On Supergraphs Satisfying CMSO Properties*

Mateus de Oliveira Oliveira

Department of Informatics, University of Bergen, Bergen, Norway
mateus.oliveira@uib.no

Abstract

Let CMSO denote the counting monadic second order logic of graphs. We give a constructive proof that for some computable function f, there is an algorithm $\mathfrak{A}$ that takes as input a CMSO sentence φ, a positive integer t, and a connected graph G of maximum degree at most Δ, and determines, in time $f(|\varphi|, t) \cdot 2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$, whether G has a supergraph G' of treewidth at most t such that $G' \models \varphi$.

The algorithmic metatheorem described above sheds new light on certain unresolved questions within the framework of graph completion algorithms. In particular, using this metatheorem, we provide an explicit algorithm that determines, in time $f(d) \cdot 2^{O(\Delta \cdot d)} \cdot |G|^{O(d)}$, whether a connected graph of maximum degree Δ has a planar supergraph of diameter at most d. Additionally, we show that for each fixed k, the problem of determining whether G has an k-outerplanar supergraph of diameter at most d is strongly uniformly fixed parameter tractable with respect to the parameter d.

This result can be generalized in two directions. First, the diameter parameter can be replaced by any contraction-closed effectively CMSO-definable parameter $\mathbf{p}$. Examples of such parameters are vertex-cover number, dominating number, and many other contraction-bidimensional parameters. In the second direction, the planarity requirement can be relaxed to bounded genus, and more generally, to bounded local treewidth.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, F.4 Mathematical Logic and Formal Languages

Keywords and phrases CMSO Logic, Algorithmic Metatheorems, Graph Completion, Bidimensionality

Digital Object Identifier 10.4230/LIPIcs.CSL.2017.33

1 Introduction

A parameterized problem $\mathcal{L} \subseteq \Sigma^* \times \mathbb{N}$ is said to be *fixed parameter tractable* (FPT) if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that for each $(x, k) \in \Sigma^* \times \mathbb{N}$, one can decide whether $(x, k) \in \mathcal{L}$ in time $f(k) \cdot |x|^{O(1)}$, where $|x|$ is the size of x [10]. Using non-constructive methods derived from Robertson and Seymour's graph minor theory, one can show that certain problems can be solved in time $f(k) \cdot |x|^{O(1)}$ for some function $f : \mathbb{N} \to \mathbb{N}$. The caveat is that the function f arising from these non-constructive methods is often *not known to be computable*. Interestingly, for some problems it is not even clear how to obtain algorithms running in time $f_1(k) \cdot |x|^{f_2(k)}$ for some *computable* functions f_1 and f_2. In this work we will use techniques from automata theory and structural graph theory to provide constructive FPT and XP algorithms for problems for which only non-constructive parameterized algorithms were known.

* This work was supported by the Bergen Research Foundation.

26th EACSL Annual Conference on Computer Science Logic (CSL 2017).
Editors: Valentin Goranko and Mads Dam; Article No. 33; pp. 33:1–33:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The counting monadic second-order logic of graphs (CMSO) extends first order logic by allowing quantifications over sets of vertices and sets of edges, and by introducing the notion of modular counting predicates. This logic is expressive enough to define several interesting graph properties, such as Hamiltonicity, 3-colorability, connectivity, planarity, fixed genus, minor embeddability, etc. Additionally, when restricted to graphs of constant treewidth, CMSO logic is able to define precisely those properties that are recognizable by finite state tree-automata operating on encodings of tree-decompositions, or equivalently, those properties that can be described by equivalence relations with finite index [7, 1, 3, 4].

The expressiveness of CMSO logic has had a great impact in algorithmic theory due to Courcelle's model-checking theorem [7]. This theorem states that for some computable function $f : \mathbb{N}^2 \to \mathbb{N}$, one can determine in time[1] $f(|\varphi|, t) \cdot |G|$ whether a given graph G of treewidth at most t satisfies a given CMSO sentence φ. As a consequence of Courcelle's theorem, many combinatorial problems, such as Hamiltonicity or 3-colorability, which are NP-hard on general graphs, can be solved in linear time on graphs of constant treewidth. In this work we will consider a class problems on graphs of constant treewidth which cannot be directly addressed via Courcelle's theorem, either because it is not clear how to formulate the set of positive instances of such a problem as a CMSO-definable set, or because although the set of positive instances is CMSO-definable, it is not clear how to explicitly construct a CMSO sentence φ defining such set. For instance, sets of graphs that are closed under minors very often fall in the second category due to Robertson and Seymour's graph minor theorem.

1.1 Main Result

Let φ be a CMSO sentence, and t be a positive integer. We say that a graph G' is a (φ, t)-supergraph of a graph G if the following conditions are satisfied: G' satisfies φ, G' has treewidth at most t, and G' is a supergraph of G (possibly containing more vertices than G).

In our main result, Theorem 14, we devise an algorithm $\mathfrak{A}$ that takes as input a CMSO sentence φ, a positive integer t, and a connected graph G of maximum degree Δ, and determines in time $f(|\varphi|, t) \cdot 2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$ whether G has a (φ, t)-supergraph. We note that our algorithm determines the existence of such a (φ, t)-supergraph G' without the need of necessarily constructing G'. Therefore, no bound on the size of a candidate supergraph G' is imposed.

In the next three sub-sections we show how Theorem 14 can be used to provide partial solutions to certain long-standing open problems in parameterized complexity theory.

1.2 Planar Diameter Improvement

In the PLANAR DIAMETER IMPROVEMENT problem (PDI), we are given a graph G, and a positive integer d, and the goal is to determine whether G has a planar supergraph G' of diameter at most d. Note that the set of YES instances for the PDI problem is closed under minors. In other words, if G has a planar supergraph of diameter at most d, then any minor H of G also has such a supergraph. Therefore, using non-constructive arguments from Robertson and Seymour's graph minor theory [17, 18] in conjunction with the fact planar graphs of constant diameter have constant treewidth, one can show that for each fixed d, there exists an algorithm $\mathfrak{A}_d$ which determines in linear time whether a given G has diameter

[1] $|G|$ denotes the number of vertices in G, and $|\varphi|$, the number of symbols in φ.

at most d. The problem is that the non-constructive techniques mentioned above provide us with no clue about what the algorithm $\mathfrak{A}_d$ actually is. This problem can be partially remedied using a technique called effectivization by self-reduction introduced by Fellows and Langston [13, 10]. Using this technique one can show that for some function $f : \mathbb{N} \rightarrow \mathbb{N}$, there exists a single algorithm $\mathfrak{A}$ which takes a graph G and a positive integer d as input, and determines in time $f(d) \cdot |G|^{O(1)}$ whether G has a planar supergraph of diameter at most d. The caveat is that the function $f : \mathbb{N} \rightarrow \mathbb{N}$ bounding the influence of the parameter d in the running time of the algorithm mentioned above is not known to be computable.

Obtaining a fixed parameter tractability result for the PDI problem with a *computable* function f is a notorious and long-standing open problem in parameterized complexity theory [10, 12, 9]. Indeed, when it comes to explicit algorithms, the status of the PDI problem is much more elusive. As remarked in [5], even the problem of determining whether PDI can be solved in time $f_1(d) \cdot |G|^{f_2(d)}$ for *computable* functions $f_1, f_2 : \mathbb{N} \rightarrow \mathbb{N}$ is open.

Using Theorem 14 we provide an explicit algorithm that solves the PDI problem for connected graphs in time $f(d) \cdot 2^{O(\Delta \cdot d)} \cdot |G|^{O(d)}$ where $f : \mathbb{N} \rightarrow \mathbb{N}$ is a *computable* function, and Δ is the maximum degree of G. This result settles an open problem stated in [5] in the case in which the input graph is connected and has bounded (even logarithmic) degree. We note that our algorithm imposes no restriction on the degree of a prospective supergraph G'.

