

Abu Zaid, Faried:

Uniformly automatic classes of finite structures

Original published in: 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science / IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science 38, 2018 Ahmedabad, India. - Saarbrücken/Wadern : Schloss Dagstuhl - Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, December, 2018. - (2018), art. 10, 21 pp.
ISBN 978-3-95977-093-4
(Leibniz International Proceedings in Informatics (LIPIcs) ; 122)

Original published: December 2018

ISSN: 1868-8969

DOI: [10.4230/LIPIcs.FSTTCS.2018.10](https://doi.org/10.4230/LIPIcs.FSTTCS.2018.10)

[Visited: 2020-03-05]



This work is licensed under a [Creative Commons Attribution 3.0 Unported license](https://creativecommons.org/licenses/by/3.0/). To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/>

Uniformly Automatic Classes of Finite Structures

Fariied Abu Zaid¹

Camelot Management Consultants, CoE Artificial Intelligence for Information Management,
Munich, Germany
faza@camelot-mc.com

Abstract

We investigate the recently introduced concept of uniformly tree-automatic classes in the realm of parameterized complexity theory. Roughly speaking, a class of finite structures is uniformly tree-automatic if it can be presented by a set of finite trees and a tuple of automata. A tree t encodes a structure and an element of this structure is encoded by a labeling of t . The automata are used to present the relations of the structure. We use this formalism to obtain algorithmic meta-theorems for first-order logic and in some cases also monadic second-order logic on classes of finite Boolean algebras, finite groups, and graphs of bounded tree-depth. Our main concern is the efficiency of this approach with respect to the hidden parameter dependence (size of the formula). We develop a method to analyze the complexity of uniformly tree-automatic presentations, which allows us to give upper bounds for the runtime of the automata-based model checking algorithm on the presented class. It turns out that the parameter dependence is elementary for all the above mentioned classes. Additionally we show that one can lift the FPT results, which are obtained by our method, from a class \mathcal{C} to the closure of \mathcal{C} under direct products with only a singly exponential blow-up in the parameter dependence.

2012 ACM Subject Classification Theory of computation \rightarrow Finite Model Theory, Theory of computation \rightarrow Fixed parameter tractability, Theory of computation \rightarrow Formal languages and automata theory

Keywords and phrases Automatic Structures, Model Checking, Fixed-Parameter Tractability, Algorithmic Meta Theorems

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2018.10

1 Introduction

In this paper we investigate the use of automata in *algorithmic meta-theorems*. Algorithmic meta-theorems are general algorithmic results stating that a class of problems \mathcal{P} can be efficiently solved on a class of instances \mathcal{C} . In many cases \mathcal{P} is the class of problems definable in a certain logic \mathcal{L} . *Parameterised complexity theory* provides one of the key notions to establish algorithmic meta-theorems: we say that the model checking problem for a logic \mathcal{L} on a class of structures \mathcal{C} is fixed-parameter tractable (FPT) (in the size of the formula) if there is a computable function f and a constant c such that we can decide for every $\varphi \in \mathcal{L}$ and every $\mathfrak{A} \in \mathcal{C}$ in time $f(|\varphi|) \cdot |\mathfrak{A}|^c$ whether $\mathfrak{A} \models \varphi$.

Prototypical examples of automata-based algorithmic meta-theorems are the theorem of Courcelle [5] for MSO-definable problems (actually MSO₂, which has the additional capability to quantify over subsets of the edge relation) on graphs of bounded treewidth and the result of Courcelle, Makowsky, and Rotics [4] for MSO-definable problems on graphs of bounded cliquewidth. The basic idea is in both cases to compute from a graph \mathfrak{G} a tree-like

¹ The presented work was conducted while the author was affiliated with the Technische Universität Ilmenau.



decomposition $t_{\mathfrak{G}}$ and from an MSO-formula φ a tree-automaton \mathcal{A}_φ that accepts exactly the tree-like decompositions of graphs that model φ . Since the construction of $t_{\mathfrak{G}}$ from \mathfrak{G} can be performed efficiently, we can efficiently check if $\mathfrak{G} \models \varphi$ by checking if \mathcal{A}_φ accepts $t_{\mathfrak{G}}$. Note that many NP-complete problems, such as 3-Colourability, are definable in MSO and hence efficiently solvable on the above mentioned classes.

