wedge, \vee\}$, then

$$\phi^{-\Theta} := \psi_1^{-\Theta} \star \psi_2^{-\Theta}.$$

7. Suppose that $\phi = Qu\,\psi$ with $Q \in \{\forall, \exists\}$ and $u \in \{x_1, \dots, x_\kappa, p_1, \dots, p_\lambda\}$. In case that $Q = \forall$ and $u = x_k$, we let

$$\phi^{-\Theta} := \chi \wedge \forall \bar{u}_k \left(\chi_{\mathrm{s}}(\bar{X}, \bar{u}_k) \to \psi^{-\Theta}\right).$$

   The other cases can be dealt with similarly.
8. Assume $\phi = \#(x_{k_1}, \dots, x_{k_i}, p_{l_{j+1}}, \dots, p_{l_{j+j'}})\,\psi = (p_{l_1}, \dots, p_{l_j})$ where $k_1, \dots, k_i \in [\kappa]$ and $l_1, \dots, l_{j+j'} \in [\lambda]$. Based on the construction from the proof of Lemma 2.4.3 in [50], it is possible to construct an $\mathsf{IFP{+}C}[\tau_1]$-formula

$$\delta(\bar{X}, \bar{u}_1, \dots, \bar{u}_\kappa, \bar{q}_1, \dots, \bar{q}_\lambda, Z_1, \dots, Z_\mu, \bar{q}_{l_1}, \dots, \bar{q}_{l_j})$$

   such that for all $\bar{a}_1, \dots, \bar{a}_\kappa \in A^{\bar{u}}$, all $\bar{n}_1, \dots, \bar{n}_\lambda \in N(A)^\ell$, all $S_m \subseteq A^{Z_i}$ with $m \in [\mu]$ and all $\bar{m}_{l_1}, \dots, \bar{m}_{l_j} \in N(A)^\ell$ where $\bar{m}_{l_h} = \bar{n}_l$ if $\bar{q}_{l_h} = \bar{q}_l$, for all $h \in [j], l \in [\lambda]$,

$$\begin{aligned} A \models\ & \delta[\bar{P}, \bar{a}_1, \dots, \bar{a}_\kappa, \bar{n}_1, \dots, \bar{n}_\lambda, S_1, \dots, S_\mu, \bar{m}_{l_1}, \dots, \bar{m}_{l_j}] \\ & \iff \Big|\Big\{ \big(\bar{a}_{k_1}/_{\approx}, \dots, \bar{a}_{k_i}/_{\approx}, \langle \bar{n}_{l_{j+1}} \rangle_A, \dots, \langle \bar{n}_{l_{j+j'}} \rangle_A \big) \big| A \models \\ & \qquad \psi^{-\Theta}[\bar{P}, \bar{a}_1, \dots, \bar{a}_\kappa, \bar{n}_1, \dots, \bar{n}_\lambda, S_1, \dots, S_\mu] \Big\}\Big| \\ & \qquad = \langle \bar{m}_{l_{j+1}}, \dots, \bar{m}_{l_{j+j'}} \rangle_{A, \Theta[A, \bar{P}]}, \end{aligned}$$

   where

$$\langle \bar{m}_1, \dots, \bar{m}_{l'} \rangle_{A, \Theta[A, \bar{P}]} := \sum_{s=1}^{l'} \langle \bar{m}_s \rangle_A \cdot \left| N(\Theta[A, \bar{P}]) \right|^{s-1}.$$

   We let

$$\phi^{-\Theta} := \chi \wedge \delta.$$

9. Suppose that

$$\phi = \mathrm{ifp}\Big(Y_m(x_{k_1}, \ldots, x_{k_i}, p_{l_1}, \ldots, p_{l_j}) \leftarrow \psi\Big)(x_{k'_1}, \ldots, x_{k'_i}, p_{l'_1}, \ldots, p_{l'_j})$$

where $k_1, \ldots, k_i, k'_1, \ldots, k'_i \in [\kappa]$, $l_1, \ldots, l_j, l'_1, \ldots, l'_j \in [\lambda]$ and $m \in [\mu]$. Let index set $I := \{k_1, \ldots, k_i\}$. Then,

$$\phi^{-\Theta} \ := \ \chi \wedge \mathrm{ifp}\Big(Z_m(\bar{u}_{k_1}, \ldots, \bar{u}_{k_i}, \bar{q}_{l_1}, \ldots, \bar{q}_{l_j}) \leftarrow \psi^{-\Theta}\Big)(\bar{u}_{k'_1}, \ldots, \bar{u}_{k'_i}, \bar{q}_{l'_1}, \ldots, \bar{q}_{l'_j}).$$