1.3 k-Outerplanar Diameter Improvement

A graph is 1-outerplanar if it can be embedded in the plane in such a way that all vertices lie in the outer-face of the embedding. A graph is k-outerplanar if it can be embedded in the plane in such a way that that deleting all vertices in the outer-face of the embedding yields a $(k-1)$-outerplanar graph. The k-outerplanar diameter improvement problem (k-OPDI) is the straightforward variant of PDI in which the completion is required to be k-outerplanar instead of planar. In [5] Cohen at al. provided an explicit polynomial time algorithm for the 1-OPDI problem. The complexity of the k-outerplanar diameter improvement problem was left open for $k \geq 2$. Using Theorem 14 we show that the k-OPDI problem can be solved in time $f(k, d) \cdot 2^{O(\Delta \cdot k)} \cdot |G|^{O(k)}$ where $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is a computable function. In other words, for each fixed k, the k-outerplanar diameter improvement problem is strongly uniformly fixed parameter tractable with respect to the diameter parameter d for bounded degree connected input graphs.

1.4 Contraction-Closed Parameters

A graph parameter is a function $\mathbf{p}$ that associates a non-negative integer with each graph. We say that such a parameter is contraction-closed if $\mathbf{p}(G) \leq \mathbf{p}(G')$ whenever G is a contraction of G'. For instance, the diameter of a graph is clearly a contraction-closed parameter. We say that a graph parameter $\mathbf{p}$ is effectively CMSO-definable if there exists a computable function α, and an algorithm that takes a positive integer k as input and constructs a CMSO formula φ_k that is true on a graph G if and only if $\mathbf{p}(G) \leq k$.

The results described in the previous subsections can be generalized in two directions. First, the diameter parameter can be replaced by any effectively CMSO-definable contraction closed parameter that is unbounded on Gamma graphs. These graphs were defined in [15] with the goal to provide a simplified exposition of the theory of contraction-bidimensionality. In particular, many well studied parameters that arise often in bidimensionality theory satisfy the conditions listed above. Examples of such parameters are the sizes of a minimum vertex cover, feedback vertex set, maximal matching, dominating set, edge dominating set, connected dominating set etc. On the other direction, the planarity requirement can be relaxed to CMSO definable graph properties that exclude some appex graph as a minor. These properties are also known in the literature as bounded local-treewidth properties. For instance, embeddability on surfaces of genus g, for fixed g, is one of such properties.

1.5 Proof Sketch And Organization of the Paper

In Section 2 we state some preliminary definitions. In Section 3 we define the notions of concrete bags, and concrete tree decompositions. Intuitively, a concrete tree-decomposition is an algebraic structure that represents a graph together with one of its tree decompositions. Using such structures we are able to define infinite families of graphs via tree-automata that accept infinite sets of tree decompositions. In particular, Courcelle's theorem can be transposed to this setting. More precisely, there is a computable function f such that for each CMSO sentence φ and each $t \in \mathbb{N}$, one can construct in time $f(|\varphi|, t)$ a tree automaton $\mathcal{A}(\varphi, t)$ which accept precisely those concrete tree decompositions of width at most t that give rise to graphs satisfying φ (Theorem 4).

In Section 4 we define the notion of sub-decomposition of a concrete tree decomposition. Intuitively, if a concrete tree decomposition $\mathbf{T}$ represents a graph G, then a sub-decomposition of $\mathbf{T}$ represents a sub-graph of G. We show that given a tree-automaton $\mathcal{A}$ accepting a set $\mathcal{L}(\mathcal{A})$ of concrete tree decompositions, one can construct a tree automaton $\mathrm{Sub}(\mathcal{A})$ which accepts precisely those sub-decompositions of concrete tree decompositions in $\mathcal{L}(\mathcal{A})$ (Theorem 6).

In Section 5, we introduce the main technical tool of this work. More specifically, we show that for each connected graph G of maximum degree Δ, one an construct in time $2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$ a tree-automaton $\mathcal{A}(G, t)$ whose language $\mathcal{L}(\mathcal{A}(G, t))$ consists precisely of those concrete tree decompositions of width at most t that give rise to G (Theorem 12).

In Section 6 we argue that the problem of determining whether G has a supergraph of treewidth at most t satisfying φ is equivalent to determining whether the intersection of $\mathcal{L}(\mathcal{A}(G, t+1))$ with $\mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1)))$ is non-empty. By combining Theorems 4, 6 and 12, we infer that this problem can be solved in time $f(|\varphi|, t) \cdot 2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$ (Theorem 14). Finally, in Section 7, we apply Theorem 14 to obtain explicit algorithms for several supergraph problems involving contraction-closed parameters.

2 Preliminaries

For each $n \in \mathbb{N}$, we let $[n] = \{1, ..., n\}$. We let $[0] = \emptyset$. For each finite set U, we let $\mathcal{P}(U)$ denote the set of subsets of U. For each $r \in \mathbb{N}$ and each finite set U, we let $\mathcal{P}^{\leq}(U, r) = \{U' \subseteq U \mid |U'| \leq r\}$ be the set of subsets of U of size at most r, and $\mathcal{P}^{=}(U, r) = \{U' \subseteq U \mid |U'| = r\}$ be the set of subsets of X of size precisely r. If $A, A_1, ..., A_k$ are sets, then we write $A = A_1 \dot\cup A_2 \dot\cup ... \dot\cup A_k$ to indicate that $A_i \cap A_j = \emptyset$ for $i \neq j$, and that A is the disjoint union of $A_1, .., A_k$.

Graphs. A *graph* is a triple $G = (V_G, E_G, \mathrm{Inc}_G)$ where V_G is a set of vertices, E_G is a set of edges, and $\mathrm{Inc}_G \subseteq E_G \times V_G$ is an incidence relation. For each $e \in E_G$ we let $\mathit{endpts}(e) = \{v \mid \mathrm{Inc}_G(e, v)\}$ be the set of endpoints of e, and we assume that $|\mathit{endpts}(e)|$ is either 0 or 2. We note that our graphs are allowed to have multiple edges, but no loops. We say that a graph H is a subgraph of G if $V_H \subseteq V_G$, $E_H \subseteq E_G$ and $\mathrm{Inc}_H = \mathrm{Inc}_G \cap E_H \times V_H$. Alternatively, we say that G is a supergraph of H. The degree of a vertex $v \in V_G$ is the number $d(v)$ of edges incident with v. We let $\Delta(G)$ denote the maximum degree of a vertex of G.

A *path* in a graph G is a sequence $v_1e_1v_2...e_{n-1}v_n$ where $v_i \in V_G$ for $i \in [n]$, $e_i \in E_G$ for $i \in [n-1]$, $v_i \neq v_j$ for $i \neq j$, and $\{v_i, v_{i+1}\} = endpts(e_i)$ for each $i \in [n-1]$. We say that G is *connected* if for every two vertices $v, v' \in V_G$ there is a path whose first vertex is v and whose last vertex is v'.

Let G and H be graphs. An isomorphism from G to H is a pair of bijections $\mu = (\dot{\mu} : V_G \to V_H, \overline{\mu} : E_G \to E_H)$ such that for every $e \in E_G$ if $endpts(e) = \{v, v'\}$ then $endpts(\overline{\mu}(e)) = \{\dot{\mu}(v), \dot{\mu}(v')\}$. We say that G and H are isomorphic if there is an isomorphism from G to H.

Treewidth. A tree is an acyclic graph T containing a unique connected component. To avoid confusion we may call the vertices of a tree "nodes" and call their edges "arcs". We let $nodes(T)$ denote the set of nodes of T and $arcs(T)$ denote its set of arcs. A *tree decomposition* of a graph G is a pair (T, β) where T is a tree and $\beta : nodes(T) \to \mathcal{P}(V_G)$ is a function that labels nodes of T with subsets of vertices of G in such a way that the following conditions are satisfied.

1. $\bigcup_{u \in nodes(T)} \beta(u) = V_G$
2. For every $e \in E_G$, there exists a node $u \in nodes(T)$ such that $endpts(e) \subseteq \beta(u)$
3. For every $v \in V_G$, the set $T_v = \{u \in nodes(T) \mid v \in \beta(u)\}$, i.e., the set of nodes of T whose corresponding bags contain v, induces a connected subtree of T.

The *width* of a tree decomposition (T, β) is defined as $max_{u \in nodes(T)} |\beta(u)| - 1$, that is, the maximum bag size minus one. The treewidth of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width of a tree decomposition of G.