The idea to present structures by automata is also the basis for the field of automatic structures. Roughly speaking, a structure is called automatic if its domain can be represented as a regular set in such a way that its relations become recognisable by synchronous multi-tape automata. However, it is not very interesting to study automatic presentations on the class of all finite structures since every finite structure has an automatic presentation (since all finite languages are regular). Recently the concept of uniformly automatic classes was introduced in [1]. In this setting the automata obtain an additional input (called advice) which encodes the structure that should be presented. Therefore it is possible to present a whole class of structures by a single presentation and a set of advices. Contrary to the classical case without advice it is indeed very interesting to ask which classes of finite structures have a uniformly automatic presentation and which algorithmic consequences can be derived from the existence of such a presentation for a given class.

From a logical point of view it is worthwhile to mention that the presentations which build the core of the FPT algorithms for bounded treewidth and bounded cliquewidth graphs are obtained from MSO-interpretations on trees. Uniformly automatic presentations, however, correspond to so called set-interpretations, which are strictly more powerful than MSO-interpretations. In fact, it is not hard to construct even uniformly word-automatic classes of graphs which have unbounded tree- and cliquewidth. The power to present more complex classes of structures comes with the trade-off that we have to restrict our consideration to FO model checking instead of MSO model checking. Still having a fixed parameter tractable model checking problem for a class of structures directly leads to FPT results for many other interesting algorithmic problems. For instance, if FO model checking is FPT on a class of graphs \mathcal{C} then Independent Set is FPT on \mathcal{C} in the size of the independent set because for every $k \in \mathbb{N}$ and every graph \mathfrak{G} it holds that $\mathfrak{G} \models \exists x_1, \dots, x_k (\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \bigwedge_{i < j} \neg Ex_i x_j)$ if and only if \mathfrak{G} contains an independent set of size k .

Meta-theorems for first-order logic have been studied extensively on classes of sparse graphs. The first result in this direction is due to Seese for graphs of bounded degree [23]. Over the past decades larger and larger classes of sparse graphs have been identified for which FO model checking is FPT. This development has recently found its climax in the result of Grohe, Kreutzer, and Siebertz for nowhere dense graphs [15]. They proved that under certain complexity theoretic assumptions this is the largest possible subgraph-closed class of graphs where FO model checking is FPT.

We investigate automaticity as a generic notion of simplicity which might bring up new and interesting classes of structures for which FO model checking is FPT. Towards the theory, we are concerned with the efficiency of this approach. The concept of fixed parameter tractability is often criticized since there are no constraints on the complexity of the parameter. Note that in general the non-elementary worst-case runtime of the automaton construction process leads to a non-elementary parameter dependence in the algorithmic meta-theorems. Frick and Grohe [12] showed, unless PTIME = NP, there is no algorithm that solves the model checking problem for MSO on words or trees in time $f(|\varphi|) \cdot \text{poly}(|t|)$ for any elementary function $f: \mathbb{N} \rightarrow \mathbb{N}$. A similar statement holds for FO on words. As trees have treewidth one, this renders Courcelle's approach to model checking of graphs with bounded treewidth optimal. Moreover, the efficiency of the automata theoretic approach has

also been confirmed in practice. For instance, Langer et al. [20] implemented Courcelle’s technique and found that their implementation can compete with other approaches for specific problems such as Dominating Set.

Even more interestingly, the automata-based approach also tends to behave tamely when applied to interpretations of structures whose theory is elementary. Eisinger [8] gave a triply-exponential upper bound on the size of the minimal automaton for formulae of integer and mixed-real addition. In [6] Durand-Gasselin and Habermehl showed for word-automatic structures that the runtime of the generic algorithm can be bounded by a function which estimates how well the presentation goes along with the Ehrenfeucht-Fraïssé relations of the structure and gave runtime bounds for integer addition matching Eisinger’s bound. Additionally they gave a triply-exponential bound for automatic graphs of bounded degree complementing a result by Kuske and Lohrey who proved, using a specialised algorithm, that model checking for automatic graphs of bounded degree is solvable in doubly-exponential space [19].