It is straightforward, though tedious, to verify that $\phi^{-\Theta}$ is as desired. □

Transduction Lemma for $\mathsf{LREC}_=$

In the following we prove the Transduction Lemma for the logic $\mathsf{LREC}_=$ (Proposition 326).

Proof. In order to prove the Transduction Lemma for $\mathsf{LREC}_=$, it suffices to adjust the proof of the Transduction Lemma for $\mathsf{FP{+}C}$ (Proposition 11) from Section A.1.1. $\mathsf{LREC}_=$-formulas do not contain relational variables and fixed-point operators. Thus, we do not have to consider formulas of the form 2 and 9. Instead we consider the following case:

10. Suppose that

$$\varphi = [\mathrm{lrec}_{\bar{v}_1, \bar{v}_2, \bar{c}}\ \varphi_=,\ \varphi_{\mathtt{E}},\ \varphi_{\mathtt{C}}](\bar{v}_3, \bar{r})$$

where $\bar{c} = (p_{c_1}, \ldots, p_{c_m})$, $\bar{r} = (p_{r_1}, \ldots, p_{r_d})$ and $\bar{v}_s = (x_{k_1^s}, \ldots, x_{k_i^s}, p_{l_1^s}, \ldots, p_{l_j^s})$ for all $s \in [3]$. Then,

$$\varphi^{-\Theta} \ := \ \chi \wedge \exists \bar{r}''\Big([\mathrm{lrec}_{\bar{v}'_1, \bar{v}'_2, \bar{c}''}\ \varphi'_=,\ \varphi_{\mathtt{E}}^{-\Theta},\ \varphi'_{\mathtt{C}}](\bar{v}'_3, \bar{r}'') \ \wedge \ \beta(\bar{r}', \bar{r}'')\Big),$$

where

$$\varphi'_= := \varphi_=^{-\Theta} \vee \bigvee_{h \in [i]} \theta'_{\approx}(\bar{X}, \bar{u}_{k_h^1}, \bar{u}_{k_h^2}),$$
$$\varphi'_{\mathtt{C}} := \exists \bar{c}'(\varphi_{\mathtt{C}}^{-\Theta} \wedge \beta(\bar{c}', \bar{c}''))$$

and $\bar{c}' = (\bar{q}_{c_1}, \ldots, \bar{q}_{c_m})$, $\bar{r}' = (\bar{q}_{r_1}, \ldots, \bar{q}_{r_d})$ and $\bar{v}'_s = (\bar{u}_{k_1^s}, \ldots, \bar{u}_{k_i^s}, \bar{q}_{l_1^s}, \ldots, \bar{q}_{l_j^s})$ for all $s \in [3]$. Further, $\bar{r}''$ and $\bar{c}''$ are tuples of number variables of length $|\bar{r}'|$ and $|\bar{c}'|$, respectively, such that the number variables in $\bar{r}''$ and $\bar{c}''$ do not occur in $\bar{q}_1, \ldots, \bar{q}_\lambda$; and β is defined as follows. The formula $\beta(\bar{r}', \bar{r}'')$ has the property that for all $\bar{m} \in N(A)^{|\bar{r}''|}$,

$$A \models \beta[\bar{n}_{r_1}, \ldots, \bar{n}_{r_d}, \bar{m}] \iff \langle \bar{m} \rangle_A = \sum_{h=1}^{d} \langle \bar{n}_{r_h} \rangle_A \cdot |N(\Theta[A])|^{h-1}.$$

Formula $\beta(\bar{c}', \bar{c}'')$ is defined analogously. Constructing β as desired is a not too difficult exercise. □

A.1.2. Proof of Proposition 12 (Composition of Transductions)

In the following we prove the subsequent proposition from Section 2.5.3. Notice, that the tuples of domain variables are named differently.