CMSO Logic. The counting monadic second-order logic of graphs, here denoted by CMSO, extends first order logic by allowing quantifications over sets of vertices and edges, and by introducing the notion of modular counting predicates. More precisely, the syntax of CMSO logic includes the logical connectives $\vee, \wedge, \neg, \Leftrightarrow, \Rightarrow$, variables for vertices, edges, sets of vertices and sets of edges, the quantifiers $\exists, \forall$ that can be applied to these variables, and the following atomic predicates:

1. $x \in X$ where x is a vertex variable and X a vertex-set variable;
2. $y \in Y$ where y is an edge variable and Y an edge-set variable;
3. $\mathrm{Inc}(x, y)$ where x is a vertex variable, y is an edge variable, and the interpretation is that the edge x is incident with the edge y.
4. $card_{s,r}(Z)$ where $0 \leq s < r$, $r \geq 2$, Z is a vertex-set or edge-set variable, and the interpretation is that $|Z| = s \pmod r$;
5. equality of variables representing vertices, edges, sets of vertices and sets of edges.

A CMSO *sentence* is a CMSO formula without free variables. If φ is a CMSO sentence, then we write $G \models \varphi$ to indicate that G satisfies φ.

Terms. Let Σ be a finite set. The set $Ter(\Sigma)$ of terms over Σ is inductively defined as follows.

1. If $a \in \Sigma$, then $a \in Ter(\Sigma)$.
2. If $a \in \Sigma$, and $t \in Ter(\Sigma)$, then $a(t) \in Ter(\Sigma)$.
3. If $a \in \Sigma$, and $t_1, t_2 \in Ter(\Sigma)$, then $a(t_1, t_2) \in Ter(\Sigma)$.

Note that the alphabet Σ is unranked and the symbols in Σ may be regarded as function symbols or arity 0, 1 or 2. The set of positions of a term $t = a(t_1, ..., t_r) \in Ter(\Sigma)$ is defined as follows.

$$Pos(t) = \{\lambda\} \cup \bigcup_{j\in\{1,\ldots,r\}} \{jp \mid p \in Pos(t_j)\}.$$

Note that if $t = a$ for some $a \in \Sigma$, then $Pos(t) = \{\lambda\}$. If $p, pj \in Pos(t)$ where $j \in \{1,2\}$, then we say that pj is a *child* of p. Alternatively, we say that p is the *parent* of pj. We say that p is a *leaf* if it has no children. We let $\tau(t)$ be the tree that has $Pos(t)$ as nodes and $\{\{p, pj\} \mid j \in \{1,2\},\ p, pj \in Pos(t)\}$ as arcs. We say that a subset $P \subseteq Pos(t)$ is *connected* if the sub-tree of $\tau(t)$ induced by P is connected. If P is connected, then we say that a position $p \in P$ is the root of P if the parent of p does not belong to P.

If $t = a(t_1, ..., t_r)$ is a term in $Ter(\Sigma)$ for $r \in \{0,1,2\}$, and $p \in Pos(t)$, then the symbol $t[p]$ at position p is inductively defined as follows. If $p = \lambda$, then $t[p] = a$. On the other hand, if $p = jp'$ where $j \in \{1,2\}$, then $t[p] = t_j[p']$.

Tree Automata. Let Σ be a finite set of symbols. A *bottom-up tree-automaton* over Σ is a tuple $\mathcal{A} = (Q, \Sigma, F, \Delta)$ where Q is a set of states, $F \subseteq Q$ a set of final states, and Δ is a set of transitions of the form $(\mathfrak{q}_1, ..., \mathfrak{q}_r, a, \mathfrak{q})$ with $a \in \Sigma$, $0 \le r \le 2$, and $\mathfrak{q}_1, ..., \mathfrak{q}_r, \mathfrak{q} \in Q$. The size of $\mathcal{A}$, which is defined as $|\mathcal{A}| = |Q| + |\Delta|$, measures the number of states in Q plus the number of transitions in Δ. The set $\mathcal{L}(\mathcal{A}, \mathfrak{q}, i)$ of all terms reaching a state $\mathfrak{q} \in Q$ in depth at most i is inductively defined as follows.

$$\mathcal{L}(\mathcal{A}, \mathfrak{q}, 1) = \{a \mid (a, \mathfrak{q}) \in \Delta\}$$

$$\mathcal{L}(\mathcal{A}, \mathfrak{q}, i) = \mathcal{L}(\mathcal{A}, \mathfrak{q}, i-1) \cup \{a(t_1, ..., t_r) \mid r \in \{1,2\},\ \text{and } \exists(\mathfrak{q}_1, ..., \mathfrak{q}_r, a, \mathfrak{q}) \in \Delta,\ t_j \in \mathcal{L}(\mathcal{A}, \mathfrak{q}_j, i-1)\}$$

We denote by $\mathcal{L}(\mathcal{A}, \mathfrak{q})$ the set of all terms reaching state $\mathfrak{q}$ in finite depth, and by $\mathcal{L}(\mathcal{A})$ the set of all terms reaching some final state in F.

$$\mathcal{L}(\mathcal{A}, \mathfrak{q}) = \bigcup_{i\in\mathbb{N}} \mathcal{L}(\mathcal{A}, \mathfrak{q}, i) \qquad \mathcal{L}(\mathcal{A}) = \bigcup_{\mathfrak{q}\in F} \mathcal{L}(\mathcal{A}, \mathfrak{q}) \tag{1}$$

We say that the set $\mathcal{L}(\mathcal{A})$ is the language *accepted* by $\mathcal{A}$.

Let $\boldsymbol{\pi} : \Sigma \to \Sigma'$ be a map between finite sets of symbols Σ and Σ'. Such mapping can be homomorphically extended to a mapping $\boldsymbol{\pi} : Ter(\Sigma) \to Ter(\Sigma')$ between terms by setting $\boldsymbol{\pi}(t)[p] = \boldsymbol{\pi}(t[p])$ for each position $p \in Pos(t)$. Additionally, $\boldsymbol{\pi}$ can be further extended to a set of terms $\mathcal{L} \subseteq Ter(\Sigma)$ by setting $\boldsymbol{\pi}(\mathcal{L}) = \{\boldsymbol{\pi}(t) \mid t \in Ter(\Sigma)\}$. Below we state some well known closure and decidability properties for tree automata.

▶ **Lemma 1** (Properties of Tree Automata [6]). *Let Σ and Σ' be finite sets of symbols. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be tree automata over Σ, and $\boldsymbol{\pi} : \Sigma \to \Sigma'$ be a mapping.*

1. *One can construct in time $O(|\mathcal{A}_1| \cdot |\mathcal{A}_2|)$ a tree automaton $\mathcal{A}_1 \cap \mathcal{A}_2$ such that $\mathcal{L}(\mathcal{A}_1 \cap \mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.*
2. *One can determine whether $\mathcal{L}(\mathcal{A}_1) \neq \emptyset$ in time $|\mathcal{A}_1|^{O(1)}$.*
3. *One can construct in time $O(|\mathcal{A}_1|)$ a tree automaton $\boldsymbol{\pi}(\mathcal{A}_1)$ such that $\mathcal{L}(\boldsymbol{\pi}(\mathcal{A}_1)) = \boldsymbol{\pi}(\mathcal{L}(\mathcal{A}_1))$.*

3 Concrete Tree Decompositions

A *t-concrete bag* is a pair (B, b) where $B \subseteq [t]$, and $b \subseteq B$ with $b = \emptyset$ or $|b| = 2$. We note that B is allowed to be empty. We let $\mathcal{B}(t)$ be the set of all t-concrete bags. Note that

$|\mathcal{B}(t)| \leq t^2 \cdot 2^t$. We regard the set $\mathcal{B}(t)$ as a finite alphabet which will be used to construct terms representing tree decompositions of graphs.

A t-concrete tree decomposition is a term $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t))$. We let $\mathbf{T}[p] = (\mathbf{T}[p].B, \mathbf{T}[p].b)$ be the t-concrete bag at position p of $\mathbf{T}$. For each $s \in [t]$, we say that a subset $P \subseteq \mathit{Pos}(\mathbf{T})$ is *s-maximal* if the following conditions are satisfied.

1. P is connected in $\mathit{Pos}(\mathbf{T})$.
2. $s \in \mathbf{T}[p].B$ for every $p \in P$.
3. If P' is a connected subset of $\mathit{Pos}(\mathbf{T})$ and $s \in \mathbf{T}[p].B$ for every $s \in P'$, then $P' \subseteq P$.

