On the Satisfiability of Temporal Logics with Concrete Domains

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet
INFORMATIK

vorgelegt
von Claudia Carapelle

geboren am 15. Mai 1985 in Fiesole (Italien)

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. rer. nat. habil. Markus Lohrey, Universität Siegen
2. Prof. Dr. rer. nat. habil. Carsten Lutz, Universität Bremen

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung am 04.11.2015 mit dem Gesamtprädikat magna cum laude.

First of all I would like to thank my supervisor, Markus Lohrey. I am extremely lucky to have had the opportunity to work with such an exceptional scientist, who is at the same time always friendly and patient. Thank you for your guidance.

Another big thanks goes to Alexander Kartzow, my unofficial co-supervisor, who more than once gave me the right tip to get out of a dead end, who significantly improved my LaTeXing style, but most of all, who was always ready to spend some time discussing ideas with me and brainstorming at the white board.

I would also like to thank Prof. Carsten Lutz for reviewing this thesis and for his kind words of appreciation.

I am also very grateful to Karin, Shiguang and Oliver. Working together I learned from all of you, and had a lot of fun in the process.

Atefeh, Eric and Vitaly, it was a pleasure to share the office life with you. Whether it was automata, German bureaucracy, or simply moral support, you were always there for me.

Thank you Giovanni, for being my rock these past three years, for listening about a thousand times to all of my talks, and for bringing the sunshine with you wherever you go.

Thank you family for supporting me, encouraging me, and believing in me in the pure and total way that just a family can.

This work is dedicated to my grandparents. Their memory is forever deeply embedded in my heart, as are all the things they taught me and the way that they always made me feel so loved and protected.

Contents

Chapter 1

Introduction

Temporal logics are a very popular family of logical languages, used to specify properties of abstracted systems. Since the first appearance of linear temporal logic, better known as LTL [36], temporal logics have been intensively studied, and have become some of the most prominent specification languages used in verification and model checking.

In the last few years, many extensions of temporal logics have been proposed in order to address the need to express more than just abstract properties, see for instance [2, 4, 19, 20, 46]. In some of these studies we can find languages which allow to reason about time intervals, space regions, data values from dense domains like the real numbers or discrete domains like the integers or natural numbers.

Linear Time Temporal Logic with Constraints.
A general approach to creating such formalisms is described in [17] by Demri and D'Souza. Here they show how to extend LTL with the ability to express properties of data values from an arbitrary relational structure $\mathcal{D} = (D, R_1, \ldots, R_n)$, consisting of a domain D and relations $R_1, \ldots, R_n$, and often called *concrete domain*. An example of concrete domain can be $(\mathbb{Z}, <)$, where the integers are considered as a relational structure over the binary order relation $< = \{(a, b) \in \mathbb{Z}^2 \mid a < b\}$. The approach from [17] is also used in the field of description logics (DLs), where Baader and Hanschke first described a way to integrate arbitrary concrete domains into the knowledge-representation language $\mathcal{ALC}$ [3].

The logic defined in [17] is called Constraint-LTL, abbreviated to CLTL. The idea behind this language is the following: For a fixed relational structure $\mathcal{D} = (D, R_1, \ldots, R_n)$ one adds to standard LTL atomic formulas of the form

$$R(\mathsf{X}^{i_1} x_1, \ldots, \mathsf{X}^{i_k} x_k) \tag{1.1}$$

called *atomic constraints*. Here, R is (a name of) one of the relations of the domain

$\mathcal{D}$, $i_1, \ldots, i_k \geq 0$, and $x_1, \ldots, x_k$ are variables that range over D, the universe of $\mathcal{D}$. A CLTL-formula containing such constraints is interpreted over (generally infinite) words, where in addition every position of the word associates with each of the variables $x_1, \ldots, x_k$ an element of $\mathcal{D}$ (one can think of $\mathcal{D}$-registers attached to the system states). Such models are also known as *multi-data words*, and if one ignores the atomic propositions (which can be most of the times simulated using data values) and fixes a finite number of variables, they can be seen as infinite sequences of vectors of values from $\mathcal{D}$.

A constraint $R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ holds in a multi-data word $w = s_0s_1s_2\ldots$ if the tuple $(a_1, \ldots, a_k)$, where a_j is the value of variable x_j at state s_{i_j}, belongs to the $\mathcal{D}$-relation R. In this way, the values of variables at different system states can be compared. For example, one might choose as domain $\mathcal{D}$ the structure $(\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}})$, where $<$ is the order relation defined above, $\equiv$ is the equality relation[1] and $\equiv_a$ is the unary predicate that only holds for a. This structure has infinitely many relations, which is not a problem with respect to satisfiability because any formula can only use finitely many of those predicates. Then, one might for instance write down a formula $(x < \mathsf{X}^1 y) \,\mathsf{U}\, (y \equiv 100)$ which holds on a multi-data word if and only if there is a position where variable y holds the value 100 and for all previous positions t, the value of x at time t is strictly smaller than the value of y at time $t+1$.

Balbiani and Condotta [4] proved a general decidability result for CLTL with constraints over all concrete domains $\mathcal{D}$ satisfying certain properties:

(i) the relations of $\mathcal{D}$ are binary, pairwise disjoint and their union covers $D \times D$, where D is the universe of $\mathcal{D}$, and

(ii) for all finite and consistent sets of constraints, any partial solution (variable valuation satisfying the constraints) can be extended to a global solution.

For these domains, the satisfiability problem for CLTL is proven to be PSPACE-complete, that is, it has the same complexity of satisfiability for LTL without constraints. Instances of domains with the above properties are $(D, <, =, >)$ with $D = \mathbb{R}$ or $D = \mathbb{Q}$, and $(\mathbb{R}^2, sw, s, se, w, e, nw, n, ne, =)$, where the nine relations illustrate the mutual position of two points in the Cartesian plane (eg. (a, b) sw (c, d) iff $a < c$ and $b < d$). In these cases, the dense structure of the real and rational numbers is fundamental to prove property (ii), and in fact the domain $(\mathbb{Z}, <, =, >)$ does not satisfy such condition. This originated the question whether

[1] The reader might be surprised by the fact that we denote the equality relation with $\equiv$. The reason is that later we have to consider relational structures over the same signature, where $\equiv$ is not necessarily interpreted as the equality relation. To avoid confusion, we have decided to use the symbol $\equiv$ for the equality relation as part of relational structures.

CLTL with constraints over the integers would still be decidable, a question which was investigated in [16, 17, 19].

In [19], Demri and Gascon studied LTL extended with constraints from a language IPC*. If we disregard succinctness aspects, the logic is equivalent to CLTL with constraints over the structure

$$\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a \in \mathbb{Z}}, (\equiv_{a,b})_{0 \le a < b}), \tag{1.2}$$

where $\equiv_{a,b}$ denotes the unary relation $\{a + xb \mid x \in \mathbb{Z}\}$ (expressing that an integer is congruent to a modulo b). The main result from [19] states that satisfiability of CLTL with constraints over $\mathcal{Z}$ is decidable and in fact also PSPACE-complete. We should remark that the PSPACE upper bound from [19] even holds for the succinct IPC*-representation of constraints used in [19].

The study of temporal logics with constraints over the integers is partly motivated by the idea of analyzing counter systems. To this end it would be extremely useful to add successor constraints ($y = x + 1$) to $\mathcal{Z}$. Unfortunately this quickly leads to undecidability [17]. Nonetheless $\mathcal{Z}$ allows qualitative representation of increment, for example $x = y + 1$ can be abstracted by $(y > x) \wedge \bigvee_{i=-2^k}^{2^k - 1} (\equiv_{i,2^k}(x) \wedge \equiv_{i+1,2^k}(y))$ where k is a large natural number. This is why temporal logics extended with constraints over $\mathcal{Z}$ seem to be a good compromise between (inexpressive) total abstraction and (undecidable) high concretion.

Branching Time Temporal Logics with Constraints.
In the same way as outlined for LTL above, constraints can be also added to branching-time logics as CTL^* (computation tree logic) and even ECTL^* (extended computation tree logic), obtaining CCTL^* and CECTL^*, respectively. In this framework, formulas are interpreted over *decorated* Kripke structures, where each node (state) carries a valuation for the register variables used in the atomic constraints $R(\mathsf{X}^{i_1} x_1, \ldots, \mathsf{X}^{i_k} x_k)$. The latter become then atomic path formulas, interpreted on infinite paths of decorated Kripke structures.

A weak form of CCTL^* with constraints from $\mathcal{Z}$ (where only integer variables at the same state can be compared) was first introduced in [13], where it is used to describe properties of infinite transition systems, represented by *relational automata*. It is shown in [13] that the model checking problem for CCTL^* over relational automata is undecidable.

Demri and Gascon [19] asked whether satisfiability of CCTL^* with constraints from $\mathcal{Z}$ over decorated Kripke structures is decidable. This problem was investigated in [7, 25], where several partial results where shown: If we replace in $\mathcal{Z}$ the binary predicate $<$ by unary predicates $<_c = \{x \mid x < c\}$ for $c \in \mathbb{Z}$, then satisfiability of CCTL^* has been shown decidable by [25]. For the full structure $\mathcal{Z}$ satisfiability has been shown to be decidable for CEF^+, a fragment of CCTL^* which contains both the existential and universal fragment of CCTL^*, see [7].

Later in [8] Bozzelli and Pinchinat proved that satisfiability of the existential and universal fragment of CCTL^* over the domain $(\mathbb{Z}, \equiv, <)$ are PSPACE-complete.

Contributions of the Thesis.
In [11] we settle the question positively, and prove that CCTL^* with constraints over $\mathcal{Z}$ is decidable. We then lift this result to ECTL^* [12], a proper extension of CTL^* (see [39, 41]) in which the CTL^* path formulas are replaced by the set of all regular properties of paths, represented by Büchi-automata or MSO-formulas.

The method that we use to obtain the results from [11, 12] is divided into two steps: Firstly we individuate sufficient conditions on a relational structure $\mathcal{D}$ which guarantee that satisfiability of CECTL^* with constraints over $\mathcal{D}$ has a decidable satisfiability problem. Secondly, we prove that $\mathcal{Z}$ enjoys these properties, at which point our main result follows.

More specifically, we prove the following result, which will be explained in detail in the sequel:

Result 1 (Thm. 4.7) Let σ be a countable relational signature, and let $\mathcal{D}$ be a σ-structure which:

- is negation-closed, and
- has the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$.

Then satisfiability of CECTL^* with constraints over $\mathcal{D}$ is decidable.

By *negation-closed*, we mean that the complement of any of the relations from σ has to be definable in positive existential first-order logic over $\mathcal{D}$. For instance $(\mathbb{Z}, =, <)$ is negation-closed, because $\neg x < y$ iff $(x = y \ \vee \ y < x)$ and $\neg x = y$ iff $(x < y \ \vee \ y < x)$. Negation closure is needed in order to achieve a strong kind of negation normal form, in which the constraints only appear in a positive form.

The second condition, the *EHD-property*, expresses the fact that we can provide a characterization of all structures which allow a homomorphism into $\mathcal{D}$ using a suitable logical language. More precisely, we say that $\mathcal{D}$ has the property $\mathsf{EHD}(\mathcal{L})$ for some logic $\mathcal{L}$ if and only if there exists an $\mathcal{L}$-sentence $\varphi_{\mathcal{D}}$ such that for any σ-structure[2] $\mathcal{A}$

$$\exists h : \mathcal{A} \to \mathcal{D} \text{ homomorphism} \iff \mathcal{A} \models \varphi_{\mathcal{D}}\,.$$

In Result 1 we use $\mathcal{L} = \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ (in short BMW), which is formed by all boolean combinations of MSO and WMSO+B sentences. WMSO+B is the extension of weak monadic second-order logic (where only quantification over finite subsets is allowed) with the bounding quantifier B: A formula $\mathsf{B}X\,\varphi$ holds

[2] For simplicity we are assuming here that σ is finite, we postpone the general definition to Section 4.1.

in a structure $\mathcal{A}$ if and only if there exists a bound $b \in \mathbb{N}$ such that for every finite subset B of the domain of $\mathcal{A}$ with $\mathcal{A} \models \varphi(B)$ we have $|B| \leq b$. We use this property to reduce the satisfiability problem of CECTL^* to satisfiability of BMW over infinite node-labeled trees. Recently, Bojańczyk and Toruńczyk have shown that satisfiability of WMSO+B over infinite node-labeled trees is decidable [5]. Fortunately, the decidability proof for WMSO+B can be extended to BMW (cf. Section 2.3).

Using Result 1 we can prove:

Result 2 (Thm. 5.2) Satisfiability of CECTL^* with constraints over the concrete domain $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\leq a<b})$ is decidable.

To show this we only need to prove that $\mathcal{Z}$ is negation-closed (Ex. 4.6), and has the property EHD(BMW) (Prop. 5.1). Our proof that $\mathcal{Z}$ has the property EHD(BMW) actually only needs rather weak assumptions on the unary predicates (which are satisfied for the unary relations $\equiv_a$ and $\equiv_{a,b}$), see Section 5.3.

We call what we described above the *EHD-method*: Given any concrete domain $\mathcal{D}$, it is enough to prove that it is negation-closed and that it enjoys the property EHD(BMW) to obtain that satisfiability of CECTL^* over $\mathcal{D}$ is decidable. This is a rather general method, and the question comes naturally, whether we can apply this result to other domains.

An interesting candidate in this context (as mentioned in [19]) is the infinite order tree $\mathcal{T}_\infty = (\mathbb{N}^*, <, \perp, \equiv)$, where $<$ denotes the prefix order on $\mathbb{N}^*$ and $\perp$ denotes the incomparability relation with respect to $<$ (we add the incomparability relation $\perp$ in order to obtain a negation-closed structure). Unfortunately we proved in [10] that $\mathcal{T}_\infty$ does not satisfy the property EHD(BMW). Using an Ehrenfeucht-Fraïssé-game for WMSO+B we obtain the following:

Result 3 (Thm. 6.1) There is no BMW-sentence ψ such that for every countable structure $\mathcal{A}$ (over the signature $\{<, \perp, =\}$) we have: $\mathcal{A} \models \psi$ if and only if there is a homomorphism from $\mathcal{A}$ to $\mathcal{T}_\infty$.

In other words, BMW is not expressive enough to distinguish between those $\{<, \perp, =\}$-structures which allow a homomorphism to the infinite order tree and those who do not.

This shows that the EHD-method cannot be applied to the concrete domain $\mathcal{T}_\infty$ (equivalently, to the infinite binary tree), but it does not imply that satisfiability for CECTL^* with constraints over $\mathcal{T}_\infty$ is undecidable. In fact a recent work from Demri and Deters established decidability of satisfiability for CCTL^* with constraints over $\mathcal{T}_\infty$ and PSPACE-completeness of the corresponding CLTL-fragment [18]. The result is actually proved for a richer logic, which allows to compare the length of the longest common prefix for pairs of elements from $\mathcal{T}_\infty$. Decidability is obtained by a reduction to the satisfiability problem of CLTL and

CCTL^* over the domain $(\mathbb{N}, \equiv, <, (\equiv_a)_{a\in\mathbb{N}})$, which were proved decidable in [19] and [11] respectively. We believe that their result can be extended to CECTL^*, see Remark 6.2.

Despite the fact that the EHD-method fails on $\mathcal{T}_\infty$, we discovered that it can be applied to other tree-like structures, such as semi-linear orders, ordinal trees, and infinitely branching trees of a fixed height. *Semi-linear orders* are partial orders that are tree-like in the sense that for every element x the set of all smaller elements $\downarrow x$ forms a linear suborder. If this linear suborder $\downarrow x$ is an ordinal (for every x) then one has an *ordinal tree.* Ordinal trees are widely studied in descriptive set theory and recursion theory. Note that a tree is a particular instance of a semi-linear order which has a smallest element and where for every x the set $\downarrow x$ is finite.

In the integer-setting we investigated satisfiability for CECTL^*-formulas with constraints over one fixed structure $\mathcal{D}$. For semi-linear orders and ordinal trees it is more natural to consider satisfiability with respect to a class of concrete domains Γ (over a fixed signature σ): The question becomes, whether for a given CECTL^* formula φ there is a concrete domain $\mathcal{C} \in \Gamma$ such that φ is satisfiable by some decorated Kripke structure with concrete values from $\mathcal{C}$. If a class Γ has a universal structure[3] $\mathcal{U}$, then satisfiability with respect to the class Γ is equivalent to satisfiability with respect to $\mathcal{U}$ because obviously a formula φ has a model with some concrete domain from Γ if and only if it has a model with concrete domain $\mathcal{U}$. A typical class with a universal model is the class of all countable linear orders, for which $(\mathbb{Q}, <)$ is universal. Similarly, for the class of all countable trees the tree $\mathcal{T}_\infty$ as well as the binary infinite tree are universal.

Application of the EHD-method to semi-linear orders and ordinal trees gives the following decidability results:

Result 4 (Thm. 6.3) Satisfiability of CECTL^* with constraints over each of the following classes is decidable:

(1) the class of all semi-linear orders,

(2) the class of all ordinal trees, and

(3) for each $h \in \mathbb{N}$, the class of all order trees of height h.

Non Local Constraints.
Notice that the constraints of the form $R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ which we consider in our logic are *local*, in the sense that they can compare data-values in an n-sized neighborhood of the state in which they are evaluated, where $n = \max\{i_1, \ldots, i_k\}$.

[3] A structure $\mathcal{U}$ is universal for the class Γ if (i) $\mathcal{U} \in \Gamma$ and (ii) there is a homomorphic embedding of every structure from Γ into $\mathcal{U}$.

Other proposed extensions of temporal logics have the ability to compare data-values at arbitrary distance. Metric temporal logic (MTL), or FreezeLTL are two prominent examples of such logics (see [2, 20]). In [17], Demri and D'Souza ask whether satisfiability of CLTL with constraints over the integers is preserved when adding non-local constraints of the form $x = \mathsf{F}y$, stating that *there exists a future state where the value of y matches the current value of x.* We answer this question negatively:

Result 5 (Thm. 8.2) Satisfiability for CLTL with constraints over $(\mathbb{Z}, =, <)$ and non-local constraints of the form $x = \mathsf{F}y$ is undecidable.

At the same time, we show that it is possible to add non-local constraints involving the order relation, and maintain decidability:

Result 6 (Thm. 8.7) Satisfiability for CLTL with constraints over $\mathcal{Z}$ from (1.2) on page 7 and non-local order constraints of the form $x < \mathsf{F}y$ or $\mathsf{F}x < y$ is decidable and PSPACE-complete.

Related Work.

In the area of knowledge representation, extensions of description logics with constraints on different concrete domains have been intensively studied, see [31] for a survey. In [32], it was shown that the extension of the description logic $\mathcal{ALC}$ with constraints from $(\mathbb{Q}, <, \equiv)$ has a decidable ($\mathsf{EXPTIME}$-complete) satisfiability problem even in the presence of general TBoxes. A TBox can be seen as a second $\mathcal{ALC}$-formula that has to hold in all nodes of a model. Our decidability proof is partly inspired by the construction from [32], which in contrast to our proof is purely automata-theoretic. Further results for description logics and concrete domains can be found in [33, 34].

There are other extensions of temporal logics that allow to reason about structures with data values, especially in the linear time setting. Logical languages like MTL [29, 2] and TPTL [1] are extensions of LTL often used to specify properties of *timed words*, i.e. data words over the real numbers in which the data sequence is monotonically growing, or monotonic data words over the natural numbers. These logics have however also received some attention on non-monotonic data words [9, 24]. In general, as soon as one drops the monotonicity requirements, satisfiability for these logics becomes undecidable and research has been concentrating on some decidable fragments. An example is $\mathsf{freezeLTL}$, a syntactical restriction of TPTL that has the ability to check data values only for equality. Satisfiability for $\mathsf{freezeLTL}$ has been shown to be decidable over finite data words, but undecidable over infinite data words [20]. In contrast to CLTL, the constraints of $\mathsf{freezeLTL}$ are of the global kind.

Chapter 2

Preliminary Notions

We abbreviate the set $\{1, \dots, d\}$ by $[1, d]$. For a function $\eta : A \to B$ and elements $a \in A$ and $b \in B$, $\eta[a \mapsto b]$ indicates the function that maps a to b and otherwise coincides with η.

2.1 Structures

Let us fix from now on a countably infinite sets of atomic propositions $\mathbb{P}$ and a countably infinite set of register variables $\mathsf{Reg} = \{r_1, r_2, \dots\}$.

Definition 2.1. A **Kripke structure** (KS) over $\mathbb{P}$ is a triple $\mathcal{K} = (S, \to, \rho)$, where:

(i) S is an arbitrary set of nodes,

(ii) $\to \subseteq S \times S$ is a binary relation such that for each $u \in S$ there is a $v \in S$ with $u \to v$, i.e., $(S, \to)$ is a directed graph without dead ends, and

(iii) $\rho : S \to 2^{\mathbb{P}}_{\mathsf{fin}}$ is a labeling function that assigns to every node a finite set of atomic propositions such that $\bigcup_{v \in S} \rho(v)$ is finite, i.e., only finitely many propositions appear in $\mathcal{K}$.

Example 2.2. In Figure 2.1 we draw two examples of Kripke structures. $\mathcal{K}_1 = (S, \to, \rho)$ is a finite KS with domain $S = \{v_1, v_2, v_3, v_4\}$, the binary relation $\to$ consisting of $\{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_3, v_4), (v_4, v_1)\}$ and labeling function ρ defined as $\rho(v_1) = \{p\}$, $\rho(v_2) = \{p, q\}$, $\rho(v_3) = \{q\}$ and $\rho(v_4) = \emptyset$.

$\mathcal{K}_2$ has for domain the infinite set of finite binary words $T = \{0, 1\}^*$, while the binary relation $\to$ is defined as $w_1 \to w_2$ if and only if $w_2 = w_1 0$ or $w_2 = w_1 1$. The labeling function is defined as $\rho(\varepsilon) = \{p\}$, and $\rho(0w) = \{q_0\}$, $\rho(1w) = \{q_1\}$ for all $w \in \{0, 1\}^*$.

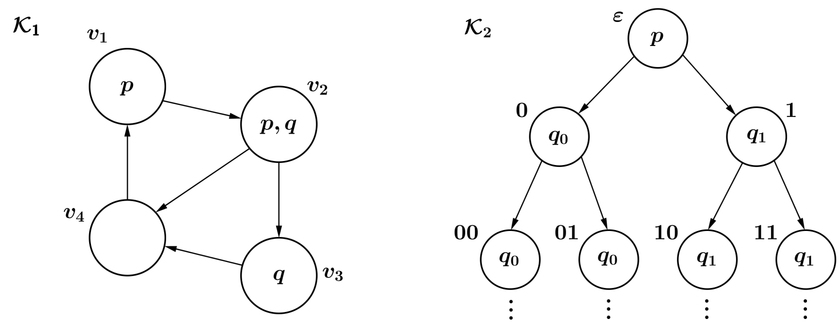

Figure 2.1: We draw here two examples of Kripke structures.

Definition 2.3. A (relational) **signature** σ is a countable (finite or infinite) set of relation symbols. Every relation symbol $R \in \sigma$ has an associated arity $\mathsf{ar}(R) \geq 1$.

A **σ-structure** is a pair $\mathcal{A} = (A, I)$, where A is a non-empty set (the universe of the structure) and I (the interpretation function) maps every $R \in \sigma$ to an $\mathsf{ar}(R)$-ary relation over A.

Example 2.4. A simple example of $\{<\}$-structure is $(\mathbb{Z}, I)$, where $I(<)$ is, as expected, the set of pairs of elements of $\mathbb{Z}$ in which the first component is smaller than the second, i.e. $I(<) = \{(a, b) \in \mathbb{Z}^2 \mid a < b\}$.

Quite often, we identify the relation $I(R)$ with the relation symbol R, and we specify a σ-structure as $(A, R_1, R_2, \ldots)$ where $\sigma = \{R_1, R_2, \ldots\}$. In the example above, then, we would simply write $(\mathbb{Z}, <)$.

Given $\mathcal{A} = (A, R_1, R_2, \ldots)$ and given a subset $B \subseteq A$, for each R_i we define $R_{i\restriction B} = R_i \cap B^{\mathsf{ar}(R)}$ to be the restriction of R_i to $B^{\mathsf{ar}(R_i)}$. We write $\mathcal{A}_{\restriction B}$ for the induced substructure $(B, R_{1\restriction B}, R_{2\restriction B}, \ldots)$.

Example 2.5. Let $\mathcal{A} = (\mathbb{Z}, <)$ be the structure from Example 2.4, then $\mathcal{A}_{\restriction \mathbb{N}} = (\mathbb{N}, <_{\restriction \mathbb{N}})$, is the obvious $\{<\}$-structure on the natural numbers.

Definition 2.6. For a subsignature $\tau \subseteq \sigma$, a τ-structure $\mathcal{B} = (B, J)$ and a σ-structure $\mathcal{A} = (A, I)$, a **homomorphism** from $\mathcal{B}$ to $\mathcal{A}$ is a mapping $h : B \to A$ such that for all $R \in \tau$ and all tuples $(b_1, \ldots, b_{\mathsf{ar}(R)}) \in B^{\mathsf{ar}(R)}$ we have

$$(b_1, \ldots, b_{\mathsf{ar}(R)}) \in J(R) \Rightarrow (h(b_1), \ldots, h(b_{\mathsf{ar}(R)})) \in I(R) .$$

We write $\mathcal{B} \preceq \mathcal{A}$ if there is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. Note that we do not require this homomorphism to be injective.

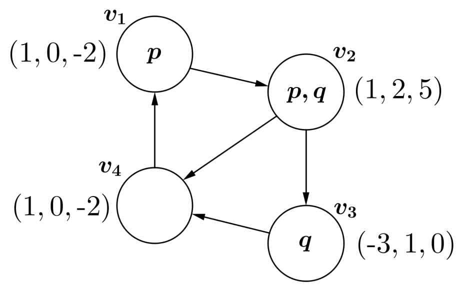

Figure 2.2: A decorated Kripke structure K.

We now introduce *decorated* Kripke structures. These are two-sorted objects where one part is a Kripke structure and the other part is some σ-structure called the *concrete domain*. The two parts are connected by a valuation function.

Definition 2.7. A **$\mathcal{D}$-decorated Kripke structure** K is a tuple $(\mathcal{D}, \mathcal{K}, \gamma)$ where:

- $\mathcal{D} = (D, I)$ is a σ-structure (the **concrete domain**),
- $\mathcal{K} = (S, \rightarrow, \rho)$ is a Kripke structure (called the **underlying Kripke structure** of K), and
- $\gamma : S \times \mathsf{Reg} \rightarrow D$ is a **valuation function**, assigning values from the concrete domain to each variable from Reg in each node of the Kripke structure.

We can imagine such objects as Kripke structures where each node v, in addition to carrying atomic propositions, also holds a (possibly infinite) vector $(a_1, a_2, \dots)$ of values from the concrete domain $\mathcal{D}$, namely $(\gamma(v, r_1), \gamma(v, r_2), \dots)$, the values assigned in v to all register variables $r_1, r_2, \dots \in \mathsf{Reg}$ from the valuation function γ. For brevity, we will usually call K a $\mathcal{D}$-Kripke structure, or a $\mathcal{D}$-KS.

Example 2.8. In Figure 2.2 we draw a $\mathcal{D}$-Kripke Structure K, where $\mathcal{D}$ is some relational structure over the set of integers. Here we suppose that the set of register variables Reg is a finite set $\{r_1, r_2, r_3\}$. The underlying KS is $\mathcal{K}_1$ from Example 2.2, and the valuation function γ on the node v_1, for instance, assigns $\gamma(v_1, r_1) = 1$, $\gamma(v_1, r_2) = 0$ and $\gamma(v_1, r_3) = -2$.

2.2 Trees and Paths

Definition 2.9. A **Kripke tree** is a particular instance of a Kripke structure of the form $\mathcal{T} = (S, \rightarrow, \rho)$, where $(S, \rightarrow)$ is a rooted tree, that is:

- $S \subseteq \Sigma^*$ is a prefix-closed set of strings over some alphabet Σ, and

- $u \to v$ if and only if $v = ua$ for some $a \in \Sigma$.

If $S = [1, n]^*$ for $n \in \mathbb{N}$, we say that $\mathcal{T}$ is a **Kripke n-tree**. If moreover $n = 1$ then we have a **Kripke path** (KP) $\mathcal{P} = (\mathbb{N}, \to, \rho)$ where $\to$ is the successor relation on the natural numbers.

Definition 2.10. We call a $\mathcal{D}$-Kripke structure $\mathfrak{T} = (\mathcal{D}, \mathcal{T}, \gamma)$ a **$\mathcal{D}$-Kripke tree** ($\mathcal{D}$-KT), a **$\mathcal{D}$-Kripke n-tree** or a **$\mathcal{D}$-Kripke path** ($\mathcal{D}$-KP) if its underlying Kripke structure $\mathcal{T}$ is a Kripke tree, a Kripke n-tree or a Kripke path, respectively

Remark 2.11. A Kripke path is nothing but a word over the alphabet $2^{\mathbb{P}}_{\mathsf{fin}}$, consisting of all finite subsets of $\mathbb{P}$. A word is more frequently represented by the sequence of labels of its nodes, in this case $\rho(1)\rho(2)\rho(3)\ldots$.

If the set of registers is finite, $\mathsf{Reg} = \{r_1, \ldots, r_n\}$, then a $\mathcal{D}$-Kripke path is exactly what we called a **multi-data word** in the introduction (page 6). We can see it as a sequence of pairs:

$$(l_0, \vec{v}_0)(l_1, \vec{v}_1)(l_2, \vec{v}_2)\ldots$$

where for each $i \in \mathbb{N}$, $l_i = \rho(i)$ is the node-label at state i and $\vec{v}_i$ is the n-vector of data values from $\mathcal{D}$ assigned by γ to $r_1, \ldots, r_n$ at state i, namely $\vec{v}_i = (\gamma(i, r_1), \ldots, \gamma(i, r_n))$.

Definition 2.12. Given a Kripke structure $\mathcal{K} = (S, \to, \rho)$, an **infinite path** is an infinite sequence $P = s_0 s_1 s_2 \cdots$ such that $s_i \in S$ and $s_i \to s_{i+1}$ for all $i \geq 0$. For $i \geq 0$ we define the node $P(i) = s_i$. A **finite path** is a finite non-empty prefix of an infinite path.

Definition 2.13. For $s \in S$, the **unfolding** of $\mathcal{K}$ from s, denoted by $\mathsf{Unf}(\mathcal{K}, s)$, is the Kripke tree $\mathcal{T} = (T, \to', \rho')$ where

- T is the set of finite paths P with $P(0) = s$,

- $\to'$ is defined to be the extension of paths by a single edge, i.e., for finite paths P_1 and P_2 from T we have $P_1 \to' P_2$ iff $P_2 = P_1 s'$ for a node $s' \in S$, and

- ρ' is given by "last-node semantics", i.e., for every $s_0 s_1 \cdots s_n \in T$ we set $\rho'(s_0 s_1 \cdots s_n) = \rho(s_n)$.

The unfolding of a Kripke structure naturally lifts to decorated KSs.

Definition 2.14. Let $\mathsf{K} = (\mathcal{D}, \mathcal{K}, \gamma)$ be a $\mathcal{D}$-Kripke structure with underlying KS $\mathcal{K} = (S, \rightarrow, \rho)$ and let $s \in S$. We denote by $\mathsf{Unf}(\mathsf{K}, s)$ the $\mathcal{D}$-Kripke tree $(\mathcal{D}, \mathcal{K}', \gamma')$ with underlying Kripke tree $\mathcal{K}' = \mathsf{Unf}(\mathcal{K}, s)$, and valuation function γ' defined again by the last node semantics $\gamma'(s_0 s_1 \cdots s_n, r) = \gamma(s_n, r)$ for all finite paths $s_0 s_1 \cdots s_n$ with $s_0 = s$ and for all $r \in \mathsf{Reg}$.

Remark 2.15. Given a Kripke Structure $\mathcal{K}$ and an infinite path $P = s_0 s_1 s_2 \cdots$, this identifies a substructure of $\mathsf{Unf}(\mathcal{K}, s_0)$ induced by the finite non-empty prefixes of P. Thus, P naturally induces a Kripke path $\mathsf{Unf}(\mathcal{K}, s_0)_{\restriction P}$, which we usually denote by $\mathcal{P}$.

For a $\mathcal{D}$-Kripke structure $\mathsf{K} = (\mathcal{D}, \mathcal{K}, \gamma)$, P also induces a $\mathcal{D}$-Kripke path $\mathsf{P} = (\mathcal{D}, \mathcal{P}, \gamma')$ in K, where γ' is obtained by restricting γ to the elements of $\mathcal{P}$. We call it the $\mathcal{D}$-Kripke path corresponding to P. Note that every $\mathcal{D}$-KP in K is an induced subgraph of the unfolding of K from some node s.

We lift the position notation for paths to Kripke paths and decorated Kripke paths by setting $\mathsf{P}(i) = \mathcal{P}(i) = s_i$ for all $i \geq 0$.

2.3 MSO and WMSO+B

Throughout this work, we fix countably infinite sets $\mathbb{V}_{\mathsf{el}}$ and $\mathbb{V}_{\mathsf{set}}$ of element variables and set variables, respectively.

Monadic second-order logic (MSO) is the extension of first-order logic where also quantification over subsets of the underlying structure is allowed. Let us fix a signature σ.

Definition 2.16 (MSO **Syntax**)**.** MSO-formulas over the signature σ are defined by the following grammar, where $R \in \sigma$, $x, y, x_1, \ldots, x_{\mathsf{ar}(R)} \in \mathbb{V}_{\mathsf{el}}$ and $X \in \mathbb{V}_{\mathsf{set}}$:

$$\varphi ::= R(x_1, \ldots, x_{\mathsf{ar}(R)}) \mid x = y \mid x \in X \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \exists x\, \varphi \mid \exists X\, \varphi\ . \tag{2.1}$$

MSO-formulas are evaluated on σ-structures, where element and set variables range respectively over elements and subsets of the domain.

Definition 2.17 (MSO **Semantics**)**.** If $\mathcal{A} = (A, I)$ is a σ-structure, the semantics of MSO-formulas on $\mathcal{A}$ are defined inductively on the structure of the formula with the help of a valuation function $\nu : \mathbb{V}_{\mathsf{el}} \cup \mathbb{V}_{\mathsf{set}} \rightarrow A \cup 2^A$ as follows:

- $(\mathcal{A}, \nu) \models R(x_1, \ldots, x_{\mathsf{ar}(R)})$ iff $(\nu(x_1), \ldots, \nu(x_{\mathsf{ar}(R)})) \in I(R)$;
- $(\mathcal{A}, \nu) \models x = y$ iff $\nu(x) = \nu(y)$;
- $(\mathcal{A}, \nu) \models x \in X$ iff $\nu(x) \in \nu(X)$;
- $(\mathcal{A}, \nu) \models \neg\varphi$ iff it is not the case that $(\mathcal{A}, \nu) \models \varphi$;

- $(\mathcal{A}, \nu) \models (\varphi_1 \wedge \varphi_2)$ iff $(\mathcal{A}, \nu) \models \varphi_1$ and $(\mathcal{A}, \nu) \models \varphi_2$;
- $(\mathcal{A}, \nu) \models \exists x\, \varphi$ iff there exists $b \in A$ such that $(\mathcal{A}, \nu[x \mapsto b]) \models \varphi$;
- $(\mathcal{A}, \nu) \models \exists X\, \varphi$ iff there exists $B \subseteq A$ such that $(\mathcal{A}, \nu[X \mapsto B]) \models \varphi$;

Remark 2.18. Introducing disjunction as

- $(\varphi_1 \vee \varphi_2) \coloneqq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$,

and universal quantification over element and set variables

- $\forall x\, \varphi \coloneqq \neg\exists x\, \neg\varphi$,
- $\forall X\, \varphi \coloneqq \neg\exists X\, \neg\varphi$,

we can associate to each formula φ its semantically equivalent negation normal form $\hat{\varphi}$, where negation only appears in front of atomic formulas and relations.

Remark 2.19. Note that, if in a formula φ no variable occurs freely, i.e. all variables appear in the scope of a quantifier, the semantics of φ do not depend on the choice of ν. We can therefore simply write $\mathcal{A} \models \varphi$.

Weak monadic second-order logic (WMSO) has the same syntax as MSO (2.1), but second-order variables are interpreted as finite subsets of the underlying universe.

WMSO+B is the extension of WMSO by the **bounding quantifier** $\mathsf{B}X\, \varphi$ for $X \in \mathbb{V}_{\mathsf{set}}$. The semantics of $\mathsf{B}X\, \varphi$ in the structure $\mathcal{A} = (A, I)$ are defined as follows: $(\mathcal{A}, \nu) \models \mathsf{B}X\, \varphi(X)$ if and only if there is a bound $b \in \mathbb{N}$ such that whenever $(\mathcal{A}, \nu) \models \varphi(B)$ for some finite subset $B \subseteq A$, then $|B| \leq b$. The dual quantifier is denoted by U. It is called the **unbounding quantifier** and $\mathsf{U}X\, \varphi = \neg\mathsf{B}X\, \varphi$ expresses that there are arbitrarily large finite sets that satisfy φ.

Example 2.20. For later use, we state some example formulas. Let $\varphi(x, y)$ be a WMSO-formula with two free first-order variables x and y. Let $\mathcal{A} = (A, I)$ be a structure and let $E_\varphi = \{(a, b) \in A \times A \mid \mathcal{A} \models \varphi(a, b)\}$ be the binary relation defined by $\varphi(x, y)$. Consider E_φ as the edge relation of the graph $\mathcal{G}_\varphi = (A, E_\varphi)$. We define the WMSO-formula $\mathsf{reach}_\varphi^Z(x_1, x_2)$ to be

$$x_1 \in Z \wedge \forall Y \subseteq Z \left[\big(x_1 \in Y \wedge \forall y \forall z\, (y \in Y \wedge z \in Z \wedge \varphi(y, z)) \rightarrow z \in Y \big) \rightarrow x_2 \in Y \right] .$$

It is easy to see that for every finite subset $B \subseteq A$, we have $\mathcal{A} \models \mathsf{reach}_\varphi^B(a, b)$ if and only if $(a, b) \in (E_\varphi^* \cap B^2)$, i.e., b is reachable from a in the subgraph $\mathcal{G}_{\varphi \upharpoonright B}$. Note that reach_φ^Z is the standard MSO-formula for reachability but restricted to the subgraph induced by Z. If we define $\mathsf{reach}_\varphi \coloneqq \exists Z\, \mathsf{reach}_\varphi^Z$, the semantics of

reach_φ seen as an MSO-formula or a WMSO-formula are the same because b is reachable from a in the graph $\mathcal{G}_\varphi$ if and only if it is in some finite subgraph of $\mathcal{G}_\varphi$.

Let $\mathsf{ECycle}_\varphi = \exists x\, \exists y\, (\mathsf{reach}_\varphi(x,y) \wedge \varphi(y,x))$ be the WMSO-formula expressing that there is a cycle in $\mathcal{G}_\varphi$. We now restrict our attention to the case that the graph $\mathcal{G}_\varphi$ defined by $\varphi(x,y)$ is acyclic. Hence, the reflexive transitive closure E_φ^* is a partial order on A. Note that a finite set $F \subseteq A$ is an E_φ-path from $a \in F$ to $b \in F$ if and only if $(F, (E_\varphi \cap (F \times F))^*)$ is a finite linear order with minimal element a and maximal element b. Define the WMSO-formula $\mathsf{Path}_\varphi(x,y,Z)$ as

$$\forall w \in Z\ \ \forall z \in Z\ \ [(\mathsf{reach}_\varphi^Z(w,z) \vee \mathsf{reach}_\varphi^Z(z,w))\ \wedge \mathsf{reach}_\varphi^Z(x,w) \wedge \mathsf{reach}_\varphi^Z(w,y)]\ .$$

For every structure $\mathcal{A}$ such that $\mathcal{G}_\varphi$ is acyclic, we have $\mathcal{A} \models \mathsf{Path}_\varphi(a,b,P)$ if and only if P contains exactly the nodes that form an E_φ-path from a to b.

We finally define the WMSO+B-formula

$$\mathsf{BPaths}_\varphi(x,y) = \mathsf{B}Z\, \mathsf{Path}_\varphi(x,y,Z)\ . \tag{2.2}$$

Under the assumption that $\mathcal{G}_\varphi$ is acyclic, $\mathcal{A} \models \mathsf{BPaths}_\varphi(a,b)$ if and only if there is a bound $k \in \mathbb{N}$ on the length of any E_φ-path from a to b.

Next, let $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ be the set of all Boolean combinations of MSO-formulas and (WMSO+B)-formulas. We use the following result:

Theorem 2.21 (cf. [5]). *One can decide whether for a given $n \in \mathbb{N}$ and a formula $\varphi \in \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ there is a Kripke n-tree $\mathcal{K}$ such that $\mathcal{K} \models \varphi$.*

This theorem follows from results of Bojańczyk and Toruńczyk [5, 6]. They introduced puzzles which can be seen as pairs $P = (A, C)$, where A is a parity tree automaton and C is an unboundedness condition C which specifies a certain set of infinite paths labeled by states of A. A puzzle accepts a tree $\mathcal{T}$ if there is an accepting run ρ of A on $\mathcal{T}$ such that for each infinite path π occurring in ρ, $\pi \in C$ holds. In particular, ordinary parity tree automata can be seen as puzzles with the trivial unboundedness condition. The proof of Theorem 2.21 combines the following results.