Proposition 12 (repeated). *Let τ_1, τ_2 and τ_3 be vocabularies. Let $\Theta_1(\bar{X}_1)$ be a parameterized $\mathsf{L}[\tau_1, \tau_2]$-transduction, and let $\Theta_2(\bar{Y})$ be a parameterized $\mathsf{L}[\tau_2, \tau_3]$-transduction where $\bar{v}$ and $\bar{w}$ are the respective tuples of domain variables. Then there exists a parameterized $\mathsf{L}[\tau_1, \tau_3]$-transduction $\Theta(\bar{X})$ with $\bar{X} = (\bar{X}_1, \bar{X}_2)$ such that $\bar{X}_2 = Y^{\ltimes \bar{v}}$, tuple $\bar{w}^{\ltimes \bar{v}}$ is the tuple of domain variables, and for all τ_1-structures A and all $\bar{P} \in A^{\bar{X}}$ with $\bar{P} = (\bar{P}_1, \bar{P}_2)$,*

$$\begin{aligned}(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X})) \iff\ & (A, \bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)), \\ & \bar{Q} := \langle \bar{P}_2 \rangle^{\bar{Y}}_{A, \approx_1} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}} \ \textit{and} \\ & (\Theta_1[A, \bar{P}_1], \bar{Q}) \in \mathrm{Dom}(\Theta_2(\bar{Y})),\end{aligned}$$

where $\approx_1$ is the equivalence relation of $(A, \bar{P}_1)$ under Θ_1, and for all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$,

$$\Theta[A, \bar{P}] \cong \Theta_2[\Theta_1[A, \bar{P}_1], \bar{Q}].$$

Proof. We can construct $\mathsf{L}[\tau_1, \tau_3]$-transduction

$$\Theta(\bar{X}) = \big(\theta_{dom}(\bar{X}), \theta_U(\bar{X}, \bar{u}), \theta_{\approx}(\bar{X}, \bar{u}, \bar{u}'), (\theta_R(\bar{X}, \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)}))_{R \in \tau_3}\big)$$

from $\mathsf{L}[\tau_1, \tau_2]$-transduction $\Theta_1(\bar{X}_1)$ and $\mathsf{L}[\tau_2, \tau_3]$-transduction $\Theta_2(\bar{Y})$,

$$\begin{aligned}\Theta_1(\bar{X}_1) &= \big(\theta_{1dom}(\bar{X}_1), \theta_{1U}(\bar{X}_1, \bar{v}), \theta_{1\approx}(\bar{X}_1, \bar{v}, \bar{v}'), (\theta_{1S}(\bar{X}_1, \bar{v}_{S,1}, \ldots, \bar{v}_{S,\mathrm{ar}(S)}))_{S \in \tau_2}\big) \quad \text{and} \\ \Theta_2(\bar{Y}) &= \big(\theta_{2dom}(\bar{Y}), \theta_{2U}(\bar{Y}, \bar{w}), \theta_{2\approx}(\bar{Y}, \bar{w}, \bar{w}'), (\theta_{2R}(\bar{Y}, \bar{w}_{R,1}, \ldots, \bar{w}_{R,\mathrm{ar}(R)}))_{R \in \tau_3}\big),\end{aligned}$$

by using the Transduction Lemma (Proposition 11) on the formulas of $\Theta_2(\bar{Y})$.

We let

$$\begin{aligned}\theta_{dom}(\bar{X}) &:= \theta_{1dom}(\bar{X}_1) \wedge \exists \bar{v}\, \theta_{1U}(\bar{X}_1, \bar{v}) \wedge \theta_{2\,\mathrm{dom}}^{-\Theta_1}(\bar{X}), \\ \theta_U(\bar{X}, \bar{u}) &:= \theta_{2\,U}^{-\Theta_1}(\bar{X}, \bar{u}), \\ \theta_{\approx}(\bar{X}, \bar{u}, \bar{u}') &:= \theta_{2\,\approx}^{-\Theta_1}(\bar{X}, \bar{u}, \bar{u}'), \quad \text{and} \\ \theta_R(\bar{X}, \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)}) &:= \theta_{2\,R}^{-\Theta_1}(\bar{X}, \bar{u}_{R,1}, \ldots, \bar{u}_{R,\mathrm{ar}(R)}) \quad \text{for all } R \in \tau_3.\end{aligned}$$

Further, we let $\approx_1$ be the equivalence relation of $(A, \bar{P}_1)$ under Θ_1.