Note that if P and P' are s-maximal then either $P = P'$, or $P \cap P' = \emptyset$. Additionally, $p \in \mathit{Pos}(\mathbf{T})$ and each $s \in \mathbf{T}[p].B$, there exists a unique subset $P \subseteq \mathit{Pos}(\mathbf{T})$ such that P is s-maximal and $p \in P$. We denote this unique set by $P(p, s)$.

▶ **Definition 2.** Let $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t))$. The graph $\mathcal{G}(\mathbf{T})$ associated with $\mathbf{T}$ is defined as follows.

1. $V_{\mathcal{G}(\mathbf{T})} = \{v_{s,P} \mid s \in [t],\ P \subseteq \mathit{Pos}(\mathbf{T}),\ P \text{ is } s\text{-maximal}\}$.
2. $E_{\mathcal{G}(\mathbf{T})} = \{e_p \mid p \in \mathit{Pos}(\mathbf{T}),\ b \neq \emptyset\}$.
3. $\mathrm{Inc}_{\mathcal{G}(\mathbf{T})} = \{(e_p, v_{s,P(p,s)}) \mid e_p \in E_{\mathcal{G}(\mathbf{T})},\ s \in \mathbf{T}[p].b\}$.

Intuitively, a t-concrete tree decomposition may be regarded as a way of representing a graph together with one of its tree decompositions. This idea is widespread in texts dealing with recognizable properties of graphs [3, 2, 8, 11, 14]. Within this framework it is customary to define a bag of width t as a graph with at most t vertices together with a function that labels the vertices of these graphs with numbers from $\{1, ..., t\}$. Our notion of t-concrete bag, on the other hand, may be regarded as a representation of a graph with at most t vertices injectively labeled with numbers from $\{1, ...t\}$ and at most *one* edge. Within this point of view, the representation used here is a syntactic restriction of the former. On the other hand, any decomposition which uses bags with arbitrary graphs of size t can be converted into a t-concrete decomposition, by expanding each bag into a sequence of t^2 concrete bags. The following observation is immediate, using the fact that if a graph has treewidth t, then it has a rooted tree decomposition in which each node has at most two children [11].

▶ **Observation 3.** *A graph G has treewidth t if and only if there exists some $(t + 1)$-concrete tree decomposition $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t + 1))$ such that $\mathcal{G}(\mathbf{T})$ is isomorphic to G.*

The next theorem may be regarded as a variant of Courcelle's theorem [8].

▶ **Theorem 4** (Courcelle's Theorem). *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for each CMSO sentence φ, and each $t \in \mathbb{N}$, there is a tree-automaton $\mathcal{A}(\varphi, t)$ accepting the following tree language.*

$$\mathcal{L}(\mathcal{A}(\varphi, t)) = \{\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t)) \mid \mathcal{G}(\mathbf{T}) \models \varphi\}. \tag{2}$$

4 Sub-Decompositions

In this section we introduce the notion of sub-decompositions of a t-concrete decomposition. Intuitively, if a t-concrete tree decomposition $\mathbf{T}$ represents a graph G then sub-decompositions of $\mathbf{T}$ represent subgraphs of G. The main result of this section states that given a tree automaton $\mathcal{A}$ over $\mathcal{B}(t)$, one can efficiently construct a tree automaton $\mathrm{Sub}(\mathcal{A})$ over $\mathcal{B}(t)$ which accepts precisely those sub-decompositions of t-concrete tree decompositions in $\mathcal{L}(\mathcal{A})$.

We say that a t-concrete bag (B, b) is a sub-bag of a t-concrete bag (B', b') if $B \subseteq B'$ and $b \subseteq b'$.

▶ **Definition 5.** We say that a t-concrete tree decomposition $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t))$ is a sub-decomposition of a t-concrete tree decomposition $\mathbf{T}' \in \mathit{Ter}(\mathcal{B}(t))$ if the following conditions are satisfied.

S1. $\mathit{Pos}(\mathbf{T}) = \mathit{Pos}(\mathbf{T}')$.

S2. For each $p \in \mathit{Pos}(\mathbf{T})$, $\mathbf{T}[p]$ is a sub-bag of $\mathbf{T}'[p]$.

S3. For each $p, pj \in \mathit{Pos}(\mathbf{T})$, and for each $s \in [t]$, if $s \in \mathbf{T}'[p].B$ and $s \in \mathbf{T}'[pj].B$, then $s \notin \mathbf{T}[p].B$ if and only if $s \notin \mathbf{T}[pj].B$.

The following theorem states that sub-decompositions of $\mathbf{T}$ are in one to one correspondence with subgraphs of $\mathcal{G}(\mathbf{T})$.

▶ **Theorem 6.** *Let G and G' be a graphs and let $\mathbf{T}' \in \mathit{Ter}(\mathcal{B}(t))$ be a t-concrete tree decomposition of G'. Then G is a subgraph of G' if and only if there exists some $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t))$ such that $\mathbf{T}$ is a sub-decomposition of $\mathbf{T}'$ with $\mathcal{G}(\mathbf{T}) = G$.*

Proof.

1. Let G be a subgraph of $\mathcal{G}(\mathbf{T}')$. We show that there exists a sub-decomposition $\mathbf{T}$ of $\mathbf{T}'$ such that $\mathcal{G}(\mathbf{T}) = G$. Since G is a subgraph of $\mathcal{G}(\mathbf{T})$, we have that $V_G \subseteq V_{\mathcal{G}(\mathbf{T}')}$, $E_G \subseteq E_{\mathcal{G}(\mathbf{T}')}$, and $\mathrm{Inc}_G = \mathrm{Inc}_{\mathcal{G}(\mathbf{T}')} \cap E_G \times V_G$. We define $\mathbf{T}$ by setting $\mathbf{T}[p]$ as follows for each $p \in \mathit{Pos}(\mathbf{T}) = \mathit{Pos}(\mathbf{T}')$.
   a. $\mathbf{T}[p].B = \mathbf{T}'[p].B \backslash \{s \mid v_{s,P(p,s)} \in V_{\mathcal{G}(\mathbf{T}')} \backslash V_G\}$.
   b. $\mathbf{T}[p].b = \emptyset$ if $e_p \in E_{\mathcal{G}(\mathbf{T}')} \backslash E_G$ and $\mathbf{T}[p].b = \mathbf{T}'[p].b$ otherwise.
   First, we note that $v_{s,P} \in V_{\mathcal{G}(\mathbf{T})}$ if and only if $v_{s,P} \in V_G$, $e_p \in E_{\mathcal{G}(\mathbf{T})}$ if and only if $e_p \in E_G$, and $(e_p, v_{i,P}) \in \mathrm{Inc}_{\mathcal{G}(\mathbf{T})}$ if and only if $(e_p, v_{i,P}) \in V_G$. Therefore, $G = \mathcal{G}(\mathbf{T})$. We will check that the t-concrete decomposition $\mathbf{T}$ defined above is indeed a sub-decomposition of $\mathbf{T}'$. In other words, we will verify that conditions **S1**, **S2** and **S3** above are satisfied. The fact that **S1** is satisfied is immediate, since we define $\mathbf{T}[p]$ for each $p \in \mathit{Pos}(\mathbf{T}')$. Therefore, $\mathit{Pos}(\mathbf{T}) = \mathit{Pos}(\mathbf{T}')$. Condition **S2** is also satisfied, since by (a) and (b) we have that $\mathbf{T}[p].B \subseteq \mathbf{T}'[p].B$ and that $\mathbf{T}[p].b$ is either $\emptyset$, or equal to $\mathbf{T}'[p].b$. Finally, condition **S3** is also satisfied, since (a) guarantees that for each number $s \in [t]$, and each s-maximal set $P \subseteq \mathit{Pos}(\mathbf{T}')$, if s is removed from $\mathbf{T}'[p].B$ for some $p \in P$, then s is indeed removed from $\mathbf{T}'[p].B$ for every $p \in P$.
2. For the converse, let $\mathbf{T}$ be a sub-decomposition of $\mathbf{T}'$. We show that the graph $\mathcal{G}(\mathbf{T})$ is a subgraph of $\mathcal{G}(\mathbf{T}')$. First, we note that condition **S3** guarantees that for each $s \in [t]$ and each $P \subseteq \mathit{Pos}(\mathbf{T})$, if P is s-maximal in $\mathbf{T}$ then P is s-maximal in $\mathbf{T}'$. Therefore, $V_{\mathcal{G}(\mathbf{T})} \subseteq V_{\mathcal{G}(\mathbf{T}')}$. Now, Condition **S2** guarantees that $e_p \in E_{\mathcal{G}(\mathbf{T})}$ implies that $e_p \in E_{\mathcal{G}(\mathbf{T}')}$. Therefore, $E_{\mathcal{G}(\mathbf{T})} \subseteq E_{\mathcal{G}(\mathbf{T}')}$. Finally, by definition $(e_p, v_{s,P}) \in \mathrm{Inc}_{\mathcal{G}(\mathbf{T})}$ if and only if $s \in \mathbf{T}[p].b$ for each $p \in P$. Since the fact that $s \in \mathbf{T}[p].b$ implies that $s \in \mathbf{T}'[p].b$, we have that $(e_p, v_{s,P}) \in \mathrm{Inc}_{\mathcal{G}(\mathbf{T})}$ implies that $(e_p, v_{s,P}) \in \mathrm{Inc}_{\mathcal{G}(\mathbf{T}')}$. Therefore, $\mathrm{Inc}_{\mathcal{G}(\mathbf{T})} \subseteq \mathrm{Inc}_{\mathcal{G}(\mathbf{T}')}$. Additionally, since $(e_p, v_{s,P(s,p)}) \in \mathrm{Inc}_{\mathcal{G}(\mathbf{T})}$ for each $e_p \in E_{\mathcal{G}(\mathbf{T})}$ and each $s \in \mathbf{T}[p].b$, we have that $\mathrm{Inc}_{\mathcal{G}(\mathbf{T})} = \mathrm{Inc}_{\mathcal{G}(\mathbf{T}')} \cap E_{\mathcal{G}(\mathbf{T})} \times V_{\mathcal{G}(\mathbf{T})}$. This shows that $\mathcal{G}(\mathbf{T})$ is a subgraph of $\mathcal{G}(\mathbf{T}')$. ◀