We adopt Durand-Gasselin’s and Habermehl’s technique and generalise their result to uniformly tree-automatic presentations. We apply this technique to the presentations that arise as the presentations of the direct product closures of uniformly tree-automatic classes. We prove that the bound of the runtime of the model checking algorithm is at most exponential in the bound of the runtime for the primal classes. Further we apply these findings in the context of FPT model checking for first order logic. We demonstrate the efficiency of the automata-theoretic approach by analysing the runtime in terms of the parameter dependence on structurally rather simple classes. Our results are as follows:

- FO model checking is FPT on the class of all finite Boolean algebras that are succinctly encoded by the number of atoms and can be performed in $\exp_2(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{B}|$. Unless $\text{NEXP} = \bigcup_{c \in \mathbb{N}} \text{STA}(*, 2^{cn}, n)$, this parameter dependence is optimal.
- FO model checking is FPT on the class of all finite abelian groups that are succinctly encoded by the orders of the direct product factors and can be performed in $\exp_4(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{G}|$. We generalise this result to finite groups of *bounded non-abelian decomposition width*, that is groups whose non-abelian direct product factors are of bounded size. We obtain the same asymptotic runtime on these classes.

This provides some first results towards Grohe’s question on which classes of algebraic structures FO model checking is FPT [14]. The mere FPT result for FO model checking on abelian groups was independently also discovered by Bova and Martin [2]. Their algorithm assumes that the groups are encoded by their multiplication tables and yields a non-elementary parameter dependence. Therefore our approach has the two advantages that it works for succinct encodings and yields an elementary parameter dependence.

- MSO model checking is FPT on every class of graphs with tree-depth at most h and can be performed in $\exp_{h+2}(\text{poly}(|\varphi|)) \cdot \text{poly}(|\mathfrak{G}|)$. This matches the runtime of the best known algorithm for these classes, which is due to Gajarsky and Hliněný [13]. Their algorithm uses a kernelisation procedure. Our proof makes use of their analysis.

2 Preliminaries

For natural numbers ℓ, m, n we write $m =_{\ell} n$ if $m = n$ or $m, n \geq \ell$. We assume that the reader is familiar with first-order logic (FO) as well as with the connection between monadic second-order logic (MSO) and tree-automata. Therefore we use this section mainly to fix our notation.

A **signature** is a finite set of relation symbols $\tau = \{R_1, \dots, R_k\}$, where every symbol $R_i \in \tau$ has an assigned arity r_i . A τ -**structure** is a tuple $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \dots, R_k^{\mathfrak{A}})$, where A is a set and $R_i^{\mathfrak{A}} \subseteq A^{r_i}$ for all $i \in \{1, \dots, k\}$. From now on we will tacitly assume that all structures under consideration are finite. The class of all finite τ -structures is denoted by $\text{Str}(\tau)$.

Let Σ, Γ be alphabets. A **(labeled binary) tree** is a function $t: \text{dom}_t \rightarrow \Sigma$, where $\text{dom}_t \subseteq \{0, 1\}^*$ is a finite prefix-closed set. The set of all trees with labels from Σ is denoted by T_Σ . Let $t_1 \in T_\Sigma, t_2 \in T_\Gamma$ with $\text{dom}_{t_1} = \text{dom}_{t_2} =: \text{dom}$. The **convolution** $t_1 \otimes t_2 \in T_{\Sigma \times \Gamma}$ is defined by $(t_1 \otimes t_2)(w) = (t_1(w), t_2(w))$ for all $w \in \text{dom}$. When we apply the convolution to several trees at once we will often write $\langle t_1, t_2, \dots, t_k \rangle$ instead of $t_1 \otimes t_2 \otimes \dots \otimes t_k$. A reader that is familiar with automatic presentations might notice that we define the convolution only for trees with the same domain. This allows us to circumvent the introduction of a padding symbol. For trees t, t_1, \dots, t_n and pairwise distinct $w_1, \dots, w_n \in \text{dom}_t$ we define $t[w_1/t_1, \dots, w_n/t_n]$ to be the tree which is obtained by replacing the subtree rooted in w_i by t_i for all $i \leq n$.