Then for all τ_1-structures A and all $\bar{P} \in A^{\bar{X}}$ with $\bar{P} = (\bar{P}_1, \bar{P}_2)$, we have

$$\begin{aligned}
& (A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X})) \\
\iff & A \models \theta_{dom}[\bar{P}] \quad \text{and} \quad \theta_U[A, \bar{P}; \bar{u}] \neq \emptyset \\
\iff & A \models \theta_{1dom}[\bar{P}_1],\ \exists \bar{a} \in A^{\bar{v}} \colon A \models \theta_{1U}[\bar{P}_1, \bar{a}],\ A \models {\theta_2}_{\mathrm{dom}}^{-\Theta_1}[\bar{P}] \quad \text{and} \quad \theta_U[A, \bar{P}; \bar{u}] \neq \emptyset \\
\iff & A \models \theta_{1dom}[\bar{P}_1],\ \theta_{1U}[A, \bar{P}_1; \bar{v}] \neq \emptyset,\ A \models {\theta_2}_{\mathrm{dom}}^{-\Theta_1}[\bar{P}] \quad \text{and} \quad \exists \bar{a} \in A^{\bar{u}} \colon A \models \theta_U[\bar{P}, \bar{a}] \\
\iff & (A, \bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)),\ A \models {\theta_2}_{\mathrm{dom}}^{-\Theta_1}[\bar{P}] \quad \text{and} \quad \exists \bar{a} \in A^{\bar{u}} \colon A \models {\theta_2}_U^{-\Theta_1}[\bar{P}, \bar{a}] \\
\iff & (A, \bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)),\ \bar{Q} = \langle \bar{P}_2 \rangle_{A,\approx_1}^{\bar{Y}} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}},\ \Theta_1[A, \bar{P}_1] \models \theta_{2\mathrm{dom}}[\bar{Q}] \\
& \text{and} \quad \exists \bar{a} \in A^{\bar{u}} \colon \Big(\langle \bar{a} \rangle_{A,\approx_1}^{\bar{w}} \in \Theta_1[A, \bar{P}_1]^{\bar{w}} \text{ and } \Theta_1[A, \bar{P}_1] \models \theta_{2U}[\bar{Q}, \langle \bar{a} \rangle_{A,\approx_1}^{\bar{w}}] \Big) \\
\iff & (A, \bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)),\ \bar{Q} = \langle \bar{P}_2 \rangle_{A,\approx_1}^{\bar{Y}} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}},\ \Theta_1[A, \bar{P}_1] \models \theta_{2\mathrm{dom}}[\bar{Q}] \\
& \text{and} \quad \exists \bar{b} \in \Theta_1[A, \bar{P}_1]^{\bar{w}} \colon \Theta_1[A, \bar{P}_1] \models \theta_{2U}[\bar{Q}, \bar{b}] \\
\iff & (A, \bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)),\ \bar{Q} = \langle \bar{P}_2 \rangle_{A,\approx_1}^{\bar{Y}} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}},\ \Theta_1[A, \bar{P}_1] \models \theta_{2\mathrm{dom}}[\bar{Q}] \\
& \text{and} \quad \theta_{2U}[\Theta_1[A, \bar{P}_1], \bar{Q}; \bar{w}] \neq \emptyset \\
\iff & (A, \bar{P}_1) \in \mathrm{Dom}(\Theta_1(\bar{X}_1)),\ \bar{Q} = \langle \bar{P}_2 \rangle_{A,\approx_1}^{\bar{Y}} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}} \\
& \text{and} \quad (\Theta_1[A, \bar{P}_1], \bar{Q}) \in \mathrm{Dom}(\Theta_2(\bar{Y}))
\end{aligned}$$