The following theorem states that given a tree automaton $\mathcal{A}$ over $\mathcal{B}(t)$, one can efficiently construct a tree automaton $\mathrm{Sub}(\mathcal{A})$ which accepts precisely those sub-decompositions of t-concrete tree decompositions in $\mathcal{L}(\mathcal{A})$.

▶ **Theorem 7** (Sub-Decomposition Automaton)**.** *Let $\mathcal{A}$ be a tree automaton over $\mathcal{B}(t)$. Then one can construct in time $2^{O(t)} \cdot |\mathcal{A}|$ a tree automaton $\mathrm{Sub}(\mathcal{A})$ over $\mathcal{B}(t)$ accepting the following language.*

$$\mathcal{L}(\mathrm{Sub}(\mathcal{A})) = \{\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t)) \mid \exists \mathbf{T}' \in \mathcal{L}(\mathcal{A}) \textit{ s.t. } \mathbf{T} \textit{ is a sub-decomposition of } \mathbf{T}'\}.$$

Proof. Let $\mathcal{A} = (Q, \mathcal{B}(t), \Delta, F)$ be a tree automaton over $\mathcal{B}(t)$. As a first step we create an intermediate tree automaton $\mathcal{A}' = (Q', \mathcal{B}(t), \Delta', F')$ which accepts the same language as $\mathcal{A}$. The tree automaton $\mathcal{A}'$ is defined as follows.

$$Q' = \{\mathfrak{q}_B \mid \mathfrak{q} \in Q,\ B \subseteq [t]\} \qquad F' = \{\mathfrak{q}_B \mid \mathfrak{q} \in F,\ B \subseteq [t]\}$$

$$\Delta' = \{(\mathfrak{q}^1_{B_1}, ..., \mathfrak{q}^r_{B_r}, (B, b), \mathfrak{q}_B) \mid (\mathfrak{q}^1, ..., \mathfrak{q}^r, (B, b), \mathfrak{q}) \in \Delta,\ B_i \subseteq [t] \text{ for } i \in [r]\}.$$

Note that for each $\mathfrak{q} \in Q$, each $B \subseteq [t]$, and each $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t))$, $\mathbf{T}$ reaches state $\mathfrak{q}_B$ in $\mathcal{A}'$ if and only if $\mathbf{T}$ reaches state $\mathfrak{q}$ in $\mathcal{A}$ and $\mathbf{T}[\lambda].B = B$, where $\mathbf{T}[\lambda]$ is the t-concrete bag at the root of $\mathbf{T}$. In particular, this implies that a term $\mathbf{T}$ belongs to $\mathcal{L}(\mathcal{A}')$ if and only if $\mathbf{T} \in \mathcal{L}(\mathcal{A})$.

Now, consider the tree automaton $\mathrm{Sub}(\mathcal{A}) = (Q'', \mathcal{B}(t), \Delta'', F'')$ over $\mathcal{B}(t)$ where

$$Q'' = \{\mathfrak{q}_{B,B'} \mid \mathfrak{q} \in Q, B \subseteq B' \subseteq [t]\} \qquad F'' = \{\mathfrak{q}_{B,B'} \mid \mathfrak{q} \in F, B \subseteq B' \subseteq [t]\}$$

$$\begin{aligned}\Delta'' = \{(\mathfrak{q}^1_{B_1,B'_1}, ..., \mathfrak{q}^r_{B_r,B'_r}, (B, b), \mathfrak{q}_{B,B'}) \mid\ & \exists(\mathfrak{q}^1_{B'_1}, ..., \mathfrak{q}^r_{B'_r}, (B', b'), \mathfrak{q}_{B'}) \in \Delta' \text{ such that}\\ & B_i \subseteq B'_i,\ B \subseteq B',\\ & (B, b) \text{ is a sub-bag of } (B', b')\\ & \text{for each } j \in [r], \text{ if } s \in B' \wedge s \in B'_j \text{ then } s \notin B \Leftrightarrow s \notin B_j\}.\end{aligned}$$

It follows by induction on the height of terms that a term $\mathbf{T} \in \mathit{Ter}(\mathcal{B}(t))$ reaches a state $\mathfrak{q}_{B,B'}$ in $\mathrm{Sub}(\mathcal{A})$ if and only if there exists some term $\mathbf{T}' \in \mathit{Ter}(\mathcal{B}(t))$ such that $\mathbf{T}'$ reaches state $\mathfrak{q}_{B'}$ in $\mathcal{A}'$, $\mathbf{T}[\lambda].B = B$, $\mathbf{T}'[\lambda].B = B'$, and $\mathbf{T}$ is a sub-decomposition of $\mathbf{T}'$. In particular, $\mathbf{T}$ reaches a final state of $\mathrm{Sub}(\mathcal{A})$ if and only if $\mathbf{T}$ is a sub-decomposition of some $\mathbf{T}'$ which reaches a final state of $\mathcal{A}'$. ◀

5 Representing All Tree Decompositions of a Given Graph

In this section we show that given a connected graph G of maximum degree Δ, and a positive integer t, one can construct in time $2^{O(\Delta \cdot t)} \cdot |V_G|^{O(t)}$ a tree automaton $\mathcal{A}(G, t)$ over $\mathcal{B}(t)$ that accepts precisely those t-concrete tree decompositions of G.

Let G be a graph. A (G, t)-concrete bag is a tuple $(B, b, \nu, \eta, y, \rho)$ where (B, b) is a t-concrete bag; $\nu : B \to V_G$ is a function that assigns a vertex of G to each element of B; $\eta : B \to \mathcal{P}^{\leq}(E_G, \Delta(G))$ is a function that assigns to each element $s \in B$, a set of edges incident with $\nu(s)$ of size at most $\Delta(G)$; y is a subset of E_G such that $|y| \leq 1$ and $y \subseteq \eta(s)$ whenever $s \in b$; and ρ is a subset of B.

We let $\mathcal{B}(G, t)$ be the set of all (G, t)-concrete bags. Note that $\mathcal{B}(G, t)$ has at most $2^{O(\Delta(G) \cdot t)} \cdot |V_G|^{O(t)}$ elements. We let $\mathit{Ter}(\mathcal{B}(G, t))$ be the set of all terms over $\mathcal{B}(G, t)$. If $\hat{\mathbf{T}}$ is a term in $\mathcal{B}(G, t)$ then for each $p \in \mathit{Pos}(\mathbf{T})$, the (G, t)-concrete bag of $\hat{\mathbf{T}}$ at position p is denoted by the tuple

$$(\hat{\mathbf{T}}[p].B, \hat{\mathbf{T}}[p].b, \hat{\mathbf{T}}[p].\nu, \hat{\mathbf{T}}[p].\eta, \hat{\mathbf{T}}[p].y, \hat{\mathbf{T}}[p].\rho).$$

▶ **Definition 8.** We say that a term $\hat{\mathbf{T}} \in \mathit{Ter}(\mathcal{B}(G, t))$ is a (G, t)-concrete tree decomposition if the following conditions are satisfied for each each $p \in \mathit{Pos}(\hat{\mathbf{T}})$ and each $s \in [t]$.