A Σ -**context** is a tree $c \in T_{\Sigma \uplus \{x\}}$ such that all nodes except for exactly one *leaf* w are labeled with letters from Σ and $c(w) = x$. The unique leaf w with label x is denoted by $c^{-1}(x)$. For a Σ -context c and a tree $t \in T_\Sigma$ the **composition** $(c \circ t) \in T_\Sigma$ is defined as $c[c^{-1}(x)/t]$.

Let us now introduce tree-automata with advice. Formally, these are just ordinary tree-automata which read letters from a composed alphabet. But since our automatic presentations will assign special semantics to the first component of such a letter it makes sense to handle these components differently in our notation.

► **Definition 2.1.** A **(deterministic bottom-up) tree-automaton with advice** is a finite state tree-automaton $\mathcal{A} = (Q, \Sigma \times \Gamma, \delta, F)$. The language that \mathcal{A} recognizes with advice $\alpha \in T_\Sigma$ is $L(\mathcal{A}[\alpha]) = \{t \in T_\Gamma \mid \text{dom}_t = \text{dom}_\alpha \wedge \alpha \otimes t \in L(\mathcal{A})\}$. A tree-language T is called regular with advice α if there is a tree-automaton \mathcal{A} with advice such that $T = L(\mathcal{A}[\alpha])$.

For the sake of brevity we usually just speak of an automaton instead of a tree-automaton with advice. The complement automaton of \mathcal{A} is denoted by $\overline{\mathcal{A}} = (Q, \Sigma \times \Gamma, \delta, Q \setminus F)$. Finally we define uniformly tree-automatic presentations.

► **Definition 2.2.** Let $\tau = \{R_1, \dots, R_k\}$ be a finite relational signature. A **uniformly tree-automatic presentation** of a class of τ -structures is a tuple $\mathfrak{c} = (\mathcal{A}, \mathcal{A}_{R_1}, \dots, \mathcal{A}_{R_k})$ of tree-automata with advice such that $L(\mathcal{A}[\alpha]) \subseteq T_{\{0,1\}}$ and $L(\mathcal{A}_{R_i}[\alpha]) \subseteq \{\langle t_1, \dots, t_{r_i} \rangle \mid t_1, \dots, t_{r_i} \in L(\mathcal{A}[\alpha])\}$ for all $\alpha \in T_\Sigma$ and all $i \in \{1, \dots, k\}$. Each $\alpha \in T_\Sigma$ with $L(\mathcal{A}[\alpha]) \neq \emptyset$ presents (the isomorphism type of) a structure $\mathcal{S}(\mathfrak{c}[\alpha]) := \left(L(\mathcal{A}[\alpha]), \left(R_i^{\mathcal{S}(\mathfrak{c}[\alpha])} \right)_{1 \leq i \leq k} \right)$, where $R_i := \{\langle t_1, \dots, t_{r_i} \rangle \mid \langle t_1, \dots, t_{r_i} \rangle \in L(\mathcal{A}_{R_i}[\alpha])\}$. The set $\{\alpha \in T_\Sigma \mid L(\alpha) \neq \emptyset\}$ of all advices that present a structure with respect to \mathfrak{c} is denoted by $P^{\mathfrak{c}}$. The class that is presented by \mathfrak{c} is $\{\mathcal{S}(\mathfrak{c}[\alpha]) \mid \alpha \in P^{\mathfrak{c}}\}$.