For all $(A, \bar{P}) \in \mathrm{Dom}(\Theta(\bar{X}))$ and $\bar{a} \in A^{\bar{u}}$ we have

$$\begin{aligned}
& A \models \theta_U[\bar{P}, \bar{a}] \\
\iff & A \models {\theta_2}_U^{-\Theta_1}[\bar{P}, \bar{a}] \\
\iff & A \models {\theta_2}_U^{-\Theta_1}[\bar{P}_1, \bar{P}_2, \bar{a}] \\
\iff & \Theta_1[A, \bar{P}_1] \models \theta_{2U}[\bar{Q}, \bar{b}] \text{ where } \bar{Q} := \langle \bar{P}_2 \rangle_{A,\approx_1}^{\bar{Y}} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}} \\
& \qquad \text{and } \bar{b} := \langle \bar{a} \rangle_{A,\approx_1}^{\bar{w}} \in \Theta_1[A, \bar{P}_1]^{\bar{w}}
\end{aligned}$$

and

$$\begin{aligned}
& A \models \theta_{\approx}[\bar{P}, \bar{a}, \bar{a}'] \\
\iff & A \models {\theta_2}_{\approx}^{-\Theta_1}[\bar{P}, \bar{a}, \bar{a}'] \\
\iff & A \models {\theta_2}_{\approx}^{-\Theta_1}[\bar{P}_1, \bar{P}_2, \bar{a}, \bar{a}'] \\
\iff & \Theta_1[A, \bar{P}_1] \models \theta_{2\approx}[\bar{Q}, \bar{b}, \bar{b}'] \text{ where } \bar{Q} := \langle \bar{P}_2 \rangle_{A,\approx_1}^{\bar{Y}} \in \Theta_1[A, \bar{P}_1]^{\bar{Y}} \\
& \qquad \bar{b} := \langle \bar{a} \rangle_{A,\approx_1}^{\bar{w}} \in \Theta_1[A, \bar{P}_1]^{\bar{w}}, \\
& \qquad \text{and } \bar{b}' := \langle \bar{a}' \rangle_{A,\approx_1}^{\bar{w}} \in \Theta_1[A, \bar{P}_1]^{\bar{w}}.
\end{aligned}$$

Thus,

$$\bar{a} \in \theta_U[A, \bar{P}; \bar{u}] \iff \langle \bar{a} \rangle_{A,\approx_1}^{\bar{w}} \in \theta_{2U}[\Theta_1[A, \bar{P}_1], \bar{Q}; \bar{w}] \quad \text{and} \tag{A.1}$$

$$(\bar{a}, \bar{a}') \in \theta_{\approx}[A, \bar{P}; \bar{u}, \bar{u}'] \iff \Big(\langle \bar{a} \rangle_{A,\approx_1}^{\bar{w}}, \langle \bar{a}' \rangle_{A,\approx_1}^{\bar{w}} \Big) \in \theta_{2\approx}[\Theta_1[A, \bar{P}_1], \bar{Q}; \bar{w}, \bar{w}']. \tag{A.2}$$

Let $\approx$ be the equivalence relation generated by $\theta_{\approx}[A, \bar{P}; \bar{u}, \bar{u}']$ and $\approx_2$ be the equivalence relation generated by $\theta_{2\approx}[\Theta_1[A, \bar{P}_1], \bar{Q}; \bar{w}, \bar{w}']$. Then we also have

$$\bar{a} \approx \bar{a}' \iff \langle \bar{a} \rangle^{\bar{w}}_{A,\approx_1} \approx_2 \langle \bar{a}' \rangle^{\bar{w}}_{A,\approx_1}.$$

Now, it is not hard to see that mapping h which assigns each equivalence class $\bar{a}/_{\approx}$ to the equivalence class $\langle \bar{a} \rangle^{\bar{w}}_{A,\approx_1} /_{\approx_2}$ is a bijection between the universe $U(\Theta[A, \bar{P}]) = \theta_U[A, \bar{P}; \bar{u}]/_{\approx}$ of $\Theta[A, \bar{P}]$ and the universe $U(\Theta_2[\Theta_1[A, \bar{P}_1], \bar{Q}]) = \theta_{2U}[\Theta_1[A, \bar{P}_1], \bar{Q}; \bar{u}]/_{\approx_2}$ of $\Theta_2[\Theta_1[A, \bar{P}_1], \bar{Q}]$