C1. If $pj \in \mathit{Pos}(\hat{\mathbf{T}})$ and $s \in \hat{\mathbf{T}}[p].B \cap \hat{\mathbf{T}}[pj].B$ then $\hat{\mathbf{T}}[p].\nu(s) = \hat{\mathbf{T}}[pj].\nu(s)$.

C2. If $\hat{\mathbf{T}}[p].b = \{s, s'\}$ then $\hat{\mathbf{T}}[p].y = \{e\}$ for some edge e with

$$\mathit{endpts}(e) = \{\hat{\mathbf{T}}[p].\nu(s), \hat{\mathbf{T}}[p].\nu(s')\}$$

.

C3. Let $r \in \{0,1,2\}$, and $p1, ..., pr$ be the children[2] of p, then

$$\hat{\mathbf{T}}[p].\eta(s) = \hat{\mathbf{T}}[p].y \,\dot\cup\, \hat{\mathbf{T}}[p1].\eta(s) \,\dot\cup\, ... \,\dot\cup\, \hat{\mathbf{T}}[pr].\eta(s).$$

C4. If $s \in \hat{\mathbf{T}}[p].\rho$ then $\hat{\mathbf{T}}[p].\eta(s) = \{e \mid (e, \hat{\mathbf{T}}[p].\nu(s)) \in \mathrm{Inc}_G\}$.

C5. If $p = \lambda$ then $\hat{\mathbf{T}}[p].\rho = \hat{\mathbf{T}}[p].B$. If $pj \in Pos(\hat{\mathbf{T}})$ then $s \in \hat{\mathbf{T}}[pj].\rho$ if and only if $s \in \hat{\mathbf{T}}[pj].B$ and $s \notin \hat{\mathbf{T}}[p].B$.

Let $\boldsymbol{\pi} : \mathcal{B}(G,t) \to \mathcal{B}(t)$ be a function such that $\boldsymbol{\pi}(B, b, \nu, \eta, y, \rho) = (B, b)$ for each (G,t)-concrete bag $(B, b, , \nu, \eta, y, \rho) \in \mathcal{B}(G,t)$. In other words, $\boldsymbol{\pi}$ transforms a (G,t)-concrete bag into a t-concrete bag by erasing the four last coordinates of the former. If $\hat{\mathbf{T}}$ is a term in $Ter(\mathcal{B}(G,t))$ then we let $\boldsymbol{\pi}(\hat{\mathbf{T}})$ be the term in $Ter(\mathcal{B}(t))$ which is obtained by setting $\boldsymbol{\pi}(\hat{\mathbf{T}})[p] = \boldsymbol{\pi}(\hat{\mathbf{T}}[p])$ for each position $p \in Pos(\hat{\mathbf{T}})$.

▶ **Theorem 9.** *Let G be a connected graph and let $\mathbf{T} \in Ter(\mathcal{B}(t))$. Then $\mathbf{T}$ is a t-concrete tree decomposition of G if and only if $|V_{\mathcal{G}(\mathbf{T})}| = |V_G|$ and there exists a (G,t)-concrete tree decomposition $\hat{\mathbf{T}} \in Ter(\mathcal{B}(G,t))$ such that $\mathbf{T} = \boldsymbol{\pi}(\hat{\mathbf{T}})$.*

Note that conditions **C1-C5** are local in the sense that they may be verified at each position $p \in Pos(\hat{\mathbf{T}})$ by analysing only the concrete bags $\hat{\mathbf{T}}[p], \hat{\mathbf{T}}[p1], ..., \hat{\mathbf{T}}[pr]$ where $p1, ..., pr$ are the children of p. This allows us to define a tree automaton $\hat{\mathcal{A}}(G,t)$ over $\mathcal{B}(G,t)$ that accepts a term $\hat{\mathbf{T}} \in Ter(\mathcal{B}(G,t))$ if and only if $\hat{\mathbf{T}}$ is a (G,t)-concrete tree decomposition.

▶ **Lemma 10.** *For each positive integer t and each graph G of maximum degree Δ, one can construct in time $2^{O(\Delta)} \cdot |V_G|^{O(t)}$ a tree automaton $\hat{\mathcal{A}}(G,t)$ over $\mathcal{B}(G,t)$ accepting the following language.*

$$\mathcal{L}(\hat{\mathcal{A}}(G,t)) = \{\hat{\mathbf{T}} \in Ter(\mathcal{B}(G,t)) \mid \hat{\mathbf{T}} \textit{ is a } (G,t)\textit{-concrete tree decomposition.}\} \tag{3}$$

The next lemma states that for each positive integers t and n, one can efficiently construct a tree automaton $\mathcal{A}(t,n)$ which accepts precisely those t-concrete tree decompositions which give rise to graphs with n vertices.

▶ **Lemma 11.** *Let t and n be positive integers with $t \leq n$. One can construct in time $2^{O(t)} \cdot n^3$ a tree automaton $\mathcal{A}(t,n)$ over $\mathcal{B}(t)$ accepting the following language.*

$$\mathcal{L}(\mathcal{A}(t,n)) = \{\mathbf{T} \in Ter(\mathcal{B}(t)) \mid |V_{\mathcal{G}(\mathbf{T})}| = n\}$$

The main result of this section (Theorem 12), follows by a combination of Theorem 9, Lemma 10 and Lemma 11.

▶ **Theorem 12.** *Let G be a connected graph of treewidth t and maximum degree Δ. Then one can construct in time $2^{O(\Delta \cdot t)} \cdot |V_G|^{O(t)}$ a tree automaton $\mathcal{A}(G,t)$ over $\mathcal{B}(t)$ such that for each $\mathbf{T} \in Ter(\mathcal{B}(t))$, $\mathbf{T} \in \mathcal{L}(\mathcal{A}(G,t))$ if and only if $\mathbf{T}$ is a concrete tree decomposition of G.*

6 (φ, t)-Supergraphs

Let φ be a CMSO sentence, and t be a positive integer. Let G and G' be graphs. We say that G' is a (φ, t)-supergraph of G if the following three conditions are satisfied: $G' \models \varphi$, G' has treewidth at most t, and G is a subgraph of G'.

---

[2] If $r = 0$ then p has no child.

▶ **Lemma 13.** *Let φ be a CMSO sentence and t be a positive integer. Then a graph G has a (φ, t)-supergraph if and only if there exists a $(t+1)$-concrete tree decomposition $\mathbf{T} \in \mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1))$ such that $\mathcal{G}(\mathbf{T})$ is isomorphic to G.*

Proof. Assume that G is a graph that has a (φ, t)-supergraph G'. Then G' satisfies φ, G' has treewidth at most t, and G is a subgraph of G'. By Observation 3, G' has a $(t+1)$-concrete tree decomposition $\mathbf{T}' \in \mathit{Ter}(\mathcal{B}(t+1))$, and therefore by Theorem 4, $\mathbf{T}' \in \mathcal{L}(\mathcal{A}(\varphi, t))$. Since G is a subgraph of G', by Theorem 6, $\mathbf{T}'$ has a sub-decomposition $\mathbf{T}$ which is a $(t+1)$-concrete tree decomposition of G. Therefore, $\mathbf{T}$ belongs to $\mathrm{Sub}(\mathcal{A}(\varphi, t+1))$.