Lemma 2.22 ([5]). *From a given (WMSO+B)-formula φ and $n \in \mathbb{N}$ one can construct a puzzle P_φ such that φ is satisfied by some Kripke n-tree iff P_φ is nonempty.*

Lemma 2.23 ([5]). *Emptiness of puzzles is decidable.*

Lemma 2.24 (Lemma 17 of [6]). *Puzzles are effectively closed under intersection.*

Using these results, it is easy to prove Theorem 2.21:

Proof. Let $\varphi \in \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$. First, φ can be effectively transformed into a disjunction $\bigvee_{i=1}^{n}(\varphi_i \wedge \psi_i)$ where $\varphi_i \in \mathsf{MSO}$ and $\psi_i \in \mathsf{WMSO{+}B}$ for all i. By Lemma 2.22, we can construct a puzzle P_i for ψ_i. The MSO-formula φ_i can be translated into a parity tree automaton A_i [37]. Using Lemma 2.24 we compute a puzzle P_i' recognizing the intersection of P_i and A_i. Clearly, φ is satisfiable over Kripke n-trees if and only if there is an i such that $\varphi_i \wedge \psi_i$ is satisfiable over Kripke n-trees, if and only if there is an i such that P_i' is nonempty. By Lemma 2.23, the latter condition is decidable which concludes the proof of the theorem. □

2.4 $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO} + \mathsf{B})$ and the k-Copy Operation

In this section we show a technical result stating that $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ (BMW) is compatible with the k-copy operation. The proof basically copies the known proofs for MSO and WMSO extended by a translation of bounding quantifiers. Readers that are not interested in the proof details can safely skip them. We will need this result later in Section 4.2

We first define the k-copy operation:

Definition 2.25. Let $k \in \mathbb{N}$ and let $\mathcal{A} = (A, I)$ be a structure over the signature σ that does not contain relation symbols $\sim, P_1, P_2, \ldots, P_k$ ($\sim$ is binary and all P_i are unary). The **k-copy of** $\mathcal{A}$, denoted by $\mathsf{copy}_k(\mathcal{A})$, is the $(\sigma \cup \{\sim, P_1, P_2, \ldots, P_k\})$-structure $(A \times \{1, 2, \ldots, k\}, J)$ where

- for all $R \in \sigma$ of arity m,

$$J(R) = \{((a_1, i), (a_2, i), \ldots, (a_m, i)) \mid (a_1, a_2, \ldots, a_m) \in I(R), 1 \le i \le k\} \ ,$$

- $J(\sim) = \{((a, i_1), (a, i_2)) \mid a \in A, 1 \le i_1, i_2 \le k\}$, and
- for each $1 \le m \le k$, $J(P_m) = \{(a, m) \mid a \in A\}$.

Given a structure $\mathcal{A}$, the k-copy operation creates a new structure, $\mathsf{copy}_k(\mathcal{A})$, which contains k many copies of $\mathcal{A}$: there are k disjoint substructures of $\mathsf{copy}_k(\mathcal{A})$ (identifiable through the predicates $P_1, \ldots, P_k$) which, seen as σ-structures, are isomorphic to $\mathcal{A}$. The additional binary predicate $\sim$ relates all those members of $\mathsf{copy}_k(\mathcal{A})$ which are a duplicate of the same element in $\mathcal{A}$.

In the following proposition we prove that $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ is *compatible* with the k-copy operation, i.e., whatever property we can specify on a structure $\mathcal{A}$ using BMW can also be expressed about its k-copy.

Proposition 2.26. *Let $k \in \mathbb{N}$ be some number, $\mathcal{A} = (A, I)$ some infinite structure over the signature σ, and $\tau = \sigma \cup \{\sim, P_1, P_2, \dots, P_k\}$ an extension of σ by one fresh binary relation symbol $\sim$ and k fresh unary relation symbols $P_1, \dots, P_k$. Given a BMW-sentence φ over τ, we can compute a BMW-sentence φ^k over σ such that $\mathsf{copy}_k(\mathcal{A}) \models \varphi$ if and only if $\mathcal{A} \models \varphi^k$.*

Proof. The proof is in 3 steps. We only do it for $\mathsf{WMSO{+}B}$ in order to avoid handling a finite and an infinite version of existential set quantification. The extension to $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ is straightforward. Instead of dealing with the bounding quantifier B directly, we deal with the unbounding quantifier U. This suffices since a bounding quantifier is equivalent to a negated unbounding quantifier. First we define a formula $\hat{\varphi}$. It uses element variables x, x' (respectively, set variables $X^1, \dots, X^k$) for every element variable x (respectively, set variable X) used in φ. In addition, $\hat{\varphi}$ uses element variables $y_1, \dots, y_k$ that identify the k different copies of $\mathcal{A}$ from the k-copy of $\mathcal{A}$ (for this purpose $y_1, \dots, y_k$ are always assigned pairwise different values). Then we prove a strong connection between evaluations of φ on $\mathsf{copy}_k(\mathcal{A})$ and of $\hat{\varphi}$ on $\mathcal{A}$. Finally, we create φ^k from $\hat{\varphi}$ by quantification over the parameters $y_1, y_2, \dots, y_k$ and show that φ^k has the desired property.

Step 1. We define $\hat{\varphi}$ from φ by case distinction on the structure of φ.

1. If $\varphi = P_i(x)$ for some $1 \leq i \leq k$, then $\hat{\varphi} \coloneqq (x' = y_i)$.
2. If $\varphi = x_1 \sim x_2$ then $\hat{\varphi} \coloneqq (x_1 = x_2)$.
3. If $\varphi = R(x_1, \dots, x_r)$ for $R \in \sigma$, then $\hat{\varphi} \coloneqq R(x_1, \dots, x_r) \wedge (x'_1 = \dots = x'_r)$.
4. If $\varphi = x \in X$, then $\hat{\varphi} \coloneqq \bigvee_{i=1}^{k} (x' = y_i \wedge x \in X^i)$.
5. If $\varphi = \psi \wedge \chi$, then $\hat{\varphi} \coloneqq \hat{\psi} \wedge \hat{\chi}$.
6. If $\varphi = \neg\psi$ then $\hat{\varphi} \coloneqq \neg\hat{\psi}$.
7. If $\varphi = \exists x\, \psi$ then $\hat{\varphi} = \exists x\, \exists x' \, (\bigvee_{i=1}^{k} x' = y_i \wedge \hat{\psi})$.
8. If $\varphi = \exists X\, \psi$ then $\hat{\varphi} = \exists X^1\, \exists X^2\, \cdots \exists X^k\, \hat{\psi}$.
9. If $\varphi = \mathsf{U} X\, \psi$ then $\hat{\varphi} = \bigvee_{i=1}^{k} \mathsf{U} X^i\, \exists X^1\, \dots \exists X^{i-1}\, \exists X^{i+1}\, \cdots \exists X^k\, \hat{\psi}$.

Step 2. Let $\varphi(x_1, \dots, x_n, X_1, \dots, X_m)$ be a $\mathsf{WMSO{+}B}$ formula. Fix some values $\hat{a}_1, \dots, \hat{a}_k \in A$ such that $\hat{a}_i \neq \hat{a}_j$ for $i \neq j$ (recall that we assume A to be infinite), $a_1, \dots, a_n \in A$, $k_1, \dots, k_n \in \{1, \dots, k\}$, and finite subsets $A^1_1, \dots, A^k_m \subseteq A$. Fix a variable assignment η_k (in $\mathsf{copy}_k(\mathcal{A})$) such that $\eta_k(x_i) = (a_i, k_i)$ and

$\eta_k(X_i) = \bigcup_{j=1}^k A_i^j \times \{j\}$. Fix another variable assignment η (in $\mathcal{A}$) such that $\eta(y_i) = \hat{a}_i$, $\eta(x_i) = a_i$, $\eta(x_i') = \hat{a}_{k_i}$ and $\eta(X_i^j) = A_i^j$. We claim that

$$(\mathsf{copy}_k(\mathcal{A}), \eta_k) \models \varphi \text{ if and only if } (\mathcal{A}, \eta) \models \hat{\varphi} .$$

The proof is by structural induction. Most cases are straightforward and can be copied from compatibility proofs of (W)MSO with the k-copy operation (see [14]). The new case is the unbounding quantifier. For this case assume that $\varphi = \mathsf{U}X\,\psi$. By definition $(\mathsf{copy}_k(\mathcal{A}), \eta_k) \models \varphi$ if and only if for all $n \in \mathbb{N}$ there is a finite set $S \subseteq A \times \{1, \ldots, k\}$ such that $|S| \geq n$ and $(\mathsf{copy}_k(\mathcal{A}), \eta_k[X \mapsto S]) \models \psi$. By induction hypothesis this is the case if and only if for all $n \in \mathbb{N}$ there are finite sets $S^1, \ldots, S^k \subseteq A$ such that $|S^1| + \cdots + |S^k| \geq n$ and

$$(\mathcal{A}, \eta[X^1 \mapsto S^1, \ldots, X^k \mapsto S^k]) \models \hat{\psi} .$$

Noting that this means that one of the sets has size at least $\frac{n}{k}$, this statement is equivalent to the statement that for all $n' \in \mathbb{N}$ there are a $1 \leq j \leq k$ and finite sets $S^1, \ldots, S^k$ such that $|S^j| \geq n'$ and

$$(\mathcal{A}, \eta[X^1 \mapsto S^1, \ldots, X^k \mapsto S^k]) \models \hat{\psi} .$$

By the pigeon hole principle, we can rewrite this to the statement that there is a $1 \leq j \leq k$ such that

$$(\mathcal{A}, \eta) \models \mathsf{U}X^j\, \exists X^1 \ldots \exists X^{j-1}\, \exists X^{j+1} \ldots \exists X^k\, \hat{\psi} .$$

This is evidently equivalent to

$$(\mathcal{A}, \eta) \models \bigvee_{i=1}^{k} \mathsf{U}X^i\, \exists X^1\, \exists X^2 \ldots \exists X^{i-1}\, \exists X^{i+1} \ldots \exists X^k\, \hat{\psi} ,$$

i.e., $(\mathcal{A}, \eta) \models \hat{\varphi}$.

Step 3. Finally, for a sentence φ set

$$\varphi^k = \exists y_1 \exists y_2 \cdots \exists y_k \bigwedge_{1 \leq i < j \leq k} y_i \neq y_j \wedge \hat{\varphi} .$$

Using the claim from Step 2, it is clear that for all structures $\mathcal{A}$ with at least k elements we have

$$\mathsf{copy}_k(\mathcal{A}) \models \varphi \text{ if and only if } \mathcal{A} \models \varphi^k .$$

This concludes the proof. $\square$

2.5 Temporal Logics

Throughout this work we will often refer to LTL (linear temporal logic) and CTL^* (computation tree logic). We define here their syntax and semantics, above all to fix notation. The reader familiar with these logics can safely skip this section.

Definition 2.27. LTL formulas over $\mathbb{P}$ are defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \ ,$$

where $p \in \mathbb{P}$.

LTL formulas are interpreted over a Kripke path $\mathcal{P} = (\mathbb{N}, \to, \rho)$, or equivalently, the infinite word over the alphabet $2^{\mathbb{P}}_{\mathsf{fin}}$ given by the sequence of labels of the nodes of $\mathcal{P}$: $w = \rho(0)\rho(1)\rho(2)\ldots$. The semantics for each position $i \in \mathbb{N}$ of $\mathcal{P}$ is defined inductively as follows:

- $(\mathcal{P}, i) \models p$ iff $p \in \rho(i)$,
- $(\mathcal{P}, i) \models \neg\varphi$ iff it is not the case that $(\mathcal{P}, i) \models \varphi$,
- $(\mathcal{P}, i) \models (\varphi_1 \wedge \varphi_2)$ iff $(\mathcal{P}, i) \models \varphi_1$ and $(\mathcal{P}, i) \models \varphi_2$,
- $(\mathcal{P}, i) \models \mathsf{X}\varphi$ iff $(\mathcal{P}, i+1) \models \varphi$, and
- $(\mathcal{P}, i) \models \varphi_1 \,\mathsf{U}\, \varphi_2$ iff there exists a position $j \geq i$ such that $(\mathcal{P}, j) \models \varphi_2$ and for all $i \leq k < j$ we have $(\mathcal{P}, k) \models \varphi_1$.

Definition 2.28. We define CTL^*-state formulas φ and CTL^*-path formulas ψ by the following grammar, where $p \in \mathbb{P}$:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathsf{E}\psi \\ \psi &::= \varphi \mid \neg\psi \mid (\psi \wedge \psi) \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \end{aligned}$$

CTL^* state and path formulas are interpreted respectively on nodes and paths of Kripke structures. So, given a KS $\mathcal{K} = (S, \to, \rho)$, a node $v \in S$ and an infinite path $P = p_0 p_1 p_2 \ldots$ in $\mathcal{K}$, we define the satisfaction relation as follows (we omit the cases which are analogous to LTL):

- $(\mathcal{K}, v) \models p$ iff $p \in \rho(v)$,
- $(\mathcal{K}, v) \models \mathsf{E}\psi$ iff there is a path $P = p_0 p_1 p_2 \ldots$ with $p_0 = v$ and $(\mathcal{K}, P) \models \psi$,
- $(\mathcal{K}, P) \models \varphi$ iff $(\mathcal{K}, p_0) \models \varphi$.

For both LTL and CTL^* we define the usual abbreviations:

- $\top := p \vee \neg p$,
- $\vartheta_1 \vee \vartheta_2 := \neg(\neg\vartheta_1 \wedge \neg\vartheta_2)$ (for both state and path formulas),
- $\mathsf{F}\varphi := \top\,\mathsf{U}\,\varphi$ (finally operator),
- $\mathsf{G}\varphi := \neg\mathsf{F}\neg\varphi$ (globally operator),
- $\mathsf{A}\psi := \neg\mathsf{E}\neg\psi$ (universal path quantifier),
- $\psi_1\,\mathsf{Rel}\,\psi_2 := \neg(\neg\psi_1\,\mathsf{U}\,\neg\psi_2)$ (the release operator).

Example 2.29. LTL and CTL* can express all sorts of interesting specifications, for instance the LTL formula $\mathsf{FG}p$ interpreted on a path P states that starting in some future position of P, the atomic proposition p always holds. The CTL* formula $\mathsf{EFG}p$ asks for the existence of at least one path on which the above LTL specification holds. This can be seen as asking that at least one of the possible computations described from the paths of the Kripke structures satisfies the required specification.

Chapter 3

ECTL* with constraints

Extended computation tree logic (ECTL*) is a branching time temporal logic first introduced in [39, 41] as an extension of CTL*. As the latter, ECTL* is interpreted on Kripke structures, has both *state*- and *path*-formulas and allows existential and universal quantifications on infinite paths. But while CTL* path-formulas allow to specify LTL properties, ECTL* can describe regular (i.e., MSO-definable) properties of paths. In its original formulation, ECTL* uses Büchi automata to replace the classical CTL* path formulas. In this work, instead of automata, we use MSO-formulas. Given the famous result of Büchi that MSO and Büchi automata are equi-expressive on paths, we obtain an expressively equivalent logic. We choose the formulation using MSO because it provides a simpler framework to add constraints.

What we present in this chapter is an enhanced version of ECTL*, which we call Constraint-ECTL*, or in short CECTL*. In CECTL* path-formulas come from Constraint-Path-MSO which we define below. Suppose we are interested in a particular concrete domain $\mathcal{D}$ over a relational structure τ and let us fix such signature for the rest of this section.

3.1 Constraint Path MSO (CMSO)

To build ECTL* with constraints we use a constraint version of MSO, interpreted on decorated Kripke paths, which we call **Constraint-Path-**MSO, denoted as CMSO. To define it, we start from MSO for infinite paths (words) with the successor function S. This is simply MSO as in Definition 2.16, where the signature σ is set to $\{S\} \cup \mathbb{P}$. Here the atomic propositions from $\mathbb{P}$ are seen as unary predicates and S is the binary predicate for the successor relation. Writing $x_1 = S(x_2)$ instead of $S(x_1, x_2)$ to improve readability, we have that MSO over $\sigma = \{S\} \cup \mathbb{P}$

is defined by the following grammar:

$$\psi ::= p(x) \mid x_1 = x_2 \mid x_1 = S(x_2) \mid x \in X \mid \neg\psi \mid (\psi \wedge \psi) \mid \exists x\, \psi \mid \exists X\, \psi \ , \tag{3.1}$$

where $p \in \mathbb{P}$, $x, x_1, x_2 \in \mathbb{V}_{\mathsf{el}}$ are element variables and $X \in \mathbb{V}_{\mathsf{set}}$ is a set variable. We interpret MSO on Kripke paths $\mathcal{P} = (\mathbb{N}, \to, \rho)$, where $\to$ is the successor function on $\mathbb{N}$ (the interpretation for S), and the labeling function ρ gives the interpretation of the unary predicate $p \in \mathbb{P}$ as $\{n \in \mathbb{N} \mid p \in \rho(n)\}$. This logic is also known as *the monadic second-order theory of S1S* (see [40]).

To obtain CMSO (over the signature τ) we extend S1S MSO by atomic formulas that describe local constraints over the concrete domain, that we call **atomic constraints**. These are built using the relations of the signature τ and the register variables from Reg. Atomic constraints have the following shape:

$$R(S^{i_1} r_1, \ldots, S^{i_k} r_k)(x) \ , \tag{3.2}$$

where $R \in \tau$ has arity k, $r_1, \ldots, r_k \in \mathsf{Reg}$, $i_1, \ldots, i_k \in \mathbb{N}$ and $x \in \mathbb{V}_{\mathsf{el}}$. Here the successor function S, with exponent i, is used to indicate that we are referring to the value of a register variable r in the i-th successor positions of the current one. Atomic constraints (an the whole CMSO) are interpreted on $\mathcal{D}$-decorated Kripke paths, for some τ-structure $\mathcal{D}$. The idea is that for a $\mathcal{D}$-KP $\mathfrak{P}$ and a position n of such path, $R(S^{i_1} r_1, \ldots, S^{i_k} r_k)(n)$ will hold if the k-tuple formed by the values assigned to the register variables r_j at position $n + i_j$ belongs to the relation R in $\mathcal{D}$.

Remark 3.1. The constraints which we introduce in (3.2) are the exact analogous of the ones from (1.1) on page 5 presented in the introduction in the context of LTL, only transported to the realm of MSO. Here we use the successor function (S) instead of the next operator (X) to point to a register variable in the next position. We also have a free variable x which represents the position at which we want to apply the constraint. We don't need this in LTL, as only the temporal operators X and U are used to navigate the models.

The constraints that we have just introduced are **local** in the sense that we can only compare concrete values assigned to registers variables at a fixed distance. In fact, given $\vartheta = R(S^{i_1} r_1, \ldots, S^{i_k} r_k)$, we can define $d(\vartheta) = \max\{i_1, \ldots, i_k\}$ to be the depth of ϑ.

As already mentioned, CMSO-formulas are interpreted over $\mathcal{D}$-Kripke paths for some τ-structure $\mathcal{D} = (D, I)$. Let $\mathfrak{P}$ be a $\mathcal{D}$-KP with underlying Kripke path $\mathcal{P} = (\mathbb{N}, \to, \rho)$.

So let $\eta : (\mathbb{V}_{\mathsf{el}} \cup \mathbb{V}_{\mathsf{set}}) \to (\mathbb{N} \cup 2^{\mathbb{N}})$ be a valuation function mapping element variables to positions and set variables to sets of positions respectively. The satisfaction relation $\models_{\mathsf{CMSO}}$ is mostly defined as expected, and we only present the most interesting cases below:

- $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} p(x)$ iff $p \in \rho(\eta(x))$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} x_1 = S(x_2)$ iff $\eta(x_1) = \eta(x_2) + 1$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} x \in X$ iff $\eta(x) \in \eta(X)$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} R(S^{i_1}r_1, \ldots, S^{i_k}r_k)(x)$ iff

$$(\gamma(\eta(x) + i_1, r_1), \ldots, \gamma(\eta(x) + i_k, r_k)) \in I(R).$$

For a CMSO-formula ψ the satisfaction relation only depends on the variables occurring freely in ψ. This motivates the following notation. If $\psi(X_1, X_2, \ldots, X_m)$ is an CMSO-formula where $X_1, \ldots, X_m$ are the only free variables, we write $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m)$ if and only if, for every valuation function η such that $\eta(X_i) = A_i$, we have $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \psi$. Moreover, we write $\models$ instead of $\models_{\mathsf{CMSO}}$ if no confusion arises.

We use some abbreviations in CMSO with the obvious semantics. In particular, we write formulas like $p(x+1)$ for $p \in \mathbb{P}$, to replace $\exists y\, (y = S(x) \wedge p(y))$, stating that the node p is satisfied in the position following x, or its generalization $p(x+i)$ for $i \in \mathbb{N}$.

Example 3.2. Consider the following CMSO-formula over the signature $\{\equiv, <\}$ (we use the infix notation for $\equiv$ and $<$):

$$\forall x\ [p(x) \wedge (r \equiv Sr)(x)] \vee [q(x) \wedge (r < Sr)(x)]\ .$$

This formula states that in all positions of a possible model, either p holds and the value of register variable r is kept equal in the next state ($r \equiv Sr$), or q holds and the value of r is increased in the next state ($r < Sr$). Interpreted over $(\mathbb{Z}, <)$-decorated Kripke paths, this formula satisfied, for instance, by the following model:

$$(p, 2)(p, 2)(q, 2)(q, 3)(p, 5)(q, 5) \ldots .$$

3.2 Constraint ECTL^* (CECTL^*)

We define CECTL^* (over the signature τ) by the following grammar:

$$\varphi ::= \mathsf{E}\psi(\underbrace{\varphi, \ldots, \varphi}_{\text{m times}}) \mid (\varphi \wedge \varphi) \mid \neg\varphi \tag{3.3}$$

where $\psi(X_1, \ldots, X_m)$ is a CMSO-formula over the signature τ in which only the set variables $X_1, \ldots, X_m \in \mathbb{V}_{\mathsf{set}}$ are allowed to occur freely.

CECTL^*-formulas are evaluated over some node of a $\mathcal{D}$-decorated Kripke structure where $\mathcal{D}$ is some τ-structure . Let $\mathfrak{K}$ be such a $\mathcal{D}$-KS with underlying Kripke structure $\mathcal{K} = (S, \to, \rho)$. Given $s \in S$, for a CECTL^*-formula φ, we define the semantics for the existential quantification as follows (the other cases are trivial):

Definition 3.3. $(\mathfrak{K}, s) \models \mathsf{E}\psi(\varphi_1, \ldots, \varphi_m)$ if and only if there exists an infinite path $P = s_0 s_1 s_2 \ldots$ with $s_0 = s$, whose corresponding $\mathcal{D}$-Kripke Path $\mathfrak{P}$ satisfies $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m)$ where $A_i = \{n \in \mathbb{N} \mid (\mathfrak{K}, s_n) \models \varphi_i\}$ for $1 \leq i \leq m$.

The intuition behind this, is that the sets $A_1, \ldots, A_m$ collect all the positions of the path $\mathfrak{P}$ in which the formulas $\varphi_1, \ldots, \varphi_m$ hold. The free variables $X_1, \ldots, X_m$ from ψ are then interpreted as $A_1, \ldots, A_m$, so that the formula $x \in X_i$ stands to mean that x should belong to the sets of positions which satisfy φ_i.

Note that for checking $(\mathfrak{K}, s) \models \varphi$ we may ignore all propositions $p \in \mathbb{P}$ and all registers $r \in \mathsf{Reg}$ that do not occur in φ.

Remark 3.4. The reader might miss atomic propositions $p \in \mathbb{P}$ in (3.3). They can be obtained using CMSO. More precisely, MSO can express the fact that a position x is the initial position of a path using the formula $\mathsf{pos}_0(x) = \forall y\, (x \neq S(y))$, then the CECTL^*-formula $\mathsf{E}[\exists x\, (\mathsf{pos}_0(x) \wedge p(x))]$ states that from the current node originates a path whose first node satisfies p, i.e., the current node satisfies p.

Note that the role of the concrete domain $\mathcal{D}$ and of the valuation function γ, for both CMSO and CECTL^* are restricted to the semantics of atomic constraints. Ordinary ECTL^*-formulas are defined as in (3.3), with the exception that in $\mathsf{E}\psi(\varphi, \ldots, \varphi)$, the formula $\psi(X_1, \ldots, X_m)$ is a classical MSO formula, i.e., without atomic constraints.

ECTL^* is interpreted over a pair $(\mathcal{K}, s)$, where $\mathcal{K}$ is a Kripke structure and s an element of its domain, and the rules are the same as above (just ignoring the concrete domain and γ).

We define the usual abbreviations:

$$\begin{aligned}
\vartheta_1 \vee \vartheta_2 &:= \neg(\neg\vartheta_1 \wedge \neg\vartheta_2)\,, \\
\vartheta_1 \to \vartheta_2 &:= \neg\vartheta_1 \vee \vartheta_2\,, \\
\mathsf{A}\psi &:= \neg\mathsf{E}\neg\psi \text{ (universal path quantifier)}\,, \\
\forall x\, \psi &:= \neg\exists x\, \neg\psi\,, \\
\forall X\, \psi &:= \neg\exists X\, \neg\psi\,.
\end{aligned}$$

Note that $(\mathfrak{K}, s) \models \mathsf{A}\psi(\varphi_1, \ldots, \varphi_m)$ if and only if for all infinite paths in $\mathcal{K}$ $P = s_0 s_1 s_2 \cdots$ with $s_0 = s$, we have for the corresponding decorated KP $\mathfrak{P}$: $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m)$ where $A_i = \{n \in \mathbb{N} \mid (\mathfrak{K}, s_n) \models \varphi_i\}$ for $1 \leq i \leq m$.

Using this extended set of operators we can put every formula into a semantically equivalent **negation normal form**, where $\neg$ only occurs in front of atomic CMSO-formulas (i.e., formulas of the form $p(x)$, $x = S(y)$, $x \in X$ or atomic constraints).

Remark 3.5. If $\psi(X_1, \ldots, X_m)$ is a CMSO-subformula which occurs after a path quantifier in a CECTL^*-formula, as for instance $\mathsf{E}\psi(\varphi_1, \ldots, \varphi_m)$, to obtain the negation normal form we additionally eliminate negated subformulas as $\neg(x \in X_i)$ where X_i is one of the set variables $X_1, \ldots, X_m$ that occurs freely in φ as follows: we replace ϑ with the equivalent formula $\mathsf{E}\psi'(\varphi_1, \ldots, \varphi_m, \neg\varphi_1, \ldots, \neg\varphi_m)$, where $\psi'(X_1, \ldots, X_m, Y_1, \ldots, Y_m)$ is obtained from ψ by replacing all occurrences of $\neg(x \in X_i)$ by $x \in Y_i$ for $1 \leq i \leq m$.

We give in the following some examples of classical CTL^* expressible specifications formulated in ECTL^*. Recall that in monadic second-order logic the binary predicate $<$ can be derived from the successor function.

Example 3.6. Response to an impulse: In all computations, every occurrence of p is eventually followed by an occurrence of q.

$$\mathsf{CTL}^* : \mathsf{AG}(p \to \mathsf{F}q) \qquad \mathsf{ECTL}^* : \mathsf{A}[\forall x\,(p(x) \to \exists y\,(x < y \wedge q(y)))].$$

Absence of unsolicited responses: In all computations q does not occur unless preceded by p.

$$\mathsf{CTL}^* : \mathsf{A}(\mathsf{F}q \to (\neg q)\mathsf{U}p) \qquad \mathsf{ECTL}^* : \mathsf{A}[\forall x\,(q(x) \to \exists y\,(y \leq x \wedge p(y)))].$$

Existence of a stabilizing computation: There is a computation where eventually p holds in every state.

$$\mathsf{CTL}^* : \mathsf{EFG}p \qquad \mathsf{ECTL}^* : \mathsf{E}[\exists x\,\forall y\,(x < y \to p(y))].$$

We illustrate in the following example that the nesting of path quantifiers in a CTL^*-formula results in the nesting of MSO-formulas inside the corresponding ECTL^*-formula.

Example 3.7. The CTL^*-formula $\mathsf{EG}(p \to \mathsf{AX}q)$ expresses the existence of a path P such that every successor of a p-labeled node on P is labeled with q. Let φ be the ECTL^*-formula stating that on all paths q holds in the next state: $\varphi = \mathsf{A}\exists x\ (\mathsf{pos}_0(x) \wedge q(x+1))$, where we use pos_0 to denote the first position of a path (see Remark 3.4). Then the required property is expressed by the formula $\mathsf{E}\psi(\varphi)$, where $\psi(X) = \forall z\,(p(z) \to z \in X)$. All together we obtain the formula

$$\mathsf{E}\forall z\ \big(p(z) \to z \in \big[\mathsf{A}\,\exists x\ (\mathsf{pos}_0(x) \wedge q(x+1))\,\big]\big)\,.$$

In the following example we exploit the higher expressive power of ECTL^* to express a system requirement which cannot be formulated in CTL^*.

Example 3.8. There is a computation path where p holds in all even positions. The following MSO-formula describes the set X of even positions of a path:

$$\mathsf{even}(X) := \exists x \ (\mathsf{pos}_0(x) \wedge x \in X) \wedge \forall x \ (S(x) \in X \leftrightarrow \neg x \in X)\,.$$

The following ECTL^*-formula describes the required property:

$$\mathsf{E}[\exists X \,\mathsf{even}(X) \wedge \forall z \,(z \in X \to p(z))].$$

Wolper [45] proved that no CTL^*-formula expresses this property.

Example 3.9. We show that it is possible, using constraints over $(\mathbb{Z}, <)$, to write a CECTL^*-formula which can only be satisfied by an infinite $(\mathbb{Z}, <)$-Kripke structure (we use the infix notation for $<$):

$$\varphi = \mathsf{E}[\forall x \,(r < Sr)(x)]. \tag{3.4}$$

We are forcing the existence of a $(\mathbb{Z}, <)$-Kripke path $\mathfrak{P}$ along which the value of the register variable r monotonically decreases, and this ensures that the domain of $\mathfrak{P}$ is infinite.

We remark that the last example shows that CECTL^* is strictly more expressive than ECTL^* in the following sense: Let us denote with $L(\varphi)$ the set of all underlying Kripke structures of $(\mathbb{Z}, <)$-KSs which satisfy φ. Then $L(\varphi)$ for φ from (3.4) is not empty and it does not contain any finite Kripke structure. On the other hand it is well known that ECTL^* enjoys the finite model property, and therefore cannot define $L(\varphi)$.

3.3 CECTL^* has the Tree Model Property

In the following we show that every satisfiable CECTL^*-formula always has a nice model, namely a tree-model where the branching degree is bounded by a constant that can be computed from the formula. The proof of this property is similar to the proof of the tree model property for ECTL^* or CTL^*. One has to additionally deal with the constraints, but this does not create particular problems.

Lemma 3.10. *Let $\mathfrak{K} = (\mathcal{D}, \mathcal{K}, \gamma)$ be a $\mathcal{D}$-Kripke structure, s_0 a node of $\mathcal{K}$ and φ a CECTL^*-formula. If $P = s_0 s_1 \cdots s_n$ is an element of $\mathsf{Unf}(\mathfrak{K}, s_0)$, then $(\mathfrak{K}, s_n) \models \varphi$ if and only if $(\mathsf{Unf}(\mathfrak{K}, s_0), P) \models \varphi$.*

Proof. The proof is an easy induction on the structure of the formula using the fact that any $\mathcal{D}$-Kripke path in $\mathfrak{K}$ starting at a node reachable from s_0 corresponds to a $\mathcal{D}$-Kripke path in $\mathsf{Unf}(\mathfrak{K}, s_0)$ and vice versa. $\square$

For similar reasons, we can duplicate subtrees of a $\mathcal{D}$-Kripke tree $\mathfrak{T}$ without affecting the set of satisfied formulas which allows to increase the branching degree of the model arbitrarily.

Lemma 3.11. *Let $\mathfrak{T}$ be a $\mathcal{D}$-Kripke tree. There exists another $\mathcal{D}$-KT $\mathfrak{T}_\omega$ such that*

- *every node of $\mathfrak{T}_\omega$ has infinitely many successors,*
- *$\mathfrak{T}$ and $\mathfrak{T}_\omega$ satisfy the same CECTL^*-formulas at their roots, and*
- *if s is a node and $\varphi = \mathsf{E}\psi(\varphi_1, \ldots, \varphi_k)$ is a formula such that $(\mathfrak{T}_\omega, s) \models \varphi$, then there are infinitely many paths starting in s which witness the path quantifier, i.e., there are infinitely many paths $P = s_0 s_1 s_2 \cdots$ in $\mathfrak{T}_\omega$ with $s_0 = s$ and $\mathfrak{P} \models \psi(A_1, \ldots, A_k)$ for $A_i = \{n \in \mathbb{N} \mid (\mathfrak{T}_\omega, s_n) \models \varphi_i\}$ $(1 \leq i \leq k)$.*

Proof. Let $\mathfrak{T} = (\mathcal{D}, \mathcal{T}, \gamma)$ be some $\mathcal{D}$-Kripke tree where $\mathcal{T} = (T, \rightarrow, \rho)$. Without loss of generality we assume that $T \subseteq S^*$ for some set S such that $\rightarrow$ is the extension of words over S by one letter.

We define a Kripke tree $\mathcal{T}_\omega = (T_\omega, \rightarrow_\omega, \rho_\omega)$ and a $\mathcal{D}$-Kripke tree $\mathfrak{T}_\omega = (\mathcal{D}, \mathcal{T}_\omega, \gamma_\omega)$ where

- $T_\omega \subseteq (S \times \mathbb{N})^*$ such that $s = (s_1, n_1)(s_2, n_2) \ldots (s_i, n_i) \in T_\omega$ if and only if $\pi(s) = s_1 s_2 \ldots s_i \in T$ (π is the projection to the first component on $(S \times \mathbb{N})^*$),
- $\rightarrow_\omega$ is extension by one element from $S \times \mathbb{N}$,
- $\rho_\omega(s) = \rho(\pi(s))$, and
- $\gamma_\omega(s, r) = \gamma(\pi(s), r)$ for all $r \in \mathsf{Reg}$.

Since $\mathfrak{T}_\omega$ is basically an infinite copy of $\mathfrak{T}$ everywhere, where projection to the first component translates between the two structures, rather simple inductions prove the following facts.

1. Let P_ω be an infinite path in $\mathfrak{T}_\omega$, $A_1, A_2, \ldots, A_k$ sets of positions of P_ω and $\varphi \in \mathsf{CMSO}$. Let $\mathfrak{P}_\omega$ be the $\mathcal{D}$-KP corresponding to P_ω and let $P = \pi(P_\omega)$ be the path in $\mathfrak{T}$ obtained by element-wise projection of P_ω to the first component. Finally, let $\mathfrak{P}$ be the $\mathcal{D}$-KP corresponding to P. Then we have
$$\mathfrak{P}_\omega \models_{\mathsf{CMSO}} \varphi(A_1, \ldots, A_k) \iff \mathfrak{P} \models_{\mathsf{CMSO}} \varphi(\pi(A_1), \ldots, \pi(A_k)) \; .$$

2. For $s = (s_1, n_1)(s_2, n_2) \dots (s_k, n_k) \in T_\omega$ and φ some CECTL^*-formula we have
$$(\mathfrak{T}_\omega, s) \models \varphi \iff (\mathfrak{T}, \pi(s)) \models \varphi \ .$$

The second part implies that every path quantifier that is satisfied at some node in $\mathfrak{T}_\omega$ is witnessed by infinitely many paths in $\mathfrak{T}_\omega$ starting at this node. $\square$

From now on, let $\#_{\mathsf{E}}(\varphi)$ denote the number of different subformulas of the form $\mathsf{E}\psi$ in the CECTL^*-formula φ. We can state the following strong tree model property:

Theorem 3.12. *Let φ be a CECTL^*-formula in negation normal form and let $\mathcal{D} = (D, I)$ be a τ-structure. Then φ is satisfiable by a $\mathcal{D}$-Kripke Structure if and only if there is a $\mathcal{D}$-*Kripke $(\#_{\mathsf{E}}(\varphi) + 1)$-tree *$\mathfrak{T}$ with root r and $(\mathfrak{T}, r) \models \varphi$.*

Proof. Let $e = \#_{\mathsf{E}}(\varphi) + 1$. Due to the previous lemmas, we can assume that $\mathfrak{K} = (\mathcal{D}, \mathcal{K}, \gamma)$ is a $\mathcal{D}$-Kripke tree with $\mathcal{K} = (S, \rightarrow, \rho)$ a KT with root r where every node $s \in S$ has infinitely many successors such that for every formula $\mathsf{E}\psi$ with $(\mathfrak{K}, s) \models \mathsf{E}\psi$ there are infinitely many pairwise disjoint (except for node s) paths starting at s that witness this path quantifier. We prune $\mathfrak{K}$ such that it is isomorphic to a $\mathcal{D}$-Kripke e-tree model of φ.

We inductively define the domain S' of our new tree. For the initial step, choose a $\mathcal{D}$-KP $\mathfrak{P}$ of $\mathfrak{K}$ arbitrarily and add its domain to S'.

For the inductive step we repeat the following procedure until every node has exactly e successors. Let $s \in S'$ be a node with less than e successors (in S'). Our inductive definition ensures that s then has exactly 1 successor. Let $\mathsf{E}\psi_1(\varphi^1_1, \dots, \varphi^1_{m_1}), \dots, \mathsf{E}\psi_k(\varphi^k_1, \dots, \varphi^k_{m_k})$ be the existential subformulas of φ which hold true in $(\mathfrak{K}, s)$. Then for each $1 \leq j \leq k$ there is a $\mathcal{D}$-Kripke path $\mathfrak{P}_j$ in $\mathfrak{K}$ with $\mathfrak{P}_j(0) = s$ and disjoint from $S' \setminus \{s\}$ such that $\mathfrak{P}_j \models_{\mathsf{CMSO}} \psi(A^j_1, \dots, A^j_{m_j})$, where $A^j_i = \{n \in \mathbb{N} \mid (\mathfrak{K}, \mathfrak{P}_j(n)) \models \varphi^j_i\}$. For each $j \in \{k+1, k+2, \dots, e-1\}$ choose further $\mathcal{D}$-Kripke paths $\mathfrak{P}_j$ of $\mathfrak{K}$ with $\mathfrak{P}_{\mathsf{j}}(0) = s$ that are disjoint from $S' \setminus \{s\}$ and the other paths (except for their origin s). Add the domains of $\mathfrak{P}_1, \mathfrak{P}_2, \dots, \mathfrak{P}_{e-1}$ to S' and continue the construction with the next node of the resulting S' that has only one successor.

The limit of this process results in a subset $S' \subseteq S$ such that S' induces a Kripke e-subtree $\mathcal{T} = \mathcal{K}_{\restriction S'}$ and a $\mathcal{D}$-Kripke subtree $\mathfrak{T} = (\mathcal{D}, \mathcal{T}, \gamma')$, where $\gamma' = \gamma_{\restriction (S' \times \mathsf{Reg})}$. We prove $(\mathfrak{T}, r) \models \varphi$ by showing the following stronger claim using structural induction on the formula φ.

(i) Given a subformula $\vartheta \in \mathsf{CECTL}^*$ of φ and $s \in S'$ such that $(\mathfrak{K}, s) \models \vartheta$, then $(\mathfrak{T}, s) \models \vartheta$.

(ii) For all CMSO-formulas $\psi(X_1, \ldots, X_m)$ that are subformulas of φ, all $\mathcal{D}$-Kripke paths $\mathfrak{P}$ in $\mathfrak{T}$ and all sets $A_1, \ldots, A_m, B_1, \ldots, B_m$ of positions of $\mathfrak{P}$ such that $A_j \subseteq B_j$ for all $1 \leq j \leq m$, and all valuation functions η,

$$(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m) \implies (\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \psi(B_1, \ldots, B_m) \,, \tag{3.5}$$

where we assume that X_i only occurs freely in ψ (we can rename bounded occurrences).

Recall that φ is in negation normal form. Hence, the proof only needs to consider the following cases, starting from (i):

- The cases $\vartheta = \varphi_1 \wedge \varphi_2$ and $\vartheta = \varphi_1 \vee \varphi_2$ are straightforward by induction.
- Let $\vartheta = \mathsf{E}\psi(\varphi_1, \ldots, \varphi_m)$. Note that only $X_1, \ldots, X_m$ are allowed to occur freely in ψ. Let $s \in S'$ be such that $(\mathfrak{K}, s) \models \mathsf{E}\psi(\varphi_1, \ldots, \varphi_m)$. By construction of $\mathfrak{T}$, there is a $\mathcal{D}$-Kripke path $\mathfrak{P}$ in $\mathfrak{T}$ (which is simultaneously in $\mathfrak{K}$) with $\mathfrak{P}(0) = s$ such that $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m)$, where $A_i = \{n \in \mathbb{N} \mid (\mathfrak{K}, \mathfrak{P}(n)) \models \varphi_i\}$. Let $B_i = \{\mathfrak{P}(n) \mid n \geq 0, (\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i\}$. Applying the inductive hypothesis from (i) to each φ_i, we have $A_i \subseteq B_i$. Thus, using the inductive hypothesis from (ii) for ψ, we obtain $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(B_1, \ldots, B_m)$. Hence, $(\mathfrak{T}, s) \models \mathsf{E}\psi(\varphi_1, \ldots, \varphi_m)$ as desired.
- Let $\vartheta = \mathsf{A}\psi(\varphi_1, \ldots, \varphi_m)$. In order to get $(\mathfrak{T}, s) \models \vartheta$, we need to show that every $\mathcal{D}$-Kripke path $\mathfrak{P}$ in $\mathfrak{T}$ that starts in s satisfies $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(B_1, \ldots, B_m)$, where $B_i = \{n \in \mathbb{N} \mid (\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i\}$. So let $\mathfrak{P}$ be a $\mathcal{D}$-Kripke path in $\mathfrak{T}$ (and hence in $\mathfrak{K}$ as well) with $\mathfrak{P}(0) = s$. Since $(\mathfrak{K}, s) \models \vartheta$, we can deduce that $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m)$, where $A_i = \{n \in \mathbb{N} \mid (\mathfrak{K}, \mathfrak{P}(n)) \models \varphi_i\}$. By inductive hypothesis for point (i), we conclude that $A_i \subseteq B_i$ and by the inductive hypothesis for point (ii) we conclude that $\mathfrak{P} \models_{\mathsf{CMSO}} \psi(B_1, \ldots, B_m)$. Hence, we get $(\mathfrak{T}, s) \models \vartheta$.