Similarly to the equivalences (A.1) and (A.2), we obtain

$$\begin{aligned} &(\bar{a}_{R,1}, \dots, \bar{a}_{R,\mathrm{ar}(R)}) \in \theta_R\big[A, \bar{P}; \bar{u}_{R,1}, \dots, \bar{u}_{R,\mathrm{ar}(R)}\big] \\ \iff &\Big(\langle \bar{a}_{R,1} \rangle^{\bar{w}}_{A,\approx_1}, \dots, \langle \bar{a}_{R,\mathrm{ar}(R)} \rangle^{\bar{w}}_{A,\approx_1}\Big) \in \theta_{2R}\big[\Theta_1[A, \bar{P}_1], \bar{Q}; \bar{w}_{R,1}, \dots, \bar{w}_{R,\mathrm{ar}(R)}\big]. \end{aligned}$$

Therefore, mapping h is an isomorphism. □

A.2. Proof omitted in Section 9

In the following we prove Observation 316 from Section 9.1.

Observation 316 (repeated). *Let φ be an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula, p be a number variable and $\bar{u}$ be a non-empty tuple of individual variables with exactly k occurrences of structure variables. Then there exists an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula ψ of rank at most k such that*

$$(A, \alpha) \models \psi \iff \alpha(p) = |\{\bar{a} \in A^{\bar{u}} \mid (A, \alpha[\bar{a}/\bar{u}]) \models \varphi\}| < \infty.$$

Proof. Without loss of generality, let $\varphi(\bar{q}, \bar{x}, \bar{u}')$ be an $\mathcal{L}^*_{\infty\omega}(\mathbf{C})[\tau]$-formula and $\bar{u} = \bar{q}\bar{x}\bar{y}$ where

- $\bar{q}$ is an ℓ-tuple of number variables,
- $\bar{x}$ is a k_1-tuple of structure variables that occur free in φ,
- $\bar{y}$ is a k_2-tuple of structure variables that do not occur free in φ,
- $\bar{u}'$ is an enumeration of all free variables that are not listed in $\bar{u}$.

We have $k = k_1 + k_2$.

If $k_1 = 0$, let

$$\psi'(\bar{u}', p) := \bigvee_{m} \bigvee_{M \in \binom{\mathbb{N}^\ell}{m}} \Bigg(\bigwedge_{\bar{i} \in M} \varphi(\bar{i}, \bar{u}') \wedge \bigwedge_{\bar{i} \notin M} \neg\varphi(\bar{i}, \bar{u}') \wedge p = m \Bigg).$$

If $k_1 > 0$, let

$$\psi'(\bar{u}', p) := \Big(\varphi_{\mathrm{sum}}(\bar{u}', p) \wedge \forall p' \Big(\varphi_{\mathrm{sum}}(\bar{u}', p') \to p' \leq p\Big)\Big),$$

where

$$\varphi_{\text{sum}}(\bar{u}', p) := \bigvee_{n>0} \, (\exists p_{\bar{i}})_{\bar{i}\in[0,n]^\ell} \left(\bigwedge_{\bar{i}\in[0,n]^\ell} \#\bar{x}\, \varphi(\bar{i}, \bar{x}, \bar{u}') = p_{\bar{i}} \;\wedge \sum_{\bar{i}\in[0,n]^\ell} p_{\bar{i}} \, = \, p \right).$$

If $k_2 = 0$, then

$$\psi(\bar{u}', p) := \psi'(\bar{u}', p)$$

If $k_2 > 0$, then

$$\psi(\bar{u}', p) := \exists r \exists p' \Big(\#x \, (x = x) = r \;\wedge\; \psi'(\bar{u}', p') \;\wedge\; p = p' \cdot r^{k_2} \Big). \qquad \square$$

Bibliography

[1] M. Beaudry and P. McKenzie. Circuits, matrices, and nonassociative computation. *Journal of Computer and System Sciences*, 50(3):441–455, 1995.

[2] J. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In A. George, J. Gilbert, and J. Liu, editors, *Graph Theory and Sparse Matrix Computation*, volume 56 of *The IMA Volumes in Mathematics and its Applications*, pages 1–29. Springer New York, 1993.