For the converse, let $\mathbf{T} \in \mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1)))$ and let $\mathbf{T}$ be a $(t+1)$-concrete tree decomposition of G. Then $\mathbf{T}$ is a sub-decomposition of some $(t+1)$-concrete tree decomposition $\mathbf{T}'$ in $\mathcal{L}(\mathcal{A}(\varphi, t+1))$. By Theorem 4, $\mathbf{T}'$ is a $(t+1)$-concrete tree decomposition of some graph G' of treewidth at most t such that $G' \models \varphi$. Since $\mathbf{T}$ is a sub-decomposition of $\mathbf{T}'$, by Theorem 6, G is a subgraph of G'. Therefore, G' is a (φ, t)-supergraph of G. ◀

We note that Lemma 13 alone does not provide us with an efficient algorithm to determine whether a graph G has a (φ, t)-supergraph. If G does not admit such a supergraph, then no $(t+1)$-concrete tree decomposition G belongs to $\mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1)))$. However, if G does admit a (φ, t)-supergraph, then Theorem 6 only guarantees that some $(t+1)$-concrete tree decomposition $\mathbf{T}$ of G belongs to $\mathrm{Sub}(\mathcal{A}(\varphi, t+1))$. The problem is that G may have exponentially[3] many such decompositions, and we do not know a priori which of these should be tested for membership in $\mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1)))$.

The issue described above can be remedied with the results from Section 5. More specifically, from Theorem 12 we have that for any given connected graph G of treewidth t and maximum degree Δ, one can construct a tree automaton $\mathcal{A}(G, t+1)$ over $\mathcal{B}(t+1)$ which accepts a $(t+1)$-concrete tree decomposition $\mathbf{T}$ if and only if the graph $\mathcal{G}(\mathbf{T})$ is isomorphic to G. Therefore, a connected graph G has a (φ, t)-supergraph if and only if

$$\mathcal{L}(\mathcal{A}(G, t+1)) \cap \mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1))) \neq \emptyset. \tag{4}$$

The next theorem states that Equation 4 yields an efficient algorithm for testing whether connected graphs of bounded degree have a (φ, t)-supergraph.

▶ **Theorem 14** (Main Theorem). *There is a computable function f, and an algorithm $\mathfrak{A}$ that takes as input a CMSO sentence φ, a positive integer t, and a connected graph G of maximum degree Δ, and determines in time $f(|\varphi|, t) \cdot 2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$ whether G has a (φ, t)-supergraph.*

Proof. By Lemma 13, G has a (φ, t)-supergraph if and only if there exists some $\mathbf{T} \in \mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1)))$ such that $\mathbf{T}$ is a $(t+1)$-concrete tree decomposition of G. By Theorem 12, $\mathcal{L}(\mathcal{A}(G, t+1))$ accepts a $(t+1)$-tree decomposition of G if and only if $\mathcal{G}(\mathbf{T})$ is isomorphic to G. Therefore, G has a (φ, t)-supergraph if and only the intersection of $\mathcal{L}(\mathcal{A}(G, t+1))$ with $\mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1)))$ is nonempty.

By Theorem 12, the tree-automaton $\mathcal{A}(G, t+1)$ can be constructed in time $2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$, and therefore the size of $\mathcal{A}(G, t+1)$ is bounded by $2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$. By Theorem 6 and Theorem 4, the tree-automaton $\mathrm{Sub}(\mathcal{A}(\varphi, t+1))$ can be constructed in time $f(|\varphi|, t)$ for some computable function $f : \mathbb{N}^2 \to \mathbb{N}$, and therefore, the size of $\mathrm{Sub}(\mathcal{A}(\varphi, t))$ is bounded by $f(|\varphi|, t)$.

[3] In fact a graph G of treewidth t may have infinitely many $(t+1)$-concrete tree decompositions, but we only need to consider those which have at most $|V_G| + |E_G|$ nodes.

Finally, given tree automata $\mathcal{A}_1$ and $\mathcal{A}_2$, one can determine whether $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) \neq \emptyset$ in time $O(|\mathcal{A}_1| \cdot |\mathcal{A}_2|)$ (Lemma 1). In particular, one can determine whether $\mathcal{L}(\mathcal{A}(G, t+1)) \cap \mathcal{L}(\mathrm{Sub}(\mathcal{A}(\varphi, t+1))) \neq \emptyset$ in time $f(|\varphi|, t) \cdot 2^{O(\Delta \cdot t)} \cdot |G|^{O(t)}$. □ ◀

7 Contraction Closed Graph Parameters

In this section we deal with simple graphs, i.e., graphs without loops or multiple edges. Therefore, we may write $\{u, v\}$ to denote an edge e whose endpoints are u and v.

Let G be a graph and $\{u, v\}$ be an edge of G. We let G/uv denote the graph that is obtained from G by deleting the edge $\{u, v\}$ and by merging vertices u and v into a single vertex x_{uv}. We say that G/uv is obtained from G by an edge-contraction. We say that a graph G' is a *contraction* of G if G' is obtained from G by a sequence of edge contractions. We say that G' is a minor of G if G' is a contraction of some subgraph of G. We say that a graph G is an appex graph if after deleting some vertex of G the resulting graph is planar.

A graph parameter is a function $\mathbf{p}$ mapping graphs to non-negative integers in such a way that $\mathbf{p}(G) = \mathbf{p}(G')$ whenever G is isomorphic to G'. We say that $\mathbf{p}$ is contraction closed if $\mathbf{p}(G') \leq \mathbf{p}(G)$ whenever G' is a contraction of G.

A graph property is simply a set $\mathcal{P}$ of graphs. We say that a property $\mathcal{P}$ is contraction-closed if for every two graphs G, G' for which G' is a contraction of G, the fact that $G \in \mathcal{P}$ implies that $G' \in \mathcal{P}$.

7.1 Diameter Improvement Problems

Let u and v be vertices in an graph G. The distance from u to v, denoted by $dist(u, v)$ is the number of edges in the shortest path from u to v. If no such path exists, we set $dist(u, v) = \infty$. The diameter of G is defined as $diam(G) = \max_{u,v} dist(u, v)$. In the PLANAR DIAMETER IMPROVEMENT problem (PDI), we are given an graph G and a positive integer d, and the goal is to determine whether G has a planar supergraph G' of diameter at most d. As mentioned in the introduction, there is an algorithm that solves the PDI problem in time $f(d) \cdot |G|^{O(1)}$, where $f : \mathbb{N} \to \mathbb{N}$ is not known to be computable. Additionally, even the problem of determining whether PDI admits an algorithm running in time $f_1(d) \cdot |G|^{f_2(d)}$ for computable functions f_1, f_2 remains open for more than two decades [13, 5]. The next theorem solves this problem in when the input graphs are connected and have bounded degree.

▶ **Theorem 15.** *There is a computable function $f : \mathbb{N} \to \mathbb{N}$, and an algorithm $\mathfrak{A}$ that takes as input, a positive integer d, and a connected graph G of maximum degree Δ, and determines in time $f(d) \cdot 2^{O(\Delta \cdot d)} \cdot |G|^{O(d)}$ whether G has a planar supergraph G' of diameter at most d.*

Proof. It should be clear that there is an algorithm that takes a positive integer d as input, and constructs, in time $O(d)$, a CMSO formula $Diam_d$ which is true on a graph G' if and only if G' has diameter at most d. Additionally, using Kuratowski's theorem, and the fact that minor relation is CMSO expressible, one can define a CMSO formula *Planar* which is true on a graph G' if and only if G' is planar. Finally, it can be shown that any planar graph of diameter at most d has treewidth $O(d)$. Therefore, by setting $\varphi = Diam_d \wedge Planar$, $t = O(d)$, and by renaming $f(|\varphi|, t)$ to $f(d)$ in Theorem 14, we have an algorithm running in time $f(d) \cdot 2^{O(\Delta \cdot d)} \cdot |G|^{O(d)}$ to determine whether G has a planar supergraph G' of diameter at most d. ◀

We note that the algorithm $\mathfrak{A}$ of Theorem 15 does not impose any restriction on the degree of a prospective supergraph G' of G. Theorem 15 can be generalized to the setting of graphs of constant genus as follows.

▶ **Theorem 16.** *There is a computable function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, and an algorithm $\mathfrak{A}$ that takes as input, positive integers d, g, and a connected graph G of maximum degree Δ, and determines in time $f(d,g) \cdot 2^{O(\Delta \cdot d)} \cdot |G|^{O(d \cdot g)}$ whether G has a supergraph G' of genus at most g and diameter at most d.*

A graph is 1-outerplanar if it can be embedded in the plane in such a way that every vertex lies in the outer face of the embedding. A graph is k-outerplanar if it can be embedded in the plane in such a way that after deleting all vertices in the outer face, the remaining graph is $(k-1)$-outerplanar. In [5] Cohen et al. have considered the k-OUTERPLANAR DIAMETER IMPROVEMENT problem (k-OPDI), a variant of the PDI problem in which the target supergraph is required to be k-outerplanar instead of planar. In particular, they have shown that the 1-OPDI problem can be solved in polynomial time. The complexity of the k-OPDI problem with respect to explicit algorithms was left as an open problem for $k \geq 2$. The next theorem states that for each fixed k, k-OPDI is strongly uniformly fixed parameter tractable with respect to the parameter d on connected graphs of bounded degree.