This completes the inductive step for point (i). We continue with the inductive step for point (ii), i.e., for CMSO-subformulas ψ of φ. To simplify notation we write η_A for $\eta[(X_j \to A_j)_{1 \leq j \leq m}]$ and η_B for $\eta[(X_j \to B_j)_{1 \leq j \leq m}]$.

- If $\psi = p(x)$ ($\neg p(x)$, respectively) for some atomic proposition $p \in \mathbb{P}$, then by definition we have: $(\mathfrak{P}, \eta_A) \models_{\mathsf{CMSO}} p(x)$ if and only if $p \in \rho(\eta_A(x))$ if and only if $p \in \rho(\eta_B(x))$ if and only if $(\mathfrak{P}, \eta_B) \models_{\mathsf{CMSO}} p(x)$.
- Similarly, if ψ is of the form $x = S(y)$, $x \neq S(y)$, $x \in X$, or $x \notin X$ for $X \in \mathbb{V}_{\mathsf{set}} \setminus \{X_1, \ldots, X_m\}$, we have $(\mathfrak{P}, \eta_A) \models_{\mathsf{CMSO}} \psi$ if and only if $(\mathfrak{P}, \eta_B) \models_{\mathsf{CMSO}} \psi$ because ψ does not depend on the interpretations of $X_1, \ldots, X_m$.

- Let $\psi = (x \in X_i)$ for some $1 \leq i \leq m$. Then $(\mathsf{P}, \eta_A) \models_{\mathsf{CMSO}} x \in X_i$ implies $\eta_A(x) \in A_i$ and whence (using $A_i \subseteq B_i$) $\eta_B(x) \in B_i$, i.e., $(\mathsf{P}, \eta_B) \models_{\mathsf{CMSO}} x \in X_i$.
- The cases $\psi = \psi_1 \vee \psi_2$ and $\psi = \psi_1 \wedge \psi_2$ follow from inductive hypothesis.
- Assume that $\psi = \exists x\, \psi_1$ and that $(\mathsf{P}, \eta_A) \models \psi$. Then there is a position n of P such that $(\mathsf{P}, \eta_A[x \mapsto n]) \models_{\mathsf{CMSO}} \psi_1$. By the inductive hypothesis this implies $(\mathsf{P}, \eta_B[x \mapsto n]) \models_{\mathsf{CMSO}} \psi_1$ and therefore $(\mathsf{P}, \eta_B) \models \exists x\, \psi_1$.
- The cases $\psi = \forall x\, \psi_1$, $\psi = \exists X\, \psi_1$ and $\psi = \forall X\, \psi_1$ (note that X must be different from $X_1, \ldots, X_m$ since X only occurs freely in ψ) are proven analogously to the previous case.
- Let $\psi = R(S^{i_1} r_1, \ldots, S^{i_k} r_k)(x)$ or its negation. Since η_A and η_B agree on x, this case is trivial.

This concludes the proof of the theorem. $\square$

Chapter 4

Satisfiability of CECTL^*

Let us now formally define the notion of satisfiability. Chosen a domain $\mathcal{D}$ we ask, given a CECTL^*-formula, whether it has a model having $\mathcal{D}$ as concrete domain (i.e. a $\mathcal{D}$-Kripke structure).

Definition 4.1. We say that a CECTL^*-formula φ is **$\mathcal{D}$-satisfiable** if there is a $\mathcal{D}$-Kripke structure K with underlying KS $\mathcal{K} = (S, \to, \rho)$ and a node $v \in S$ such that $(\mathsf{K}, v) \models \varphi$.

With $\mathcal{D}$-SAT we denote the following computational problem: *Is a given formula $\varphi \in \mathsf{CECTL}^*$ $\mathcal{D}$-satisfiable?*

In this section we will show that the above problem is decidable, provided that the domain $\mathcal{D}$ satisfies certain properties.

4.1 The EHD-Property

We now introduce one of the central notions of this work: the **EHD-property**. EHD stands for "the existence of a homomorphism is definable". This is a property of a relational structure $\mathcal{A}$, expressing the ability of a logic $\mathcal{L}$ to distinguish between those structures $\mathcal{B}$ which can be mapped to $\mathcal{A}$ by a homomorphism ($\mathcal{B} \preceq \mathcal{A}$) and those who cannot. Recall the definition of homomorphism (Definition 2.6).

Definition 4.2. Let $\mathcal{L}$ be a logic (e.g. MSO). A σ-structure $\mathcal{A}$ has the property $\mathsf{EHD}(\mathcal{L})$ if there is a computable function that maps every finite subsignature $\tau \subseteq \sigma$ to an $\mathcal{L}$-sentence φ_τ such that for every countable τ-structure $\mathcal{B}$ we have:

$$\mathcal{B} \preceq \mathcal{A} \iff \mathcal{B} \models \varphi_\tau \ .$$

Example 4.3. The structure $\mathcal{Q} = (\mathbb{Q}, <, \equiv)$, where $\equiv$ is equality[1], has the property $\mathsf{EHD}(\mathsf{WMSO})$ (and $\mathsf{EHD}(\mathsf{MSO})$). In fact we can show that, for any countable $\{<, \equiv\}$-structure $\mathcal{B} = (B, J)$, $\mathcal{B} \preceq \mathcal{Q}$ if and only if there does not exist $(a, b) \in J(<)$ such that $(b, a) \in (J(<) \cup J(\equiv) \cup J(\equiv)^{-1})^*$.

If we look at $\mathcal{B}$ as a graph with two kinds of edges, $<$-edges and $\equiv$-edges, we are ultimately excluding the presence of cycles with at least one $<$-edge. It is clear why the presence of such cycles would make it impossible to build a homomorphism into $\mathcal{Q}$. It is not hard to show that the converse holds: it is enough to know that $\mathcal{B}$ does not present such cycles to guarantee the existence of a homomorphism into $\mathcal{Q}$ (see [32]). This particular acyclicity condition can be easily expressed in WMSO or MSO using the reach-construction from Example 2.20.

Notation 4.4. Throughout this work, we will mostly make use of the property $\mathsf{EHD}(\mathcal{L})$ where $\mathcal{L}$ is $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ or one of its fragments. In this case, we will sometimes refer to it simply as to the EHD-property.

4.2 The EHD Method

The main result of this section gives a criterion on the concrete domain $\mathcal{D}$ that implies decidability of $\mathcal{D}$-SAT. To state this criterion, we need one further technical condition:

Definition 4.5. A σ-structure $\mathcal{A} = (A, I)$ is **negation-closed** if, for every $R \in \sigma$, the complement of $I(R)$ is effectively definable by a positive existential first-order formula, i.e., if there is a computable function that maps each relation symbol $R \in \sigma$ to a positive existential first-order formula $\varphi_R(x_1, \ldots, x_{\mathsf{ar}(R)})$ (i.e., a formula that is built up from atomic formulas using $\wedge$, $\vee$, and $\exists$) such that

$$A^{\mathsf{ar}(R)} \setminus I(R) = \{(a_1, \ldots, a_{\mathsf{ar}(R)}) \mid \mathcal{A} \models \varphi_R(a_1, \ldots, a_{\mathsf{ar}(R)})\}.$$

Example 4.6. The structure $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a \in \mathbb{Z}}, (\equiv_{a,b})_{0 \leq a < b})$ from (1.2) on page 7 is negation-closed (to improve readability we write $x \equiv a$ instead of $\equiv_a(x)$ and similarly for $\equiv_{a,b}$). We have in fact:

- $\neg x \equiv y$ if and only if $x < y \vee y < x$.
- $\neg x < y$ if and only if $x \equiv y \vee y < x$.

[1] A remark concerning the equality relation should be made at this point. In the structure $\mathcal{Q}$, we mean with $\equiv$ the equality relation, whereas in $\mathcal{B}$, the relation $J(\equiv)$ can be any binary relation. Nevertheless, in MSO we have a built-in equality, see the MSO-syntax (2.1) on page 16. This is one more reason to denote the equality relation as part of a structure with $\equiv$ instead of $=$. In the structure $\mathcal{B}$ the MSO-formulas $x = y$ and $x \equiv y$ have, in general, different semantics.

- $\neg x \equiv a$ if and only if $\exists y \in \mathbb{Z}\,(y = a \wedge (x < y \vee y < x))$.
- $\neg x \equiv a \bmod b$ if and only if $x \equiv c \bmod b$ for some $0 \leq c < b$ with $a \neq c$:

$$\bigvee_{\substack{0 \leq c < b \\ a \neq c}} x \equiv c \bmod b \ .$$

We are now ready to present the first main result of our work:

Theorem 4.7. *Let σ be a countable signature and let $\mathcal{D}$ be a σ-structure which:*

- *is negation-closed, and*
- *has the property* $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$*.*

Then the problem $\mathcal{D}$-SAT is decidable.

This is a very broad and general result: It is enough, given any concrete domain $\mathcal{D}$, to prove that it enjoys these two properties, to guarantee decidability of the $\mathcal{D}$-SAT problem for CECTL^*. In the following, we refer to this criterion as **the EHD method**.

Let us fix an CECTL^*-formula φ in negation normal form and a negation-closed σ-structure $\mathcal{D}$ for the rest of this section. We want to check whether φ is $\mathcal{D}$-satisfiable. First, we reduce our problem to formulas in strong negation normal form:

Definition 4.8. We say that a CECTL^*-formula φ is in **strong negation normal form** if it is in negation normal form and additionally there is no subformula $\neg\vartheta(x)$ where $\vartheta(x)$ is an atomic constraint.

Lemma 4.9. *If $\mathcal{D} = (D, I)$ is negation-closed, given a CECTL^*-formula φ one can compute a CECTL^*-formula $\hat{\varphi}$ in strong negation normal form such that φ is $\mathcal{D}$-satisfiable if and only if $\hat{\varphi}$ is $\mathcal{D}$-satisfiable.*

Proof. We can assume that φ is in negation normal form. Using induction, it suffices to eliminate one negated atomic constraint $\vartheta(x) = \neg R(S^{i_1}r_1, \ldots, S^{i_k}r_k)(x)$ in φ, where $k = \mathsf{ar}(R)$. Let $d = \max\{i_1, \ldots, i_k\}$ be the depth of the constraint ϑ. Since $\mathcal{D}$ is negation-closed, we can compute a positive quantifier-free first-order formula $\psi(y_1, y_2, \ldots, y_k, z_1, z_2, \ldots, z_m)$ such that

$$(a_1, \ldots, a_k) \notin I(R) \iff \mathcal{D} \models \exists z_1 \cdots \exists z_m \, \psi(a_1, \ldots, a_k, z_1, \ldots, z_m) \ .$$

Let $s_1, \ldots, s_m \in \mathsf{Reg}$ be fresh register variables not occurring in φ. We define the CECTL^*-formula $\hat{\varphi}$ by replacing in φ every occurrence of the negated constraint ϑ by the formula

$$\psi(S^{i_1} r_1, \ldots, S^{i_k} r_k, S^d s_1, \ldots, S^d s_m)^2 ,$$

i.e., we replace in the positive quantifier-free formula $\psi(y_1, \ldots, y_k, z_1, \ldots, z_m)$ every occurrence of a variable y_j (respectively, z_j) by $S^{i_j} r_j$ (respectively, $S^d s_j$).

The idea is the following: Given a negated constraint ϑ, we substitute it with a boolean combination of positive ones, involving the same registers appearing in ϑ ($S^{i_j} r_j$) and new - existentially quantified - ones. We have then to "install" new registers ($s_1, \ldots, s_m$) which we place at the deepest node involved in the constraint (depth d) to use as these existentially quantified variables.

Now we have to prove that

$$\varphi \text{ is } \mathcal{D}\text{-satisfiable} \iff \hat{\varphi} \text{ is } \mathcal{D}\text{-satisfiable} \quad .$$

Proof of $\Longrightarrow$. If φ is $\mathcal{D}$-satisfiable, then by Theorem 3.12 there is a $\mathcal{D}$-Kripke e-tree $\mathfrak{T} = (\mathcal{D}, \mathcal{T}, \gamma)$ with $(\mathfrak{T}, \varepsilon) \models \varphi$ and underlying Kripke e-tree $\mathcal{T} = ([1, e]^*, \to, \rho)$. We modify $\mathfrak{T}$ and obtain a new $\mathcal{D}$-Kripke tree $\mathfrak{S} = (\mathcal{D}, \mathcal{T}, \gamma')$ by defining the interpretation function γ' on the fresh register variables $s_1, \ldots, s_m$ (and leaving otherwise $\gamma' = \gamma$) as follows: Consider $w, v \in [1, e]^*$ such that $|v| = d$ and let v_p be the prefix of v of length i_p for $1 \leq p \leq k$,

- If $(\gamma(wv_1, r_1) \ldots, \gamma(wv_k, r_k)) \notin I(R)$, we can fix values $b_1, \ldots, b_m \in D$ such that
$$\mathcal{D} \models \psi(\gamma(wv_1, r_1) \ldots, \gamma(wv_k, r_k), b_1, \ldots, b_m) \ .$$
In this case we set $\gamma'(wv, s_q) = b_q$ for all $1 \leq q \leq m$.

- If $(\gamma(wv_1, r_1) \ldots, \gamma(wv_k, r_k)) \in I(R)$, we choose $\gamma'(wv, s_q) \in D$ arbitrarily for all $1 \leq q \leq m$.

- Finally, for all $w \in [1, e]^*$ such that $|w| < d$ we choose $\gamma'(wv, s_q) \in D$ arbitrarily for all $1 \leq q \leq m$.

By induction on the structure of φ we prove that $(\mathfrak{S}, \varepsilon) \models \hat{\varphi}$. All steps are trivial except for the case that the subformula is precisely $\vartheta(x) = \neg R(S^{i_1} r_1, \ldots, S^{i_k} r_k)(x)$. In this case let $P = p_0 p_1 p_2, \ldots$ be a path in $\mathcal{T}$ inducing the $\mathcal{D}$-Kripke path $\mathfrak{P}$ in $\mathfrak{T}$ and the $\mathcal{D}$-Kripke path $\mathfrak{R}$ in $\mathfrak{S}$ (which only differ for the values of γ and γ' on $s_1, \ldots, s_m$) and let η be a valuation function such that $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \vartheta(x)$. Thus, setting $p_j = \eta(x)$, we get $(\gamma(p_{j+i_1}, r_1), \ldots, \gamma(p_{j+i_k}, r_k)) \notin I(R)$. According

[2] In case ψ is a conjunction or disjunction of atomic constraints, they all apply to the same element variable x. For instance if $\psi(y_1, y_2) = y_1 < y_2 \vee y_2 < y_1$, we substitute $\vartheta(x)$ by $(r_1 < r_2)(x) \vee (r_2 < r_1)(x)$.

to our definition of γ', we have set $\gamma(p_{j+d}, s_q) = b_q$ for all $1 \leq q \leq m$, where $b_1, \ldots, b_m \in D$ such that

$$\mathcal{D} \models \psi(\gamma(p_{j+1}, r_1) \ldots, \gamma(p_{j+k}, r_k), b_1, \ldots, b_m) \; .$$

Thus, it follows that

$$(\mathfrak{R}, \eta) \models_{\mathsf{CMSO}} \psi(S^{i_1} r_1, \ldots, S^{i_k} r_k, S^d s_1, \ldots, S^d s_m)(x) \; ,$$

which concludes the first direction.

Proof of $\Longleftarrow$. In order to prove that φ is $\mathcal{D}$-satisfiable if $\hat{\varphi}$ is $\mathcal{D}$-satisfiable, let us assume that $\mathfrak{K}$ is a $\mathcal{D}$-Kripke Structure such that $(\mathfrak{K}, d) \models \hat{\varphi}$ for some node d. In order to show $(\mathfrak{K}, d) \models \varphi$ by induction on the structure of φ, we end up (after several trivial steps) with the following claim: For every path $P = p_0 p_1 p_2 \ldots$ in $\mathfrak{K}$ (inducing the $\mathcal{D}$-Kripke path $\mathfrak{P}$) and valuation function η,

$$(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \psi(S^{i_1} r_1, \ldots, S^{i_k} r_k, S^d s_1, \ldots, S^d s_m)(x) \tag{4.1}$$

implies $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \vartheta(x)$. Assuming (4.1) and $\eta(x) = p_t$, there are values, namely $\gamma(p_{t+d}, r_i)$ $(1 \leq i \leq m)$ witnessing

$$\mathcal{D} \models \exists z_1 \cdots \exists z_m \, \psi(a_1, \ldots, a_k, z_1, \ldots, z_m),$$

where $a_j = \gamma(p_{t+i_j}, r_i)$. By choice of ψ this implies that $\mathcal{D} \models \neg R(a_1, \ldots, a_k)$. Hence, we have $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \neg R(S^{i_1} r_1, \ldots, S^{i_k} r_k)(x)$, i.e., $(\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \vartheta(x)$. $\square$

Example 4.10. Consider the domain $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a \in \mathbb{Z}}, (\equiv_{a,b})_{0 \leq a < b})$.

Let $\varphi = \mathsf{E}[\forall x \, \neg(r = 3)(x)]$ be the CECTL^*-formula expressing the fact that there exists a path on which r never assumes value 3 (we write $r = 3$ instead of $\equiv_3 (r)$).

As we saw in Example 4.6, $\mathcal{Z}$ is negation-closed, and we can find an existentially quantified positive first order formula, namely

$$\psi(a) = \exists z \, (z = 3 \wedge (a < z \vee z < a)) \; ,$$

such that $\neg r = 3$ if and only if $\psi(r)$ holds.

The strong negation normal form of φ is

$$\hat{\varphi} = \mathsf{E}\big[\forall x \, \big(s = 3 \wedge (r < s \vee s < r)\big)(x)\big] \; .$$

As you can see we have introduced a new register variable s and used it to replace the existentially quantified variable z.

Before we start with technical details, let us briefly sketch how we relate satisfiability for formulas in strong negation normal form with the property $\mathsf{EHD}(\mathcal{L})$ where $\mathcal{L}$ is some logic like $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ (in short BMW). The leading idea for solving satisfiability for CECTL^* formulas is to split the search for a model into two steps.

The first step is to describe all the Kripke trees that satisfy the structural requirements of a given CECTL^* formula φ over the signature σ. With structural requirements we mean, roughly speaking, all those parts of φ that can also be expressed in pure ECTL^*. The second step is to assign a value to all register variables $r \in \mathsf{Reg}$ in each node of the model, such that also the constraints from φ are satisfied by the resulting decorated Kripke tree.

In order to accomplish the first step, we define from φ a pure ECTL^*-formula φ^a which we call "the abstraction of φ". This formula φ^a results from φ by replacing each atomic constraint with a fresh atomic proposition not occurring in φ. Every Kripke tree $\mathcal{T}$ that satisfies φ^a satisfies all the structural requirements of φ and is marked by new propositions at those positions where a model of φ would have to satisfy certain requirements on the values held by the register variables.

For the second step we can use the new propositions to extract from every tree model $\mathcal{T}$ of φ^a a σ-structure $\mathcal{B}$ the *constraint graph* that encodes all the constraints imposed by φ on the register variables in the following sense: If there is a homomorphism from $\mathcal{B}$ to $\mathcal{D}$ then we can equip $\mathcal{T}$ with a valuation function γ such that the resulting $\mathcal{D}$-Kripke tree satisfies φ (and vice-versa).

If $\mathcal{D}$ has property $\mathsf{EHD}(\mathsf{BMW})$ we can compile our two steps into the question whether a certain BMW-formula has a tree model. This BMW-formula requires that all its models encode the Kripke tree $\mathcal{T}$ satisfying φ^a as well as the the corresponding σ-structure $\mathcal{B}$ allowing a homomorphism to $\mathcal{D}$.

For the following definitions let us fix a CECTL^*-formula φ in which only the atomic constraints $\vartheta_1, \ldots, \vartheta_n$ occur. Let d_i be the depth of ϑ_i. Moreover, let Reg_φ be the set of register variables from Reg appearing in φ.

Definition 4.11. We define φ^a, the **abstraction of** φ, as the ordinary ECTL^*-formula that is obtained from φ by replacing every occurrence of a constraint $\vartheta_i(x)$ by the MSO-formula $p_i(x + d_i)$, for some fresh atomic proposition p_i which does not appear in φ. The same definition is also used for a CMSO-subformula of φ.

Notice how we use the fact that the constraints from CECTL^* are local to individuate the "lower" node involved in the constraint (the one at depth d_i) and mark it with the fresh propositional variable p_i. This way, when navigating a tree-model of the abstracted formula φ^a, we know that all paths which go through a node marked with p_i should satisfy the constraint ϑ_i. This would not work if the constraints were non-local.

Example 4.12. Given the CECTL^*-formula over the signature $\{<, \equiv\}$

$$\varphi = \mathsf{E}[\forall x\, q_1(x) \rightarrow (r_1 < S^2 r_2)(x)] \wedge \mathsf{A}[\exists x\, q_2(x) \wedge (Sr_1 \equiv r_2)(x)]$$

we replace the atomic constraints with the propositional variables p_1 and p_2 to obtain the abstracted ECTL^*-formula

$$\varphi^a = \mathsf{E}[\forall x\, q_1(x) \rightarrow p_1(x+2)] \wedge \mathsf{A}[\exists x\, q_2(x) \wedge p_2(x+1)].$$

Definition 4.13. Given a decorated Kripke e-tree $\mathbf{T}$ with underlying Kripke e-tree $\mathcal{T} = ([1,e]^*, \rightarrow, \rho)$ and $\rho(u) \cap \{p_1, \dots, p_n\} = \emptyset$ for all $u \in [1,e]^*$, we define the Kripke e-tree $\mathbf{T}^a = ([1,e]^*, \rightarrow, \rho^a)$, where $\rho^a(u)$ contains

- all propositions from $\rho(u)$ and
- all propositions p_j $(1 \leq j \leq n)$ such that, if $\vartheta_j = R(S^{i_1}r_1, \dots, S^{i_k}r_k)$ and $d_j = \max\{i_1, \dots, i_k\}$, we have:
  - $u = wv$ with $|v| = d_j$, and
  - $(\gamma(wv_1, r_1), \dots, \gamma(wv_k, r_k)) \in I(R)$, where v_l denotes the prefix of v of length i_l.

Hence, the fact that proposition p_j labels node wv with $|v| = d_j$ means that the constraint ϑ_j holds along every path that starts in node w and descends in the tree down via node wv.

Definition 4.14. Given a Kripke e-tree $\mathcal{T} = ([1,e]^*, \rightarrow, \rho)$ where the new propositions $p_1, \dots, p_n$ are allowed to occur in $\mathcal{T}$, we define a countable σ-structure $\mathcal{G}_{\mathcal{T}} = ([1,e]^* \times \mathsf{Reg}_\varphi, J)$ (the **constraint graph**) as follows: The interpretation $J(R)$ of the relation symbol $R \in \sigma$ contains all k-tuples $((wv_1, r_1), \dots, (wv_k, r_k))$ (where $k = \mathsf{ar}(R)$) for which there are $1 \leq j \leq n$ and $v \in [1,e]^{d_j}$ such that $p_j \in \rho(wv)$, and $\vartheta_j = R(S^{i_1}r_1, \dots, S^{i_k}r_k)$, where v_l still denotes the prefix of v of length i_l.

A constraint graph is in charge of "remembering" the relations connecting the register variables that we "forgot" when we abstracted the constraints using propositional variables. The domain of $\mathcal{G}_{\mathcal{T}}$ has one element for each register pair (v, r) where v is a node of the tree $\mathcal{T}$ and r is a register variable appearing in φ, and the relations of $\mathcal{G}_{\mathcal{T}}$ are the ones which should hold between the register variables in order to build a model for φ. It is called a constraint graph because (when all relations are binary) it is just like a graph with different kinds of edges (one for each relation) representing all the constraints from φ, and we will use this view of it in the following chapters.

Remark 4.15. Recall the k-copy operation from Definition 2.25 and consider the Kripke tree $\mathcal{T} = ([1, e]^*, \rightarrow, \rho)$ from Definition 4.14 as a relational structure over the signature $\mathbb{P} \cup \{\rightarrow, p_1, \dots, p_n\}$, where $\rightarrow$ is seen as a binary predicate and the atomic propositions from $\mathbb{P}$ and $\{p_1, \dots, p_n\}$ as unary predicates. We see that $\mathsf{copy}_k(\mathcal{T})$ defines a structure with domain $([1, e]^* \times \{1, \dots, k\})$ over the signature $\tau = \mathbb{P} \cup \{\rightarrow, \sim, p_1, \dots, p_n, P_1, \dots, P_k\}$. Then setting $k = |\mathsf{Reg}_\varphi|$ one can easily write down a one-dimensional first-order interpretation which interprets $\mathcal{G}_\mathcal{T}$ in $\mathsf{copy}_k(\mathcal{T})$. This means that we can find a substructure $\mathcal{S} = (D, I)$ of $\mathsf{copy}_k(\mathcal{T})$ which is isomorphic to $\mathcal{G}_\mathcal{T}$, and which is definable inside $\mathsf{copy}_k(\mathcal{T})$ using one-dimensional first order formulas.

In particular, the domain of $\mathcal{G}_\mathcal{T}$ is trivially in a bijection with the domain of $\mathsf{copy}_k(\mathcal{T})$ via the mapping $(w, r_i) \mapsto (w, i)$ for all $w \in [1, e]^*$ and all $i = 1 \dots k$. We can therefore choose $D = \{x \in \mathsf{copy}_k(\mathcal{T}) \mid \mathsf{copy}_k(\mathcal{T}) \models \mathsf{true}\}$.

Successively for each relation R of arity t which appears in some constraint ϑ we can find a first order formula φ_R on the signature τ such that defining

$$I(R) = \{(x_1, \dots, x_t) \in D^t \mid \mathsf{copy}_k(\mathcal{T}) \models \varphi_R(x_1, \dots, x_t)\}$$

one obtains that $\mathcal{S} = (D, I)$ is isomorphic to $\mathcal{G}_\mathcal{T}$. Let us see how: Suppose that $\vartheta = R_s(S^{i_1} r_1, \dots, S^{i_t} r_t)$ is a constraint of depth $h = \max\{i_1, \dots, i_t\}$, then we define

$$\varphi_\vartheta(x_1, \dots, x_t) = \exists y \, \exists y_1 \, \dots \exists y_t \;\; p_s(y + h) \bigwedge_{j=1,\dots,t} \left[y_j \sim y \wedge P_j(x_j) \wedge y_j \rightarrow^{i_j} x_j \right],$$

where $y \rightarrow^i x$ stands for $\exists a_1 \dots a_{i-1} \, (y \rightarrow a_1 \wedge a_1 \rightarrow a_2 \wedge \dots \wedge a_{i-1} \rightarrow x)$. Then, if the relation R appears in the constraints $\vartheta_1 \dots \vartheta_m$, we define $\varphi_R = \varphi_{\vartheta_1} \vee \dots \vee \varphi_{\vartheta_m}$.

It is then simple to verify that $\mathcal{G}_\mathcal{T}$ is isomorphic to $\mathcal{S}$. Since BMW is a superset of first order logic, this implies that given a BMW-formula φ, one can compute a formula φ' such that $\mathcal{G}_\mathcal{T} \models \varphi$ if and only if $\mathsf{copy}_k(\mathcal{T}) \models \varphi'$. For more on this subject we refer the reader to [23, Section 12.3]

Example 4.16. Figure 4.1 shows an example, where we assume that $e = 2$ and $n = 2$, $\vartheta_1 = (r_1 < Sr_2)$, and $\vartheta_2 = (Sr_1 \equiv Sr_2)$. The figure shows a portion of an $(\mathbb{N}, <, \equiv)$-Kripke 2-tree $\mathbf{T} = ((\mathbb{N}, <, \equiv), \mathcal{T}, \gamma)$. The edges of the Kripke 2-tree $\mathcal{T}$ are dotted. We assume that $\mathcal{T}$ is defined over the empty set of propositions. The node to the left (respectively, right) of a tree node w is labeled by the value $\gamma(w, r_1)$ (respectively, $\gamma(w, r_2)$). The figure shows the labeling of tree nodes with the two new propositions p_1 and p_2 (corresponding to ϑ_1 and ϑ_2) as well as the $\{<, \equiv\}$-structure $\mathcal{G}_{\mathbf{T}^a}$.

Lemma 4.17. *Let φ be a CECTL^*-formula in strong negation normal form. Then φ is $\mathcal{D}$-satisfiable if and only if there is a Kripke $(\#_\mathsf{E}(\varphi) + 1)$-tree $\mathcal{T}$ such that $(\mathcal{T}, \varepsilon) \models \varphi^a$ and $\mathcal{G}_\mathcal{T} \preceq \mathcal{D}$.*

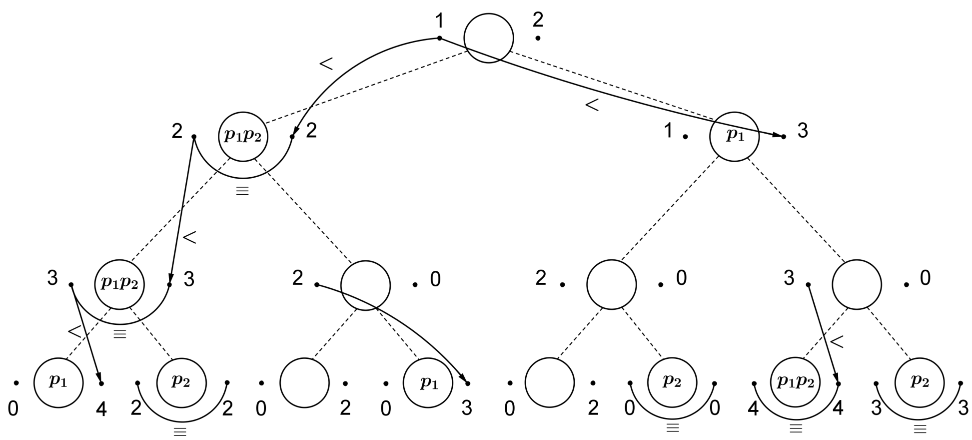

Figure 4.1: The $(\mathbb{N}, <, \equiv)$-decorated Kripke 2-tree $\mathfrak{T}$ from Example 4.16, the Kripke 2-tree $\mathfrak{T}^a$, and the constraint graph $\mathcal{G}_{\mathfrak{T}^a}$.

Proof. Let $\mathcal{D} = (D, I)$, $e = (\#_{\mathsf{E}}(\varphi) + 1)$, and let Reg_φ, n, ϑ_j, and d_j $(1 \leq j \leq n)$ be defined as above.

Proof of $\Rightarrow$. Assume that φ is $\mathcal{D}$-satisfiable. By Theorem 3.12 there is a $\mathcal{D}$-Kripke e-tree $\mathfrak{T} = (\mathcal{D}, \mathcal{T}, \gamma)$ with $\mathcal{T} = ([1, e]^*, \to, \rho)$ such that $(\mathfrak{T}, \varepsilon) \models \varphi$. Take the Kripke e-tree $\mathcal{T}^a = ([1, e]^*, \to, \rho^a)$.

We claim that $\gamma : [1, e]^* \times \mathsf{Reg}_\varphi \to D$ is a homomorphism from $\mathcal{G}_{\mathcal{T}^a}$ to $\mathcal{D}$. For this, assume that the tuple $((wv_1, r_1), \ldots, (wv_k, f_k))$ belongs to the interpretation of R in $\mathcal{G}_{\mathcal{T}^a}$. By Definition 4.14 there are $1 \leq j \leq n$ and $v \in [1, e]^{d_j}$ such that $p_j \in \rho^a(wv)$, $\vartheta_j = R(S^{i_1} r_1, \ldots, S^{i_k} r_k)$, and v_l is the prefix of v of length j_l for each $1 \leq l \leq k$. Since $p_j \in \rho^a(wv)$, Definition 4.13 implies

$$(\gamma(wv_1, r_1), \ldots, \gamma(wv_k, r_k)) \in I(R) \ .$$

Hence, γ is indeed a homomorphism.

In order to show $(\mathcal{T}^a, \varepsilon) \models \varphi^a$ we prove simultaneously by structural induction on the formula that

(1) For all CECTL^*-subformulas χ of φ and $v \in [1, e]^*$, if $(\mathfrak{T}, v) \models \chi$, then $(\mathcal{T}^a, v) \models \chi^a$, and

(2) for all CMSO-subformulas $\psi(X_1, \ldots, X_m)$ of φ (we assume that $X_1, \ldots, X_m$ only appear freely), all paths P in $\mathcal{T}$, all valuation functions η, and all sets $A_1, \ldots, A_m$, $B_1, \ldots, B_m \subseteq \mathbb{N}$ of positions of $\mathfrak{P}$ such that $A_i \subseteq B_i$ for all $1 \leq i \leq m$,

$$\text{if } (\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m) \text{ then } (\mathcal{P}^a, \eta) \models_{\mathsf{CMSO}} \psi^a(B_1, \ldots, B_m) \ ,$$

where $\mathfrak{P}$ is the $\mathcal{D}$-Kripke path induced by P in $\mathfrak{T}$ and $\mathcal{P}^a$ is the Kripke path induced by P in $\mathcal{T}^a$.

We have to consider the following cases, where we write η_A and η_B for the valuations $\eta[(X_j \to A_j)_{1\leq j\leq m}]$ and $\eta[(X_j \to B_j)_{1\leq j\leq m}]$, respectively.

- Assume that $\chi = \mathsf{E}\psi(\varphi_1, \ldots, \varphi_m)$, whence $\chi^a = \mathsf{E}\psi^a(\varphi_1^a, \ldots, \varphi_m^a)$. Since $(\mathfrak{T}, v) \models \chi$, we know that there is an infinite $\mathcal{D}$-Kripke path $\mathfrak{P}$ with $\mathfrak{P}(0) = v$ such that $\mathfrak{P} \models \psi(A_1, \ldots, A_m)$, where $A_i = \{n \in \mathbb{N} \mid (\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i\}$. We can use the induction hypothesis (point (1)) on $\varphi_1, \ldots, \varphi_m$ to obtain that, for all $n \in \mathbb{N}$ and $1 \leq i \leq m$, if $(\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i$ then $(\mathcal{T}^a, \mathfrak{P}(n)) \models \varphi_i^a$. So, if we define $B_i = \{n \in \mathbb{N} \mid (\mathcal{T}^a, \mathfrak{P}(n)) \models \varphi_i^a\}$, we can deduce that $A_i \subseteq B_i$. Applying point (2) of the induction hypothesis we conclude that $\mathcal{P}^a \models \psi^a(B_1, \ldots, B_m)$, where $\mathcal{P}^a$ denotes the Kripke path in $\mathcal{T}^a$ that corresponds to $\mathfrak{P}$.

- The case $\chi = \mathsf{A}\psi(\varphi_1, \ldots, \varphi_m)$ is treated analogously to the previous one replacing "there is" by "for all".

- Assume that $\psi = \vartheta_j(x)$ for some atomic constraint $\vartheta_j = R(S^{i_1}r_1, \ldots, S^{i_k}r_k)$ of depth $d_j = \max\{i_1, \ldots, i_k\}$. We want to show that $(\mathfrak{P}, \eta_A) \models_{\mathsf{CMSO}} \vartheta_j(x)$ implies $(\mathcal{P}^a, \eta_B) \models_{\mathsf{CMSO}} p_j(x + d_j)$. Let $n = \eta(x)$. Note that $\eta_A(x) = \eta_B(x) = n$. If $(\mathfrak{P}, \eta_A) \models_{\mathsf{CMSO}} R(S^{i_1}r_1, \ldots, S^{i_k}r_k)(x)$, then $(\gamma(n + i_1, r_1), \ldots, \gamma(n + i_k, r_k)) \in I(R)$, and this, by Definition 4.13 implies that $p_j \in \rho^a(n + d_j)$ which implies $(\mathcal{P}^a, \eta_B) \models_{\mathsf{MSO}(\sigma)} p_j(x + d_j)$.

- All other steps are trivial.

Proof of $\Leftarrow$. For the other direction, assume that there are a Kripke e-tree $\mathcal{T} = ([1, e]^*, \to, \rho_{\mathcal{T}})$ such that $(\mathcal{T}, \varepsilon) \models \varphi^a$, and a homomorphism $h : [1, e]^* \times \mathsf{Reg}_\varphi \to A$ from $\mathcal{G}_{\mathcal{T}}$ to $\mathcal{D} = (D, I)$. Define the $\mathcal{D}$-Kripke structure $\mathfrak{T} = (\mathcal{D}, \mathcal{T}', \gamma)$, where

- $\mathcal{T}'$ is obtained from $\mathcal{T}$ by removing the propositions corresponding to atomic constraints, i.e., $\mathcal{T}' = ([1, e]^*, \to, \rho)$ with $\rho(v) = \rho_{\mathcal{T}}(v) \setminus \{p_1, \ldots, p_n\}$ for all $v \in [1, e]^*$,

- $\gamma(v, r) = h(v, r)$ for all $r \in \mathsf{Reg}$, and

- γ is defined arbitrarily on all $r \in \mathsf{Reg} \setminus \mathsf{Reg}_\varphi$.

We claim that $(\mathfrak{T}, \varepsilon) \models \varphi$. Again by structural induction, we prove the following claim.

1. For all CECTL^*-subformulas χ of φ and $v \in [1, e]^*$, if $(\mathcal{T}, v) \models \chi^a$, then $(\mathfrak{T}, v) \models \chi$, and

2. for all CMSO-subformulas ψ of φ, all paths P in $\mathcal{T}$, all valuation functions η, and all sets $A_1, \ldots, A_m$, $B_1, \ldots, B_m \subseteq \mathbb{N}$ of positions of P such that $A_i \subseteq B_i$ for all $1 \le i \le m$,

$$\text{if } (\mathcal{P}, \eta) \models_{\mathsf{CMSO}} \psi^a(B_1, \ldots, B_m) \text{ then } (\mathfrak{P}, \eta) \models_{\mathsf{CMSO}} \psi(A_1, \ldots, A_m) ,$$

where $\mathcal{P}$ is the Kripke path induced by P in $\mathcal{T}$, $\mathfrak{P}$ is the $\mathcal{D}$-KP induced by P in $\mathfrak{T}$, and we assume that $X_1, \ldots, X_m$ only appear freely in φ.

All steps are trivial except for the case that χ is the atomic constraint $\vartheta_j = R(S^{i_1}r_1, \ldots, S^{i_k}r_k)$ of depth d_j. In this case, we have $\chi^a = p_j(x + d_j)$. Assume that $(\mathcal{P}, \eta[(X_i \to B_i)_{1 \le i \le k}]) \models_{\mathsf{CMSO}} p_j(x + d_j)$ and let $n = \eta(x)$. By definition of $\mathcal{G}_{\mathcal{T}}$ (Definition 4.14), this implies that the tuple $((\mathcal{P}(n + i_1), f_1), \ldots, (\mathcal{P}(n + i_k), f_k))$ belongs to the interpretation of R in $\mathcal{G}_{\mathcal{T}}$. Now, since h is a homomorphism and $\gamma_{\restriction \mathsf{Reg}_\varphi} = h$, we conclude that

$$(\gamma(\mathcal{P}(n + i_1), r_1), \ldots, \gamma(\mathcal{P}(n + i_k), r_k)) \in I(R) ,$$

and thus

$$(\mathfrak{P}, \eta[(X_i \to A_i)_{1 \le i \le k}]) \models_{\mathsf{CMSO}} R(S^{i_1}r_1, \ldots, S^{i_k}r_k)(x)$$

as desired.

□

We can now piece everything together to conclude the proof of Theorem 4.7: $\mathcal{D}$-SAT is decidable, provided that $\mathcal{D}$ is negation-closed and has the property EHD(Bool(MSO, WMSO+B)).

Proof (Theorem 4.7). Let φ be a CECTL*-formula. Thanks to Lemma 4.9 we can assume without loss of generality, that φ is in strong negation normal form. Let $\vartheta = \varphi^a$ be the abstraction of φ for the further discussion. Hence, ϑ is an ordinary ECTL*-formula, where negation only occur in front of formulas of the form $p(x)$ where $p \in \mathbb{P} \setminus \{p_1, \ldots, p_n\}$ or atomic MSO-formulas, and $e = \#_{\mathsf{E}}(\vartheta) + 1$. By Lemma 4.17, we have to check, whether there is a Kripke e-tree $\mathcal{T}$ such that

$$(\mathcal{T}, \varepsilon) \models \vartheta \text{ and } \mathcal{G}_{\mathcal{T}} \preceq \mathcal{D} .$$

Let $\tau \subseteq \sigma$ be the finite subsignature consisting of all relation symbols that occur in our initial CECTL*-formula φ. Note that $\mathcal{G}_{\mathcal{T}}$ is actually a countable τ-structure. Since the concrete domain $\mathcal{D}$ has the property EHD(BMW), one can compute from τ a BMW-sentence α such that for every countable τ-structure $\mathcal{B}$ we have $\mathcal{B} \models \alpha$ if and only if $\mathcal{B} \preceq \mathcal{D}$. Our new goal is to decide whether there is a Kripke e-tree $\mathcal{T}$ such that

$$(\mathcal{T}, \varepsilon) \models \vartheta \text{ and } \mathcal{G}_{\mathcal{T}} \models \alpha .$$

Given the fact that every ECTL^*-formula can be effectively transformed into an equivalent MSO-formula with a single free first-order variable [15, 27], and since the root ε of a tree is first-order definable, we get an MSO-sentence ψ such that $(\mathcal{T}, \varepsilon) \models \vartheta$ if and only if $\mathcal{T} \models \psi$. Hence, we have to check whether there is a Kripke e-tree $\mathcal{T}$ such that

$$\mathcal{T} \models \psi \text{ and } \mathcal{G}_{\mathcal{T}} \models \alpha .$$

Since BMW is compatible with first-order interpretations (Remark 4.15) and with the k-copy operation (Proposition 2.26), we can compute from α a BMW-sentence α' and from α' another BMW-sentence β such that for $k = |\mathsf{Reg}_\varphi|$ we have:

$$\mathcal{G}_{\mathcal{T}} \models \alpha \xLeftrightarrow{\text{Rmk. 4.15}} \mathsf{copy}_k(\mathcal{T}) \models \alpha' \xLeftrightarrow{\text{Prop. 2.26}} \mathcal{T} \models \beta .$$

Thus, we have to check whether there is a Kripke e-tree $\mathcal{T}$ such that $\mathcal{T} \models \psi \wedge \beta$, where $\psi \wedge \beta$ is a BMW-sentence. By Theorem 2.21 this is decidable, which completes the proof.