[3] A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals Pure and Applied Logic*, 100(1-3):141–187, 1999.

[4] A. Brandstädt, V. Le, and J. Spinrad. *Graph Classes: A Survey.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[5] P. Buneman. A characterisation of rigid circuit graphs. *Discrete Mathematics*, 9:205–212, 1974.

[6] S. Buss, S. Cook, A. Gupta, and V. Ramachandran. An optimal parallel algorithm for formula evaluation. *SIAM Journal on Computing*, 21:755–780, 1992.

[7] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.

[8] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[9] M. Chudnovsky and P. Seymour. The structure of claw-free graphs. In *Surveys in Combinatorics, 2005 [invited lectures from the Twentieth British Combinatorial Conference, Durham, UK, July 2005]*, pages 153–171, 2005.

[10] D. Corneil, Y. Perl, and L. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.

[11] A. Cournier and M. Habib. *Trees in Algebra and Programming — CAAP'94: 19th International Colloquium Edinburgh, U.K., April 11–13, 1994 Proceedings*, chapter A new linear algorithm for Modular Decomposition, pages 68–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[12] A. Curtis, C. Izurieta, B. Joeris, S. Lundberg, and R. McConnell. An implicit representation of chordal comparabilty graphs in linear-time. In *Graph-Theoretic Concepts in Computer Science, 32nd International Workshop, WG 2006, Bergen, Norway, June 22-24, 2006, Revised Papers*, pages 168–178, 2006.

[13] E. Dahlhaus, J. Gustedt, and R. McConnell. Efficient and practical modular decomposition. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 26–35, 1997.

[14] A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with rank operators. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science LICS*, pages 113–122, 2009.

[15] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2012.

[16] B. Duschnik and E. Miller. Partially ordered sets. *American Journal of Mathematics*, 63:600–610, 1941.

[17] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1999.

[18] K. Etessami and N. Immerman. Tree canonization and transitive closure. *Information and Computation*, 157(1–2):2–24, 2000.

[19] S. Even, A. Pnueli, and A. Lempel. Permutation graphs and transitive graphs. *Journal of the ACM*, 19(3):400–410, 1972.

[20] R. Fagin. Generalized first–order spectra and polynomial–time recognizable sets. In R. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings 7*, pages 43–73, 1974.

[21] T. Gallai. Transitiv orientierbare Graphen. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(1):25–66, 1967.

[22] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974.

[23] P. Gilmore and A. Hoffman. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics*, 16:539–548, 1964.

[24] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Computer science and applied mathematics. Academic Press, 1991.

[25] E. Grädel, P. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Springer-Verlag, 2007.

[26] E. Grädel and W. Pakusa. Rank logic is dead, long live rank logic! In *Proceedings of the 24th EACSL Annual Conference on Computer Science Logic (CSL)*, pages 390–404, 2015.

[27] M. Grohe. Fixed-point logics on planar graphs. In *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 6–15, 1998.

[28] M. Grohe. Definable tree decompositions. In *Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 406–417, 2008.

[29] M. Grohe. Fixed-point definability and polynomial time on chordal graphs and line graphs. In A. Blass, N. Dershowitz, and W. Reisig, editors, *Fields of Logic and Computation, Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*, pages 328–353, 2010.

[30] M. Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2010.

[31] M. Grohe. Descriptive complexity, canonisation, and definable graph structure theory. http://lii.rwth-aachen.de/images/Mitarbeiter/pub/grohe/cap/all.pdf, 2013. [Online; accessed 2016-03-03].

[32] M. Grohe, B. Grußien, A. Hernich, and B. Laubner. L-recursion and a new logic for logarithmic space. In *Proceedings of the 25th International Workshop on Computer Science Logic (CSL)*, pages 277–291, 2011.

[33] M. Grohe, B. Grußien, A. Hernich, and B. Laubner. L-recursion and a new logic for logarithmic space. *Logical Methods in Computer Science*, 9(1), 2012.

[34] M. Grohe and J. Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *Proceedings of the 7th International Conference on Database Theory (ICDT)*, pages 70–82, 1999.