3 Model Checking Revisited

Since the class of all words is uniformly tree-automatic it is clear by the previously mentioned result of Frick and Grohe [12] that every algorithm that solves the model checking problem for structures given by a uniformly tree-automatic presentation has an unavoidable non-elementary worst-case runtime behaviour. On the other hand, for many important examples of automatic structures the situation is much better. For instance it is known that the

Algorithm 1 Model Checking on Uniformly Tree-Automatic Classes.

Input: Tree-automatic presentation $\mathbf{c} = (\mathcal{A}, (\mathcal{A}_R)_{R \in \tau})$, FO-formula $\varphi(x_1, \dots, x_m)$
Output: Tree-automaton \mathcal{A}_φ

```

1: procedure COMPOSE( $\mathbf{c}, \varphi$ )
2:   if  $\varphi(x_1, \dots, x_m) = R(x_{i_1}, \dots, x_{i_k}), R \in \tau \cup \{=\}$  then
3:      $\mathcal{A}'_R \leftarrow \text{EXTEND}(\mathcal{A}_R, m, i_1, \dots, i_k)$ 
4:      $\mathcal{A}_D \leftarrow \text{DOMAIN}(\mathcal{A}, m)$ 
5:      $\mathcal{A}_\varphi \leftarrow \text{INTERSECT}(\mathcal{A}'_R, \mathcal{A}_D)$ 
6:     minimise  $\mathcal{A}_\varphi$ 
7:     return  $\mathcal{A}_\varphi$ 
8:   else if  $\varphi(x_1, \dots, x_m) = \psi(x_1, \dots, x_m) \wedge \theta(x_1, \dots, x_m)$  then
9:      $\mathcal{A}_\psi \leftarrow \text{COMPOSE}((\mathcal{A}, (\mathcal{A}_R)_{R \in \tau}), \psi)$ 
10:     $\mathcal{A}_\theta \leftarrow \text{COMPOSE}((\mathcal{A}, (\mathcal{A}_R)_{R \in \tau}), \theta)$ 
11:    return  $\text{INTERSECT}(\mathcal{A}_\psi, \mathcal{A}_\theta)$ 
12:   else if  $\varphi(x_1, \dots, x_m) = \neg\psi(x_1, \dots, x_m)$  then
13:      $\mathcal{A}_\psi \leftarrow \text{COMPOSE}((\mathcal{A}, (\mathcal{A}_R)_{R \in \tau}), \psi(x_1, \dots, x_m))$ 
14:      $\mathcal{A}_D \leftarrow \text{DOMAIN}(\mathcal{A}, r)$ 
15:     return  $\text{INTERSECT}(\overline{\mathcal{A}_\psi}, \mathcal{A}_D)$ 
16:   else if  $\varphi(x_1, \dots, x_m) = \exists x_{m+1} : \psi(x_1, \dots, x_{m+1})$  then
17:      $\mathcal{A}_\psi \leftarrow \text{COMPOSE}((\mathcal{A}, (\mathcal{A}_R)_{R \in \tau}), \psi(x_1, \dots, x_m))$ 
18:      $\mathcal{A}'_\varphi \leftarrow \text{PROJECT}(\mathcal{A}_\psi)$ 
19:      $\mathcal{A}_\varphi \leftarrow \text{DETERMINIZE}(\mathcal{A}'_\varphi)$ 
20:     return  $\mathcal{A}_\varphi$ 
21:   end if
22: end procedure

```

first-order theory of Presburger Arithmetic can be decided in three-fold exponential time [21]. It is therefore very natural to analyse the runtime of a given model checking algorithm for automatic structures with respect to some fixed presentation.

In [6] Durand-Gasselin and Habermehl proposed a method to estimate the time that the generic automata based model checking algorithm for structures given by a *word-automatic* presentation needs when it is used to solve the first order theory of a single structure. They showed that for certain presentations of $(\mathbb{Z}, +)$ the running time of the algorithm is only triply exponential in the formula. Similar bounds were established for arbitrary word-automatic presentations of structures of bounded degree.

In the following we want to extend their method to *uniformly tree-automatic* presentations of *classes of structures*. Fortunately this generalization goes through very well because of the nice analogue of the Myhill-Nerode congruence for regular tree-languages.