□

4.2.1 The EHD Method for Classes of Structures

In a variant of the notion of $\mathcal{D}$-satisfiability (introduced in Definition 4.1) we are given a class of structures Δ, and we look for an $\mathcal{A}$-Kripke structure, for some $\mathcal{A} \in \Delta$, satisfying our formula.

Definition 4.18. We say that $\varphi \in \mathsf{CECTL}^*$ is Δ-satisfiable if and only if there exists a member $\mathcal{A}$ of Δ such that φ is $\mathcal{A}$-satisfiable.

Clearly, choosing $\Delta = \{\mathcal{D}\}$ gives us back the original problem. Suppose now Δ has a **Universal Structure** (some $\mathcal{U} \in \Delta$ such that every $\mathcal{A} \in \Delta$ admits an injective homomorphism into $\mathcal{U}$), then Δ-satisfiability is equivalent to $\mathcal{U}$-satisfiability. In fact it is easy to see that a formula φ has a model with some concrete domain from Δ if and only if it has a model with concrete domain $\mathcal{U}$: Suppose $\mathfrak{K} = (\mathcal{A}, \mathcal{K}, \gamma)$ is a model for φ, and let h be the injective homomorphism from $\mathcal{A}$ to $\mathcal{U}$. Then define $\gamma' = h \circ \gamma$. It is easy to show that $\mathfrak{U} = (\mathcal{U}, \mathcal{K}, \gamma')$ is a model for φ as well.

A typical class with a universal model is the class of all countable linear orders, for which $(\mathbb{Q}, <)$ is universal.

In order to approach the problem of satisfiability with respect to a class of structures, we need to adapt some of the notions that we introduced in the previous sections. First of all we need to extend the definition of the EHD-property to this setting: Given a countable signature σ and a class Δ of σ-structures, we say that Δ has the property $\mathsf{EHD}(\mathcal{L})$ if there is a computable function that maps every finite subsignature $\tau \subseteq \sigma$ to an $\mathcal{L}$-sentence φ_τ such that for every countable τ-structure $\mathcal{B}$ we have:

$$\text{there exists } \mathcal{A} \in \Delta \text{ such that } \mathcal{B} \preceq \mathcal{A} \iff \mathcal{B} \models \varphi_\tau \ .$$

Analogously, we need to give a suitable definition of negation closure: Δ is negation-closed if and only if for every $R \in \sigma$, there exists a positive existential first-order formula φ_R, defining for all $\mathcal{A} = (A, I) \in \Delta$ the complement of $I(R)$. Note that the formula φ_R does not depend on the particular structure $\mathcal{A}$, but only on the relation R.

One can then retrace all the steps of Theorem 4.7 and prove that

Theorem 4.19. *Given a class Δ of σ-structures, Δ-satisfiability is decidable for* CECTL^**, provided that Δ has the* EHD*-property and is negation-closed.*

Chapter 5

Concrete domains over the integers

Having introduced the EHD method, we now apply it successfully to several concrete domains over the integers. The final goal of this section is to show:

Proposition 5.1. *The concrete domain* $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\leq a<b})$ *has the property* $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$.

Since $\mathcal{Z}$ is also negation-closed (see Example 4.6), the following result is an immediate consequence of Theorem 4.7:

Theorem 5.2. $\mathcal{Z}$-SAT *is decidable.*

We prove Proposition 5.1 in three steps. First, we show that $(\mathbb{Z}, <)$ has the property $\mathsf{EHD}(\mathsf{WMSO{+}B})$. In a second step we extend this result to the structure $(\mathbb{Z}, <, \equiv)$. Finally we add a countable set of unary predicates satisfying certain computability requirements, and show that $\mathcal{Z}$ is an instance of this case.

5.1 $\mathbb{Z}$ with Order-Constraints

Our first goal is to prove:

Proposition 5.3. *The structures* $(\mathbb{Z}, <)$*,* $(\mathbb{N}, <)$ *and* $(\mathbb{Z} \setminus \mathbb{N}, <)$ *have the property* $\mathsf{EHD}(\mathsf{WMSO{+}B})$.

As a preparation of the proof, we first define some terminology and successively we characterize all $\{<\}$-structures that allow a homomorphism to $(\mathbb{Z}, <)$ in terms of their paths. Let $\mathcal{A} = (A, I)$ be a countable $\{<\}$-structure. Setting $E = I(<)$, we can see (A, E) as a directed graph. When talking about paths, we always refer to finite directed E-paths inside (A, E). The length of a path $(a_0, a_1, \dots, a_n)$ is n. For $S \subseteq A$ and $x \in A \setminus S$, a path from x to S is a path from x to some node $y \in S$. A path from S to x is defined symmetrically.

Lemma 5.4. *Given a $\{<\}$-structure $\mathcal{A}$, we have $\mathcal{A} \preceq (\mathbb{Z}, <)$ if and only if*

(H1) $\mathcal{A}$ does not contain cycles, and

(H2) for all $a, b \in A$ there is an $n \in \mathbb{N}$ such that the length of each path from a to b is bounded by n.

Proof.

($\Rightarrow$) Let us first show the "only if" direction of the lemma. Suppose h is a homomorphism from $\mathcal{A}$ to $(\mathbb{Z}, <)$. Heading for a contradiction, suppose that there is a cycle $(a_0, \dots, a_k)$ in $\mathcal{A}$, i.e., $(a_0, \dots, a_k)$ is a path such that $(a_k, a_0) \in E$. Setting $z_i = h(a_i)$, this implies $z_i < z_{i+1}$ for $0 \le i \le k-1$ and $z_k < z_0$ which is a contradiction. Hence, (H1) holds.

Suppose now that $a, b \in A$ are such that for every $n \in \mathbb{N}$ there is a path of length at least n from a to b. If $d = h(b) - h(a)$, we can find a path $(a_0, a_1 \dots, a_k)$ with $a_0 = a$, $a_k = b$ and $k > d$. Since h is a homomorphism, this path is mapped to an increasing sequence of integers $h(a) = h(a_0) < h(a_1) < \dots < h(a_k) = h(b)$. But this contradicts $h(b) - h(a) = d < k$. Hence, (H2) holds.

($\Leftarrow$) For the "if" direction of the lemma assume that $\mathcal{A}$ is acyclic (property (H1)) and that (H2) holds. Fix an enumeration $a_0, a_1, a_2, \dots$ of the countable set A. For $n \ge 0$ let

$$S_n = \{a \in A \mid \exists i, j \le n : (a_i, a), (a, a_j) \in E^*\}.$$

We claim that S_n has the following properties.

(P1) S_n is convex w.r.t. the partial order E^*: If $a, c \in S_n$ and $(a, b), (b, c) \in E^*$, then $b \in S_n$.

(P2) For $a \in A \setminus S_n$ all paths between a and S_n are "one-way", i.e., there are not $b, c \in S_n$ such that $(b, a), (a, c) \in E^*$.

(P3) For all $a \in A \setminus S_n$ there is a bound $c \in \mathbb{N}$ such that all paths between a and S_n have length at most c. Let $c_n^a \in \mathbb{N}$ be the smallest such bound (hence, we have $c_n^a = 0$ if there is no path between a and S_n).

(P1) is obvious and moreover implies (P2). To see (P3), assume that there are only paths from S_n to a but not the other way round (see (P2)); the other case is symmetric. If there is no bound on the length of paths from S_n to a, then by definition of S_n, there is no bound on the length of paths from $\{a_0, \dots, a_n\}$ to a. By the pigeon principle, there is a $0 \leq i \leq n$ such that there is no bound on the length of paths from a_i to a. But this contradicts property (H2).

We build the homomorphism h inductively. For every $n \geq 0$ we define functions $h_n : S_n \to \mathbb{Z}$ such that the following invariants hold for all $n \geq 0$.

(I1) If $n > 0$ then $h_n(a) = h_{n-1}(a)$ for all $a \in S_{n-1}$.

(I2) $h_n(S_n)$ is bounded in $\mathbb{Z}$, i.e., there are $z_1, z_2 \in \mathbb{Z}$ such that $h_n(S_n) \subseteq [z_1, z_2]$.

(I3) h_n is a homomorphism from the induced subgraph $\mathcal{A}_{\restriction S_n}$ to $(\mathbb{Z}, <)$.

For $n = 0$ we have $S_0 = \{a_0\}$. We set $h_0(a_0) = 0$ (any other integer would be also fine). Properties (I1)–(I3) are easily verified. For $n > 0$, there are four cases.

1. $a_n \in S_{n-1}$, thus $S_n = S_{n-1}$. We set $h_n = h_{n-1}$. Clearly, (I1)–(I3) hold for n.
2. $a_n \notin S_{n-1}$ and there is no path from a_n to S_{n-1} or vice versa. We set $h_n(a_n) \coloneqq 0$. Since $S_n = S_{n-1} \cup \{a_n\}$, (I1)–(I3) follow easily from the induction hypothesis.
3. $a_n \notin S_{n-1}$ and there is a path from a_n to S_{n-1}. Then, by (P2) there are no paths from S_{n-1} to a_n. Hence, we have

   $$S_n = S_{n-1} \cup \{a \in A \mid \exists b \in S_{n-1} \, : (a_n, a), (a, b) \in E^*\}.$$

   We have to define the value $h_n(a)$ for all $a \in A \setminus S_{n-1}$ that lie along a path from a_n to S_{n-1}. By (I2) there are $z_1, z_2 \in \mathbb{Z}$ with $h_{n-1}(S_{n-1}) \subseteq [z_1, z_2]$. Recall the definition of c^a_{n-1} from (P3). For all $a \in A \setminus S_{n-1}$ that lie on a path from a_n to S_{n-1}, we set $h_n(a) = z_1 - c^a_{n-1}$. Since there are paths from a to S_{n-1}, we have $c^a_{n-1} > 0$. Hence, for all $a \in S_n \setminus S_{n-1}$, $h_n(a) < z_1$. Let us check that $h_n : S_n \to \mathbb{Z}$ satisfy (I1)– (I3): Invariant (I1) holds by definition of h_n. For (I2) note that $h_n(S_n) \subseteq [z_1 - c^{a_n}_{n-1}, z_2]$.
   It remains to show (I3), i.e., that h_n is a homomorphism from $\mathcal{A}\restriction S_n$ to $(\mathbb{Z}, <)$. Hence, we have to show that $h(b_1) < h(b_2)$ for all $(b_1, b_2) \in E \cap (S_n \times S_n)$.
   - If $b_1, b_2 \in S_{n-1}$, then $h_n(b_1) = h_{n-1}(b_1) < h_{n-1}(b_2) = h_n(b_2)$ by the induction hypothesis.

– If $b_1 \in S_n \setminus S_{n-1}$ and $b_2 \in S_{n-1}$, we know that $h_n(b_2) = h_{n-1}(b_2) \geq z_1$ while $h_n(b_1) < z_1$ by construction. Hence, we have $h_n(b_1) < h_n(b_2)$.
– If $b_2 \in S_n \setminus S_{n-1}$ and $b_1 \in S_{n-1}$, then $(b_1, b_2) \in E$ contradicts (P2) because b_2 is on a path from a_n to S_{n-1} and (b_1, b_2) is a path in the opposite direction.
– If both b_1 and b_2 belong to $S_n \setminus S_{n-1}$ then $h_n(b_i) = z_1 - c^{b_i}_{n-1}$ for $i \in \{1, 2\}$. Since $(b_1, b_2) \in E$, we have $c^{b_1}_{n-1} > c^{b_2}_{n-1}$. This implies $h_n(b_1) < h_n(b_2)$.

4. $a_n \notin S_{n-1}$ and there is a path from S_{n-1} to a_n. For all

$$a \in S_n \setminus S_{n-1} = \{a \in A \setminus S_{n-1} \mid a \text{ belongs to a path from } S_{n-1} \text{ to } a_n\},$$

set $h_n(a) = z_2 + c^a_{n-1}$. The rest of the argument is analogous to the previous case.

This concludes the construction of h_n. Thanks to (I1), the limit function $h = \bigcup_{i \in \mathbb{N}} h_i$ exists. By (I3) and $A = \bigcup_{i \in \mathbb{N}} S_i$, h is a homomorphism from $\mathcal{A}$ to $(\mathbb{Z}, <)$.

□

A result similar to Lemma 5.4 holds for $(\mathbb{N}, <)$. Here the characterization of homomorphisms relies on the fact that if some element a is mapped to $n \in \mathbb{N}$ by some homomorphism, then a path leading to a is at most of length n.

Lemma 5.5. *We have $\mathcal{A} \preceq (\mathbb{N}, <)$ if and only if*

(H1) $\mathcal{A}$ does not contain cycles, and

(H2) for all $a \in A$ there is an $n \in \mathbb{N}$ such that the length of each path ending in a is bounded by n.

Proof. If $h : A \to \mathbb{N}$ is a homomorphism and $a \in A$ then every path ending in a can be of length at most $h(a)$. Moreover, $\mathcal{A}$ must be acyclic by the same argument that we used for $(\mathbb{Z}, <)$.

For the other direction assume that $\mathcal{A}$ is acyclic and for each $a \in A$ there is some $c_a \in \mathbb{N}$ such that the longest path leading to a has length c_a. Define $h(a) = c_a$. It is rather straightforward to show that h is a homomorphism. □

To prove Proposition 5.3, stating that $(\mathbb{Z}, <)$, $(\mathbb{N}, <)$ and $(\mathbb{Z} \setminus \mathbb{N}, <)$ have the property $\mathsf{EHD}(\mathsf{WMSO{+}B})$, we need to show that we can use $\mathsf{WMSO{+}B}$ to distinguish between all those $\{<\}$-structures which allow a homomorphism to $(\mathbb{Z}, <)$ and those which do not. Thanks to the above results, we just need to express in $\mathsf{WMSO{+}B}$ the easy hypothesis of Lemma 5.4 and Lemma 5.5.

Proof (Proposition 5.3). For $(\mathbb{Z}, <)$, we translate the conditions (H1) and (H2) from Lemma 5.4 into $\mathsf{WMSO{+}B}$. Cycles are excluded by the sentence $\neg\mathsf{ECycle}_<$ (see Example 2.20). Moreover, for an acyclic $\{<\}$-structure $\mathcal{A}$ we have $\mathcal{A} \models \forall x\, \forall y\, \mathsf{BPaths}_<(x, y)$ (see also Example 2.20) if and only if for all $a, b \in A$ there is a bound $n \in \mathbb{N}$ on the length of paths from a to b. Thus,

$$\mathcal{A} \preceq (\mathbb{Z}, <) \text{ if and only if } \mathcal{A} \models \neg\mathsf{ECycle}_< \wedge \forall x\, \forall y\, \mathsf{BPaths}_<(x, y)\ .$$

Similarly, using Lemma 5.5, we obtain that

$$\mathcal{A} \preceq (\mathbb{N}, <) \text{ if and only if } \mathcal{A} \models \neg\mathsf{ECycle}_< \wedge \forall y\, \mathsf{B}Z\, \exists x\, \mathsf{Path}_<(x, y, Z)\ .$$

Since $(\mathbb{Z} \setminus \mathbb{N}, <)$ is $(\mathbb{N}, <)$ with reversed order, one proves analogously that

$$\mathcal{A} \preceq (\mathbb{Z} \setminus N, <) \text{ if and only if } \mathcal{A} \models \neg\mathsf{ECycle}_< \wedge \forall x\, \mathsf{B}Z\, \exists y\, \mathsf{Path}_<(x, y, Z)\ .$$

□

5.2 $\mathbb{Z}$ with Order- and Equality-Constraints

In this section, we extend Proposition 5.3 to the negation-closed structure $(\mathbb{Z}, <, \equiv)$, where $\equiv$ is the equality relation on $\mathbb{Z}$. For this purpose, given a structure $\mathcal{A} = (A, I)$ over the signature $\{\equiv\} \uplus \sigma$ we define *the quotient of* $\mathcal{A}$, obtained by *contracting* all the $\equiv$-paths (note that $I(\equiv)$ is usually not the identity relation on A).

Definition 5.6. Let $\sim\, = (I(\equiv) \cup I(\equiv)^{-1})^*$ be the smallest equivalence relation on A that contains $I(\equiv)$. We denote the $\sim$-quotient of $\mathcal{A}$ by $\tilde{\mathcal{A}} = (\tilde{A}, \tilde{I})$: It is a σ-structure with domain $\tilde{A} = \{[a] \mid a \in A\}$ the set of all $\sim$-equivalence classes. For all $R \in \sigma$ of arity k, we define $\tilde{I}(R)$ as the set of k-tuples $([a_1], \ldots, [a_k])$ for which there are $b_1 \in [a_1], \ldots, b_k \in [a_k]$ such that $(b_1, \ldots, b_k) \in I(R)$.

Remark 5.7. Let $\mathcal{A} = (A, I)$ be a $\{<, \equiv\}$-structure and $\tilde{\mathcal{A}} = (\tilde{A}, \tilde{I})$ its quotient. In this case we have that $([a], [b]) \in \tilde{I}(<)$ iff there are $a' \sim a$ and $b' \sim b$ such that $(a', b') \in I(<)$. Since $\sim$ is the reflexive and transitive closure of the first-order definable relation $I(\equiv) \cup I(\equiv)^{-1}$, we can construct a WMSO-formula $\tilde{\varphi}(x, y)$ (using the reach-construction from Example 2.20) that defines $\sim$. That is, $\sim$ is WMSO-definable (and MSO-definable as well).

Let $\mathcal{C} = (C, J)$ be a structure over the signature $\{\equiv\} \cup \sigma$ where $J(\equiv)$ is real equality, i.e., $J(\equiv) = \{(c, c) \mid c \in C\}$. In this case the quotient $\tilde{\mathcal{C}} = (\tilde{C}, \tilde{J})$ is isomorphic to the reduct of $\mathcal{C}$ with signature σ. Whenever $\equiv$ is interpreted as real equality in the target structure, taking the quotient of a structure is compatible with the existence of homomorphisms in the following sense.

Lemma 5.8. *Let $\mathcal{C} = (C, J)$ be a concrete domain over $\{\equiv\} \uplus \sigma$ where $J(\equiv)$ is real equality. Then, for every $\tau \subseteq \sigma$, and every $(\{\equiv\} \cup \tau)$-structure $\mathcal{A} = (A, I)$,*

1. *$\mathcal{A} \preceq \mathcal{C}$ if and only if $\tilde{\mathcal{A}} \preceq \tilde{\mathcal{C}}$.*

2. *$\mathcal{C}$ has the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ if and only if $\tilde{\mathcal{C}}$ does.*

Proof.

1. For the direction ($\Rightarrow$) let $h : \mathcal{A} \to \mathcal{C}$ be a homomorphism. We show that $g : \tilde{\mathcal{A}} \to \tilde{\mathcal{C}}$, defined by $g([a]) = h(a)$, is a homomorphism as well. Notice that the mapping g is well defined: $[a] = [b]$ implies $(a, b) \in (I(\equiv) \cup I(\equiv)^{-1})^*$. Since h is a homomorphism, $(h(a), h(b)) \in J(\equiv)$, i.e., $h(a) = h(b)$.

   Then let $([a_1], \ldots, [a_k]) \in \tilde{I}(R)$ for some $R \in \tau$. By definition of $\tilde{I}$, there are $b_1 \in [a_1], \ldots, b_k \in [a_k]$ such that $(b_1, \ldots, b_k) \in I(R)$. Therefore $(g([a_1]), \ldots, g([a_k])) = (h(b_1), \ldots, h(b_k))$, and since h is a homomorphism, $(h(b_1), \ldots, h(b_k)) \in J(R) = \tilde{J}(R)$ as wanted.

   For the direction ($\Leftarrow$) let $h : \tilde{\mathcal{A}} \to \tilde{\mathcal{C}}$ be a homomorphism. We define $g : \mathcal{A} \to \mathcal{C}$ by $g(a) = h([a])$. Then let $R \in \tau$ and $b_1, \ldots, b_k \in A$ such that $(b_1, \ldots, b_k) \in I(R)$. This implies that $([b_1], \ldots, [b_k]) \in \tilde{I}(R)$ and therefore $(h([b_1]), \ldots, h([b_k])) = (g(b_1), \ldots, g(b_k)) \in J(R)$.

   Finally, if $a, b \in A$ are such that $(a, b) \in I(\equiv)$, then $[a] = [b]$. Therefore $g(a) = h([a]) = h([b]) = g(b)$, i.e., $(g(a), g(b)) \in J(\equiv)$. This proves that g is a homomorphism.

2. Let $\mathcal{L} = \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ in the following arguments. Since $\tilde{\mathcal{C}}$ is a reduct of $\mathcal{C}$, it is clear that property $\mathsf{EHD}(\mathcal{L})$ for $\mathcal{C}$ implies property $\mathsf{EHD}(\mathcal{L})$ for $\tilde{\mathcal{C}}$. For the other direction, assume that $\tilde{\mathcal{C}}$ has the property $\mathsf{EHD}(\mathcal{L})$. Let $\tau \subseteq \sigma \uplus \{\equiv\}$ be a finite subsignature. If τ does not contain $\equiv$ then, by the property $\mathsf{EHD}(\mathcal{L})$ for $\tilde{\mathcal{C}}$, there exists an $\mathcal{L}$-sentence ψ_τ such that for every τ-structure $\mathcal{A}$ we have $\mathcal{A} \models \psi_\tau$ if and only if $\mathcal{A} \preceq \tilde{\mathcal{C}}$. But the latter is equivalent to $\mathcal{A} \preceq \mathcal{C}$ (since τ does not contain $\equiv$).

   Hence, we can assume that τ contains $\equiv$. Let $\tau' = \tau \setminus \{\equiv\}$. Since $\tilde{\mathcal{C}}$ has property $\mathsf{EHD}(\mathcal{L})$, we can find an $\mathcal{L}$-sentence $\psi_{\tau'}$, such that every τ'-structure $\mathcal{A}$,
$$\mathcal{A} \models \psi_{\tau'} \iff \mathcal{A} \preceq \tilde{\mathcal{C}}. \tag{5.1}$$

   By Remark 5.7 there is an $\mathcal{L}$-formula $\tilde{\varphi}(x, y)$ that defines the equivalence relation $\sim$. Let $\tilde{\vartheta}_\tau$ be the $\mathcal{L}$-sentence obtained by replacing in $\psi_{\tau'}$ every occurrence of an atomic formula $R(x_1, \ldots, x_k)$ for $R \in \tau'$ by
$$\tilde{R}(x_1, \ldots, x_k) := \exists z_1 \cdots \exists z_k \, (\tilde{\varphi}(z_1, x_1) \wedge \cdots \wedge \tilde{\varphi}(z_k, x_k) \wedge R(z_1, \ldots, z_k)).$$

We claim that for every τ-structure $\mathcal{B} = (B, I)$,

$$\mathcal{B} \models \tilde{\vartheta}_\tau \iff \tilde{\mathcal{B}} \models \psi_{\tau'}. \tag{5.2}$$

Using this claim, we obtain

$$\mathcal{B} \models \tilde{\vartheta}_\tau \iff \tilde{\mathcal{B}} \models \psi_{\tau'} \overset{(5.1)}{\iff} \tilde{\mathcal{B}} \preceq \tilde{\mathcal{C}} \overset{1.}{\iff} \mathcal{B} \preceq \mathcal{C},$$

which implies that $\mathcal{B}$ has the property $\mathsf{EHD}(\mathcal{L})$, as wanted.

The proof of (5.2) is by induction on the structure of the formula, the only non-trivial case being $\mathcal{B} \models \tilde{R}(a_1, \ldots, a_k)$ if and only if $\tilde{\mathcal{B}} \models R([a_1], \ldots, [a_k])$. Note that $\tilde{\mathcal{B}} \models R([a_1], \ldots, [a_k])$ if and only if there are $b_1, \ldots, b_k \in B$ such that $b_j \sim a_j$ and $\mathcal{B} \models R(b_1, \ldots, b_k)$, which is exactly what $\tilde{R}(a_1, \ldots, a_k)$ expresses.

□

An application of the previous lemma to Proposition 5.3 directly yields the following results.

Proposition 5.9. *$(\mathbb{Z}, <, \equiv)$, $(\mathbb{N}, <, \equiv)$, and $(\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$ satisfy the property* $\mathsf{EHD}(\mathsf{WMSO{+}B})$.

5.3 Adding Unary Predicates

We extend now our result to expansions of $(\mathbb{Z}, <, \equiv)$ by unary predicates that satisfy some computability assumptions. For the rest of this section, we fix a signature σ of unary predicates (not containing the symbols $\equiv$ and $<$) and a $(\sigma \cup \{\equiv, <\})$-structure $\mathcal{Z}_\sigma = (\mathbb{Z}, I)$ where $I(\equiv)$ and $I(<)$ are interpreted as expected.

Definition 5.10. We call a finite subset $\bar{P} \subseteq \sigma$ *bounded below* (*bounded above*, respectively) if $\bigcap_{P \in \bar{P}} I(P)$ is bounded below (bounded above, respectively).

We next define two conditions, (C1) and (C2), that imply the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ for $\mathcal{Z}_\sigma$.

(C1) The bounds of $\bar{P} \subseteq \sigma$ are effectively computable in the following sense: We can decide, given a finite subset $\bar{P} \subseteq \sigma$, whether $\bar{P}$ is bounded below (above, respectively). Additionally we can compute, given a finite subset $\bar{P} \subseteq \sigma$ that is bounded below (above, respectively), a bound $\ell(\bar{P}) \in \mathbb{Z}$ ($u(\bar{P}) \in \mathbb{Z}$, respectively) such that $\ell(\bar{P}) \leq z$ ($u(\bar{P}) \geq z$, respectively) for all $z \in \bigcap_{P \in \bar{P}} I(P)$.

(C2) For all finite subsets $\bar{P}_1, \bar{P}_2 \subseteq \sigma$ and all predicates $P \in \sigma$, if $\bar{P}_1$ is bounded below and $\bar{P}_2$ is bounded above, then we can effectively compute the finite set $I(P) \cap [\ell(\bar{P}_1), u(\bar{P}_2)]$.

The main result of this section is the following proposition.

Proposition 5.11. *If σ and I are chosen in such a way that $\mathcal{Z}_\sigma$ satisfies conditions (C1) and (C2), then $\mathcal{Z}_\sigma$ has property* EHD(BMW)*. The analogous result holds for $\mathcal{N}_\sigma = \mathcal{Z}_{\sigma \restriction \mathbb{N}}$.*

We fix a finite subsignature $\tau \subseteq \sigma$. Due to (C1), we can compute $m < M \in \mathbb{Z}$ such that m is a lower bound for all $\bar{P} \subseteq \tau$ that are bounded below and M is an upper bound for all $\bar{P} \subseteq \tau$ that are bounded above. We fix the numbers m and M for the rest of this section.

Let $\mathcal{A} = (A, J)$ be a $(\tau \cup \{<, \equiv\})$-structure. The proof of Proposition 5.11 uses a decomposition of $\mathcal{A}$ into four parts, called "the bounded part", "the greater part", "the smaller part" and "the rest".

Intuitively, an element $a \in A$ belongs to the bounded part if we know a priori that any homomorphism h from $\mathcal{A}$ to $\mathcal{Z}_\tau$ (we write $\mathcal{Z}_\tau$ for the reduct of $\mathcal{Z}_\sigma$ with signature $\tau \cup \{\equiv, <\}$) maps a to an element in the interval $[m, M]$. Similarly, the greater part consists of all elements $a \in A$ that do not belong the bounded part but any homomorphism to $\mathcal{Z}_\tau$ must map a above m, and the smaller part consists of all elements $a \in A$ that do not belong to the bounded part but any homomorphism to $\mathcal{Z}_\tau$ must map a below M.

We then reduce the question whether $\mathcal{A}$ can be embedded into $\mathcal{Z}_\tau$ to the questions whether the bounded part satisfies a certain MSO-formula and whether the $\{<, \equiv\}$-reducts of "the greater part", "the smaller part" and "the rest", allow a homomorphism to $(\mathbb{N}, <, \equiv)$, $(\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$, and $(\mathbb{Z}, <, \equiv)$ respectively.

Definition 5.12. Let $\mathcal{A} = (A, J)$ be a $(\tau \cup \{<, \equiv\})$-structure. We denote by $\tilde{\mathcal{A}} = (\tilde{A}, \tilde{J})$ the $\sim$-quotient of $\mathcal{A}$ (cf. Definition 5.6).

We call $a \in A$ *bounded below* if there is some $b \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[b]$ to $[a]$, and a subset $\bar{P} \subseteq \tau$ which is bounded below such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$.

We call $a \in A$ *bounded above* if there is some $b \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[a]$ to $[b]$, and a subset $\bar{P} \subseteq \tau$ which is bounded above such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$.

An example will help clarify the meaning of this definition.

Example 5.13. Suppose $\tau = \{\equiv_0\}$ and $\mathcal{Z}_\tau = (\mathbb{Z}, I)$ is the $(\tau \cup \{<, \equiv\})$-structure where $I(\equiv_0) = \{0\}$, as expected. In Figure 5.1, we show a $(\tau \cup \{<, \equiv\})$-structure $\mathcal{A}$, where by definition 5.12, a_0, a_1, a_2 are bounded above, while a_3, a_4, a_5 are

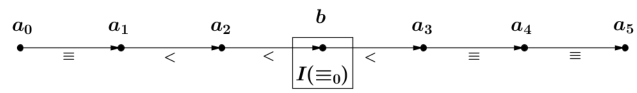

Figure 5.1: We represent here a $\{<, \equiv, \equiv_0\}$-structure $\mathcal{A} = (A, I)$. Labeled directed edges are used to depict the binary relations while the interpretation of the unary predicate $\equiv_0$ is shown as a set.

bounded below. Notice that, by the fact that $b \in I(\equiv_0)$, any homomorphism h from $\mathcal{A}$ to $\mathcal{Z}_\tau$ should satisfy $h(b) = 0$. Being there a $<$-path from a_1 to b, then we know that $h(a_1) < h(b) = 0$. This is the reason behind calling a_1 bounded above. The same kind of reasoning can be applied to the other elements.

With these preparations, we can easily define the four substructures mentioned above.

Definition 5.14. For a $(\tau \cup \{<, \equiv\})$-structure $\mathcal{A} = (A, J)$ we define

- the bounded part $B = \{a \in A \mid a \text{ is bounded below and bounded above}\}$,
- the greater part $G = \{a \in A \mid a \text{ is bounded below but not bounded above}\}$,
- the smaller part $S = \{a \in A \mid a \text{ is bounded above but not bounded below}\}$, and
- the rest $R = \{a \in A \mid a \text{ is neither bounded above nor bounded below}\}$.

Let us start with two simple lemmas.

Lemma 5.15. *Let $h : \mathcal{A} \to \mathcal{Z}_\tau$ be a homomorphism. Then the following holds:*

- *If $a \in B$ then $m \leq h(a) \leq M$.*
- *If $a \in S$ then $h(a) \leq M$.*
- *If $a \in G$ then $m \leq h(a)$*

Proof. It suffices to show that if a is bounded below (bounded above, respectively), then $m \leq h(a)$ ($h(a) \leq M$, respectively). If a is bounded below, then if there is some $b \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[b]$ to $[a]$, and a subset $\bar{P} \subseteq \tau$ which is bounded below such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$. We get $m \leq h(b) \leq h(a)$. If a is bounded above, we can argue in the same way. □

Lemma 5.16. *The following relations are disjoint from $J(<)$: $B \times S$, $B \times R$, $G \times B$, $R \times B$, $G \times S$, $G \times R$, $R \times S$.*

Proof. Assume for instance $(b, s) \in J(<)$ for some $b \in B$ and $s \in S$. Since $b \in B$, b is bounded from below. Hence, there is some $c \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[c]$ to $[b]$, and a subset $\bar{P} \subseteq \tau$ which is bounded below such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$. Hence, there is also a $<$-path in $\tilde{\mathcal{A}}$ from $[c]$ to $[s]$, i.e., s is bounded below, which contradicts $s \in S$. The other cases can be proved analogously. $\square$

Remark 5.17. The parts B, G, S, and R are all MSO- and WMSO-definable in the sense that there are MSO-formulas $\chi_i(x)$ for $i \in \{B, G, S, R\}$ with one free first-order variable x such that $\mathcal{A} \models \chi_i(a)$ for each $a \in A$ if and only if a belongs to the part i (and the same holds if we interpret $\chi_i(x)$ as a WMSO-formula).

We next state three lemmas that allow to prove Proposition 5.11.

Lemma 5.18. *We have $\mathcal{A} \preceq \mathcal{Z}_\tau$ if and only if $\mathcal{A}_{\restriction B} \preceq \mathcal{Z}_{\tau\restriction[m,M]}$ and $\mathcal{A}_{\restriction G\cup S\cup R} \preceq \mathcal{Z}_\tau$.*

Lemma 5.19. *Given a finite $\tau \subseteq \sigma$ we can compute an* MSO*-sentence ψ_B such that $\mathcal{A}_{\restriction B} \preceq \mathcal{Z}_{\tau\restriction[m,M]}$ if and only if $\mathcal{A}_{\restriction B} \models \psi_B$.*

Lemma 5.20. *The following four conditions are equivalent:*

1. *There is a homomorphism $h : \mathcal{A}_{\restriction G\cup S\cup R} \to \mathcal{Z}_{\tau\restriction\mathbb{Z}\setminus[m,M]}$ with $h(G) \subseteq [M+1, \infty)$, $h(S) \subseteq (-\infty, m-1]$.*
2. *$\mathcal{A}_{\restriction G\cup S\cup R} \preceq \mathcal{Z}_\tau$*
3. *$(G \cup S \cup R, J(<), J(\equiv)) \preceq (\mathbb{Z}, <, \equiv)$, $(G, J(<), J(\equiv)) \preceq (\mathbb{N}, <, \equiv)$, and $(S, J(<), J(\equiv)) \preceq (\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$*
4. *There is a homomorphism $h : (G \cup S \cup R, J(<), J(\equiv)) \to (\mathbb{Z}, <, \equiv)$ with $h(G) \subseteq \mathbb{N}$, $h(S) \subseteq \mathbb{Z} \setminus \mathbb{N}$.*

Before we prove these lemmas, we show how they imply Proposition 5.11.

Proof (Proposition 5.11). Fix a finite subsignature $\tau \subseteq \sigma$. By Lemma 5.18 we have $\mathcal{A} \preceq \mathcal{Z}_\tau$ if and only if $\mathcal{A}_{\restriction B} \preceq \mathcal{Z}_{\tau\restriction[m,M]}$ and $\mathcal{A}_{\restriction G\cup S\cup R} \preceq \mathcal{Z}_\tau$. By Lemma 5.19 we can compute from τ an MSO-sentence ψ_B such that $\mathcal{A}_{\restriction B} \models \psi_B$ if and only if $\mathcal{A}_{\restriction B} \preceq \mathcal{Z}_{\tau\restriction[m,M]}$. Moreover, from Lemma 5.20 we know that $\mathcal{A}\restriction G \cup S \cup R \preceq \mathcal{Z}_\tau$ if and only if

- $(G \cup S \cup R, J(<), J(\equiv)) \preceq (\mathbb{Z}, <, \equiv)$,
- $(G, J(<), J(\equiv)) \preceq (\mathbb{N}, <, \equiv)$, and
- $(S, J(<), J(\equiv)) \preceq (\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$.

The structures $(\mathbb{Z}, <, \equiv), (\mathbb{N}, <, \equiv)$, $(\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$ have the property $\mathsf{EHD}(\mathsf{BMW})$. Hence, there are BMW-sentences ψ_G, ψ_S, and ψ_R such that $\mathcal{A}_{\restriction G \cup S \cup R} \preceq \mathcal{Z}_\tau$ if and only if

- $(G \cup S \cup R, J(<), J(\equiv)) \models \psi_R$,
- $(G, J(<), J(\equiv)) \models \psi_G$, and
- $(S, J(<), J(\equiv)) \models \psi_S$.

Since the subsets B, G, S and R of $\mathcal{A}$ are MSO-definable as well as WMSO-definable (cf. Remark 5.17), we can compute relativizations of ψ_i to part i for $i \in \{B, G, S, R\}$ and obtain BMW-sentences $\varphi_B, \varphi_R, \varphi_G$ and φ_S such that

- $\mathcal{A} \models \varphi_B$ if and only if $\mathcal{A}_{\restriction B} \models \psi_B$,
- $\mathcal{A} \models \varphi_R$ if and only if $(G \cup S \cup R, J(<), J(\equiv)) \models \psi_R$,
- $\mathcal{A} \models \varphi_G$ if and only if $(G, J(<), J(\equiv)) \models \psi_G$, and
- $\mathcal{A} \models \varphi_S$ if and only if $(S, J(<), J(\equiv)) \models \psi_S$.

Putting everything together, we have $\mathcal{A} \preceq \mathcal{Z}_\tau$ if and only if $\mathcal{A} \models \varphi_B \wedge \varphi_G \wedge \varphi_S \wedge \varphi_R$. $\square$

We now prove the auxiliary lemmas (in a different order).

Proof (Lemma 5.20). The direction $(1 \Rightarrow 2)$ is trivial. Let us prove $(2 \Rightarrow 3)$, $(3 \Rightarrow 4)$, and $(4 \Rightarrow 1)$.

$(2 \Rightarrow 3)$ Let $h : \mathcal{A}_{\restriction G \cup S \cup R} \to \mathcal{Z}_\tau$ be a homomorphism. It follows immediately that h is also a homomorphism from the reduct $(G \cup S \cup R, J(<), J(\equiv))$ to $(\mathbb{Z}, <, \equiv)$.

Let $a \in G$. Then $h(a) \geq m$ by Lemma 5.15. Setting $h' : G \to \mathbb{N}$ with $h'(a) = h(a) - m$ yields a homomorphism from $(G, J(<), J(\equiv))$ to $(\mathbb{N}, <, \equiv)$.

The proof for $(S, J(<), J(\equiv)) \preceq (\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$ is analogous.

$(3 \Rightarrow 4)$ Assume that there are homomorphisms

$$\begin{aligned} h : (G \cup S \cup R, J(<), J(\equiv)) &\to (\mathbb{Z}, <, \equiv), \\ h_G : (G, J(<), J(\equiv)) &\to (\mathbb{N}, <, \equiv), \text{ and} \\ h_S : (S, J(<), J(\equiv)) &\to (\mathbb{Z} \setminus \mathbb{N}, <, \equiv). \end{aligned}$$

Define the mapping $h' : G \cup S \cup R \to \mathbb{Z}$ by

$$h'(a) = \begin{cases} h(a) & \text{if } a \in R, \\ \max(h(a), h_G(a)) & \text{if } a \in G, \\ \min(h(a), h_S(a)) & \text{if } a \in S. \end{cases}$$

With Lemma 5.16 one easily concludes that this is the desired homomorphism.

(4 $\Rightarrow$ 1) Let $h : (G \cup S \cup R, J(<), J(\equiv)) \to (\mathbb{Z}, <, \equiv)$ be the homomorphism from 4. Let $\mathcal{P}^+$ be the set of subsets of τ that are not bounded above and let $\mathcal{P}^-$ be the set of subsets of τ that are not bounded below. We define a sequence $(\eta_i)_{i\in\mathbb{Z}}$ of integers as follows:

- $\eta_0 = M + 1$,
- η_{-1} is the maximal number such that for each $\bar{P} \in \mathcal{P}^-$ there is a $\eta_{-1} \leq z < m$ with $z \in I(P)$ for all $P \in \bar{P}$ (we set $\eta_{-1} = m - 1$ if $\mathcal{P}^- = \emptyset$),
- for $i > 0$ let η_i be minimal such that for each $\bar{P} \in \mathcal{P}^+$ there is a $\eta_{i-1} \leq z < \eta_i$ with $z \in I(P)$ for all $P \in \bar{P}$ (we set $\eta_i = \eta_{i-1} + 1$ if $\mathcal{P}^+ = \emptyset$),
- for $i < -1$ let η_i be maximal such that for each $\bar{P} \in \mathcal{P}^-$ there is a $\eta_i \leq z < \eta_{i+1}$ with $z \in I(P)$ for all $P \in \bar{P}$ (we set $\eta_i = \eta_{i+1} - 1$ if $\mathcal{P}^- = \emptyset$).

For all $a \in G \cup S \cup R$ let $\bar{P}_a = \{P \in \tau \mid [a] \in \tilde{J}(P)\}$. Note that for all $r \in R$, $\bar{P}_r$ is neither bounded above or below (otherwise r would be bounded above or below, respectively), for all $g \in G$, $\bar{P}_g$ is not bounded above and for all $s \in S$, $\bar{P}_s$ is not bounded below. We conclude that the following map $h' : G \cup S \cup R \to \mathbb{Z}$ is well defined:

$$h'(a) = \min\{z \in \mathbb{Z} \mid \eta_{h(a)} \leq z < \eta_{h(a)+1} \text{ and } z \in I(P) \text{ for all } P \in \bar{P}_a\}.$$

Since h preserves $<$ and $\equiv$, h' does the same. Moreover, h' is defined in such a way that it preserves all unary predicates from τ.