[35] Y. Gurevich. Toward logic tailored for computational complexity. In *Computation and Proof Theory*, pages 175–216. Springer, 1984.

[36] M. Habib and C. Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.

[37] W.-L. Hsu and T.-H. Ma. Substitution decomposition on chordal graphs and applications. In *Proceedings of the 2nd International Symposium on Algorithms*, ISA '91, pages 52–60, London, UK, 1991. Springer-Verlag.

[38] W.-L. Hsu and T.-H. Ma. Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs. *SIAM Journal on Computing*, 28(3):1004–1020, 1999.

[39] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[40] N. Immerman. Expressibility as a complexity measure: Results and directions. In *Proceedings of 2nd Conference on Structure in Complexity Theory*, pages 194–202, 1987.

[41] N. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16:760–778, 1987.

[42] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17(5):935–938, 1988.

[43] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[44] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In A. Selman, editor, *Complexity Theory Retrospective*, pages 59–81. Springer-Verlag, 1990.

[45] H. Kierstead, W. Trotter, and J. Qin. The dimension of cycle-free orders. *Order*, 9(2):103–110, 1992.

[46] D. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms, 2nd Edition*. Addison-Wesley, 1973.

[47] J. Köbler, S. Kuhnert, B. Laubner, and O. Verbitsky. Interval graphs: Canonical representation in logspace. In *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming (ICALP)*, pages 384–395, 2010.

[48] D. Kozen. *Theory of Computation*. Springer, New York, 2006.

[49] B. Laubner. Capturing polynomial time on interval graphs. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 199–208, 2010.

[50] B. Laubner. *The Structure of Graphs and New Logics for the Characterization of Polynomial Time*. PhD thesis, Humboldt-Universität zu Berlin, 2011.

[51] L. Libkin. Logics with counting and local properties. *ACM Transactions on Computational Logic*, 1(1):33–59, 2000.

[52] L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, 2004.

[53] S. Lindell. A logspace algorithm for tree canonization. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 400–404, 1992.

[54] T.-H. Ma and J. Spinrad. Cycle-free partial orders and chordal comparability graphs. *Order*, 8(1):49–61, 1991.

[55] R. McConnell and J. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 536–545, 1994.

[56] R. McConnell and J. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1-3):189–241, 1999.

[57] R. Möhring. *Graphs and Order*, volume 147 of *NATO ASI Series C, Mathematical and Physical Sciences*, pages 41–102. D. Reidel, 1984.

[58] R. Möhring and F. Radermacher. Substitution Decomposition for Discrete Structures and Connections with Combinatorial Optimization. *Annals of Discrete Mathematics*, 19:257–356, 1984.

[59] R. Möhring, D. Wagner, and F. Wagner. VLSI network design. In M. Ball, T. Magnanti, C. Monma, and G. Nemhauser, editors, *Handbooks in Operations Research and Management Sciences: Network Routing*, volume 8, chapter 8, pages 625–712. Elsevier, Amsterdam, 1995.

[60] M. Otto. Bounded variable logics and counting – A study in finite models. Habilitationsschrift RWTH Aachen, 1995, 190 pages.

[61] A. Pnueli, A. Lempel, and S. Even. Transitive orientation of graphs and identification of permutation graphs. *Canadian Journal of Mathematics*, 23:376–385, 1971.

[62] O. Reingold. Undirected ST-connectivity in log-space. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 376–385, 2005.

[63] R. Sedgewick. *Algorithms in Java: Parts 1-4.* Addison-Wesley, 3rd edition, 2002.

[64] M. Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, 2nd, international edition, 1996.

[65] J. Spinrad. *Efficient Graph Representations*, volume 19 of *Fields Institute monographs.* American Mathematical Society, Providence, RI, 2003.

[66] M. Vardi. The complexity of relational query languages. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC)*, pages 137–146, 1982.

[67] J. Walter. *Representations of Rigid Cycle Graphs.* PhD thesis, Wayne State University, 1972.

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Dissertationsschrift “Capturing Polynomial Time and Logarithmic Space using Modular Decompositions and Limited Recursion” selbständig, ohne unerlaubte Hilfe und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Berlin, den 23. März 2016

Berit Grußien