▶ **Theorem 17.** *There is a computable function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, and an algorithm $\mathfrak{A}$ that takes as input, positive integers d, k, and a connected graph G of maximum degree Δ, and determines in time $f(k,d) \cdot 2^{O(\Delta \cdot k)} \cdot |G|^{O(k)}$ whether G has a k-outerplanar supergraph G' of diameter at most d.*

Finally, the SERIES-PARALLEL DIAMETER IMPROVEMENT problem (SPDI) consists in determining whether a graph G has a series parallel supergraph of diameter at most d. The parameterized complexity of this problem was left as an open problem in [5]. The next theorem states that SPDI is strongly uniformly fixed parameter tractable with respect to the parameter d on connected graphs of bounded degree.

▶ **Theorem 18.** *There is a computable function $f : \mathbb{N} \to \mathbb{N}$, and an algorithm $\mathfrak{A}$ that takes as input, a positive integer d and a connected graph G of maximum degree Δ, and determines in time $f(d) \cdot 2^{O(\Delta)} \cdot |G|^{O(1)}$ whether G has a series-parallel supergraph G' of diameter at most d.*

7.2 Contraction Bidimensional Parameters

Fomin, Golovach and Thilikos [15] have defined a sequence $\{\gneqq_k\}_{k \in \mathbb{N}}$ of graphs and have shown that these graphs serve as obstructions for small treewidth on H-minor free graphs, whenever H is an apex graph. More precisely, they have proved the following result.

▶ **Theorem 19** (Fomin-Golovach-Thilikos [15]). *For every apex graph H, there is a $c_H > 0$ such that every connected H-minor-free graph of treewidth at least $c_H \cdot k$ contains $\gneqq_k$ as a contraction.*

We say that a graph parameter $\mathbf{p}$ is Gamma-unbounded if there is a computable function $\alpha : \mathbb{N} \to \mathbb{N}$ such that $\alpha \in \omega(1)$, and $\mathbf{p}(\gneqq_k) \geq \alpha(k)$ for every $k \in \mathbb{N}$.

We say that a parameter $\mathbf{p}$ is effectively CMSO definable if there is a computable function $f : \mathbb{N} \to \mathbb{N}$, and an algorithm $\mathfrak{A}$ that takes as input a positive integer k and constructs, in time at most $f(k)$, a CMSO-sentence φ which is true on an graph G if and only if $\mathbf{p}(G) \leq k$. The following theorem is a corollary of Theorem 14 and Theorem 19.

▶ **Theorem 20.** *Let $\mathbf{p}$ be a Gamma-unbounded effectively CMSO definable graph parameter, and let $\mathcal{P}$ be a CMSO definable graph property excluding some apex graph H as a minor.*

Then there is a computable function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm $\mathfrak{A}$ that takes as input a positive integer k, and a connected graph G of maximum degree Δ, and determines, in time $f(k) \cdot 2^{O(\Delta \cdot f(k))} \cdot |G|^{f(k)}$, whether G has a supergraph G' such that $G' \in \mathcal{P}$ and $\mathbf{p}(G') \leq k$.

Note that similarly to the case of diameter improvement problem, if $\mathbf{p}$ is an unbounded effectively CMSO definable graph parameter, then we can determine whether a graph G has an r-outerplanar supergraph G' with $\mathbf{p}(G') \leq k$ in time $f(r,k) \cdot 2^{O(\Delta \cdot r)} \cdot |G|^{O(r)}$ for some computable function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. In other words, this problem, for connected bounded degree graphs, is strongly uniformly fixed parameter tractable with respect to the parameter $\mathbf{p}$ for each fixed r.

▶ **Definition 21.** A graph parameter $\mathbf{p}$ is contraction-bidimensional if the following conditions are satisfied.

1. $\mathbf{p}$ is contraction-closed.
2. If G is a graph which has $\gneqq_k$ as a contraction, then $\mathbf{p}(G) \geq \Omega(k^2)$.

For instance, the following parameters are contraction bidimensional.

1. Size of a vertex cover.
2. Size of a feedback vertex set.
3. Size of a minimum maximal matching.
4. Size of a dominating set.
5. Size of a edge dominating set.
6. Size of a clique traversal set.

▶ **Theorem 22** ([15, 16]). *Let $\mathbf{p}$ be a bidimensional parameter. Then if $\mathbf{p}(G) \leq k$, the treewidth of $\mathbf{p}$ is at most $O(\sqrt{k})$.*

▶ **Theorem 23.** *For each effectively CMSO-definable contraction-bidimensional parameter $\mathbf{p}$, there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm $\mathfrak{A}$ that takes as input a positive integer k, and a connected graph G of maximum degree Δ, and determines in time $f(k) \cdot 2^{O(\Delta \cdot \sqrt{k})} \cdot |G|^{O(\sqrt{k})}$ whether G has a planar supergraph G' with $\mathbf{p}(G') \leq k$.*

For instance, Theorem 23 states that for some computable function $f : \mathbb{N} \to \mathbb{N}$, one can determine in time $f(k) \cdot 2^{O(\Delta \cdot \sqrt{k})} \cdot |G|^{O(\sqrt{k})}$ whether G has a planar supergraph G' with feedback vertex set at most k. We note that in view of Theorem 22, the planarity requirement of Theorem 23 can be replaced for any CMSO definable property $\mathcal{P}$ which excludes some apex graph as a minor.

Acknowledgements. This work was supported by the Bergen Research Foundation. The author thanks Michael Fellows, Fedor Fomin, Petr Golovach, Daniel Lokshtanov and Saket Saurabh for interesting discussions.

References

1 Karl Abrahamson and Michael Fellows. Finite automata, bounded treewidth, and well-quasiordering. *Contemporary Mathematics*, 147:539–539, 1993.
2 Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In *Proc. of SODA 2008*, pages 641–650. SIAM, 2008.
3 Mikołaj Bojańczyk and Michal Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *Proc. of LICS 2016*, pages 407–416. ACM, 2016.
4 Mikołaj Bojańczyk and Michal Pilipczuk. Optimizing tree decompositions in MSO. In *Proc. of STACS 2017 (To appear)*, 2017.

5 Nathann Cohen, Daniel Gonçalves, Eunjung Kim, Christophe Paul, Ignasi Sau, Dimitrios M. Thilikos, and Mathias Weller. A polynomial-time algorithm for outerplanar diameter improvement. In *Proc. of the 10th International Computer Science Symposium in Russia (CSR 2015)*, volume 9139 of *LNCS*, pages 123–142, 2015.

6 H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available at `http://www.grappa.univ-lille3.fr/tata`, 2007. Release October, 12th 2007.

7 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

8 Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic. A language-theoretic approach.* HAL, June 14 2012. URL: `http://hal.archives-ouvertes.fr/hal-00646514`.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

10 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

11 Michael Elberfeld. Context-free graph properties via definable decompositions. In *Proc. of the 25th Conference on Computer Science Logic (CSL 2016)*, volume 62 of *LIPIcs*, pages 17:1–17:16, 2016.

12 Michael R. Fellows and Rodney G. Downey. Parameterized computational feasibility. *Feasible Mathematics II*, 13:219–244, 1995.

13 Michael R. Fellows and Michael A. Langston. On search decision and the efficiency of polynomial-time algorithms. In *Proc. of STOC 1989*, pages 501–512. ACM, 1989.

14 Jörg Flum, Markus Frick, and Martin Grohe. Query evaluation via tree-decompositions. *Journal of the ACM (JACM)*, 49(6):716–752, 2002.

15 Fedor V. Fomin, Petr Golovach, and Dimitrios M. Thilikos. Contraction obstructions for treewidth. *Journal of Combinatorial Theory, Series B*, 101(5):302–314, 2011.

16 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *Proc. of SODA 2010*, pages 503–510, 2010.

17 Neil Robertson and Paul D. Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995.

18 Neil Robertson and Paul D Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.