Next we show that the image of h' has empty intersection with the interval $[m, M]$. By definition of η_{-1}, η_0 and h', $h'(a) \in [m, M]$ would imply $h(a) = -1$ Note that by by our assumptions on h, this implies $a \in R \cup S$. In particular, $\bar{P}_a$ cannot be bounded below, i.e., $\bar{P}_a \in \mathcal{P}^-$. Thus, there is a minimal $\eta_{-1} \leq z < m$ such that $z \in I(P)$ for all $P \in \bar{P}_a$. This implies $h'(a) = z < m$ which completes our claim. Thus, h' is a homomorphism from $\mathcal{A}_{\restriction G\cup S\cup R}$ to $\mathcal{Z}_{\tau\restriction \mathbb{Z}\setminus[m,M]}$.

To show that $h'(G) \subseteq [M + 1, \infty)$ and $h(S) \subseteq (-\infty, m - 1]$ note that $h(G) \subseteq \mathbb{N}$ and $h(S) \subseteq \mathbb{Z} \setminus \mathbb{N}$. This implies $h'(G) \subseteq [M + 1, \infty)$ and $h'(S) \subseteq (-\infty, M]$. Hence, $h'(S) \subseteq (-\infty, m - 1]$ by the previous paragraph.

□

Proof (Lemma 5.18). If $h : \mathcal{A} \to \mathcal{Z}_\tau$ is a homomorphism, then the restrictions of h to B and $G \cup S \cup R$ witness $\mathcal{A}_{\restriction B} \preceq \mathcal{Z}_{\tau\restriction[m,M]}$ (here we use Lemma 5.15) and $\mathcal{A}_{\restriction G\cup S\cup R} \preceq \mathcal{Z}_\tau$.

Now assume that $h_1 : \mathcal{A}_{\restriction B} \to \mathcal{Z}_{\tau\restriction B}$ and $h_2 : \mathcal{A}_{\restriction G\cup S\cup R} \to \mathcal{Z}_\tau$ are homomorphisms. By Lemma 5.20 there exists a homomorphism $h_2' : \mathcal{A}_{\restriction G\cup S\cup R} \to \mathcal{Z}_\tau$ such that $h(G) \subseteq [M+1,\infty)$ and $h(S) \subseteq (-\infty, m-1]$. We define the mapping $h : A \to \mathbb{Z}$ by $h(b) = h_1(b) \in [m, M]$ for $b \in B$ and $h(a) = h_2'(a)$ for $a \in G\cup S\cup R$. This mapping preserves $\equiv$ and all unary predicated. Moreover, using Lemma 5.16 it follows easily that it preserves also the relation $<$.

□

Proof (Lemma 5.19). A homomorphism $h : \mathcal{A}_{\restriction B} \to \mathcal{Z}_{\tau\restriction[m,M]}$ can be identified with a partition of B into $M - m + 1$ sets $B_m, \dots, B_M$, where $B_i = \{a \in B \mid h(a) = i\}$. Hence, the MSO-sentence ψ_B from Lemma 5.19 states that there is a partition of B into $M - m + 1$ sets $B_m, \dots, B_M$ such that the corresponding mapping $h : B \to [m, M]$ preserves all relations from τ. Fixing a tuple of $M-m+1$ many set variables $\overline{X} = (X_m, \dots, X_M)$, we want to define formulas with the following properties:

- $\psi_{\text{part}}(\overline{X})$ expresses that $\overline{X}$ forms a finite partition.
- $\psi_<(\overline{X})$ expresses that the partition preserves the relation $I(<)$.
- $\psi_=(\overline{X})$ expresses that the partition preserves the relation $I(\equiv)$.
- $\psi_\tau(\overline{X})$ expresses that the partition preserves every unary relation $P \in \tau$.

These formulas can be defined as follows:

$$\psi_{\text{part}} = \forall x \bigvee_{i\in[m,M]} \Big(x \in X_i \wedge \bigwedge_{\substack{j\in[m,M]\\ i\neq j}} x \notin X_j \Big),$$

$$\psi_< = \forall x\, \forall y \left(x < y \to \bigvee_{\substack{i,j\in[m,M]\\ i<j}} (x \in X_i \wedge y \in X_j) \right),$$

$$\psi_= = \forall x\, \forall y \left(x \equiv y \to \bigvee_{i\in[m,M]} (x \in X_i \wedge y \in X_i) \right),$$

$$\psi_\tau = \bigwedge_{P\in\tau} \forall x \left(x \in P \to \bigvee_{i\in I(P)\cap[m,M]} x \in X_i \right).$$

Note that the formulas of the last form are all computable due to condition (C2). Now we can define $\psi_B = \psi_{\text{part}} \wedge \psi_< \wedge \psi_= \wedge \psi_\tau$. □

5.4 Expansions of $\mathbb{Z}$ that satisfy Conditions (C1) and (C2)

In this section, we will present concrete examples of unary relations that satisfy the conditions (C1) and (C2) from the previous section.

Definition 5.21. Define the signature

$$\sigma = \{F_S, C_S \mid S \subseteq \mathbb{Z} \text{ finite}\} \cup \{\equiv_{a,b} \mid a, b \in \mathbb{N}, a < b\},$$

where all symbols are unary. We define the structure $\mathcal{Z}_\sigma = (\mathbb{Z}, I)$ where $I(F_S) = S$, $I(C_S) = \mathbb{Z} \setminus S$, and $\equiv_{a,b}$ holds at all $z \in \mathbb{Z}$ such that $z = a \bmod b$.

Note that the $\mathcal{Z}_\sigma$ (which is defined over the signature $\sigma \cup \{<, \equiv\}$, see the first paragraph of Section 5.3) is an expansion of $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\le a<b})$ from (1.2) because $\equiv_a$ is the same relation as $F_{\{a\}}$.

Lemma 5.22. *$\mathcal{Z}_\sigma$ satisfies the conditions (C1) and (C2).*

Proof. The condition (C2) holds trivially because all set $I(P)$ for $P \in \sigma$ are computable sets and the map $P \mapsto I(P)$ is computable.

It remain to show (C1). Let $\bar{P} = \bar{F} \cup \bar{C} \cup \bar{M}$ be a finite set where $\bar{F} \subseteq \{F_S \mid S \subseteq \mathbb{Z} \text{ finite}\}$, $\bar{C} \subseteq \{C_S \mid S \subseteq \mathbb{Z} \text{ finite}\}$, and $\bar{M} \subseteq \{\equiv_{a,b} \mid a, b \in \mathbb{N}, a < b\}$.

Note that $\bar{F} \neq \emptyset$ implies that $\bar{P}$ is bounded above and below. Otherwise $\bar{P}$ is bounded above (below) if and only if $\bar{M}$ is bounded above (below). Let $\bar{M} = \{\equiv_{a_1,b_1}, \dots, \equiv_{a_k,b_k}\}$ and

$$S = \bigcap_{i=1}^{k} \{a_i + zb_i \mid z \in \mathbb{Z}\}.$$

The set S is either empty or of the form $\{y + z \cdot \mathrm{lcm}(b_1, b_2, \dots, b_k) \mid z \in \mathbb{Z}\}$ for some $y \in \mathbb{Z}$ (and hence neither bounded below nor bounded above), where $\mathrm{lcm}(b_1, b_2, \dots, b_k)$ is the least common multiple of the numbers $b_1, b_2, \dots, b_k$. The latter holds if and only if $a_i = a_j \bmod \gcd(b_i, b_j)$ for all $i \neq j$ (see [21, Theorem 2.4.2]).

For those $\bar{P}$ that are bounded above (bounded below), it is easy to compute an upper bound (a lower bound). If $\bar{F}$ is nonempty, take an $F_S \in \bar{F}$ and use $\min(S)$ and $\max(S)$ as bounds. If $\bar{P}$ is bounded and $\bar{F}$ is empty, then 0 is a lower and upper bound. □

We can add further unary predicates and still have conditions (C1) and (C2). Let Prim be the set of prime numbers. Consider the expansion $\mathcal{Z}_{\sigma\cup\{\pi,\overline{\pi}\}}$ of the structure $\mathcal{Z}_\sigma$ from Definition 5.21, where $I(\pi) = \mathsf{Prim}$ and $I(\overline{\pi}) = \mathbb{Z} \setminus \mathsf{Prim}$. The following result of Dirichlet relates prime numbers to modulo constraints. Recall that two natural numbers n_1, n_2 are *coprime* if there is no prime p such that $p \mid n_1$ and $p \mid n_2$.

Theorem 5.23 (Dirichlet's Theorem)**.** *Let $a < b \in \mathbb{N}$. The equation $x = a \bmod b$ has infinitely many solutions that are prime if and only if a and b are coprime.*

If a and b are not coprime, let $p \neq 1$ be a common divisor of both. Every solution of $x = a \bmod b$ is a multiple of p whence there is at most 1 solution that is a prime which can be computed from a and b.

There is an easy observation relating the complement of the primes with the modulo predicates.

Lemma 5.24. *For all numbers $a < b \in \mathbb{N}$ there are infinitely many solutions of $x = a \bmod b$ that are not prime numbers.*

Proof. There are three cases:

- If $a = 0$, all solutions above b are not prime.
- If $a = 1$, assume that $n \in \mathbb{N}$ is a solution of $x = 1 \bmod b$. Then $n^k = 1 \bmod b$ for all $k \geq 2$ and we obtain infinitely many non-prime solutions.
- If $a > 1$, then for all $n \in \mathbb{N}$ we have $a + bna = a \bmod b$ and $a + bna$ is not a prime because it is a multiple of a.

□

Corollary 5.25. *The structure $\mathcal{Z}_{\sigma \cup \{\pi, \overline{\pi}\}}$ has the property* EHD(BMW)*.*

Proof. Take a subset $\bar{P}$ of the unary relations from $\sigma \cup \{\pi, \overline{\pi}\}$, where σ is from Definition 5.21. Then, we first determine whether the intersection of all unary relations from $\sigma \cap \bar{P}$ is finite or not, as in the proof of Lemma 5.22. If the intersection is infinite then it is of the form $S = \{c + z \cdot b \mid z \in \mathbb{Z}\} \setminus F$ for $c < b \in \mathbb{Z}$ and a finite set $F \setminus \mathbb{Z}$, which can be computed. Clearly, $\mathsf{Prim} \cap S$ is bounded below by 0 and by Dirichlet's Theorem it is bounded above if and only if c and b are not coprime, in which case an upper bound can be computed from b and c. The set $(\mathbb{Z} \setminus \mathsf{Prim}) \cap S$ is neither bounded below nor bounded above (by Lemma 5.24). Since Prim and $\mathbb{Z} \setminus \mathsf{Prim}$ are computable, properties (C1) and (C2) hold. □

Since $\mathcal{Z}_{\sigma \cup \{\pi, \overline{\pi}\}}$ is also negation-closed, we get:

Corollary 5.26. *The problem $\mathcal{Z}_{\sigma \cup \{\pi, \overline{\pi}\}}$-*SAT *is decidable.*

At the end of this section, we want to briefly mention that the expansion of $\mathbb{Z}$ under consideration may contain undecidable unary predicates. Take some undecidable set $H \subseteq \mathbb{N}$, e.g., the halting problem. Consider the structure $\mathcal{Z}' = (\mathbb{Z}, <, \equiv, H, \bar{H})$, where $\bar{H} = \mathbb{Z} \setminus H$. Then $\{H, \bar{H}\}$ satisfies conditions (C1) and (C2). Just note that H is bounded below but not above, $\bar{H}$ is neither bounded below nor above and $H \cap \bar{H} = \emptyset$. Thus, the bounds m and M for the bounded part can be chosen to be $m = 0, M = -1$. The conditions on the bounded part reduce to the fact that it has to be empty. Since $\mathcal{Z}'$ is also negation-closed, we conclude that $\mathcal{Z}'$-SAT is decidable.

5.5 A Concrete Domain over $\mathbb{Q}$

A simple adaptation of our proof for the concrete domain $\mathcal{Z}$ shows that the negation-closed structure $\mathcal{Q} = (\mathbb{Q}, <, \equiv, (\equiv_q)_{q\in\mathbb{Q}})$ has the property EHD(MSO): It can easily be shown that for any $\{<, \equiv, (\equiv_q)_{q\in\mathbb{Q}}\}$-structure $\mathcal{A}$, $\mathcal{A} = (A, I) \preceq \mathcal{Q}$ if and only if

- (A, E) is acyclic, where $E =$ Id$\circ I(<)\circ$Id and Id is the symmetric transitive closure of the equality relation, i.e. $(I(\equiv) \cup I(\equiv)^{-1})^*$,
- there is no $(a, b) \in E^+$ with $a \in I(\equiv_p)$, $b \in I(\equiv_q)$ and $q \leq p$, and
- there is no $(a, b) \in$ Id with $a \in I(\equiv_p)$, $b \in I(\equiv_q)$, and $q \neq p$.

Since these conditions are easily expressible in MSO, it follows that $\mathcal{Q}$-SAT is decidable.

Thanks to the density of the rational numbers there is no need to use the bounding quantifier to bound the number of elements between any two given rational numbers.

Chapter 6

"Tree-Like" Concrete Domains

We have developed the EHD-method with the initial intention to solve the satisfiability problem of CECTL^* (and CCTL^*) with constraints over the integers. But the general nature of this method raised the question whether this could be successfully applied to other interesting concrete domains. The complete infinite binary tree $\mathcal{T}_2$, i.e. the set of finite words over the alphabet $\{0,1\}$, or the infinitely branching infinite tree $\mathcal{T}_\infty$, equipped with the prefix relation $<$ seemed to be good candidates for domains, and the problem was also mentioned in [19]. Imagine to want to describe all those $\{<\}$-structures which allow a homomorphism into $\mathcal{T}_\infty$. At first glance it would seem to be enough to avoid cycles and bound the number of predecessors of each element to obtain our purpose, and both these properties can be expressed using boolean combinations of MSO and WMSO+B sentences. But recall that, aside the EHD-property, a second condition is necessary to apply the EHD-method: negation closure. To obtain this, we need to consider both $\mathcal{T}_\infty$ and $\mathcal{T}_2$ as structures over the extended signature $\{<, \equiv, \perp\}$, where $\equiv$ is the equality relation and $\perp$ contains all pairs of incomparable nodes. Difficulties arise once we add the "incomparability" relation: It turns out that MSO and WMSO+B are not sufficient to distinguish between those $\{<, \equiv, \perp\}$-structures which allow homomorphism to $\mathcal{T}_\infty$ and those which do not.

Nevertheless, the study of this case gave rise to two results. Firstly, we prove that the infinitely branching infinite tree $\mathcal{T}_\infty$ does not enjoy the property EHD(BMW) and, since both $\mathcal{T}_\infty \preceq \mathcal{T}_2$ and $\mathcal{T}_2 \preceq \mathcal{T}_\infty$, this implies that the binary tree does not either.

Theorem 6.1. *There is no* Bool(MSO, WMSO+B)*-sentence ψ such that for every countable structure $\mathcal{A}$ (over the signature $\{<, \perp, \equiv\}$) we have: $\mathcal{A} \models \psi$ if and only if there is a homomorphism from $\mathcal{A}$ to $\mathcal{T}_\infty$.*

To obtain this result we develop an Ehrenfeucht-Fraïssé-game for WMSO+B and use it to show that the logic BMW cannot distinguish between two structures, one which allows homomorphism into $\mathcal{T}_\infty$ and one which does not.

Remark 6.2. We want to remark that the fact that the EHD-method is not applicable to the case of the binary tree does not imply the undecidability of the satisfiability problem for temporal logics with constraints over such structure. In fact there have been some interesting developments in recent work.

In [28], Kartzow and Weidner prove PSPACE-completeness of the satisfiability problem for CLTL with constraints over trees, enriched with lexicographic ordering, using an automata-theoretic approach.

At the same time Demri and Deters prove decidability of both CLTL and CCTL* with constraints over the tree, enriched with the ability to compare lengths of longest common prefixes [18]. They also establish PSPACE-completeness in the linear time case.

The result is obtained by a reduction to satisfiability of CLTL and CCTL* with constraints over the domain $(\mathbb{N}, \equiv, <, (\equiv_a)_{a\in\mathbb{N}})$, where the linear time case is decidable by [19], and the corresponding result for the branching time setting is established in [11], using the EHD-method.

The idea behind the reduction is to express a prefix constraint $x < y$ (where x and y are interpreted as words from $\mathbb{N}^*$) using the equivalent formula $\mathrm{clen}(x, y) = \mathrm{clen}(x, x)$, which states that the length of the longest common prefix of x and y is the same as the length of x. One can successively translate these formulas using constraints on the natural numbers by associating to each term $\mathrm{clen}(x, y)$ a register variable $r_{x,y}$ and asking that $r_{x,y} = r_{x,x}$. In [18] the authors individuate conditions which ensure that a valuation for the register variables $r_{x,y}$ can be compiled into a correct assignment for the string variables x and y, and are able to express such conditions in CLTL and CCTL*. In the branching time case, one has to additionally take care of the correct propagation of the values from the register variables in the different branches of the $\mathcal{T}$-Kripke Structures. It is essential here that CCTL* has the bounded tree-model property ([25]): From a given formula φ we can compute a number d such that φ is satisfiable if and only if it admits a d-tree model. Then the issue is solved by creating d copies of each $r_{x,y}$ and regulating their behavior with CCTL* formulas (see Lemma 10 from [18]). Since also CECTL* has the bounded tree-model property, the same idea should work for this logic. Being CCTL* a fragment of CECTL*, the formulas from Lemma 10 in [18] would guarantee a correct reduction also in this case.

As a second main result of this chapter, we identify three classes of "tree-like" structures which do enjoy the EHD-property, namely *semi-linear orders, ordinal trees* and *infinitely branching trees of height h* for each fixed h. In the following section we introduce such classes and prove that they are negation-closed and enjoy the property $\mathsf{EHD}(\mathcal{L})$ in the sense of Section 4.2.1, where $\mathcal{L}$ is either MSO or WMSO, depending on the class. This implies:

Theorem 6.3. *Δ-satisfiability (see Def. 4.18) of CECTL^*-formulas is decidable when Δ is one of the following:*

(1) the class of all semi-linear orders,

(2) the class of all ordinal trees, and

(3) for each $h \in \mathbb{N}$, the class of all order trees of height h.

We would like to remark that for this result (in particular to establish the EHD-property), we do not need to assume that the domain of the considered structures is countable. For example, in the case of semi-linear orders, we can find a WMSO-sentence φ such that for all $\{<, \perp\}$-structures $\mathcal{A}$, may they be countable or not, $\mathcal{A} \preceq \mathcal{B}$ for some $\mathcal{B} \in \Gamma$ if and only if $\mathcal{A} \models \varphi$. This does not in fact have any effect on the decidability procedure, as the constraint graphs that we obtain through the EHD-method (for which we need to check whether they satisfy φ) have countable domain. Nonetheless what we present here is a more general result.

6.1 "Tree-like" Structures

We now introduce the structures we will be dealing with in this chapter.

Definition 6.4. A **semi-linear order** (in Wolk's work [43] simply a *tree*) is a partial order $\mathcal{P} = (P, <)$ with the additional property that for all $p \in P$ the suborder induced by $\{p' \in P \mid p' \leq p\}$ forms a linear order.

This property is equivalent to the one formulated by Wolk [43]: Given incomparable elements $p_1, p_2 \in P$, there is no $q \in P$ such that $p_1 < q$ and $p_2 < q$, i.e., two incomparable elements cannot have a common descendant. Clearly all trees (in the usual sense) satisfy this property, but not vice-versa.

Definition 6.5. We call a semi-linear order $\mathcal{P} = (P, <)$ an **ordinal forest** if for all $p \in P$ the linear suborder induced by $\{p' \in P \mid p' \leq p\}$ is an ordinal.

We call $\mathcal{P}$ a **forest** if if for all $p \in P$ the linear suborder induced by $\{p' \in P \mid p' \leq p\}$ is finite.

A forest (ordinal forest) is a **tree (ordinal tree)** if it has a unique minimal element.

A forest $\mathcal{F}$ of **height** h (for $h \in \mathbb{N}$) is a forest that contains a linear suborder with $h + 1$ many elements but no linear suborder with $h + 2$ elements. We say that an element $x \in P$ is **at level** i if $|\{y \in P \mid y < x\}| = i$. Thus, every minimal element is at level 0.

Given a partial order $(P, <)$, we denote by $\perp_<$ the **incomparability relation** defined by $p \perp_< q$ iff $p \neq q$ and neither $p < q$ nor $q < p$ hold. Given a $\{<, \perp, \equiv\}$-structure $\mathcal{P} = (P, <, \perp, \equiv)$ such that $(P, <)$ is a semi-linear order (resp., ordinal tree, tree of height h), $\equiv$ is the equality relation on P, and $\perp = \perp_<$, then we also say that $\mathcal{P}$ is a semi-linear order (resp. ordinal tree, tree of height h).

Proposition 6.6. *For any class Δ of $\{<, \perp, \equiv\}$-structures such that in every $\mathcal{A} \in \Delta$*

(i) $<$ is interpreted as a strict partial order,

(ii) $\perp$ is interpreted as the incomparability with respect to $<$ (i.e., $\perp = \perp_<$), and

(iii) $\equiv$ is the equality relation,

Δ is negation-closed. In particular, the class of all semi-linear orders and all its subclasses are negation-closed.

Proof. For every $\mathcal{A} \in \Delta$ the following equalities hold, where A is the universe of $\mathcal{A}$:

$$
\begin{aligned}
(A^2 \setminus <) &= \{(x, y) \mid \mathcal{A} \models y < x \vee y \equiv x \vee x \perp y\} \ , \\
(A^2 \setminus \perp) &= \{(x, y) \mid \mathcal{A} \models x < y \vee x \equiv y \vee y < x\} \ , \\
(A^2 \setminus \equiv) &= \{(x, y) \mid \mathcal{A} \models x < y \vee x \perp y \vee y < x\} \ .
\end{aligned}
$$

□

Note that for this it is crucial that we add the incomparability relation $\perp$.

Remark 6.7. Recall the definition of the $\sim$-quotient $\tilde{\mathcal{A}}$ of a structure $\mathcal{A}$ (Def. 5.6) and the results of Lemma 5.8, establishing a connection between the EHD-property of a structure and the EHD-property of its $\sim$-quotient. It is not hard to see that this result applies also in the setting of Δ-satisfiability, where Δ is the class of semi-linear orders (ordinal trees, trees of fixed height). Therefore, in order to prove Theorem 6.3 it is enough to show that the class of all semi-linear orders (and its subclasses) seen as $\{<, \perp\}$-structures have the property EHD($\mathcal{L}$) for a suitable $\mathcal{L}$.

6.2 The EHD-Property for Semi-Linear Orders

Let Γ denote the class of all semi-linear orders (over $\{<, \perp\}$). The aim of this section is to prove that Γ has the property EHD(WMSO). For this purpose, we characterize all those structures that admit a homomorphism to some member of Γ. The resulting criterion can be easily translated into WMSO. Hence, we do not need the bounding quantifier from WMSO+B (the same will be true in the following Sections 6.3 and 6.4).

Let us prove one first reduction: It turns out that, in the case of semi-linear orders (and also ordinal forests) the existence of a homomorphism to a semi-linear order is in fact equivalent to the existence of a **compatible extension**. We say that a graph[1] $(A, <, \perp)$ can be **extended** to a semi-linear order (an ordinal forest) if there is a partial order $\lhd$ such that $(A, \lhd, \perp_\lhd)$ is a semi-linear order (an ordinal forest) **compatible** with $(A, <, \perp)$, i.e.,

$$x < y \Rightarrow x \lhd y \text{ and } x \perp y \Rightarrow x \perp_\lhd y \ . \tag{6.1}$$

We prove that, in this setting, restricting our attention to such compatible expansions is not a loss of generality:

Lemma 6.8. *The following are equivalent for every structure $\mathcal{A} = (A, <, \perp)$:*

1. *$\mathcal{A}$ can be extended to a semi-linear order (to an ordinal forest, respectively);*
2. *$\mathcal{A} \preceq \mathcal{B}$ for some semi-linear order (ordinal tree, respectively) $\mathcal{B}$.*

Proof. We start with the implication ($1 \Rightarrow 2$). Assume that $\mathcal{A}$ can be extended to a compatible semi-linear order (ordinal forest, resp.) $\mathcal{A}' = (A, \lhd, \perp_\lhd)$. Thanks to compatibility, the identity is a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$. In the case of an ordinal forest, one can add one common minimal element to obtain an ordinal tree.

Let us now prove ($2 \Rightarrow 1$). Suppose h is a homomorphism from $\mathcal{A} = (A, <, \perp)$ to some semi-linear order $\mathcal{B} = (B, \prec, \perp_\prec)$. We extend $\mathcal{A}$ to a compatible semi-linear order $(A, \lhd, \perp_\lhd)$. Let us fix an arbitrary well-order $<_{\text{wo}}$ on the set A (which exists by the axiom of choice). We define the binary relation $\lhd$ on A as follows:

$$x \lhd y \text{ if and only if } h(x) \prec h(y) \text{ or } (h(x) = h(y) \text{ and } x <_{\text{wo}} y),$$

As usual, we denote with $\perp_\lhd$ the incomparability relation for $\lhd$, i.e., $x \perp_\lhd y$ if and only if neither $x \lhd y$ nor $y \lhd x$ nor $x = y$ holds. We show that $(A, \lhd, \perp_\lhd)$ is a semi-linear order. In fact, irreflexivity and transitivity are easy consequences of the definition of $\lhd$ and of the fact that $\prec$ is a partial order. To show that $\lhd$ is semi-linear, assume that $x_1 \lhd x$ and $x_2 \lhd x$. By definition $h(x_1) \prec h(x)$ or $h(x_1) = h(x)$ and $h(x_2) \prec h(x)$ or $h(x_2) = h(x)$. By semi-linearity of $\mathcal{B}$, we

[1] We call $(A, <, \perp)$ a graph to emphasize that here the binary relation symbols $<$ and $\perp$ can have arbitrary interpretations and they need not be a partial order and its incomparability relation. We can instead see them as two different kinds of edges in an arbitrary graph (as they are in a constraint graph).

deduce that $h(x_1)$ and $h(x_2)$ are comparable and, by definition of $\lhd$, so are x_1 and x_2. It remains to show that $(A, \lhd, \perp_\lhd)$ is compatible with $\mathcal{A}$. Let $x < y$. Then, by the fact that h is a homomorphism, $h(x) \prec h(y)$ which guarantees that $x \lhd y$. If $x \perp y$, then $h(x) \perp_\prec h(y)$. Since $\mathcal{B}$ is a semi-linear order, this implies that neither $h(x) \prec h(y)$ nor $h(y) \prec h(x)$ nor $h(x) = h(y)$ holds. As a consequence none of $x \lhd y$, $y \lhd x$ and $x = y$ holds. Therefore we have $x \perp_\lhd y$.

The case in which $\mathcal{B}$ is an ordinal tree is dealt with similarly. It is enough to notice that $\lhd$ does not contain any infinite decreasing chains, since $\prec$ is well-founded and $<_{\mathrm{wo}}$ is a well-order.

□

Inspired by Wolk's work on comparability graphs [43, 44] we use Rado's selection lemma [38] in order to obtain the compactness result that a graph can be extended to a semi-linear order iff every one of its finite subgraphs is. Recall that a **choice function** for a family of sets $X = \{X_i \mid i \in I\}$ is a function f with domain I such that $f(i) \in X_i$ for all $i \in I$, i.e., it *chooses* one element from each set X_i.

Lemma 6.9 (Rado's selection lemma, cf. [26, 38])**.** *Let I be an arbitrary index set and let $X = \{X_i \mid i \in I\}$ be a family of finite sets. For each finite subset A of I, let f_A be a choice function for the family $\{X_i \mid i \in A\}$. Then there is a choice function f for X such that, for all finite $A \subseteq I$, there is a finite set B such that $A \subseteq B \subseteq I$ with $f(i) = f_B(i)$ for all $i \in A$.*

Lemma 6.10 (extension of Theorem 2 in [44])**.** *A structure $\mathcal{A} = (A, <, \perp)$ can be extended to a semi-linear order if and only if every finite substructure of $\mathcal{A}$ can be extended to a semi-linear order.*

Proof. The direction $(\Rightarrow)$ is trivial. For the direction $(\Leftarrow)$, let

$$I = \{\{x, y\} \subseteq A \mid x \neq y\}$$

be the set of pairs of distinct elements of A. For all $i = \{x, y\} \in I$ we define $Z_i = \{(x, y), (y, x), \#\}$. We want to find a choice function for the family of sets $\{Z_i \mid i \in I\}$ which is in some sense *compatible* with the relations $\perp$ and $<$. In fact, choosing for each $i \in I$ one element of Z_i corresponds intuitively to deciding whether the two elements x and y are comparable, and in which order, or if they are incomparable.

Each finite subset J of I defines a set $\bar{J} = \{x \in j \mid j \in J\}$ and a substructure $\mathcal{A}_{\restriction \bar{J}} = (A_{\restriction \bar{J}}, <_{\restriction \bar{J}}, \perp_{\restriction \bar{J}})$. Since $\mathcal{A}_{\restriction \bar{J}}$ is finite, by hypothesis it can be extended to a semi-linear order. Hence, we can find a partial order $\lhd_J$ on $\bar{J}$ such that $(A_{\restriction \bar{J}}, \lhd_J, \perp_{\lhd_J})$ is a semi-linear order compatible with $\mathcal{A}_{\restriction \bar{J}}$ as in (6.1) on page 67.

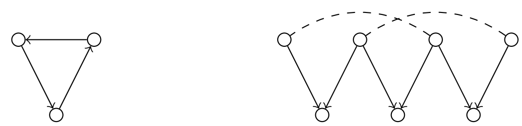

Figure 6.1: A $<$-cycle of three elements and an "incomparable triple-u", where dashed lines are $\perp$-edges.

Let f_J be the choice function for $\{Z_j \mid j \in J\}$ defined as follows:

$$f_J(\{x,y\}) = \begin{cases} (y,x) & \text{iff } y \lhd_J x, \\ (x,y) & \text{iff } x \lhd_J y, \\ \# & \text{otherwise.} \end{cases}$$

By Lemma 6.9 we can find a choice function f for $\{Z_i \mid i \in I\}$ such that for all finite $J \subseteq I$ there is a finite set K such that

$$J \subseteq K \subseteq I \text{ and } f(j) = f_K(j) \text{ for all } j \in J.$$

Define $x \lhd y$ iff $(x,y) \in f(I)$. We need to prove that $(A, \lhd, \perp_{\lhd})$ is an extension of $\mathcal{A}$ to a semi-linear order. But all the properties that we need to check are local, and thanks to Rado's selection lemma, $\lhd$ always coincides, on every finite subset of A, with some $\lhd_J$, which is a semi-linear order compatible with $<$ and $\perp$. $\square$

Thanks to Lemma 6.10, given a $\{<, \perp\}$-structure $\mathcal{A}$ we now only need to find necessary and sufficient conditions which guarantee that every finite substructure of $\mathcal{A}$ admits a homomorphism into a semi-linear order.

Definition 6.11. Let $\mathcal{A} = (A, <, \perp)$ be a graph. Given $A' \subseteq A$, we say A' is **connected (with respect to $<$)** if and only if, for all $a, a' \in A'$, there are $a_1, \ldots, a_n \in A'$ such that $a = a_1$, $a' = a_n$ and $a_i < a_{i+1}$ or $a_{i+1} < a_i$ for all $1 \leq i \leq n-1$. A **connected component** of $\mathcal{A}$ is a maximal (with respect to inclusion) connected subset of A. Given a subset $A' \subseteq A$ and $c \in A'$, we say that c is a **central point of** A' if and only if for every $a \in A'$ neither $a \perp c$ nor $c \perp a$ nor $a < c$ holds.

In other words, a central point of a subset $A' \subseteq A$ is a node of the structure $\mathcal{A} = (A, <, \perp)$ which has no incoming or outgoing $\perp$-edges, and no incoming $<$-edges within A'.

Example 6.12. A $<$-cycle (of any number of elements) does not have a central point, nor does an *incomparable triple-u*, see Figure 6.1. Both structures do not admit homomorphism into a semi-linear order. While this statement is obvious for the cycle, we leave the proof for the incomparable triple-u as an exercise.

Lemma 6.13. *A finite structure $\mathcal{A} = (A, <, \perp)$ can be extended to a semi-linear order if and only if every non-empty connected $B \subseteq A$ has a central point.*

Let us extract the main argument for the ($\Rightarrow$)-part of the proof for later reuse. It establishes a connection between minimal elements of connected subsets of A and central points:

Lemma 6.14. *Let $(A, \lhd, \perp_\lhd)$ be a semi-linear order extending $\mathcal{A} = (A, <, \perp)$. If a connected subset $B \subseteq A$ (with respect to $<$) contains a minimal element m with respect to $\lhd$, then m is central in B (again with respect to $\mathcal{A}$).*

Proof. Let $b \in B$. Since B is connected, there are $b_1, \ldots, b_n \in B$ such that $b_1 = m$, $b_n = b$ and $b_i < b_{i+1}$ or $b_{i+1} < b_i$ for all $1 \leq i \leq n-1$. As $\lhd$ is compatible with $<$, this implies that $b_i \lhd b_{i+1}$ or $b_{i+1} \lhd b_i$ for all $1 \leq i \leq n-1$. Given that m is minimal, applying semi-linearity of $\lhd$, we obtain that $m = b_i$ or $m \lhd b_i$ for all $1 \leq i \leq n$. In particular, we have $m = b$ or $m \lhd b$. Since $(A, \lhd, \perp_\lhd)$ is a semi-linear order, compatible with $(A, <, \perp)$, we cannot have $b < m$, $m \perp b$ or $b \perp m$ (since this would imply $b \lhd m$ or $m \perp_\lhd b$). Hence, m is central. $\square$

Proof of Lemma 6.13. For the direction ($\Rightarrow$) let B be any non-empty connected subset of A. Since B is finite, there is a minimal element m. Using the previous lemma we conclude that m is central in B.

We prove the direction ($\Leftarrow$) by induction on $n = |A|$. Suppose $n = 1$ and let $A = \{a\}$. The fact that $\{a\}$ has a central point implies that neither $a < a$ nor $a \perp a$ holds. Hence, $\mathcal{A}$ is a semi-linear order.

Suppose $n > 1$ and assume the statement to be true for all $i < n$. If $\mathcal{A}$ is not connected with respect to $<$, then we apply the induction hypothesis to every connected component. The union of the resulting semi-linear orders extends $\mathcal{A}$. Now assume that $\mathcal{A}$ is connected and let c be a central point of A. By the inductive hypothesis we can find $\lhd'$ such that $(A \setminus \{c\}, \lhd', \perp_{\lhd'})$ is a semi-linear order extending $\mathcal{A} \setminus \{c\}$. We define $\lhd := \lhd' \cup \{(c, x) \mid x \in A \setminus \{c\}\}$ (i.e., we add c as a smallest element), which is obviously a partial order on A.

To prove that $\lhd$ is semi-linear, let $a_1, a_2, a \in A$ such that $a_1 \lhd a$ and $a_2 \lhd a$. If $a_1 = c$ or $a_2 = c$, then a_1 and a_2 are comparable by definition. Otherwise, we conclude that $a_1, a_2, a \in A \setminus \{c\}$. Hence, $a_1 \lhd' a$ and $a_2 \lhd' a$, and semi-linearity of $\lhd'$ settles the claim.

We finally show compatibility. Suppose that $a < b$. If $a = c$, then $a \lhd b$. The case $b = c$ cannot occur, because c is central in A. The remaining possibility $a \neq c \neq b$ implies that $a \lhd' b$ and hence $a \lhd b$ as desired. Finally, suppose that $a \perp b$. Then $a \neq c \neq b$, because c is central. We conclude that $a \perp_{\lhd'} b$ and also $a \perp_\lhd b$. $\square$

We are finally ready to state the main result of this section which (together with Theorem 4.19) completes the proof of the first part of Theorem 6.3:

Proposition 6.15. *The class of all semi-linear orders* Γ *enjoys the property* $\mathsf{EHD}(\mathsf{WMSO})$.

Proof. Take $\mathcal{A} = (A, <, \perp)$. Thanks to Lemmas 6.8, 6.10 and 6.13, it is enough to show that WMSO can express the condition that every finite and non-empty connected substructure of $\mathcal{A}$ has a central point. Using the the WMSO-formula $\mathsf{reach}_\varphi^X(x, y)$, as defined in Example 2.20, where $\varphi(z, w) := z < w \vee w < z$, we can see that $\mathcal{A} \models \mathsf{reach}_\varphi^B(a, b)$ if and only if a and b are in the same connected component of $\mathcal{A}_{\restriction B}$. Then, we define the following WMSO-formulas:

$$\begin{aligned} \mathsf{connected}(X) &:= \forall x \in X\ \forall y \in X\ \mathsf{reach}_\varphi^X(x, y)\ , \\ \mathsf{central}(x, X) &:= x \in X \wedge \forall y \in X\ \neg(x \perp y \vee y \perp x \vee y < x)\ , \text{ and} \\ \psi &:= \forall X\,(\mathsf{connected}(X) \wedge X \neq \emptyset \rightarrow \exists x\,\mathsf{central}(x, X))\ . \end{aligned}$$

It is straightforward to verify that $\mathcal{A} \models \psi$ if and only if every finite non-empty connected subset of A has a central point. ☐

6.3 The EHD-Property for Ordinal Trees

Let Ω denote the class of all ordinal trees (over the signature $\{<, \perp\}$). The aim of this section is to prove that Ω has the property $\mathsf{EHD}(\mathsf{MSO})$. We use again the notions of a connected subset and a central point as introduced in Definition 6.11 to characterize those structures which admit a homomorphism into an ordinal tree. Here, in contrast with the case of semi-linear orders, the final condition will be that all connected sets (not just the finite ones) have a central point.

Lemma 6.16. *Let* $\mathcal{A} = (A, <, \perp)$ *be a structure. There exists* $\mathcal{O} \in \Omega$ *such that* $\mathcal{A} \preceq \mathcal{O}$ *if and only if every non-empty and connected* $B \subseteq A$ *has a central point.*

Proof. We start with the direction ($\Rightarrow$). Due to Lemma 6.8 we can assume that there is a relation $\lhd$ that extends $(A, <, \perp)$ to an ordinal forest. Let $B \subseteq A$ be a non-empty connected set. Since $(A, \lhd, \perp_\lhd)$ is an ordinal forest, B has a minimal element c with respect to $\lhd$. By Lemma 6.14, c is a central point of B.

For the direction ($\Leftarrow$) we first define a partition of the domain of $\mathcal{A}$ into subsets C_β for $\beta \sqsubset \chi$, where χ is an ordinal (whose cardinality is bounded by the cardinality of A). Here $\sqsubset$ denotes the natural order on ordinals. Assume that the pairwise disjoint subsets C_β have been defined for all $\beta \sqsubset \alpha$ (which is true for $\alpha = 0$ in the beginning). Then we define C_α as follows. Let $C_{\sqsubset\alpha} = \bigcup_{\beta \sqsubset \alpha} C_\beta \subseteq A$.

If $A \setminus C_{\sqsubset\alpha}$ is not empty, then we define $\mathcal{C}_\alpha$ as the set of connected components of $A \setminus C_{\sqsubset\alpha}$. Let

$$C_\alpha = \{c \in A \setminus C_{\sqsubset\alpha} \mid c \text{ is a central point of some } B \in \mathcal{C}_\alpha\}.$$

Clearly, C_α is not empty. Hence, there must exist a smallest ordinal χ such that $A = C_{\sqsubset\chi}$.

For every ordinal $\alpha \sqsubset \chi$ and each element $c \in C_\alpha$ we define the sequence of connected components $\mathsf{road}(c) = (B_\beta)_{(\beta \sqsubseteq \alpha)}$, where $B_\beta \in \mathcal{C}_\beta$ is the unique connected component with $c \in B_\beta$. This ordinal-indexed sequence keeps record of the *road* we took to reach c by storing information about the connected components to which c belongs at each stage of our process.

Given $\mathsf{road}(c) = (B_\beta)_{(\beta \sqsubseteq \alpha)}$ and $\mathsf{road}(c') = (B'_\beta)_{(\beta \sqsubseteq \alpha')}$ for some $c \in C_\alpha$ and $c' \in C_{\alpha'}$, let us define $\mathsf{road}(c) \lhd \mathsf{road}(c')$ if and only if $\alpha \sqsubset \alpha'$ and $B_\beta = B'_\beta$ for all $\beta \sqsubseteq \alpha$. Basically this is the *prefix order* for ordinal-sized sequences of connected components.

Now let $O = \{\mathsf{road}(c) \mid c \in A\}$. Note that $\mathcal{O} = (O, \lhd, \perp_\lhd)$ is an ordinal forest, because for each $c \in C_\alpha$ the order $(\{\mathsf{road}(c') \mid \mathsf{road}(c') \unlhd \mathsf{road}(c)\}, \unlhd)$ forms the ordinal α (for each $\beta \sqsubset \alpha$ it contains exactly one road of length β).

Now we show that the mapping h with $h(c) = \mathsf{road}(c)$ is a homomorphism from $\mathcal{A}$ to $\mathcal{O}$. Take elements $a, a' \in A$ with $a \in C_\alpha$, and $a' \in C_{\alpha'}$ for some $\alpha, \alpha' \sqsubset \chi$. Let $\mathsf{road}(a) = (B_\beta)_{(\beta \sqsubseteq \alpha)}$ and $\mathsf{road}(a') = (B'_\beta)_{(\beta \sqsubseteq \alpha')}$.

- If $a < a'$, then (i) $\alpha \sqsubset \alpha'$, because a' cannot be central point of a set which contains a, and (ii) $B_\beta = B'_\beta$ for all $\beta \sqsubseteq \alpha$ because a and a' belong to the same connected component of $A \setminus C_{\sqsubset\beta}$ for all $\beta \sqsubseteq \alpha$. By these observations we deduce that $\mathsf{road}(a) \lhd \mathsf{road}(a')$.

- If $a \perp a'$, then, without loss of generality, suppose that $\alpha \sqsubseteq \alpha'$. At stage α, a is a central point of $B_\alpha \in \mathcal{C}_\alpha$. Since $\alpha \sqsubseteq \alpha'$, the connected component B'_α exists. We must have $B_\alpha \neq B'_\alpha$, since otherwise we would have $a \perp a' \in B_\alpha$ contradicting the fact that a is central for B_α. Therefore, $\mathsf{road}(a) \perp_\lhd \mathsf{road}(a')$.

We finally add one extra element road_0 and make this the minimal element of $\mathcal{O}$, thus finding a homomorphism from $\mathcal{A}$ into an ordinal tree. $\square$

We can now complete the proof of the second part of Theorem 6.3

Proposition 6.17. *The class Ω of all ordinal trees has the* EHD*-property.*

Proof. Given a $\{<, \perp\}$-structure $\mathcal{A}$, it suffices by Lemma 6.16 to find an MSO-formula expressing the fact that every non-empty connected subset of $\mathcal{A}$ has a central point. Recall the WMSO-formula ψ from Theorem 6.15. Seen as an MSO-formula, ψ clearly does the job. $\square$

Remark 6.18. The procedure described in the proof of Lemma 6.16 can be also used to embed a structure $\mathcal{A} = (A, <, \perp)$ into an ordinary tree. Note that, in fact, a tree in the classical sense is just an ordinal tree where, for every x, the set of all elements smaller than x forms a finite linear order. To satisfy this extra condition, the ordinal χ has to satisfy $\chi \leq \omega$, i.e., every element $a \in A$ has to belong to a set C_n for some finite n. We use this observation in Section 6.5. Unfortunately, our results from Section 6.5 imply that such condition cannot be expressed in $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$.

6.4 The EHD-Property for Trees of Fixed Height

Fix $h \in \mathbb{N}$. The aim of this section is to show that the class Θ_h of all trees of height h (over $\{<, \perp\}$) has the property $\mathsf{EHD}(\mathsf{MSO})$. The proof relies on the fact that we can unfold the fix-point procedure on the central points from the ordinal tree setting for h steps in MSO.

For this section, we fix an arbitrary structure $\mathcal{A} = (A, <, \perp)$. We first define subsets $A_0, A_1, \ldots, A_h \subseteq A$ that are pairwise disjoint. The elements of A_0 are the central points of A (this set is possibly empty) and, for each $i \geq 1$, A_i contains the central points of each connected component of $A \setminus (A_0 \cup \cdots \cup A_{i-1})$. Note that A_0 contains exactly those nodes of $\mathcal{A}$ that a homomorphism from $\mathcal{A}$ to some tree can map to the root of the tree because elements from A_0 are neither incomparable to any other element nor below any other element, while all element outside of A_0 have to be incomparable to some other element or have to be below some other element. Hence they cannot be mapped to the root by any homomorphism. Thus, there is a homomorphism from $\mathcal{A}$ to some element of Θ_h if and only if $\mathcal{A} \setminus A_0$ can be embedded into some forest of height $h-1$. Now the sets A_i for $1 \leq i \leq h$ collect exactly those elements which are chosen in the i-th step of the fix-point procedure from the proof of Lemma 6.16 (where this set is called C_i). Thus, if $A_0, A_1, \ldots, A_h$ form a partition of A, then $\mathcal{A}$ allows a homomorphism to some $\mathcal{T} \in \Theta_h$. It turns out that the converse is also true. If $\mathcal{A} \preceq \mathcal{T}$ for some $\mathcal{T} \in \Theta_h$ then $A_0, A_1, \ldots, A_h$ form a partition of A. Thus, it suffices to show that each A_i is MSO-definable. To do this, we define for all $i \in \mathbb{N}$ the formulas

$$\begin{aligned}
\psi_0(x) &:= \forall y\, \neg(y < x \vee y \perp x \vee x \perp y) \,, \\
\mathsf{con}_{i+1}(x, y) &:= \exists Z\, \forall z\, (z \in Z \to \bigwedge_{j=0}^{i} \neg\psi_j(z)) \wedge \mathsf{reach}_{\varphi}^{Z}(x, y) \,, \\
\psi_{i+1}(x) &:= \forall y\, (\mathsf{con}_{i+1}(x, y) \to \neg(y < x \vee y \perp x \vee x \perp y)) \,,
\end{aligned}$$

where $\mathsf{reach}_{\varphi}^{X}(x, y)$ is as defined in Example 2.20, with $\varphi(z, w) := z < w \vee w < z$.

Let A_0 be the set of nodes $a \in A$ such that $\mathcal{A} \models \psi_0(a)$ and let A_{i+1} be the set of nodes $a \in A$ such that $\mathcal{A} \models \psi_{i+1}(a)$.

Clearly A_0 is the set of central points of A. Inductively, one shows that $\mathsf{con}_{i+1}(x, y)$ expresses the fact that x and y are connected through a path that does not intersect any A_j for $j \leq i$, and that A_{i+1} is the set of central points of the connected components of $A \setminus (A_0 \cup \cdots \cup A_i)$.

Lemma 6.19. *There exists $\mathcal{T} \in \Theta_h$ such that $\mathcal{A} \preceq \mathcal{T}$ if and only if $A_0, \ldots, A_h$ is a partition of A.*

Proof. For the direction $(\Rightarrow)$ take a homomorphism g from $\mathcal{A}$ to a tree $\mathcal{T} = (T, \lhd, \perp_\lhd) \in \Theta_h$. By induction we prove that if g maps a to the i-th level of $\mathcal{T}$ then $a \in A_j$ for some $j \leq i$. For $i = 0$ assume that $g(a)$ is the root of the tree. Then a cannot be incomparable or greater than any other element. Thus, it is a central point of A, i.e., $a \in A_0$.

For the inductive step, assume that $g(a)$ is on the i-th level for $i > 0$. Heading for a contradiction, assume that a is neither in $A_0 \cup \cdots \cup A_{i-1}$ nor a central point of some connected component of $A \setminus (A_0 \cup \cdots \cup A_{i-1})$. Then there is some $a' \in A \setminus (A_0 \cup \cdots \cup A_{i-1})$ such that a and a' are in the same connected component of $A \setminus (A_0 \cup \cdots \cup A_{i-1})$ and one of $a' < a$, $a' \perp a$ or $a \perp a'$ holds. Since g is a homomorphism, we get $g(a') \lhd g(a)$ or $g(a') \perp_\lhd g(a)$. If $g(a') \lhd g(a)$, then a' has to be mapped by g to some level $j < i$, whence $a' \in A_0 \cup \cdots \cup A_j$ by the inductive hypothesis. This contradicts our assumption on a'. Now, assume that $g(a') \perp_\lhd g(a)$. Let $a = a_0, a_1, \ldots, a_m = a'$ be a path connecting a and a' in $A \setminus (A_0 \cup \cdots \cup A_{i-1})$. Since $a_i \notin A_0 \cup \cdots \cup A_{i-1}$, the inductive hypothesis shows that all $g(a_i)$ are on level i or larger. But then, since $a_0, a_1, \ldots, a_m$ is a path, all $g(a_i)$ must belong to the subtree rooted at $g(a)$. This leads to the contradictions that $g(a_m) = g(a')$ is in the subtree rooted at $g(a)$ and hence is not incomparable to $g(a)$. Thus, we can conclude that $a \in A_0 \cup \cdots \cup A_i$.

For the direction $(\Leftarrow)$ assume that $A_0 \cup \cdots \cup A_h = A$. Applying the same construction described in the proof of Lemma 6.16 for ordinal trees, it is not hard to see that we find a homomorphism g from $\mathcal{A}$ to some tree of height h which maps the elements of A_i to elements on level i. Should A_0 be empty, then A would not be connected, and we would have a forest of height $h - 1$. Adding a minimal element we still get a tree of height h. $\square$

Theorem 6.20. *Θ_h has the* EHD*-property.*

Proof. Let $\mathcal{A}$ be any $\{<, \perp\}$-structure. Then, by Lemma 6.19, $\mathcal{A} \preceq \mathcal{T}$ for some $\mathcal{T} \in \Theta_h$ if and only if

$$\mathcal{A} \models \forall x \bigvee_{i=0}^{h} \psi_i(x).$$

□

6.5 Trees do not have the EHD-Property

Let Θ be the class of all countable trees (over $\{<, \perp\}$). In this section, we prove that the logic $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ (the most expressive logic for which the EHD-technique currently works) cannot distinguish between $\{<, \perp\}$-structures that admit a homomorphism to some element of Θ and those that do not. Heading for a contradiction, assume that φ is a sentence such that a countable structure $\mathcal{A} = (A, <, \perp)$ satisfies φ if and only if there is a homomorphism from $\mathcal{A}$ to some $\mathcal{T} \in \Theta$. Let k be the quantifier rank of φ. We construct two graphs $\mathcal{E}_k$ and $\mathcal{U}_k$ such that $\mathcal{E}_k$ admits a homomorphism into a tree while $\mathcal{U}_k$ does not. We then use an Ehrenfeucht-Fraïssé game for $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ to show that φ cannot separate these two structures, contradicting our assumption. This contradiction shows that Θ does not have EHD, proving Theorem 6.1.

6.5.1 The WMSO+B-Ehrenfeucht-Fraïssé-Game

The k-round $\mathsf{WMSO}{+}\mathsf{B}$-EF-game on a pair of structures $(\mathcal{A}, \mathcal{B})$ over the same finite relational signature σ is played by Spoiler and Duplicator as follows.[2] In the following, A denotes the domain of $\mathcal{A}$ and B the domain of $\mathcal{B}$.

The game starts in position

$$p_0 := (\mathcal{A}, \emptyset, \emptyset, \mathcal{B}, \emptyset, \emptyset).$$

In general, before playing the i-th round (for $1 \leq i \leq k$) the game is in a position

$$p = (\mathcal{A}, a_1, \dots, a_{i_1}, A_1, \dots, A_{i_2}, \mathcal{B}, b_1, \dots, b_{i_1}, B_1, \dots, B_{i_2}),$$

where

1. $i_1, i_2 \in \mathbb{N}$ satisfy $i_1 + i_2 = i - 1$,
2. $a_j \in A$ for all $1 \leq j \leq i_1$,
3. $b_j \in B$ for all $1 \leq j \leq i_1$,
4. $A_j \subseteq A$ is a finite set for all $1 \leq j \leq i_2$, and
5. $B_j \subseteq B$ is a finite set for all $1 \leq j \leq i_2$.

In the i-th round Spoiler and Duplicator produce the next position as follows. Spoiler chooses to play one of the following three possibilities:

[2] For the ease of presentation we assume that $\mathcal{A}$ and $\mathcal{B}$ are infinite structures.

1. Spoiler can play an *element move*. For this he chooses either some $a_{i_1+1} \in A$ or $b_{i_1+1} \in B$. Duplicator then responds with an element from the other structure, i.e., with $b_{i_1+1} \in B$ or $a_{i_1+1} \in A$. The position in the next round is

$$(\mathcal{A}, a_1, \dots, a_{i_1}, a_{i_1+1}, A_1, \dots, A_{i_2}, \mathcal{B}, b_1, \dots, b_{i_1}, b_{i_1+1}, B_1, \dots, B_{i_2}).$$

2. Spoiler can play a *set move*. For this he chooses either some finite $A_{i_2+1} \subseteq A$ or some finite $B_{i_2+1} \subseteq B$. Duplicator then responds with a finite set from the other structure, i.e., with $B_{i_2+1} \subseteq B$ or $A_{i_2+1} \subseteq A$. The position in the next round is

$$(\mathcal{A}, a_1, \dots, a_{i_1}, A_1, \dots, A_{i_2}, A_{i_2+1}, \mathcal{B}, b_1, \dots, b_{i_1}, B_1, \dots, b_{i_2}, B_{i_2+1}).$$

3. Spoiler can play a *bound move*. For this he chooses one of the structures $\mathcal{A}$ or $\mathcal{B}$ and chooses a natural number $l \in \mathbb{N}$. Duplicator responds with another number $m \in \mathbb{N}$. Then the rest of the round is played as in the case of a set move with the restrictions that Spoiler has to choose a subset of size at least m from his chosen structure and Duplicator has to respond with a set of size at least l.

After k rounds, the game ends in a position

$$p = (\mathcal{A}, a_1, \dots, a_{i_1}, A_1, \dots, A_{i_2}, \mathcal{B}, b_1, \dots, b_{i_1}, B_1, \dots, B_{i_2}),$$

where $i_1 + i_2 = k$. Duplicator wins the game if

1. $a_j \in A_t \Leftrightarrow b_j \in B_t$ for all $1 \leq j \leq i_1$ and all $1 \leq t \leq i_2$,
2. $a_j = a_t \Leftrightarrow b_j = b_t$ for all $1 \leq j < t \leq i_1$, and
3. for all relation symbols $R \in \sigma$ (of arity n) $(a_{j_1}, a_{j_2}, \dots, a_{j_n}) \in R^{\mathcal{A}} \Leftrightarrow (b_{j_1}, b_{j_2}, \dots, b_{j_n}) \in R^{\mathcal{B}}$ for all $j_1, j_2, \dots, j_n \in \{1, \dots, i_1\}$.

As one would expect, the $\mathsf{WMSO{+}B}$-EF-game can be used to show undefinability results for $\mathsf{WMSO{+}B}$ due to the relationship between winning strategies in the k-round game and equivalence with respect to formulas up to quantifier rank k.

Proposition 6.21. *For given σ-structures $\mathcal{A}$ and $\mathcal{B}$, elements $a_1, \dots, a_{i_1} \in \mathcal{A}$, $b_1, \dots, b_{i_1} \in \mathcal{B}$ and finite sets $A_1, \dots, A_{i_2} \subseteq \mathcal{A}$, $B_1, \dots, B_{i_2} \subseteq \mathcal{B}$, define the position*

$$p = (\mathcal{A}, a_1, \dots, a_{i_1}, A_1, \dots, A_{i_2}, \mathcal{B}, b_1, \dots, b_{i_1}, B_1, \dots, B_{i_2}) \ .$$

Then, $(\mathcal{A}, a_1, \dots, a_{i_1}, A_1, \dots, A_{i_2})$ and $(\mathcal{B}, b_1, \dots, b_{i_1}, B_1, \dots, B_{i_2})$ are indistinguishable by any $\mathsf{WMSO{+}B}$*-formula $\varphi(x_1, \dots, x_{i_1}, X_1, \dots, X_{i_2})$ of quantifier rank k if and only if Duplicator has a winning strategy in the k-round* $\mathsf{WMSO{+}B}$*-EF-game started in p.*

Proof. First of all note that, since we are considering a finite relational signature, up to logical equivalence there are only finitely many different $\mathsf{WMSO{+}B}$-formulas $\varphi(x_1,\dots,x_{i_1},X_1,\dots,X_{i_2})$ of quantifier rank k. This fact is proved in a completely analogous way to the case of first-order or monadic second-order logic.

The proof is by induction on k. The base case $k=0$ is trivial. Assume now that the proposition holds for $k-1$. We use the abbreviations $\bar{a}=(a_1,\dots,a_{i_1})$, $\bar{A}=(A_1,\dots,A_{i_2})$, $\bar{b}=(b_1,\dots,b_{i_1})$, and $\bar{B}=(B_1,\dots,B_{i_2})$ in the following. First assume that there is a $\mathsf{WMSO{+}B}$-formula $\varphi(x_1,\dots,x_{i_1},X_1,\dots,X_{i_2})$ of quantifier rank k such that

$$\mathcal{A}\models\varphi(\bar{a},\bar{A}) \tag{6.2}$$

and

$$\mathcal{B}\not\models\varphi(\bar{b},\bar{B})\ . \tag{6.3}$$

We show that Spoiler has a winning strategy in the k-round game by a case distinction on the structure of φ. We only consider the case $\varphi=\mathsf{B}X\,\psi$ (all other cases can be handled exactly as in the WMSO-EF-game, see e.g. [23]). Let $l\in\mathbb{N}$ be a strict bound witnessing (6.2), in the sense that there is no set A_{i_2+1} of size at least l such that $\mathcal{A}\models\varphi(\bar{a},\bar{A},A_{i_2+1})$. Then Spoiler chooses structure $\mathcal{B}$ and bound l. Duplicator responds with some bound $m\in\mathbb{N}$. Due to (6.3)

$$\mathcal{B}\models\neg\mathsf{B}X\,\psi(\bar{b},\bar{B},X)\ .$$

Hence, there is a set B_{i_2+1} of size at least m such that

$$\mathcal{B}\models\psi(\bar{b},\bar{B},B_{i_2+1})\ .$$

Spoiler chooses this set B_{i_2+1}. Duplicator must answer with a set A_{i_2+1} of size at least l. By the choice of l we conclude that

$$\mathcal{A}\not\models\psi(\bar{a},\bar{A},A_{i_2+1})\ .$$

By the inductive hypothesis, Spoiler has a winning strategy in the resulting position for the $(k-1)$-round game.

For the other direction, assume that $(\mathcal{A},\bar{a},\bar{A})$ and $(\mathcal{B},\bar{b},\bar{B})$ are indistinguishable by $\mathsf{WMSO{+}B}$-formulas of quantifier rank k. Duplicator's strategy is as follows.

- If Spoiler plays an element move choosing without loss of generality $a_{i_1+1}\in\mathcal{A}$, let Φ be the set of all $\mathsf{WMSO{+}B}$-formulas φ of quantifier rank $k-1$ such that $\mathcal{A}\models\varphi(\bar{a},a_{i_1+1},\bar{A})$. Since Φ is finite up to logical equivalence, there is a $\mathsf{WMSO{+}B}$-formula ψ of quantifier rank $k-1$ such that $\psi\equiv\bigwedge_{\varphi\in\Phi}\varphi$. By the assumption (indistinguishableness up to quantifier rank k) and the fact that $\mathcal{A}\models\exists x\,\psi(\bar{a},x,\bar{A})$ we conclude that $\mathcal{B}\models\exists x\,\psi(\bar{b},x,\bar{B})$. Hence, there is an element $b_{i_1+1} \in \mathcal{B}$ such that $\mathcal{B} \models \psi(\bar{b}, b_{i_1+1}, \bar{B})$. Thus, Duplicator can respond with b_{i_1+1} and obtain a position for which he has a winning strategy by the induction hypothesis.

- If Spoiler plays a set move, we use the same strategy as in the element move. We only have to replace the element a_{i_1+1} by Spoiler's set A_{i_1+1} and the first-order quantifier by a set quantifier.

- Assume that Spoiler plays a bound move, choosing $\mathcal{B}$ and bound $l \in \mathbb{N}$. Let

$$\Phi_A = \left\{ \varphi \;\middle|\; \mathrm{rank}(\varphi) = k-1, \forall M \subseteq \mathcal{A} \left(|M| \geq l \Rightarrow \mathcal{A} \not\models \varphi(\bar{a}, \bar{A}, M)\right) \right\}.$$

Note that $\mathcal{A} \models \mathsf{B}X\, \varphi(\bar{a}, \bar{A}, X)$ for all $\varphi \in \Phi_A$. Thus, $\mathcal{B} \models \mathsf{B}X\, \varphi(\bar{b}, \bar{B}, X)$ for all $\varphi \in \Phi_A$. Since Φ_A is finite up to equivalence we can fix a number $m \in \mathbb{N}$ that serves as a bound in $(\mathcal{B}, \bar{b}, \bar{B})$ for all $\varphi \in \Phi_A$. Thus, for the set

$$\Phi_B = \left\{ \varphi \;\middle|\; \mathrm{rank}(\varphi) = k-1, \forall M \subseteq \mathcal{B} \left(|M| \geq m \Rightarrow \mathcal{B} \not\models \varphi(\bar{b}, \bar{B}, M)\right) \right\}$$

we have $\Phi_A \subseteq \Phi_B$. Duplicator answers Spoiler's challenge with this number m. Then Spoiler has to choose a set $B_{i_2+1} \subseteq \mathcal{B}$ of size at least m. Let

$$\Psi_B = \{\varphi \mid \mathrm{rank}(\varphi) = k-1, \mathcal{B} \models \varphi(\bar{b}, \bar{B}, B_{i_2+1})\}\ .$$

Note that $\Phi_B \cap \Psi_B = \emptyset$. Since Ψ_B is finite up to equivalence, there is a WMSO+B-formula $\psi \in \Psi_B$ of quantifier rank $k-1$ such that $\psi \equiv \bigwedge_{\varphi \in \Psi_B} \varphi$. In particular, $\psi \notin \Phi_B$. Hence, $\psi \notin \Phi_A$ (since $\Phi_A \subseteq \Phi_B$). By the definition of Φ_A this means that there is a subset $A_{i_2+1} \subseteq \mathcal{A}$ such that $|A_{i_2+1}| \geq l$ and $\mathcal{A} \models \psi(\bar{a}, \bar{A}, A_{i_2+1})$. Duplicator chooses this set A_{i_2+1}. The resulting position allows a winning strategy for Duplicator by the induction hypothesis.

$\square$

6.5.2 Two Structures that WMSO+B cannot Distinguish

In this section we define a class of finite structures $\mathcal{G}_{n,m}$ for $n, m \in \mathbb{N}$. Using these structures, we define for every $k \geq 0$ structures $\mathcal{E}_k$ and $\mathcal{U}_k$. We show that for every $k \geq 0$, $\mathcal{E}_k$ can be mapped homomorphically into a tree, whereas $\mathcal{U}_k$ cannot. In the next section, we will show that Duplicator wins the k-round EF-game for both WMSO+B and MSO.

The *standard plain triple-u* is the structure $\mathcal{P} = (P, <^P, \perp^P)$, where

$$\begin{aligned} P &= \{l, r, a_1, a_2, b_1, b_2, b_3\}, \\ <^P &= \{(l, b_1), (a_1, b_1), (a_1, b_2), (a_2, b_2), (a_2, b_3), (r, b_3)\}, \text{ and} \\ \perp^P &= \{(l, r), (r, l)\}. \end{aligned}$$

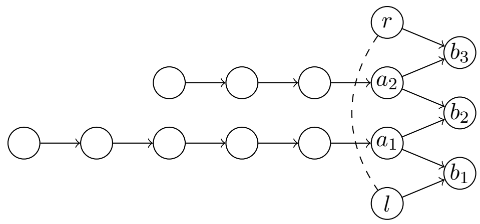

Figure 6.2: The standard $(5, 3)$-triple-u, where we only draw the Hasse diagram for $<^D$, and where dashed edges are $\perp$-edges.

We call a structure $(V, <, \perp)$ a *plain triple-u* if it is isomorphic to the standard plain triple-u. For $n, m \in \mathbb{N}$, the *standard* (n, m)*-triple-u* is the structure $\mathcal{G}_{n,m} = (D, <^D, \perp^D)$, where

$$D = \{l, r, a_1, a_2, b_1, b_2, b_3\} \cup (\{1, 2, \ldots, n\} \times \{a_1\}) \cup (\{1, 2, \ldots, m\} \times \{a_2\}),$$

and $<^D, \perp^D$ are the minimal relations such that

- $\mathcal{G}_{n,m}$ restricted to $\{l, r, a_1, a_2, b_1, b_2, b_3\}$ is the standard plain triple-u,
- $(a_1, 1) < (a_1, 2) < \cdots < (a_1, n) < a_1$, and
- $(a_2, 1) < (a_2, 2) < \cdots < (a_2, m) < a_2$.

We call a graph $(V, <, \perp)$ an (n, m)*-triple-u* if it is isomorphic to the standard (n, m)-triple-u. Figure 6.2 depicts a $(5, 3)$-triple-u.

Remark 6.22. For all $n, m \in \mathbb{N}$ and each (n, m)-triple-u $\mathcal{W}$ we fix an isomorphism $\psi_{\mathcal{W}}$ between $\mathcal{W}$ and the standard (n, m)-triple-u. Note that this isomorphism is unique if $n \neq m$. If $n = m$, then there is an automorphism of $\mathcal{G}_{n,n}$ exchanging the nodes l and r. Thus, choosing an isomorphism means to choose the left node of the triple-u. For $x \in \{l, r, a_1, a_2, b_1, b_2, b_3\}$ we denote by $\mathcal{W}.x$ the unique node $w \in \mathcal{W}$ such that $\psi_{\mathcal{W}}(w) = x$. Furthermore, we call the linear order of size n (resp., m) that consists of all proper ancestors of $\psi_{\mathcal{W}}^{-1}(a_1)$ (resp., $\psi_{\mathcal{W}}^{-1}(a_2)$) the *left order* (resp., *right order*) of $\mathcal{W}$.

Let $k \in \mathbb{N}$ be a natural number. Fix a strictly increasing sequence $(n_{k,i})_{i \in \mathbb{N}}$ such that the linear order of length $n_{k,i}$ and the linear order of length $n_{k,j}$ are equivalent with respect to WMSO+B-formulas of quantifier rank up to k for all $i, j \in \mathbb{N}$. Such a sequence exists because there are (up to equivalence) only finitely many WMSO+B-formulas of quantifier rank k. Since the linear orders of length $n_{k,i}$ are finite, they are equivalent with respect to both MSO-formulas and WMSO-formulas of quantifier rank up to k.

Definition 6.23 (The embeddable triple-u)**.** Let $\mathcal{E}_k$ be the structure that consists of

1. the disjoint union of infinitely many $(n_{k,1}, n_{k,j})$-triple-u's and infinitely many $(n_{k,j}, n_{k,1})$-triple-u's for all $j \geq 2$,

2. one additional node d, and

3. for each triple-u $\mathcal{W}$ an $<$-edge from $\mathcal{W}.l$ to d.

In the following we call d the *final node* of $\mathcal{E}_k$

Lemma 6.24. *For all $k \in \mathbb{N}$, $\mathcal{E}_k$ admits a homomorphism to a tree.*

Proof. Using the procedure on the central points from the ordinal tree setting described in the proof of Lemma 6.16, we first start adding the chains of each triple-u to the tree. In step $n_{k,1}$ we finally have placed all the chains of length $n_{k,1}$. Thus, for each triple-u $\mathcal{W}$ either $\mathcal{W}.a_1$ or $\mathcal{W}.a_2$ becomes central. Thus, in step $n_{k,1} + 1$ all the triple-u's split into two disconnected components and the incomparability edges, which were avoiding that $\mathcal{W}.l$ became central, now cease having such an effect. We can therefore map $\mathcal{W}.l$ at stage $n_{k,1} + 2$ and the final node d in step $n_{k,1} + 3$. Thus, it is easy to prove that the fix-point procedure from the proof of Lemma 6.16 terminates at stage ω. Whenever this happens, the given structure admits a homomorphism to a tree, see Remark 6.18. $\square$

Definition 6.25 (The unembeddable triple-u)**.** Let $\mathcal{U}_k$ be the structure that consists of

1. the disjoint union of infinitely many $(n_{k,j}, n_{k,j})$-triple-u's for all $j \geq 2$,

2. one additional node d, and

3. for each triple-u $\mathcal{W}$ an $<$-edge from $W.l$ to d.

In the following we call d the *final node* of $\mathcal{U}_k$

Lemma 6.26. *For all $k \in \mathbb{N}$, $\mathcal{U}_k$ does not admit a homomorphism to a tree.*

Proof. Again, we consider the fix-point procedure from the proof of Lemma 6.16. Assume that $\mathcal{U}_k$ admits a homomorphism to a tree. Then, the final node d has to be placed at some stage i into the tree, i.e., in the notation of the proof of Lemma 6.16, d belongs to some set C_i for $i < \omega$. But there is a $(n_{k,i}, n_{k,i})$-triple-u $\mathcal{W}$ and $\mathcal{W}.l < d$. Hence, $\mathcal{W}.l$ has to be placed into the tree in one of the first $i-1$ stages. But $\mathcal{W}.a_1$ and $\mathcal{W}.a_2$ are the target nodes of chains of length $n_{k,i} \geq i$. Hence, after i stages they are still not mapped into the tree. Therefore, after i

stages, $\mathcal{W}.l$ and $\mathcal{W}.r$ are in the same connected component and they are linked by an $\perp$-edge. This contradicts the fact that $\mathcal{W}.l$ was placed into the tree in one of the first $i-1$ stages.

$\square$

6.5.3 Duplicators Strategies in the k-Round Game

We show that Θ does not have the EHD-property by showing that Duplicator has a winning strategy for the k-round MSO-EF-game and $\mathsf{WMSO{+}B}$-EF-game on the pair $(\mathcal{E}_k, \mathcal{U}_k)$ for each $k \in \mathbb{N}$. Hence, the two structures are not distinguishable by $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$-formulas of quantifier rank k.

For MSO this is rather simple. Since the linear orders of length $n_{k,i}$ and $n_{k,j}$ are indistinguishable up to quantifier rank k, it is straightforward to compile the strategies on these pairs of paths into a strategy on the whole structures for the k-round game. It is basically the same proof as the one showing that a strategy on a pair $(\biguplus_{i\in I}\mathcal{A}_i, \biguplus_{i\in I}\mathcal{B}_i)$ of disjoint unions can be compiled from strategies on the pairs $(\mathcal{A}_i, \mathcal{B}_i)$. In our situation there is an $i \in I$ such that $\mathcal{A}_i = \mathcal{B}_i$ consists of infinitely many plain triple-u's together with the final node, and the other pairs $(\mathcal{A}_j, \mathcal{B}_j)$ for $j \in I \setminus \{i\}$ consist of two linear orders that are indistinguishable by MSO-formulas of quantifier rank k.

Compiling local strategies to a global strategy in the $\mathsf{WMSO{+}B}$-EF-game is much more difficult because strategies are not closed under infinite disjoint unions. For instance, let $\mathcal{A}$ be the disjoint union of infinitely many copies of the linear order of size $n_{k,1}$ and $\mathcal{B}$ be the disjoint union of all linear orders of size $n_{k,j}$ for all $j \in \mathbb{N}$. Clearly, Duplicator has a winning strategy in the k-round game starting on the pair that consists of the linear order of size $n_{k,1}$ and the linear order of size $n_{k,j}$. But in $\mathcal{A}$ every linear suborder has size bounded by $n_{k,1}$, while $\mathcal{B}$ has linear suborders of arbitrary finite size. This difference is of course expressible in $\mathsf{WMSO{+}B}$.

Even though strategies in $\mathsf{WMSO{+}B}$-games are not closed under disjoint unions, we can obtain a composition result for disjoint unions on certain restricted structures as follows. Let $\mathcal{A} = \biguplus_{i\in\mathbb{N}}\mathcal{A}_i$ and $\mathcal{B} = \biguplus_{i\in\mathbb{N}}\mathcal{B}_i$ be disjoint unions of structures $\mathcal{A}_i$ and $\mathcal{B}_i$ satisfying the following conditions:

1. All $\mathcal{A}_i$ and $\mathcal{B}_i$ are finite structures.

2. For every $i \in \mathbb{N}$, Duplicator has a winning strategy in the k-round MSO-EF-game on $\mathcal{A}_i$ and $\mathcal{B}_i$.

3. There is a constant $c \in \mathbb{N}\setminus\{0\}$ such that whenever Spoiler starts the MSO-EF-game on $(\mathcal{A}_i, \mathcal{B}_i)$ with a set move choosing a set of size n in $\mathcal{A}_i$ or $\mathcal{B}_i$, then Duplicator's strategy answers with a set of size at least $\frac{n}{c}$.

In this case Duplicator has a winning strategy in the k-round $\mathsf{WMSO{+}B}$-EF-game on $\mathcal{A}$ and $\mathcal{B}$. To substantiate this claim, we sketch his strategy. For an element or set move, Duplicator just uses the local strategies from the MSO-game to give an answer to any challenge. For a bound move, Duplicator does the following. If Spoiler's chooses the bound $l \in \mathbb{N}$, then Duplicator chooses the number m, which is the total number of elements in all substructures $\mathcal{A}_i$ or $\mathcal{B}_i$ in which some elements have been chosen in one of the previous rounds plus $c \cdot l$. This forces Spoiler to choose $c \cdot l$ elements in fresh substructures. Then Duplicator uses his strategy in each local pair of structures to give an answer to Spoiler's challenge. Since Spoiler choses $c \cdot l$ elements in fresh substructures, Duplicator answers with at least $\frac{c \cdot l}{c} = l$ many elements in fresh substructures. This is a valid move and it preserves the existence of local winning strategies between each pair $(\mathcal{A}_i, \mathcal{B}_i)$ for the rounds yet to play.

From now on, we consider a fixed number $k \in \mathbb{N}$ and the game on the structures $\mathcal{E}_k$ and $\mathcal{U}_k$. We use a variant of the closure under restricted disjoint unions, sketched above, to provide a winning strategy for Duplicator. In order to reduce notational complexity we just write $\mathcal{E}$ for $\mathcal{E}_k$, $\mathcal{U}$ for $\mathcal{U}_k$ and n_i for $n_{k,i}$ (for all $i \in \mathbb{N}$). With $\bar{E}$ (resp. $\bar{U}$) we denote the set of all maximal subgraphs that are (n, m)-triple-u's occurring in $\mathcal{E}$ (resp., $\mathcal{U}$) where n and m range over $\mathbb{N}$. Note that $\mathcal{E}$ is the disjoint union of all $W \in \bar{E}$ together with the final node, and similarly of $\mathcal{U}$. Unfortunately, we cannot apply the result on restricted disjoint unions directly because of the following problems.

- Due to the final nodes of $\mathcal{E}$ and $\mathcal{U}$, the structures are not disjoint unions of triple-u's. But since the additional structure in both structures is added in a uniform way this does not pose a problem for the proof.

- The greater cause for trouble is that there is no constant c as in condition 3 that applies uniformly to all MSO-EF-games on an (n_j, n_1)-triple-u of $\mathcal{E}$ and an (n_j, n_j)-triple-u of $\mathcal{U}$ for all $j \in \mathbb{N}$. The problem is that if Spoiler chooses in his first move all elements of the right order of the (n_j, n_j)-triple-u, then the only possible answer of Duplicator is to choose the set of the n_1 many elements of the right order of the (n_j, n_1)-triple-u. But since the numbers n_j grow unboundedly, there is not constant c such that the inequation $n_1 \geq cn_j$ holds for all j. This problem does not exist for moves where Spoiler chooses many elements in the left order of the (n_j, n_j)-triple-u. Duplicator's strategy allows to exactly choose the same subset of the left order of the (n_j, n_1)-triple-u. This allows to overcome the problem that Duplicator should answer challenges where Spoiler chooses a large set with an equally large set (up to some constant factor): Instead of assigning each triple-u in $\bar{E}$ a fixed corresponding triple-u in $\bar{U}$, we do this dynamically.

If Spoiler chooses a lot of elements from the left order of a fresh (n_j, n_j)-triple-u, then Duplicator answers this challenge in a (n_j, n_1)-triple-u and we consider these two structures as forming one pair of the disjoint unions. On the other hand, if Spoiler chooses a lot of elements from the right order of a fresh (n_j, n_j)-triple-u, then Duplicators corresponding structure is chosen to be a fresh (n_1, n_j)-triple-u. In any case Duplicator's local winning strategy may copy most of Spoiler's choice (i.e., all elements chosen from the plain triple-u and from the order of length n_j from which Spoiler has chosen more elements), thus producing a set which is at least half as big as Spoiler's challenge.

In our prove we encode this dynamic choice of corresponding structures as a partial map $\varphi : \bar{E} \to \bar{U}$. The following definition of a locally-i-winning position describes the requirements on a position obtained after playing some rounds that allow to further use local winning strategies in order to compile a winning strategy for the next i-rounds. It basically requires that the map φ is such that for each triple-u $W \in \mathsf{dom}(\varphi)$ the restriction of the current position to W and $\varphi(W)$ is a valid position in the i-round $\mathsf{WMSO{+}B}$-EF-game on $(W, \varphi(W))$ which is winning for Duplicator and that $\mathsf{dom}(\varphi)$ and $\mathsf{im}(\varphi)$ covers all elements that have been chosen so far (in an element move or as a member of some set).

Definition 6.27. A position

$$p = (\mathcal{E}, e_1, \dots, e_{i_1}, E_1, \dots, E_{i_2}, \mathcal{U}, u_1, \dots, u_{i_1}, U_1, \dots, U_{i_2})$$

in the $\mathsf{WMSO{+}B}$-EF-game on $(\mathcal{E}, \mathcal{U})$ is called locally-i-winning (for Duplicator) if there is a partial bijection $\varphi : \bar{E} \to \bar{U}$ such that

- $\mathsf{dom}(\varphi)$ is finite,
- for all $W \in \bar{E}$, $W' \in \bar{U}$, and $1 \leq j \leq i_1$,
  1. if $e_j \in W$ then $W \in \mathsf{dom}(\varphi)$ and $u_j \in \varphi(W)$, and
  2. if $u_j \in W'$ then $W' \in \mathsf{im}(\varphi)$ and $e_j \in \varphi^{-1}(W')$,
- for all $W \in \bar{E}$, $W' \in \bar{U}$, and $1 \leq j \leq i_2$,
  1. if $E_j \cap W \neq \emptyset$ then $W \in \mathsf{dom}(\varphi)$ and
  2. if $U_j \cap W' \neq \emptyset$ then $W' \in \mathsf{im}(\varphi)$, and
- φ is compatible with local strategies in the following sense:
  1. For all $W \in \mathsf{dom}(\varphi)$, $x \in \{l, r, a_1, a_2, b_1, b_2, b_3\}$, $1 \leq j \leq i_1$ and $1 \leq k \leq i_2$ we have

- $e_j = W.x \Leftrightarrow u_j = \varphi(W).x$, and
- $W.x \in E_k \Leftrightarrow \varphi(W).x \in U_k$.

2. For all $W \in \mathsf{dom}(\varphi)$ and $1 \leq j \leq i_1$, e_j belongs to the left (resp., right) order of W if and only if u_j belongs to the left (resp., right) order of $\varphi(W)$.
3. For each $W \in \mathsf{dom}(\varphi)$, the restriction of the position p to the left (resp., right) order of W and the left (resp., right) order of $\varphi(W)$ is a winning position for Duplicator in the i-round WMSO-EF-game.
4. For all $1 \leq j \leq i_1$, e_j is the final node of $\mathcal{E}$ if and only if u_j is the final node of $\mathcal{U}$.
5. For all $1 \leq j \leq i_2$, E_j contains the final node of $\mathcal{E}$ if and only if U_j contains the final node of $\mathcal{U}$.

Remark 6.28. Note that the WMSO+B-EF-game on ($\mathcal{E}$,$\mathcal{U}$) starts in a locally-k-winning position where the partial map φ is the map with empty domain. Moreover, for all $i \in \mathbb{N}$, every locally-i-winning position is a winning position for Duplicator in the 0-round WMSO+B-EF-game.

Proposition 6.29. *Duplicator has a winning strategy in the k-round* WMSO+B-*EF-game on* ($\mathcal{E}_k$,$\mathcal{U}_k$).

Due to the previous remark, the proposition follows directly form the following lemma.

Lemma 6.30. *Let $1 \leq i \leq k$ be a natural number and p a locally-i-winning position. Duplicator can respond any challenge of Spoiler so that the next position is locally-$(i-1)$-winning.*

Proof. Let $\varphi : \bar{E} \to \bar{U}$ be the partial bijection for the locally-i-winning position p. In the following, we say that an (n, m)-triple-u is *fresh* if it does not belong to $\mathsf{dom}(\varphi) \cup \mathsf{im}(\varphi)$. We consider the three possible types of moves for Spoiler.

1. If Spoiler plays an element move, there are the following possibilities.

   - If Spoiler chooses the final node of one of the structures, Duplicator answers with the final node of the other.
   - If Spoiler chooses some node from an (n, m)-triple-u $W \in \mathsf{dom}(\varphi)$, then the local strategies allow Duplicator to answer this move with a node from $\varphi(W)$.
   - Analogously, if Spoiler chooses some node from an (n, m)- triple-u $W \in \mathsf{im}(\varphi)$, then the local strategies allow Duplicator to answer this move with a node from $\varphi^{-1}(W)$.

- If Spoiler chooses a node from a fresh (n, m)-triple-u W then Duplicator can choose some fresh (n', m')-triple-u W' from the other structure and can use the WMSO-equivalence up to quantifier rank k of the left and right orders of W and W' to find a response to Spoilers challenge such that adding (W, W') (or (W', W) depending on whether $W \in \bar{E}$) to φ leads to a locally-$(i-1)$-winning position.

2. If Spoiler plays a set move, then he chooses a finite set containing elements from some of the triple-u's from $\mathsf{dom}(\varphi)$ or $\mathsf{im}(\varphi)$ and from l many fresh triple-u's. Choosing l fresh triple-u's from the other structure, we can find a response on each of the triple-u's corresponding to the local strategy similar to the case of the element move. The union of all these local responses is a response for Duplicator that leads to a locally-$(i-1)$-winning position.

3. If Spoiler plays a bound move, we distinguish which structure he chooses.

   - If he chooses structure $\mathcal{U}$ and the bound $l \in \mathbb{N}$, let Z_n be the (finite) set of all (n, n)-triple-u's occurring in $\mathsf{im}(\varphi)$ and set

     $$m_1 = \sum_{n \in \mathbb{N}} \sum_{W \in Z_n} (2n + 7).$$

     Duplicator responds with the bound $m = m_1 + 2l$. Note that $2n+7$ is the size of an (n, n)-triple-u. Hence m_1 is the number of nodes in non-fresh triple-u's of $\mathcal{U}$. Assume that Spoiler chooses some finite subset S of $\mathcal{U}$ with $|S| \geq m$. We construct a subset S' in $\mathcal{E}$ such that the resulting position is locally-$(i-1)$-winning. Moreover, we guarantee that for any fresh triple-u $W \in \bar{U}$ such that $S \cap W \neq \emptyset$, Duplicator's response $S' \cap W'$ in a corresponding fresh triple-u $W' \in \bar{E}$ contains at least $\frac{1}{2}|S \cap W|$ many elements. If $W_1, \dots, W_z \in \bar{U}$ are all the fresh triple-u's that intersect S non trivially, then we already argued that $|\bigcup_{i=1}^{z} (W_i \cap S)| \geq m - m_1 = 2l$. Hence, Duplicator's response S' contains at least l many elements as desired. The concrete choice of S' is done as follows.

     (a) For all $W \in \mathsf{im}(\varphi)$, Duplicator chooses a set $S'_W \subseteq \varphi^{-1}(W)$ such that S'_W is the answer to Spoiler's challenge $S \cap W$ according to a winning strategy in the i-round WMSO-EF-game on the restriction of p to $\varphi^{-1}(W)$ and W. This winning strategy exists because position p is locally i-winning.

     (b) Now consider a fresh (n, n)-triple-u $W \in \bar{U}$ with $W \cap S \neq \emptyset$. Let L (resp., R) be the nodes in the left (resp., right) order of W. If $|L \cap S| \geq |R \cap S|$, then take a fresh (n, n_1)-triple-u $W' \in \bar{E}$ (note that $n \geq n_1$) and extend the partial bijection φ by $\varphi(W') = W$. Duplicator chooses the subset $S'_W = \psi(S \cap W \setminus R) \cup T$, where ψ is the obvious isomorphism between the $(n,0)$-sub-triple-u of W (i.e., $W \setminus R$) and the $(n,0)$-sub-triple-u of W', and T is an answer to Spoilers move $S \cap R$ according to a winning strategy in the i-round WMSO-EF-game between the right order of W' and the right order of W. Note that $|S'_W| \geq \frac{1}{2}|S \cap W|$.

If $|L \cap S| < |R \cap S|$, then let W' be an (n_1, n) triple-u and use the same strategy but reverse the roles of the left and the right order of the chosen triple-u's.

(c) If the final node of $\mathcal{U}$ is in S, let S'_d be the singleton containing the final node of $\mathcal{E}$, otherwise let $S'_d = \emptyset$.

Finally, let S' be the union of S'_d and all sets S'_W defined in (a) and (b) above. Since Spoiler has chosen at least $2l-1$ many elements from fresh triple-u's, we directly conclude that $|S'| \geq l$. Moreover, since all the parts of S' were defined using local strategies, we easily conclude that the position reached by choosing S' is locally-$(i-1)$-winning.

- If Spoiler chooses structure $\mathcal{E}$ and bound $l \in \mathbb{N}$, we use a similar strategy. Let Y_n be the set of all (n_1, n)-triple-u's and all (n, n_1)-triple-u's occurring in $\mathsf{dom}(\varphi)$, and define

$$m_1 = \sum_{n \in \mathbb{N}} \sum_{W \in Y_n} n_1 + n + 7,$$

and $m_2 = l \cdot n_1$. Note that m_1 is the number of nodes from non-fresh triple-u's from $\mathcal{E}$. Duplicator responds with $m = m_1 + m_2 + l$. Let $S \subseteq \mathcal{E}$ be Spoiler's set with $|S| \geq m$. There are at least $m_2 + l$ elements in S chosen from fresh triple-u's $W_1, W_2, \ldots, W_z \in \bar{E}$. Either $z > l$ or Spoiler has chosen at least l elements from $W_1 \cup W_2 \cup \cdots \cup W_z$ that do not belong to the orders of length n_1 (which in total contain only $z \cdot n_1 \leq l \cdot n_1 = m_2$ many elements). Duplicator chooses his response S' in $\mathcal{U}$ as follows:

(a) For all $W \in \mathsf{dom}(\varphi)$, Duplicator chooses a set $S'_W \subseteq \varphi(W)$ such that S'_W is the answer to Spoiler's challenge $S \cap W$ according to a winning strategy in the i-round WMSO-EF-game on the restriction of p to W and $\varphi(W)$. This winning strategy exists because position p is locally i-winning.

(b) Now consider a fresh triple-u $W \in \bar{E}$ with $W \cap S \neq \emptyset$. If W is an (n_1, n)-triple-u or an (n, n_1)-triple-u, let $W' \in \bar{U}$ be a fresh (n, n)-triple-u of $\mathcal{U}$, and extend the partial bijection φ by $\varphi(W) = W'$.

Let us consider the case that W is an (n, n_1)-triple-u (for the other case one can argue analogously) and let R be the right order (of size n_1) of W. Duplicator chooses the subset $S'_W = \psi(S \cap W \setminus R) \cup T$, where ψ is the obvious isomorphism between the $(n, 0)$-sub-triple-u of W (i.e., $W \setminus R$) and the $(n, 0)$-sub-triple-u of W', and T is an answer to Spoiler's move $S \cap R$ according to a winning strategy in the i-round WMSO-EF-game between the right order of W and the right order of W'. We can assume that $S'_W \neq \emptyset$. because we have $S \cap W \setminus R \neq \emptyset$ or $S \cap R \neq \emptyset$ and in the latter case T can be chosen to be non-empty.

(c) If the final node of $\mathcal{E}$ is in S, let S'_d be the singleton containing the final node of $\mathcal{U}$, otherwise let $S'_d = \emptyset$.

Finally, let Duplicator's response S' be the union of S'_d and all sets S'_W defined in (a) and (b) above. By the argument before (a), Duplicator selects in (b) in total at least l elements. Moreover, since all the parts of S' where defined using local strategies, we easily conclude that the position reached by choosing S' is locally-$(i-1)$-winning.

□

Chapter 7

Extensions

7.1 Existential Interpretation Preserves Satisfiability

Let us state a simple preservation theorem for $\mathcal{A}$-SAT. Assume that $\mathcal{A} = (A, I)$ and $\mathcal{B} = (B, J)$ are structures over countable signatures $\sigma_{\mathcal{A}}$ and $\sigma_{\mathcal{B}}$, respectively. We say that $\mathcal{A}$ is **existentially interpretable** in $\mathcal{B}$ if there exist $n \geq 1$, a quantifier-free first-order formula $\varphi(y_1, \ldots, y_l, x_1, \ldots, x_n)$, and for each $R \in \sigma_{\mathcal{A}}$ with $k = \mathsf{ar}(R)$ a quantifier-free first order formula

$$\varphi_R(z_1, \ldots, z_{l_R}, x_{1,1}, \ldots, x_{1,n}, \ldots, x_{k,1}, \ldots, x_{k,n})$$

over the signature $\sigma_{\mathcal{B}}$, where the mapping $R \mapsto \varphi_R$ has to be computable, such that $\mathcal{A}$ is isomorphic to the structure (A', I'), where

$$A' = \{\bar{b} \in B^n \mid \exists \bar{c} \in B^l : \mathcal{B} \models \varphi(\bar{c}, \bar{b})\} \text{ and}$$
$$I'(R) = \{(\bar{b}_1, \ldots, \bar{b}_k) \in B^{kn} \mid \exists \bar{c} \in B^{l_R} : \mathcal{B} \models \varphi_R(\bar{c}, \bar{b}_1, \ldots, \bar{b}_k)\}$$

for each $R \in \sigma_{\mathcal{A}}$.

Proposition 7.1. *If $\mathcal{B}$-SAT is decidable and $\mathcal{A}$ is existentially interpretable in $\mathcal{B}$, then $\mathcal{A}$-SAT is decidable too.*

Proof. Let ψ be an CECTL^*-formula over $\sigma_{\mathcal{A}}$. Let Reg_ψ be the set of register variables that occur in ψ. Let us choose new register variables $s_{r,j}$, and $t_{R,m}$ for all $r \in \mathsf{Reg}_\psi$, $1 \leq j \leq l$, $R \in \sigma_{\mathcal{A}}$, and $1 \leq m \leq l_R$. Furthermore we need n copies of each $r \in \mathsf{Reg}_\psi$: r^i for $1 \leq i \leq n$. Define the CECTL^*-formula on $\sigma_{\mathcal{B}}$

$$\vartheta = \psi' \wedge \mathsf{A}\Big[\forall x \bigwedge_{r \in \mathsf{Reg}_\psi} \varphi\big(s_{r,1}, \ldots, s_{r,l}, r^1, \ldots, r^n\big)(x)\Big] \ ,$$

where ψ' is obtained from ψ by replacing every constraint $R(S^{i_1}r_1, \ldots, S^{i_k}r_k)$ in ψ (where $k = \mathsf{ar}(R)$) by the boolean formula

$$\varphi_R(S^d t_{R,1}, \ldots, S^d t_{R,l_R}, S^{i_1}r_1^1, \ldots, S^{i_1}r_1^n, \ldots, S^{i_k}r_k^1, \ldots, S^{i_k}r_k^n),$$

where $d = \max\{i_1, \ldots, i_{\mathsf{ar}(R)}\}$. Using arguments similar to those from the proof of Lemma 4.9, one can show that ψ is $\mathcal{A}$-satisfiable if and only if ϑ is $\mathcal{B}$-satisfiable. $\square$

Examples of structures $\mathcal{A}$ that are existentially interpretable in $(\mathbb{Z}, <, \equiv)$, and hence have a decidable $\mathcal{A}$-SAT-problem are:

- $(\mathbb{Z}^n, <_{\text{lex}}, \equiv)$ (for $n \geq 1$), where $<_{\text{lex}}$ denotes the strict lexicographic order on n-tuples of integers, and
- the structure $\text{Allen}_{\mathbb{Z}}$, which consists of all $\mathbb{Z}$-intervals together with Allen's relations b (before), a (after), m (meets), mi (met-by), o (overlaps), oi (overlapped by), d (during), di (contains), s (starts), si (started by), f (ends), fi (ended by). In artificial intelligence, Allen's relations are a popular tool for representing temporal knowledge.

7.2 Finite Satisfiability

Fix a signature σ and a negation-closed σ-structure as concrete domain $\mathcal{D} = (D, I)$. We say that a CECTL^*-formula φ is **finitely $\mathcal{D}$-satisfiable** if there is a $\mathcal{D}$-Kripke structure K, whose underlying Kripke structure $\mathcal{K}$ is finite, and a node v of $\mathcal{K}$ such that $(\mathsf{K}, v) \models \varphi$. We denote as $\mathsf{FINSAT}(\mathcal{D})$ the following computational problem: *Is a given formula $\varphi \in \mathsf{CECTL}^*$ finitely $\mathcal{D}$-satisfiable?* The main result of this section is the following.

Proposition 7.2. *A CECTL^*-formula ψ is finitely $\mathcal{D}$-satisfiable if and only if there is a $\mathcal{D}$-Kripke structure $\mathsf{K} = (\mathcal{D}, \mathcal{K}, \gamma)$, where $\mathcal{K}$ has domain S, and a node $v \in S$ such that*

1. *$(\mathsf{K}, v) \models \psi$ and*
2. *$\mathsf{im}(\gamma_{\restriction \mathsf{Reg}_\psi})$ is finite, where Reg_ψ is the set of register variables occurring in ψ,*

i.e., there exists a model for ψ where the valuation function γ assigns only finitely many elements of $\mathcal{D}$.

Proof. The "only-if" part is trivial because every finite model of φ satisfies conditions 1. and 2. For the "if" part let us start with a $\mathcal{D}$-Kripke structure K with underlying Kripke structure $\mathcal{K} = (S, \rightarrow, \rho)$ satisfying conditions 1. and 2. We have to find a finite model of ψ. W.l.o.g. we can assume that every node of S is reachable from v.

We now define an abstracted CECTL^*-formula ψ^a (without constraints) as follows: First take for all $r \in \mathsf{Reg}_\psi$ and all $a \in \mathsf{im}(\gamma)$ a fresh proposition $p_{r,a}$, which has the following intuitive meaning: "register variable r is mapped to the value a". Then we construct from ψ the formula ψ_0 by replacing every occurrence of an atomic constraint $R(S^{i_1}r_1, \ldots, S^{i_k}r_k)(x)$ by the CECTL^*-path formula

$$\bigvee_{(a_1,\ldots,a_k)\in B} \bigwedge_{j=1}^{k} p_{r_j,a_j}(x + i_j) \ ,$$

where $B = I(R) \cap \mathsf{im}(\gamma)^k$. Finally, we define $\psi^a = \psi_0 \wedge \psi_1$, where ψ_1 is defined as

$$\psi_1 = \mathsf{A}\forall x \Big(\bigwedge_{r\in\mathsf{Reg}_\psi} \bigvee_{a\in\mathsf{im}(\gamma)} (p_{r,a}(x) \wedge \bigwedge_{b\in\mathsf{im}(\gamma)\setminus\{a\}} \neg p_{r,b}(x)) \Big) \ .$$

It states that for every node x that is reachable from the current node and every $r \in \mathsf{Reg}_\psi$ there is exactly one $a \in \mathsf{im}(\gamma)$ such that x is labeled with $p_{r,a}$. In the intuitive sense, we are making sure that each register variable is assigned only one value from $\mathsf{im}(\gamma)$.

In a first step, we construct from the $\mathcal{D}$-KS K, which is a model for ψ, a Kripke structure K^a, which is a model of ψ^a. For this, we extend the Kripke structure $\mathcal{K} = (S, \rightarrow, \rho)$ to the Kripke structure $\mathsf{K}^a = (S, \rightarrow, \rho^a)$, where

$$\rho^a(e) = \rho(e) \cup \{p_{r,a} \mid \gamma(e, r) = a\}.$$

We clearly have $(\mathsf{K}^a, v) \models \psi_1$. Moreover, a simple induction over the structure of formulas shows that $(\mathsf{K}^a, v) \models \psi_0$.

Now, ECTL^* has the finite model property. This follows from the facts that (i) ECTL^*-formulas can be translated into equivalent modal μ-calculus formulas [15], and (ii) that the modal μ-calculus has the finite model property [30]. Therefore, there exists a finite Kripke structure $\mathcal{K}' = (S', \rightarrow', \rho')$ and $v' \in S'$ such that $(\mathcal{K}', v') \models \psi^a$. W.l.o.g. we can assume that every node of S' is reachable from the node v'.

We finally construct from $\mathcal{K}'$ a finite model K' for our original formula ψ. The underlying Kripke structure is $\mathcal{K}'$, where we can remove the new propositions $p_{r,a}$. We define the valuation function γ as follows: Let $e \in S'$ and $r \in \mathsf{Reg}_\psi$. Since e is reachable from v' and $(\mathcal{K}', v') \models \psi_1$ there must exist a unique $a \in \mathsf{im}(\gamma)$ such that $p_{r,a} \in \rho'(e)$. We set $\gamma(e, r) = a$.

We also have $(\mathcal{K}', v') \models \psi_0$. A simple induction finally shows that this implies $(\mathsf{K}', v') \models \psi$.

$\square$

Given this characterization we can prove the following result:

Corollary 7.3. *Let $\mathcal{Z}$ be the σ-structure defined in (1.2) on page 7 (or one of its expansions from the previous chapters). Then $\mathsf{FINSAT}(\mathcal{Z})$ is decidable.*

Proof. Let Reg_φ be the set of register variables appearing in φ, and choose two fresh variables s, t. Let ψ be defined as the conjunction of the following two formulas:

$$\psi_1 = \mathsf{A}\forall x\,(s = Ss)(x) \wedge (t = St)(x)$$
$$\psi_2 = \mathsf{A}\forall x \bigwedge_{r \in \mathsf{Reg}_\varphi} (s \leq r \leq t)(x)$$

It is not hard to see that φ is finitely $\mathcal{Z}$-satisfiable if and only if $(\varphi \wedge \psi)$ is $\mathcal{Z}$-satisfiable: Suppose that $(\mathsf{K}, v) \models \varphi \wedge \psi$ for a $\mathcal{Z}$-Kripke structure $\mathsf{K} = (\mathcal{Z}, \mathcal{K}, \gamma)$, where w.l.o.g. every node is reachable from v. Then ψ_1 enforces that γ assigns s and t a constant value, i.e. $\gamma(w, s) = a$ and $\gamma(w, t) = b$ for all w nodes of $\mathcal{K}$. At the same time ψ_2 requires that every other register variable r which appears in φ has assigned some value $z \in \mathbb{Z}$ that belongs to the interval $[a, b]$. By Proposition 7.2, $\varphi \wedge \psi$ has a finite model, which is also a model of φ.

Vice versa, if φ has a finite model K, then there are integers $c, d \in \mathbb{Z}$ such that $\mathsf{im}(\gamma_{\restriction \mathsf{Reg}\varphi}) \subseteq [c, d]$. We can extend K to a model for $\varphi \wedge \psi$ by defining $\gamma(w, s) = c$ and $\gamma(w, t) = d$ for every node w of $\mathcal{K}$.

Since $\mathcal{Z}$-SAT is decidable (Theorem 5.2) so is $\mathsf{FINSAT}(\mathcal{Z})$.

$\square$

We can use Corollary 7.3 to show that for every linear order $\mathcal{L}$ (extended with the equality relation), $\mathsf{FINSAT}(\mathcal{L})$ is decidable:

Corollary 7.4. *Let $(L, <)$ be a linear order and define $\mathcal{L} = (L, <, \equiv)$ where $\equiv$ is the equality relation on L. Then $\mathsf{FINSAT}(\mathcal{L})$ can be reduced to $\mathsf{FINSAT}(\mathcal{Z})$, and is therefore decidable.*

Proof. First assume that L is infinite. Let φ be a CECTL^*-formula over the signature $\{<, \equiv\}$ and let $\mathsf{K} = (\mathcal{Z}, \mathcal{K}, \gamma)$ be a finite $\mathcal{Z}$-KS in which φ holds. Choose $a, b \in \mathbb{Z}$ such that $\mathsf{im}(\gamma_{\restriction \mathsf{Reg}_\varphi}) \subseteq [a, b]$. Let $n = b - a$. Since L is infinite, there exists elements $l_0, \ldots, l_n \in L$ such that $l_0 < l_1 < \cdots < l_n$ in $(L, <)$. Let $\mathsf{K}' = (\mathcal{L}, \mathcal{K}, \gamma')$ be the $\mathcal{L}$-KS with the same underlying Kripke structure as K and $\gamma'(d, r) = l_i$ if $\gamma(d, r) = a + i$. This is clearly a finite model of φ over the domain $\mathcal{L}$. By reversing the role of $\mathcal{L}$ and $\mathcal{Z}$, we can show that φ is finitely $\mathcal{Z}$-satisfiable if φ is finitely $\mathcal{L}$-satisfiable.

If L is a finite set with $c = |L|$, then we can reduce $\mathsf{FINSAT}(\mathcal{L})$ again to $\mathsf{FINSAT}(\mathcal{Z})$ by mapping a formula $\varphi \in \mathsf{ECTL}^*(\{<, \equiv\})$ to $\varphi \wedge \psi$, where ψ is a variant of the formula from the proof of Corollary 7.3. Using the relations $\equiv_1$ and $\equiv_c$ we have to bound the value taken by each register variable $r \in \mathsf{Reg}$ that appears in φ to the interval $[1, c]$. $\square$

It is open whether there is a linear order for which $\mathcal{L}$-SAT is undecidable.

Remark 7.5. Instead of using a reduction to the satisfiability problem, one can prove all decidability results of this section with the following approach: Analogously to the definition of $\mathsf{EHD}(\mathcal{L})$ (for a logic $\mathcal{L}$), say that a σ-structure $\mathcal{A}$ has the property $\mathsf{EHD}_{\mathsf{fin}}(\mathcal{L})$ if there is a computable function that maps every finite subsignature $\tau \subseteq \sigma$ to an $\mathcal{L}$-sentence φ_τ such that for every countable τ-structure $\mathcal{B}$ we have the following: There exists a homomorphism $h : \mathcal{B} \to \mathcal{A}$ with finite image if and only if $\mathcal{B} \models \varphi_\tau$.

Then we can follow exactly all the steps relating the EHD-property of a structure $\mathcal{D}$ with decidability of $\mathcal{D}$-SAT and obtain a proof that $\mathsf{FINSAT}(\mathcal{D})$ is decidable for every negation-closed domain $\mathcal{D}$ with property $\mathsf{EHD}_{\mathsf{fin}}(\mathsf{BMW})$. The results stated above then follow from the fact that every infinite linear order has property $\mathsf{EHD}_{\mathsf{fin}}(\mathsf{BMW})$: A constraint graph allows a homomorphism with finite image to an infinite linear order if and only if there is a bound on the length of the longest $<$-chain (after contraction of $\equiv$-edges as usual).

7.3 A generalization of the EHD-method

In Theorem 4.7, connecting the notion of EHD-property to the satisfiability problem for CECTL^*, we state our result for domains which enjoy the property $\mathsf{EHD}(\mathsf{BMW})$, where BMW is short for the logical language formed by all Boolean combinations of MSO and $\mathsf{WMSO{+}B}$ sentences.

This result can be generalized. Given a negation-closed concrete domain $\mathcal{D}$, all we need for our method to work, is that $\mathcal{D}$ has the property $\mathsf{EHD}(\mathcal{L})$ for some logic $\mathcal{L}$ which satisfies the following three properties:

- P1 Satisfiability of a given $\mathcal{L}$-sentence over the class of infinite node-labeled trees is decidable.
- P2 $\mathcal{L}$ is closed under boolean combinations with MSO-formulas.
- P3 $\mathcal{L}$ is compatible with one dimensional first-order interpretations and with the k-copy operation.

For instance, MSO itself satisfies all the above properties. By Rabin's seminal tree theorem [37], satisfiability of MSO-sentences over infinite node-labeled trees is decidable and Muchnik's theorem (cf. [42]) implies compatibility of MSO with k-copying.

It is not clear, though, whether the structure that we are most interested in, $(\mathbb{Z}, <, \equiv)$, satisfies the property EHD(MSO), and we actually conjecture that it does not. Thus the need to use the logic WMSO+B, or actually its Boolean closure with MSO, BMW, in order to satisfy P2. This logic, on top of having the above properties, can in fact naturally express the condition which we have found to characterize all those constraint graphs which allow a homomorphism to $(\mathbb{Z}, <, \equiv)$: given any two elements, there is a bound on the length of all paths which connect them.

Chapter 8

Adding Non-Local Constraints

CECTL^* extends ECTL^* with constraints which allow to reason about concrete numerical values. We have remarked before that one characteristic of these constraints is that they have a fixed depth: we can compare the values assigned to the register variables at fixed positions, *e.g.*, we can express equality between the value of r_1 at the current position and the value of r_2 at the i^{th} next position along a path using the formula $r_1 \equiv S^i r_2$.

Different logics like metric temporal logic (MTL), timed propositional temporal logic (TPTL) or $\mathsf{freezeLTL}$ are all extensions of linear temporal logic (LTL) which allow to specify properties of data words. As mentioned in the introduction, data words are basically $\mathcal{A}$-Kripke paths with only one register variable, where $\mathcal{A}$ is typically the set of natural numbers or real numbers, see [2]. In these logics, one can compare the current data value with future values at arbitrary distance from the current position. For instance, we can express the property that there is a future data value which is equal to the current one with the TPTL-formula $x.\mathsf{F}(x = 0)$[1]. It is interesting whether we can add this feature to CLTL, CCTL^* or CECTL^* and preserve decidability. This question was also asked in [17].

To this end, we substitute the atomic constraints from (3.2) on page 25 with non-local ones of the form:

$$R(O_1 r_1, \ldots, O_k r_k)(x) \tag{8.1}$$

where $O_i = S^j$ for some $j \in \mathbb{N}$ or $O_i = \mathsf{F}$. Intuitively, $O_i = \mathsf{F}$ would refer to the r_i-value at some (existentially quantified) future position of the path. On the concrete domain $(\mathbb{Z}, <, \equiv)$, this would allow to express, for instance, the above

[1] TPTL and $\mathsf{freezeLTL}$ are allowed to store the data value of the current position using the *freeze* quantifier $x.\varphi$, and to later compare it with the data value at some other position: $x = k$ means that $d - v(x) = k$, where $v(x)$ is the value of x which is currently stored, and d is the data value at the current position.

mentioned property that there is a future position in which the value of the register variable r matches the one in the current position: $(r \equiv \mathsf{F}r)(x)$.

Unfortunately we can show that this leads to undecidability of the satisfiability problem, also in very restricted settings: Even if we consider as the starting point logic LTL instead of ECTL^*, adding these new constraints causes undecidability of the satisfiability problem on very simple concrete domains, like $(\mathbb{N}, <, \equiv)$ and $(\mathbb{Z}, <, \equiv)$ (Section 8.1).

On the positive side, we can regain decidability on these concrete domains by disallowing the use of non-local equality constraints (Section 8.2).

Definition 8.1. $\mathsf{CLTL}[\mathsf{F}]$ on the signature σ is the extension of CLTL defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid R(O_1 r_1, \dots, O_k r_k)$$

where $p \in \mathbb{P}$, $R \in \sigma$, $k = \mathsf{ar}(R)$, and for all $1 \leq j \leq k$, $r_j \in \mathsf{Reg}$ and $O_j = \mathsf{X}^{i_j}$ for some $i_j \in \mathbb{N}$ or $O_j = \mathsf{F}$.

$\mathsf{CLTL}[\mathsf{F}]$ is nothing but LTL extended by non-local constraints as those from (8.1). Since we now add constraints to a temporal logic (instead of MSO on paths) we go back to the syntax from (1.1) on page 5: As usual in temporal logics, we don't need variables to point to nodes of the Kripke structure, which can only be navigated using the temporal operators, and use X instead of the symbol for the successor function, i.e., in our new constraints the term $\mathsf{X}^j r_j$ replaces the CECTL^*-term $S^j r_j$. We also use the classical abbreviations, in particular $\mathsf{G}\varphi$ (φ holds globally in the future) and $\mathsf{F}\varphi$ (φ holds finally in the future).

The semantics of $\mathsf{CLTL}[\mathsf{F}]$ is mostly inherited from that of LTL, but while LTL-formulas are evaluated over words (Kripke paths), we evaluate a $\mathsf{CLTL}[\mathsf{F}]$-formula on a $\mathcal{D}$-decorated Kripke path $\mathbf{P} = (\mathcal{D}, \mathcal{P}, \gamma)$, where $\mathcal{P}$ is a Kripke path. Note that the valuation function γ and the concrete domain $\mathcal{D} = (D, I)$ only play a role in evaluating constraints: We define $(\mathbf{P}, n) \models R(O_1 r_1, \dots, O_k r_k)$ if and only if there are $i_1, \dots, i_k$ such that

$$(\gamma(n + i_1, r_1), \dots, \gamma(n + i_k, r_k)) \in I(R)$$

where $i_l = j$ if $O_l = \mathsf{X}^j$ and $i_l > 0$ if $O_l = \mathsf{F}$ for all $1 \leq l \leq k$ and $j \in \mathbb{N}$.

8.1 Undecidability of LTL with Non-Local Constraints

As anticipated, the main result of this section is the following:

Theorem 8.2. *Satisfiability for* $\mathsf{CLTL}[\mathsf{F}]$ *of the concrete domains* $(\mathbb{Z}, <, \equiv)$ *and* $(\mathbb{N}, <, \equiv)$ *is undecidable.*

To obtain this result we use *incrementing counter automata*, in short ICAs, first introduced in [20]. In contrast to their definition in [20], we use input-free ICAs, but this does not change things, since we are only interested in the emptiness problem.

Definition 8.3. An **incrementing counter automaton** (ICA) with ε-transitions and zero testing is a tuple $C = (Q, q_I, n, \delta, F)$, where:

- Q is a finite set of states,
- $q_I \in Q$ is the initial state,
- $n \in \mathbb{N}$ is the number of counters,
- $\delta \subseteq Q \times L \times Q$ is the transition relation over the instruction set $L = \{\mathsf{inc}_i, \mathsf{dec}_i, \mathsf{ifz}_i \mid 1 \leq i \leq n\}$, and
- $F \subseteq Q$ is the set of accepting states.

A **configuration** of C is a pair (q, v) where $q \in Q$ and $v : \{1, \ldots, n\} \to \mathbb{N}$ is a **counter valuation**. For configurations (q, v), (q', v'), and an instruction $l \in L$ there is an **exact transition** $(q, v) \xrightarrow{l}_{\dagger} (q', v')$ of C if and only if $(q, l, q') \in \delta$ and one of the following cases holds:

- $l = \mathsf{inc}_i$ for some i, $v(j) = v'(j)$ for $j \neq i$, and $v'(i) = v(i) + 1$
- $l = \mathsf{dec}_i$ for some i, $v(j) = v'(j)$ for $j \neq i$, $v(i) > 0$, and $v'(i) = v(i) - 1$
- $l = \mathsf{ifz}_i$ for some i, $v(i) = 0$, and $v'(j) = v(j)$ for all j.

We define a partial order $\leq$ on counter valuations as follows: $v \leq w$ if and only if $v(i) \leq w(i)$ for all $1 \leq i \leq n$.

The **transitions** of C are of the form $(q, w) \xrightarrow{l} (q', w')$ such that there are v, v' with an exact transition $(q, v) \xrightarrow{l}_{\dagger} (q', v')$, $w \leq v$, and $v' \leq w'$.

An **infinite run** of C is an infinite sequence of transitions

$$(q_0, v_0) \xrightarrow{l_0} (q_1, v_1) \xrightarrow{l_1} \cdots \text{ such that } q_0 = q_I \ .$$

An infinite run is **accepting** if and only if some accepting state occurs infinitely often.

Essentially, ICAs relax the conditions on transitions, by letting faulty increments occur at any time. The problem whether an ICA admits an accepting run is deeply connected to that of the halting problem (for finite runs) and of the recurring state problem (for infinite runs) of *insertion channel machines with* *emptiness testing*, see [35]. Their computational power is strictly weaker than that of perfect channel machines, but emptiness is still undecidable on infinite words, which makes them a useful tool for undecidability proofs.

Theorem 8.4 (see Theorem 2.9b of [20])**.** *The existence of an infinite accepting run for ICAs is undecidable and* Π^0_1*-complete.*

To prove undecidability of the satisfiability problem for $\mathsf{CLTL}[\mathsf{F}]$ over the concrete domain $(\mathbb{Z}, <, \equiv)$, we use a reduction from the infinite accepting run problem for ICAs (for the method we drew inspiration from [20]) .

Proof (Theorem 8.2). Let $C = (Q, q_I, n, \delta, F)$ be an ICA. We define an $\mathsf{CLTL}[\mathsf{F}]$-formula φ_C on the atomic proposition set $\mathbb{P} = Q \cup L$ where Q are the states of C and $L = \{\mathsf{inc}_i, \mathsf{dec}_i, \mathsf{ifz}_i \mid 1 \leq i \leq n\}$. We build φ_C so to be satisfiable over the concrete domain $\mathcal{D} = (\mathbb{Z}, <, \equiv)$ (or $\mathcal{D} = (\mathbb{N}, <, \equiv)$) if and only if C has an infinite accepting run.

To encode a successful run of C, we require that a $\mathcal{D}$-Kripke path $\mathfrak{P}$ satisfies the properties below:

- In each position of the path $\mathfrak{P}$, one and only one state from Q occurs, and one and only one operation from L occurs:

$$\varphi_{\text{struct}} := \mathsf{G}\Big(\bigvee_{q \in Q} q \wedge \bigvee_{l \in L} l \wedge \bigwedge_{\substack{q,q' \in Q \\ q \neq q'}} (q \to \neg q') \wedge \bigwedge_{\substack{l,l' \in L \\ l \neq l'}} (l \to \neg l')\Big) .$$

- The computation starts with the initial state and reaches a final state infinitely often:

$$\varphi_{\text{Büchi}} := q_I \wedge \bigvee_{q \in F} \mathsf{GF} q .$$

- The transition relations of C are encoded in the following way:

$$\varphi_{\text{trans}} := \mathsf{G} \bigwedge_{q \in Q} \Big(q \to \bigvee_{(q,l,q') \in \delta} (l \wedge \mathsf{X} q')\Big) .$$

- We fix $2n$ pairwise different register variables $r_i, s_i \in \mathsf{Reg}$ for $1 \leq i \leq n$. We use their interpretations to identify each inc_i-operation and dec_i-operation, respectively. While the identifiers are assigned univocally for the increment instructions, more than one decrement can have the same identifier value. To make sure that each inc-operation on counter i is assigned a unique value, we require that at every position of the path $\mathfrak{P}$, which corresponds to an inc_i-operation, r_i is assigned a strictly greater value than in the previous position, and otherwise remains constant.

For the sequence of values of s_i we only require that it stays constant whenever the instruction dec_i does not occur, and it is otherwise non-decreasing:

$$\varphi_{\mathsf{inc}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((\mathsf{inc}_i \rightarrow r_i < \mathsf{X} r_i) \wedge (\neg \mathsf{inc}_i \rightarrow r_i \equiv \mathsf{X} r_i) \Big)$$

$$\varphi_{\mathsf{dec}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((s_i \leq \mathsf{X} s_i) \wedge (\neg \mathsf{dec}_i \rightarrow s_i \equiv \mathsf{X} s_i) \Big) \ .$$

- Whenever a zero test on counter i occurs, the counter should be empty. To make sure that a run respects this property, we should check that, for each increase on counter i, we can find *at least* a corresponding decrease. It is not necessary that this correspondence is exact, since a faulty increase can occur at any time, making additional decreases possible. We use the identifier values r_i and s_i to match each inc_i, which is eventually followed by a zero test ifz_i, to a dec_i with the same identifier:

  $$\varphi_{\mathsf{ifz1}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((\mathsf{inc}_i \wedge \mathsf{F} \mathsf{ifz}_i) \rightarrow \mathsf{X}(r_i \equiv \mathsf{F} s_i) \Big) \ . \tag{8.2}$$

  We should also enforce the fact that, for each inc_i, the correspondent dec_i occurs after inc_i and before ifz_i. For this we require that s_i is never assigned a higher value than r_i, and that they coincide in the occurrence of a zero test instruction on counter i. Since s_i cannot decrease, this means that any dec_i with the same value of an inc_i-instruction should happen before the zero test:

  $$\varphi_{\mathsf{ifz2}} := \mathsf{G} \bigwedge_{i=1}^{n} (r_i \geq s_i \wedge (\mathsf{ifz}_i \rightarrow r_i \equiv s_i)) \ .$$

Let φ_C be the conjunction of all the above formulas. We prove the following equivalence:

$$C \text{ has an accepting run} \quad \Longleftrightarrow \quad \varphi_C \text{ is satisfiable} \ .$$

Proof of $\Longrightarrow$. Let $\mathsf{run} = (q_0, v_0) \xrightarrow{l_0} (q_1, v_1) \xrightarrow{l_1} \cdots$ be a successful run of C. Starting from this we define a $\mathcal{D}$-Kripke path $\mathsf{P} = (\mathcal{D}, \mathcal{P}, \gamma)$ which satisfies φ_C, where $\mathcal{D}$ can be $(\mathbb{N}, <, \equiv)$ or $(\mathbb{Z}, <, \equiv)$.

First of all we define $\mathcal{P} = (\mathbb{N}, \rightarrow, \rho)$, where $\rightarrow$ is the successor relation on the natural numbers, and $\rho(i) = \{q_i, l_i\}$ for all $i \in \mathbb{N}$. Since the run is successful, this ensures that $\varphi_{\text{struct}} \wedge \varphi_{\text{Büchi}} \wedge \varphi_{\mathsf{trans}}$ is satisfied.

Now we define the interpretations of r_i and s_i. For all $1 \leq i \leq n$, we define $\gamma(0, r_i) = \gamma(0, s_i) = 0$. For all other nodes $j \geq 1$ we define

$$\gamma(j, r_i) = \begin{cases} \gamma(j-1, r_i) + 1 & \text{iff } l_{j-1} = \mathsf{inc}_i \\ \gamma(j-1, r_i) & \text{otherwise} \end{cases}$$

$$\gamma(j, s_i) = \begin{cases} \gamma(j-1, s_i) + 1 & \text{iff } l_{j-1} = \mathsf{dec}_i \text{ and } \gamma(j-1, s_i) < \gamma(j-1, r_i) \\ \gamma(j-1, s_i) & \text{otherwise} \end{cases}$$

Clearly this definition of γ makes φ_{inc} and φ_{dec} true. To prove that also φ_{ifz1} and φ_{ifz2} hold, we note that, since run is a successful run of C, a transition with operation ifz_i can only occur if counter i is empty. Therefore, the number of increase instructions on counter i, between any two ifz_i, should be matched by an equal or greater number of decrease instructions. By definition of the functions, for each increase on the value of r_i which is eventually followed by a zero test on counter i, there is a corresponding increase on the value of s_i. Furthermore, whenever s_i reaches the value of r_i, the value of s_i is no longer increased until r_i grows again, thus ensuring that $\varphi_{\mathsf{ifz1}} \wedge \varphi_{\mathsf{ifz2}}$ holds.

Proof of $\Longleftarrow$. Let $\mathbf{P} = (\mathcal{D}, \mathcal{P}, \gamma)$ be a $\mathcal{D}$-Kripke path such that $\mathbf{P} \models \varphi_C$. $\mathcal{D}$ can be $(\mathbb{N}, <, \equiv)$ or $(\mathbb{Z}, <, \equiv)$, this does not change the proof. We define a run

$$\mathsf{run} = (q_0, v_0) \xrightarrow{l_0} (q_1, v_1) \xrightarrow{l_1} \cdots$$

of C and prove that it is accepting. By $\varphi_{\text{struct}} \wedge \varphi_{\text{Büchi}}$ the label $\rho(\mathbf{P}(i))$ of every node of the path $\mathbf{P}$ contains one and only one symbol q from Q and l from L. We set $q_i = q$ and $l_i = l$. Since $\varphi_{\text{Büchi}}$ holds, q_0 is the initial state q_I, and an accepting state is visited infinitely often. Since φ_{trans} holds, for every $i \in \mathbb{N}$ we have that $(q_i, l_i, q_{i+1}) \in \delta$. We set to zero the initial value of every counter $1 \leq j \leq n$: $v_0(j) = 0$. For all later positions $i \geq 1$ we define:

$$v_i(j) = \begin{cases} v_{i-1}(j) + 1 & \text{iff } l_{i-1} = \mathsf{inc}_j \\ v_{i-1}(j) - 1 & \text{iff } l_{i-1} = \mathsf{dec}_j \text{ and } v_{i-1}(j) > 0 \\ v_{i-1}(j) & \text{otherwise.} \end{cases}$$

Note that $v_i(j)$ is always positive. It remains to show that

$$(q_i, v_i) \xrightarrow{l_i} (q_{i+1}, v_{i+1}) \tag{8.3}$$

according to Definition 8.3. We only discuss the non trivial cases.

- If $l_i = \mathsf{dec}_j$ and $v_i(j) = 0$, then also $v_{i+1}(j) = 0$. Let v' be the counter valuation that assigns $v'(j) = 1$ and coincides with v_i on all other counters. Then, $(q_i, v') \xrightarrow{\mathsf{dec}_j}_{\dagger} (q_{i+1}, v_{i+1})$ is an exact transition. Since $v_i \leq v'$, we get (8.3).

- If $l_i = \mathsf{ifz}_j$, then we need to show $v_i(j) = 0$ in order to get (8.3). For this to hold, it is enough to notice that φ_{ifz1} and φ_{ifz2} ensure that for every inc_j followed by a ifz_j there is a dec_j, and this occurs before ifz_j. Hence, every time we increase $v_k(j)$ by one for some $k < i$, we also decrease it by one before the zero test. All other decreases do not alter the value of the counter.

We conclude that, since the infinite accepting run problem for ICAs is undecidable and Π^0_1-complete, satisfiability for $\mathsf{CLTL}[\mathsf{F}]$ over $(\mathbb{N}, <, \equiv)$ and $(\mathbb{Z}, <, \equiv)$ is also undecidable and Π^0_1-hard. □

Remark 8.5. In the formula φ_C we only use unary temporal operators, i.e., the until modality U never appears. This is a strict fragment of LTL, sometimes referred to as $\mathsf{unaryLTL}$ in the literature. Note also that constraints of the form $r < \mathsf{F}s$ or $\mathsf{F}r < s$ never appear in φ_C, we only use a non-local equality constraint $r \equiv \mathsf{F}s$ in (8.2). We can then state a more precise result: *Satisfiability for* $\mathsf{unaryLTL}$ *with local constraints over the signature* $\{<, \equiv\}$ *and non-local constraints over the signature* $\{\equiv\}$ *is undecidable for the concrete domains* $(\mathbb{Z}, <, \equiv)$ *and* $(\mathbb{N}, <, \equiv)$

Remark 8.6. Going in a different direction, we could instead substitute all local constraints with non-local constraints, and obtain a formula equivalent to φ_C in the following way: Using the formula

$$\varphi_{\mathsf{mon}} := \mathsf{G} \bigwedge_{i=1}^{n} [\neg(\mathsf{F}r_i < r_i) \wedge \neg(\mathsf{F}s_i < s_i)] \ ,$$

we make sure that the sequence of values assigned to the registers variables is non decreasing. We substitute the formulas φ_{inc} and φ_{dec} by the following:

$$\varphi'_{\mathsf{inc}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((\mathsf{inc}_i \rightarrow \neg r_i \equiv \mathsf{F}r_i) \wedge (\neg\mathsf{inc}_i \rightarrow r_i \equiv \mathsf{F}r_i) \Big) \ ,$$

$$\varphi'_{\mathsf{dec}} := \mathsf{G} \bigwedge_{i=1}^{n} (\neg\mathsf{dec}_i \rightarrow s_i \equiv \mathsf{F}s_i) \ .$$

Note that, given a non-decreasing sequence $(a_j)_{j\in\mathbb{N}}$, asking that the value of some a_j is the same of a later member of the sequence a_{j+k} implies that the sequence is constant in that interval, in particular $a_j = a_{j+1}$ holds. At the same time, asking that $a_j = a_{j+k}$ does not hold for any $k > 0$, implies in particular that it does not hold for $k = 1$ and therefore $a_j < a_{j+1}$.

Using these new formulas we can show that: *Satisfiability for* $\mathsf{unaryLTL}$ *with only non-local constraints of the form* $r{*}\mathsf{F}s$, *or* $\mathsf{F}r{*}s$, *for* $* \in \{<, \equiv\}$ *is undecidable for the concrete domains* $(\mathbb{Z}, <, \equiv)$ *and* $(\mathbb{N}, <, \equiv)$.

Since LTL can be seen as a fragment of CTL^* and ECTL^*, the above undecidability results also apply to such logics extended with this new kind of constraints.

8.2 Regaining Decidability by Restricting the Use of Non-Local Constraints

Looking at the proof of Theorem 8.2, one can see how the use of non-local equality constraints is essential for the reduction: The correctness of the zero tests of the ICAs needs to be guaranteed and we do it by matching the identifiers of increase and decrease instructions using constraints of the form $r \equiv \mathsf{F}s$. Since we cannot predict how many computation steps separate an increase to its matching decrease, we necessarily need to use a non-local constraint.

In this section we show that, if this matching is not possible, i.e., if we limit the use of non-local constraints to the order relation $<$, decidability is regained. From now on we refer to constraints of the form

$$r < \mathsf{F}s \ , \ \mathsf{F}r < s \ , \ \text{and} \ \mathsf{F}r < \mathsf{F}s \ ,$$

with $r, s \in \mathsf{Reg}$ as to **non-local order constraints**. Let $\mathcal{Z}$ be the structure over the signature $\sigma = \{<, \equiv, (\equiv_a)_{a \in \mathbb{Z}}, (\equiv_{a,b})_{0 \leq a < b}\}$ from (1.2) on page 7. We show the following:

Theorem 8.7. *Given a* CLTL *formula* φ *with local constraints over* σ *and non-local order constraints, one can compute an* CLTL *formula* $\hat{\varphi}$*, with only local constraints over* σ*, such that* φ *is* $\mathcal{Z}$*-satisfiable if and only if* $\hat{\varphi}$ *is* $\mathcal{Z}$*-satisfiable.*

The idea is the following: by using some auxiliary register variables and extra atomic propositions, it is possible to replace the non-local order constraints by local constraints without changing the set of Kripke paths satisfying the original formula. Suppose that $\vartheta = (r < \mathsf{F}s)$ appears in φ. On a potential model $\mathbf{P}$ of φ, whenever ϑ holds, the current value of r can be matched with a larger value of s in some future node. What we want to do, is store the current value of r in some auxiliary variable, and propagate it until we find a match. Of course ϑ could hold infinitely often, think for instance of the formula $\mathsf{GF}(r < \mathsf{F}s)$, and we cannot store infinitely many values of r. Nonetheless there is a solution, which allows, using only two auxiliary variables, to check that the constraint is satisfied. The first step is to replace any occurrence of ϑ in φ by a fresh atomic proposition p. Then we use two fresh register variables: a_c which stores the value of r which we are *currently* trying to match to some future value of s, and a_m, which stores the *maximal* values of r which we have encountered so far that needs to be matched to some future s. Then we use two additional atomic propositions, q_c and q_m, to keep track of whether the checking processes for a_c and a_m are active. So whenever p occurs, we initialize the checking processes and keep checking whether we find a match for a_c, and updating a_m whenever a larger value of r occurs in a node marked by p. If a_c is matched to a larger value of s, then we can transfer the

	$\mathsf{P}(0)$	$\mathsf{P}(1)$	$\mathsf{P}(2)$	$\mathsf{P}(3)$	$\mathsf{P}(4)$	$\mathsf{P}(5)$	$\mathsf{P}(6)$
prop	p, q_c, q_m	p, q_c, q_m	q_c, q_m	q_c		p, q_c, q_m	
r	3	5	6	6	4	3	5
s	2	2	2	5	7	2	4
a_c	3	3	3	5	$\star$	3	$\star$
a_m	3	5	$\boxed{5}$	$\star$	$\star$	3	$\star$

Table 8.1: We have replaced every occurrence of the constraint $(r < \mathsf{F}s)$ by the fresh propositional variable p. When the value of a_c or a_m is circled, it is matched to a larger value of s in the next position. When the value of a_m is inside a square, it means that we transfer the value of a_m to the following position of a_c. The star signifies that the value of that variable at that position is irrelevant. Notice how, whenever both q_c and q_m do not hold anymore, all the previous values of r in positions marked by p have been matched to future values of s.

value of a_m to a_c and stop the checking process for a_m, until p holds again. See Table 8.1 for an example.

Remark 8.8. One might think that a simpler approach is possible: using a single auxiliary variable a one could store the value of r at the first occurrence of $r < \mathsf{F}s$ and update it whenever the constraint $r < \mathsf{F}s$ appears again by setting a to the maximum between the current value of r and the previous value of a. Then one could simply set $\mathsf{G}[p \to \mathsf{F}(a < \mathsf{X}s)]$ as a final condition (where again, we use the fresh propositional variable p to mark the nodes where the non-local constraint holds). This method would indeed fail if the value of a were to be always increased before we were able to match it to a larger value of s. Take for instance the sequence of values:

	$\mathsf{P}(0)$	$\mathsf{P}(1)$	$\mathsf{P}(2)$	$\mathsf{P}(3)$	$\mathsf{P}(4)$	...
r	3	4	5	6	7	...
s	1	2	3	4	5	...

This sequence clearly satisfies $\mathsf{G}(r < \mathsf{F}s)$. Using only one auxiliary variable a as explained above one would obtain the following valuation:

	$\mathsf{P}(0)$	$\mathsf{P}(1)$	$\mathsf{P}(2)$	$\mathsf{P}(3)$	$\mathsf{P}(4)$	...
a	3	4	5	6	7	...

which would not satisfy the final condition $\mathsf{GF}(a < \mathsf{X}s)$. Using two auxiliary variables, instead, a first one can be constantly updated to new larger values of r, while the second one is used to check whether a match happens infinitely often.

Let us now show this construction in detail:

Proof of Theorem 8.7. We can assume that φ is in negation normal form. First of all notice that we can substitute in φ any constraint of the form $\mathsf{F}r < \mathsf{F}s$ with the following formula

$$\mathsf{XF}[(r < s) \vee (r < \mathsf{F}s) \vee (\mathsf{F}r < s)]$$

without changing the semantics. We can then assume that the non-local constraints in φ are of the kind $(r < \mathsf{F}s)$, $(\mathsf{F}r < s)$, or their negations.

It is enough to show that we can remove one of such constraints ϑ to obtain our result.

Case 1. Suppose $\vartheta = (r < \mathsf{F}s)$. Then let $a_c, a_m \in \mathsf{Reg}$ be two auxiliary register variables not appearing in φ, and $p, q_c, q_m \in \mathbb{P}$ be fresh atomic propositions. We define $\hat{\varphi} = \varphi[\vartheta \mapsto p] \wedge \psi$, where $\varphi[\vartheta \mapsto p]$ is obtained from φ by substituting any positive occurrence of ϑ with p, and ψ is the conjunction of ψ_0 to ψ_9 defined in the following. Note that, the values of a_c and a_m are always kept so to satisfy $a_c \leq a_m$. Requirements for the initial position:

$$\psi_0 = [\neg p \rightarrow \neg(q_c \vee q_m)] \wedge [p \rightarrow (a_c = r \wedge a_m = r)] \ .$$

Whenever p occurs, start (or continue) checking a_c and a_m:

$$\psi_1 = \mathsf{G}[p \rightarrow (q_c \wedge q_m)] \ .$$

Do not start checking a_c unless solicited by p:

$$\psi_2 = \mathsf{G}[(\neg q_c \wedge \neg \mathsf{X}p) \rightarrow \neg \mathsf{X}q_c] \ .$$

Do not start checking a_m unless solicited by p:

$$\psi_3 = \mathsf{G}[(\neg q_m \wedge \neg \mathsf{X}p) \rightarrow \neg \mathsf{X}q_m] \ .$$

If the checking process is initiated, set a_c and a_m to the value of r:

$$\psi_4 = \mathsf{G}[(\neg q_c \wedge \neg q_m \wedge \mathsf{X}p) \rightarrow (\mathsf{X}(a_c \equiv r) \wedge \mathsf{X}(a_m \equiv r))] \ .$$

If we are checking a_c, and $a_c < \mathsf{X}s$ is not satisfied, propagate the value of a_c and keep checking. Since $a_c \not< \mathsf{X}s$ implies $a_m \not< \mathsf{X}s$, if the checking process is active on a_m it should be kept active: If p does not hold in the following state we simply propagate a_m, if p does hold, then we set $\mathsf{X}a_m$ to the maximal value[2] between the current value of a_m, and the value of r in the following position. If the checking

[2] Note that $a = \max\{b, c\}$ can be expressed as $(b \leq c \wedge a \equiv c) \vee a \equiv b$.

process for a_m was not active, but p holds in the next state, we set the value of a_m to $\max\{a_c, r\}$, to keep $a_c \leq a_m$ true.

$$\begin{aligned}\psi_5 = \mathsf{G}[(q_c \wedge \neg(a_c < \mathsf{X}s)) \rightarrow \quad & \mathsf{X}q_c \wedge (a_c \equiv \mathsf{X}a_c) \\ \wedge\ & (q_m \wedge \mathsf{X}p) \rightarrow (\mathsf{X}a_m \equiv \max\{a_m, \mathsf{X}r\}) \\ \wedge\ & (q_m \wedge \neg\mathsf{X}p) \rightarrow (\mathsf{X}q_m \wedge (\mathsf{X}a_m \equiv a_m)) \\ \wedge\ & (\neg q_m \wedge \mathsf{X}p) \rightarrow \mathsf{X}(a_m \equiv \max\{a_c, r\})]\ .\end{aligned}$$

In case we are checking a_c but not a_m, and the constraint is satisfied, we either stop checking if $\mathsf{X}p$ does not hold, or we re-initialize both checking process if $\mathsf{X}p$ holds:

$$\psi_6 = \mathsf{G}[(q_c \wedge \neg q_m \wedge a_c < \mathsf{X}s) \ \rightarrow\ (\mathsf{X}(\neg p \wedge \neg q_c) \vee \mathsf{X}(p \wedge\ a_c \equiv r\ \wedge\ a_m \equiv r))]\ .$$

If both checking processes are active, and $a_c < \mathsf{X}s$ but $a_m \not< \mathsf{X}s$, then we transfer the value of a_m to a_c. If $\mathsf{X}p$ does not hold we stop checking a_m, while, if $\mathsf{X}p$ holds, we keep the checking process on a_m active and set $\mathsf{X}a_m \equiv \max\{a_m, \mathsf{X}r\}$ in order to keep $a_c \leq a_m$:

$$\begin{aligned}\psi_7 = \mathsf{G}[(q_c \wedge q_m \wedge a_c < \mathsf{X}s \wedge \neg a_m < \mathsf{X}s) \rightarrow \quad & (\mathsf{X}a_c \equiv a_m) \wedge \mathsf{X}q_c \\ \wedge\ & \neg\mathsf{X}p \rightarrow \neg\mathsf{X}q_m \\ \wedge\ & \mathsf{X}p \rightarrow \mathsf{X}a_m \equiv \max\{a_m, \mathsf{X}r\}]\ .\end{aligned}$$

If both a_m and a_c are smaller than the next value of s, we can stop all checking processes, unless p holds again in the next position, in which case we set a_c and a_m to the value of r and re-initialize the checking procedure:

$$\begin{aligned}\psi_8 = \mathsf{G}[(q_c \wedge q_m \wedge a_c < \mathsf{X}s \wedge a_m < \mathsf{X}s) \rightarrow \quad & \neg\mathsf{X}p \rightarrow \mathsf{X}(\neg q_m \wedge \neg q_c) \\ \wedge\ & \mathsf{X}p \rightarrow \mathsf{X}(a_m \equiv r \wedge a_c = r)]\ .\end{aligned}$$

We add the acceptance condition: either $a_c < \mathsf{X}s$ holds infinitely often (all $(r < \mathsf{F}s)$ are eventually satisfied) or at some point we stop checking and q_c never holds again (the last constraint is satisfied).

$$\psi_9 = \mathsf{GF}(a_c < \mathsf{X}s) \vee \mathsf{FG}\neg q_c\ .$$

Note that $\psi_5 \wedge \psi_9$ implies $\mathsf{G}[q_c \rightarrow (q_c\,\mathsf{U}\,(a_c < \mathsf{X}s))]$, a perhaps more intuitive final condition.

To complete the proof of *Case 1*, we have to show that there exists a $\mathcal{Z}$-Kripke path P such that $\mathsf{P} \models \varphi$ if and only if there exists a $\mathcal{Z}$-Kripke path $\hat{\mathsf{P}}$ such that $\hat{\mathsf{P}} \models \varphi[\vartheta \mapsto p] \wedge \bigwedge_{i=0}^{9} \psi_i$.

First we prove the direction ($\Rightarrow$). Suppose $\mathsf{P} = (\mathcal{Z}, \mathcal{P}, \gamma) \models \varphi$, where $\mathcal{P} = (\mathbb{N}, \rightarrow, \rho)$ is a Kripke-path. To build $\hat{\mathsf{P}}$ we recursively define two extensions of the labeling and valuation function respectively: $\hat{\rho}$ and $\hat{\gamma}$. In the following let us write $\gamma_i(r)$ instead of $\gamma(i, r)$. As basic step we set $\hat{\rho}(0) = \rho(0) \cup \{p, q_c, q_m\}$ and $\hat{\gamma}_0(a_c) = \hat{\gamma}_0(a_m) = \gamma_0(, r)$ if $(\mathsf{P}, 0) \models \vartheta$. Otherwise $\hat{\rho}(0) = \rho(0)$ and $\hat{\gamma}_0(a_c)$ and $\hat{\gamma}_0(a_m)$ are chosen arbitrarily.

Suppose now we have defined $\hat{\gamma}$ and $\hat{\rho}$ up to $i - 1$, we set

(a) $\hat{\rho}(i) = \rho(i) \cup \{p, q_c, q_m\}$ if $(\mathsf{P}, i) \models \vartheta$,

while, if $(\mathsf{P}, i) \not\models \vartheta$, then

(b) $\hat{\rho}(i) = \rho(i) \cup \{q_c, q_m\}$ if $q_c, q_m \in \hat{\rho}(i-1)$ and $\hat{\gamma}_{i-1}(a_c) \not< \gamma_i(s)$,

(c) $\hat{\rho}(i) = \rho(i) \cup \{q_c\}$ if $q_c \in \hat{\rho}(i-1), q_m \notin \hat{\rho}(i-1)$ and $\hat{\gamma}_{i-1}(a_c) \not< \gamma_i(s)$ or if $q_c, q_m \in \hat{\rho}(i-1)$ and $\hat{\gamma}_{i-1}(a_c) < \gamma_i(s)$ but $\hat{\gamma}_{i-1}(a_m) \not< \gamma_i(s)$,

(d) $\hat{\rho}(i) = \rho(i)$ otherwise.

We define $\hat{\gamma}(t) = \gamma(t)$ for all $t \in \mathsf{Reg}_{\varphi}$, and according to the following table for $t \in \{a_c, a_m\}$.

	$\hat{\rho}(i\text{-}1)$	$(\mathsf{P}, i) \models \vartheta$	$\hat{\gamma}_{i-1}(a_c) < \gamma_i(s)$	$\hat{\gamma}_{i-1}(a_m) < \gamma_i(s)$	$\hat{\gamma}_i(a_c)$	$\hat{\gamma}_i(a_m)$
1	q_c, q_m	yes	no	$\star$	$\hat{\gamma}_{i-1}(a_c)$	$\max\{\hat{\gamma}_{i-1}(a_m), \gamma_i(r)\}$
2	q_c, q_m	no	no	$\star$	$\hat{\gamma}_{i-1}(a_c)$	$\hat{\gamma}_{i-1}(a_m)$
3	q_c, q_m	$\star$	yes	no	$\hat{\gamma}_{i-1}(a_m)$	$\max\{\hat{\gamma}_{i-1}(a_m), \gamma_i(r)\}$
4	q_c, q_m	$\star$	yes	yes	$\gamma_i(r)$	$\gamma_i(r)$
5	q_c	no	$\star$	$\star$	$\hat{\gamma}_{i-1}(a_c)$	$\hat{\gamma}_{i-1}(a_c)$
6	q_c	yes	no	$\star$	$\hat{\gamma}_{i-1}(a_c)$	$\max\{\hat{\gamma}_{i-1}(a_c), \gamma_i(r)\}$
7	q_c	yes	yes	$\star$	$\gamma_i(r)$	$\gamma_i(r)$
8	$\emptyset$	$\star$	$\star$	$\star$	$\gamma_i(r)$	$\gamma_i(r)$

Table 8.2: In the first column we write whether q_c and q_m belong to $\hat{\rho}(i-1)$ and $\star$ means that the value is non influential.

Note the following facts from Table 8.2:

$$(\mathsf{P}, i) \models \vartheta \quad \text{implies} \quad \gamma_i(r) \in [\hat{\gamma}_i(a_c), \hat{\gamma}_i(a_m)] \; , \tag{8.4}$$

$$(\mathsf{P}, i) \models (q_c \wedge \neg a_c < \mathsf{X}s) \text{ implies } \hat{\gamma}_{i-1}(a_c) = \hat{\gamma}_i(a_c) \ . \tag{8.5}$$

Now, let $\hat{\mathsf{P}}$ be the $\mathcal{Z}$-Kripke path having $\hat{\mathcal{P}} = (\mathbb{N}, \to, \hat{\rho})$ as underlying KP and $\hat{\gamma}$ as valuation function. It is easy to see that $\mathsf{P} \models \varphi$ implies $\hat{\mathsf{P}} \models \varphi[\vartheta \mapsto p]$. This can be done by induction, using the fact that by definition of $\hat{\rho}$ in point (a), $(\mathsf{P}, i) \models \vartheta$ implies $(\hat{\mathsf{P}}, i) \models p$. The two formulas are otherwise identical, and $\hat{\rho}$ and $\hat{\gamma}$ coincide with ρ and γ except on the fresh atomic propositions and register variables which do not appear in φ.

The fact that $\hat{\mathsf{P}} \models \psi_i$ for all $i = 0, \ldots, 9$ can be derived from the definitions of $\hat{\rho}$ and $\hat{\gamma}$ as follows:

- ψ_0 is satisfied by the definition of $\hat{\gamma}_0$ and $\hat{\rho}_0$.

- ψ_1, ψ_2 and ψ_3 are easily verified from points (a)-(d) of the definition of $\hat{\rho}$. In fact, p is only added to $\hat{\rho}(i)$ together with q_c and q_m, furthermore q_c and q_m are only added to $\hat{\rho}(i)$ if they also belong to $\hat{\rho}(i-1)$ or if $p \in \hat{\rho}(i)$.

- ψ_4 is directly implied by the last row of Table 8.2.

- To prove that ψ_5 to ψ_8 are satisfied, one needs to check that $\hat{\rho}$ and $\hat{\gamma}$ have been defined appropriately. Let us do it, as an example, for ψ_5.

  Assume that $\alpha = (q_c \wedge \neg(a_c < \mathsf{X}s))$ holds, then $\mathsf{X}q_c$ is true as a consequence of points (b) and (c). To satisfy $a_c \equiv \mathsf{X}a_c$, we should have $\hat{\gamma}_i(a_c) = \hat{\gamma}_{i-1}(a_c)$, as guaranteed by rows 1,2,5,6 of Table 8.2.

  The third conjunct assumes $(q_m \wedge \mathsf{X}p)$, additionally to the original assumption α. Since $p \in \hat{\rho}(i)$ holds if and only if $(\mathsf{P}, i) \models \vartheta$, these circumstances are described by line 1 of Table 8.2, where the value for $\hat{\gamma}_i(a_m)$ is chosen appropriately.

  The forth conjunct assumes $(q_m \wedge \neg\mathsf{X}p)$ in addition to α. $\mathsf{X}q_m$ is guaranteed by (b), and the constraint on a_m holds by the second line of Table 8.2.

  In the fifth conjunct, with α and $\neg q_m \wedge \mathsf{X}p$ as assumptions, we are in the situation described by line 6 of Table 8.2, which again sets the value for $\hat{\gamma}_i(a_m)$ correctly.

  The fact that ψ_6 to ψ_8 are satisfied can be proved analogously.

- Let us now take a look at the final condition, we want to show that $\hat{\mathsf{P}} \models \psi_9$. Suppose $(\mathsf{P}, i) \models \vartheta$. Then, by (a), $q_c \in \hat{\rho}(i)$. Points (b) and (c) imply that $q_c \in \hat{\rho}(j)$ for all $i \leq j \leq k$ where k is a (possibly non existing) later position such that $(\hat{\mathsf{P}}, k) \models a_c < \mathsf{X}s$. Using (8.5) we can deduce that the value $\hat{\gamma}_j(a_c)$ is kept constant until such position k, that is

$$\hat{\gamma}_i(a_c) = \hat{\gamma}_{i+1}(a_c) = \cdots = \hat{\gamma}_k(a_c) \ .$$

Now, because $(\mathfrak{P}, i) \models \vartheta$, by (8.4), $\hat{\gamma}_i(a_c) \leq \gamma_i(r)$. Since $\vartheta = r < \mathsf{F}s$, we know that there exists a position $k \geq i$ such that $\gamma_i(r) < \gamma_{k+1}(s)$. Using all the above facts we deduce that there exists k such that

$$\hat{\gamma}_k(a_c) \leq \hat{\gamma}_i(r) < \hat{\gamma}_{k+1}(s) \ ,$$

that is $(a_c < \mathsf{X}s)$ holds at position k. Therefore, if $(\mathfrak{P}, i) \models \vartheta$ is true for infinitely many $i \in \mathbb{N}$, the fist disjunct of ψ_9 will be satisfied.

A similar reasoning can be applied to the case where there exists $i \in \mathbb{N}$ such that $(\mathfrak{P}, i) \models \vartheta$ but $(\mathfrak{P}, j) \not\models \vartheta$ for all $j > i$, to obtain that from some position on $\neg q_c$ always holds, satisfying the second disjunct of ψ_9.

Let us now show the other direction ($\Leftarrow$) of the implication. Suppose there exists some $\mathcal{Z}$-KP $\mathfrak{P}$ which is a model for $\hat{\varphi} = \varphi[\vartheta \mapsto p] \wedge \bigwedge \psi_i$. We claim $\mathfrak{P} \models \varphi$. To show this it is enough to prove that $(\mathfrak{P}, i) \models p \Rightarrow (\mathfrak{P}, i) \models \vartheta$. If p holds on some node i of $\mathfrak{P}$, then q_c and q_m also hold. Additionally we can deduce from ψ_0 and ψ_4 to ψ_8, that $\gamma_i(r) \in [\gamma_i(a_c), \gamma_i(a_m)]$. Then, according to ψ_5, both q_c and q_m are kept true until at some position j $(a_c < \mathsf{X}s)$ holds (such node exists by ψ_9). Until then, a_c is kept constant and a_m can only increase or stay the same, so $\gamma_i(r) \in [\gamma_j(a_c), \gamma_j(a_m)]$ also holds. At this point, either both $a_c < \mathsf{X}s$ and $a_m < \mathsf{X}s$ hold, in which case $(\mathfrak{P}, i) \models (r < \mathsf{F}s)$ and we have concluded our proof, or only $a_c < \mathsf{X}s$ holds. If this is the case, ψ_7 insures that the value of a_m is transfered to a_c and that the checking process q_c is kept active. Again, using ψ_5 and ψ_9, we can guarantee that there exists a later node k, where $a_c < \mathsf{X}s$ holds, and until then $\gamma_i(r) < \gamma_k(a_c)$. Therefore, also in this case, we have found that $(\mathfrak{P}, i) \models (r < \mathsf{F}s)$.

Case 2. The case where $\vartheta = (\mathsf{F}s < r)$ can be dealt with very similarly as Case 1. The only difference is that we have to match the current value of r with a *smaller* future value of s. And this can be easily done by slightly modifying ψ_5 to ψ_9 to fit this situation.

Case 3. Suppose $\vartheta = \neg(r < \mathsf{F}s)$. The semantics of this constraint are the following: Given a $\mathcal{Z}$-Kripke path $\mathfrak{P} = (\mathcal{Z}, \mathcal{P}, \gamma)$, $(\mathfrak{P}, i) \models \vartheta$ if and only if for all $j \geq i$, $\gamma(i, r) \geq \gamma(j, s)$. This is the same as saying that $\gamma(i, r)$ should be greater or equal than the maximum value of $\gamma(j, s)$ for $j \geq i$. The global nature of this constraint allows us to use only one auxiliary variable a_m which will store the *minimum* value of r for which the constraint needs to be satisfied, and one fresh propositional variable q that will record whether the checking process has started. Again we define $\varphi' = \varphi[\vartheta \mapsto p] \wedge \psi$. Here ψ is the conjunction of the formulas ψ_0 to ψ_5.

In the initial position, if p holds, we set the value of a_c to r:

$$\psi_0 = p \rightarrow a_c \equiv r \ .$$

If p holds, we start the checking process, which never ends. If p never holds, then the process q is never initiated:

$$\psi_1 = \mathsf{G}(p \rightarrow \mathsf{G}q) \land (\neg q \,\mathsf{U}\, p) \ .$$

Whenever p holds for the first time (checking process q has not been activated before), we set a_c equal to r:

$$\psi_2 = \mathsf{G}(\neg q \land \mathsf{X}p \rightarrow \mathsf{X}a_c \equiv \mathsf{X}r) \ .$$

Whenever p holds, if the checking process q had already started, we update the value of a_c to r, if this is smaller, and otherwise keep it constant:

$$\psi_3 = \mathsf{G}(q \land \mathsf{X}p \rightarrow \mathsf{X}a_c \equiv \min\{a_c, \mathsf{X}r\}) \ .$$

If p does not hold we simply propagate the value of a_c:

$$\psi_4 = \mathsf{G}(\mathsf{X}\neg p \rightarrow \mathsf{X}a_c \equiv a_c) \ .$$

Whenever the checking process q is ongoing, we make sure that a_c is greater or equal than the value of s in the following position:

$$\psi_5 = \mathsf{G}(q \rightarrow a_c \geq \mathsf{X}s) \ .$$

Using a similar but simpler procedure than the one in Case 1, we can show that φ is $\mathcal{Z}$-satisfiable if and only if so is φ'.

Case 4. The last case, $\vartheta = \neg(\mathsf{F}s < r)$ can be dealt with in the same way as for Case 3. This concludes the proof.

□

Remark 8.9. The translation we just presented from CLTL[F] with only non-local order constraints to CLTL is in LOGSPACE. Since satisfiability for CLTL with local constraints over $\mathcal{Z}$ is a PSPACE complete problem ([19]), then so is satisfiability for CLTL[F] with only non-local order constraints.

Chapter 9

Conclusion and Final Remarks

In this work we have extended the notion of temporal logic with local constraints as introduced in [17, 7] from CLTL and CCTL^* to CECTL^*. We have proved a general result stating that satisfiability of CECTL^* with constraints over any domain $\mathcal{D}$ which (i) is *negation-closed* and (ii) *satisfies the EHD-property* is decidable.

We have shown that the domains $(\mathbb{Z}, \equiv, <)$ and $(\mathbb{N}, \equiv, <)$ satisfy these properties, even if extended with constant- and periodicity-constraints, proving decidability of CECTL^* with constraints over such structures. This implies the same results for CCTL^*, whose satisfiability over integer domains with order- and equality-constraints had been open since it was first asked in [19].

We have also successfully applied this result to other domains, concentrating on classes of "tree-like" structures as semi-linear orders, ordinal trees and trees of a fixed height.

At the same time we have explored the limits of this method, showing that it cannot be applied to the infinite binary tree with the prefix and incomparability relation $\mathcal{T} = ([0,1]^*, <, \perp)$: Despite the fact that both CLTL and CCTL^* with constraints over $\mathcal{T}$ have a decidable satisfiability problem (as shown very recently in [18]), $\mathcal{T}$ does not have the EHD-property.

Successively, we have considered the idea (proposed in [17, 8]) to allow the use of non-local constraints into the logic, and discovered that this leads to undecidability of CLTL (and therefore CCTL^* and CECTL^*) with constraints over the domains $(\mathbb{Z}, \equiv, <)$ and $(\mathbb{N}, \equiv, <)$. On the positive side, we showed that restricting the use of non local constraints, allowing the ones involving order and discarding the ones involving equality, permits to regain decidability. We have established the result for CLTL, answering only partially the question in [8], which was stated for certain fragments of CCTL^*. We leave decidability for CCTL^* over $(\mathbb{Z}, \equiv, <)$ with non-local order constraints as an open problem, that we would like to explore in the future.

The other - most evident - question which we leave open, is the one concerning the complexity of the satisfiability problem for CECTL^*, or perhaps more interestingly, for CCTL^*, with constraints over the integers.

The lack of complexity bounds is due to the fact that we rely on the decidability result established in [5] for satisfiability of $\mathsf{WMSO{+}B}$ over infinite node-labeled trees, in which the authors make no statements regarding the complexity of their procedure.

At the same time, we believe that our decidability result, whose upside is its generalized nature, may not be the most effective way to devise an efficient decidability procedure for the specific case of the domain $(\mathbb{Z}, \equiv, <)$.

The reason behind this statement is that, to establish whether a given CECTL^*-formula is satisfiable using our method, we have to check whether a *constraint graph* satisfies a $\mathsf{WMSO{+}B}$-formula. This constraint graph is generated from an ECTL^* formula, and given this fact, one could assume that it has certain regular properties. Instead of doing this, we simply check that this graph allows a homomorphism to $(\mathbb{Z}, \equiv, <)$, making no assumptions on its structure. We believe that it might be more efficient to factor in these assumptions and devise another procedure, which perhaps could avoid the use of $\mathsf{WMSO{+}B}$ and allow us to derive some complexity bounds.

Finally, we would like to remark that our results show once more the deep connection between the constraint satisfaction problem (CSP) for a structure $\mathcal{D}$ and the satisfiability problem for logics with constraints over $\mathcal{D}$.

The completion property from [4] to show decidability of CLTL, or the ω-admissibility criterion from [34] used for the description logic $\mathcal{ALC}$, relate local satisfiability of a constraint satisfaction problem to global satisfiability. In some sense this is the same idea behind Lemma 6.10, in which we establish a compactness result for the CSP for semi-linear orders.

In our work, instead, the connection is established through logic: A domain $\mathcal{D}$ has the property $\mathsf{EHD}(\mathcal{L})$ if the logic $\mathcal{L}$ is able to "solve" its CSP problem by distinguishing those constraint systems which admit a satisfying assignment, and those who do not.

Bibliography

[1] R. Alur and T. A. Henzinger. A really temporal logic. In *Proc. FOCS 1989*, pages 164–169. IEEE Computer Society, 1989.

[2] R. Alur and T. Henzinger. Real-time logics: complexity and expressiveness. In *Information and Computation*, vol. 104, 390–401, 1993.

[3] F. Baader and P. Hanschke. A Scheme for Integrating Concrete Domains into Concept Languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Volume 1, pages 452–457, 1991.

[4] P. Balbiani and J. Condotta. Computational Complexity of Propositional Linear Temporal Logics Based on Qualitative Spatial or Temporal Reasoning. In *Proceedings of the 4th International Workshop on Frontiers of Combining Systems (FroCoS '02)*, pages 162–176. Springer-Verlag, 2002.

[5] M. Bojańczyk and S. Toruńczyk. Weak MSO+U over infinite trees. In *Proc. STACS 2012*, vol. 14 of *LIPIcs*, 648–660. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.

[6] M. Bojańczyk and S. Toruńczyk. Weak MSO+U over infinite trees (long version). Available at http://www.mimuw.edu.pl/~bojan/papers/wmsou-trees.pdf.

[7] L. Bozzelli and R. Gascon. Branching-time temporal logic extended with qualitative Presburger constraints. In *Proc. LPAR 2006*, LNCS 4246, 197–211. Springer, 2006.

[8] L. Bozzelli and S. Pinchinat. Verification of Gap-order Constraint Abstractions of Counter Systems. In *Theor. Comput. Sci.*, Vol. 523, pages 1–36. Elsevier, 2014

[9] C. Carapelle, S. Feng, O. Fernández and K. Quaas. Satisfiability for MTL and TPTL over Non-monotonic Data Words. In *Proceedings of Language*

and Automata Theory and Applications 2014, LNCS 8370, pages 248–259. Springer, 2014.

[10] C. Carapelle, S. Feng, A. Kartzow, and M. Lohrey. Satisfiability of ECTL* with tree constraints. In *Computer Science – Theory and Applications*, LNCS 9139, pages 94–108. Springer, 2015. http://dx.doi.org/10.1007/978-3-319-20297-6_7.

[11] C. Carapelle, A. Kartzow, and M. Lohrey. Satisfiability of CTL* with constraints. In *Proc. CONCUR 2013*, LNCS 8052, pages 455–469. Springer, 2013.

[12] C. Carapelle, A. Kartzow and M. Lohrey. Satisfiability of ECTL* with constraints. Accepted for publication in *Journal of Computer and System Sciences*, currently available at http://www.eti.uni-siegen.de/ti/veroeffentlichungen/ectl-with-constraints.pdf.

[13] K. Čerāns. Deciding properties of integral relational automata. In *Proc. ICALP 1994*, LNCS 820, pages 35–46. Springer, 1994.

[14] B. Courcelle. Monadic second-order definable graph transductions: a survey *Theor. Comput. Sci.*, 126:53–75, 1994.

[15] M. Dam. CTL* and ECTL* as fragments of the modal mu-calculus. *Theor. Comput. Sci.*, 126(1):77–96, 1994.

[16] S. Demri. LTL over Integer Periodicity Constraints. *Foundations of Software Science and Computation Structures*, 2987, pages 121–135. Springer, 2004.

[17] S. Demri and D. D'Souza. An automata-theoretic approach to Constraint LTL. In *Information and Computation* vol. 205, 3, pages 380–415. Academic press, 2007.

[18] S. Demri and M. Deters. Temporal logics on strings with prefix relation. In *Journal of Logic and Computation*, 2015. http://logcom.oxfordjournals.org/content/early/2015/06/04/logcom.exv028.abstract.

[19] S. Demri and R. Gascon. Verification of qualitative $\mathbb{Z}$ constraints. *Theor. Comput. Sci.*, 409(1):24–40, 2008.

[20] S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3), 16:1–16:30, 2009.

[21] C. Ding, D. Pei and A. Salomaa, Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography. *Word Scientific*, 1996. https://books.google.com/books?id=RQLtCgAAQBAJ,

[22] M. Droste Structure of partially ordered sets with transitive automorphism groups. *Memoirs of the American Mathematical Society*, 334, 1985

[23] H.D. Ebbinghaus and J. Flum. Finite Model Theory. Perspectives in Mathematical Logic Series, Springer, 1995.

[24] S. Feng, M. Lohrey and Karin Quaas. Path-Checking for MTL and TPTL over Data Words. Accepted for publication in *Proceedings of DLT 2015*. Currently available at http://arxiv.org/abs/1412.3644.

[25] R. Gascon. An automata-based approach for CTL* with constraints. *Electr. Notes Theor. Comput. Sci.*, 239:193–211, 2009.

[26] W. H. Gottschalk Choice functions and Tychonoff's theorem. *Proceedings of the American Mathematical Society*, 2:172, 1951.

[27] D. Janin and I. Walukiewicz On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic. In *Proc. CONCUR 1996*, LNCS 1119, 263–277. Springer, 1996.

[28] A. Kartzow and T. Weidner Model checking Constraint LTL over Trees. aivalable at http://arxiv.org/abs/1504.06105

[29] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[30] D. Kozen A finite model theorem for the propositional μ-calculus. *Studia Logica* 47(3):233–241, 1988.

[31] C. Lutz. Description logics with concrete domains-a survey. In *Advances in Modal Logic 4*, pages 265–296. King's College Publications, 2003.

[32] C. Lutz. Combining interval-based temporal reasoning with general TBoxes. *Artificial Intelligence*, 152(2):235 – 274, 2004.

[33] C. Lutz. NEXPTIME-complete description logics with concrete domains. *ACM Trans. Comput. Log.*, 5(4):669–705, 2004.

[34] C. Lutz and M. Milicic. A tableau algorithm for description logics with concrete domains and general TBoxes. *J. Autom. Reasoning*, 38(1-3):227–259, 2007.

[35] J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. *Proc. FOSSACS 2006*, LNCS 3921, pages 217–230. Springer, 2006.

[36] A. Pnueli. The Temporal Logic of Programs. *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS '77, pages 46–57. IEEE Computer Society, 1977.

[37] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[38] R. Rado. Axiomatic treatment of rank in infinite sets. *Canadian Journal of Mathematics* 1:337–343, 1949.

[39] W. Thomas. Computation tree logic and regular omega-languages. In *Proc. REX Workshop 1988*, LNCS 354, 690–713. Springer, 1988.

[40] W. Thomas. Languages, Automata, and Logic. *Handbook of Formal Languages*, 389–455, Springer 1996.

[41] M. Y. Vardi and P. Wolper. Yet another process logic (preliminary version). In *Proc. Logic of Programs 1983*, LNCS 164, 501–512. Springer, 1983.

[42] I. Walukiewicz. Monadic second-order logic on tree-like structures *Theor. Comput. Sci.*, 275(1-2):311–346, 2002.

[43] E. S. Wolk. The Comparability Graph of a Tree. In *Proceedings of the American Mathematical Society*, 13(5):789–795, 1962.

[44] E. S. Wolk. A Note on "The Comparability Graph of a Tree". In *Proceedings of the American Mathematical Society*, 16(1):17–20, 1965.

[45] P. Wolper. Temporal logic can be more expressive. In *Information and Control*, 56, 72–99, 1983.

[46] F. Wolter and M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In *Proceedings of the seventh Conference on Principles of Knowledge Representation and Reasoning*, pages 3–14, 2000.

Scientific Career

October 2015 - Present	Research assistant at the Chair for Automata Theory at the Institute for Theoretical Computer Science of the TU-Dresden. Part of the Collaborative Research Center HAEC
October 2012 - September 2015	PhD Student at Universität Leipzig - Institut für Informatik Scholarship holder for the DFG Research Training Group 1763 QuantLA (Quantitative Logics and Automata) Advisor: Prof. Dr. Markus Lohrey
April 2012	Master's degree in Mathematics Pure Mathematics Curriculum Facoltà di Scienze Matematiche Fisiche Naturali Università degli Studi di Firenze, (Italy) Final grade: 110/110 with honors
December 2008	Bachelor's degree in Mathematics Pure Mathematics Curriculum Facoltà di Scienze Matematiche Fisiche Naturali Università degli Studi di Firenze, (Italy) Final grade: 110/110
Academic Year 2007-2008	Participated to the Erasmus/LLP project attended two semesters at Universidad Autonoma de Madrid (Spain)
School Year 2002-2003	Scientific High School Diploma Bilingual Curriculum (English and French) Liceo Scientifico Antonio Gramsci Firenze (Italy) Final grade: 100/100

List of Publications

- Satisfiability of CTL* with constraints.
  In *Proc. CONCUR 2013*, LNCS 8052, pages 455-469. Springer, 2013.
  (with A. Kartzow and M. Lohrey)

- Satisfiability for MTL and TPTL over Non-monotonic Data Words.
  In *Proceedings of Language and Automata Theory and Applications 2014*, LNCS 8370, pages 248-259. Springer, 2014.
  (with S. Feng, O. Fernández and K. Quaas)

- On the Expressiveness of TPTL and MTL over omega-Data Words.
  In *Proceedings of Automata and Formal Languages 2014*, EPTCS 151, pages 174-187. 2014
  (with S. Feng, O. Fernández and K. Quaas)

- Satisfiability of ECTL* with Tree Constraints.
  In *Computer Science - Theory and Applications*. LNCS 9139, pages 94-108, 2015.
  (with S. Feng, A. Kartzow, and M. Lohrey)

- Temporal Logics with Local Constraints (Invited Contribution).
  In *24th EACSL Annual Conference on Computer Science Logic*. CSL 2015, pages 2-13.
  (with Markus Lohrey)

- Satisfiability of ECTL* with constraints.
  Accepted for publication in *Journal of Computer and System Sciences*. Available at `http://www.eti.uni-siegen.de/ti/veroeffentlichungen/ectl-with-constraints`.
  (with A. Kartzow and M. Lohrey)

Talks

29.08.2013 Concur 2013 - Buenos Aires.
Satisfiability of CTL* with constraints

21.09.2013 Highlights of Logic, Games and Automata 2013 - Paris.
Satisfiability of CTL* with constraints

21.02.2014 Almoth 2014 - Kassel.
Satisfiability of CTL* with constraints

14.03.2014 LATA 2014 - Madrid.
Satisfiability for MTL and TPTL over Non-Monotonic Data Words

06.05.2014 WATA 2014 - Leipzig.
Metric Temporal Logic and Timed Propositional Temporal Logic for Reasoning about Weighted Words over the Integers

05.09.2014 Highlights of Logic, Games and Automata 2014 - Paris
Which Comparability Graphs are Embeddable into Trees?

25.02.2015 Frontiers of Formal Methods - Aachen
Satisfiability of ECTL* with constraints

15.06.2015 GT-Verif, Journes annuelles 2015 - Paris
Satisfiability of CTL* with local constraints

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich have keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäßig aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialen oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, 29. Juni 2015

Claudia Carapelle