We start with a detailed description of the model checking algorithm on structures given by an advice α from a uniform tree-automatic presentation \mathbf{c} . Up to small optimizations it resembles the standard algorithm that constructs from \mathbf{c}, α , and $\varphi(x_1, \dots, x_m)$ an automaton \mathcal{A}_φ with $L(\mathcal{A}_\varphi) = \{\langle \alpha, t_1, \dots, t_m \rangle \mid \mathcal{S}(\mathbf{c}[\alpha]) \models \varphi(t_1, \dots, t_m)\}$ by recursion over the structure of φ . The exact procedure is given by Algorithm 1.

The subroutine $\text{EXTEND}(\mathcal{A}_R, m, i_1, \dots, i_k)$ computes the minimal automaton that checks on input $\langle \alpha, t_1, \dots, t_m \rangle$ if $\langle \alpha, t_{i_1}, \dots, t_{i_k} \rangle \in L(\mathcal{A}_R)$, that is if $(t_{i_1}, \dots, t_{i_k}) \in R^{\mathcal{S}(\mathbf{c}[\alpha])}$. The subroutine DOMAIN constructs the minimal tree-automaton that recognises exactly those trees in $T_{\Sigma \times \Gamma^m}$ that are convolutions of trees $t_0 \in T_\Sigma$ and $t_1, \dots, t_m \in T_\Gamma$ such that $t_1, \dots, t_m \in$

$\mathcal{S}(c[t_0])$. The subroutine $\text{INTERSECT}(\mathcal{A}_1, \mathcal{A}_2)$ uses the standard product construction to obtain an automaton \mathcal{A}_\cap with $L(\mathcal{A}_\cap) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. Note that it is crucial for the runtime analysis in the following section that we only construct the reachable states of the product automaton. Finally $\text{PROJECT}(\mathcal{A})$ applies the projection $(\sigma, \gamma_1, \dots, \gamma_{m+1}) \mapsto (\sigma, \gamma_1, \dots, \gamma_m)$ to the input alphabet of \mathcal{A} , which yields a non-deterministic automaton, and $\text{DETERMINIZE}(\mathcal{A})$ uses the standard determinization procedure for tree-automata (again omitting non-reachable states).

4 A Presentation Aware Runtime Analysis

The main ingredient for the runtime analysis of Algorithm 1 is the marriage of the Ehrenfeucht-Fraïssé relations (EF-relations) on the presented class of structures and the Myhill-Nerode congruences on the languages which form the presentation. Ehrenfeucht-Fraïssé relations were introduced by Fraïssé in his seminal work [11] as a purely combinatorial characterisation of elementary equivalence. His ideas were later popularised by the appealing game-theoretic presentation given by Ehrenfeucht in [7]. Even the possibility to bound the complexity of certain logical theories using EF-relations was already present in these early works. This technique was later systematically studied by Ferrante and Rackoff (see [9]). They used EF-relations to give upper bounds on the complexity of first-order theories like Presburger Arithmetic or the theory of one-to-one functions.

Klaetke used in [18] the ideas of Ferrante and Rackoff to bound the size of the automata for linear arithmetic $(\mathbb{R}, +, <)$. Eisinger picked up the techniques and showed in [8] similar bounds for a certain automata based presentation of mixed integer and mixed real addition, respectively (we remark here that his way of presenting the structures by automata differs slightly from our definition of an automatic presentation). Durand-Gasselín and Habermehl recently showed that if a refinement of the EF-relations for a structure \mathfrak{A} is compatible with an automatic presentation of \mathfrak{A} in the sense that these relations are congruences on the encodings of the elements (with respect to concatenation) then the runtime of the standard algorithm for solving the theory of an automatic structure can be bounded in terms of the index of these relations. In this section we build upon their work and generalise their result to classes with a uniform tree-automatic presentation. Therefore it is necessary to develop a suitable notion of EF-congruences for our purposes. Besides switching from automatic presentations to uniform tree-automatic presentations, there are a few subtle differences to the definition in [6] in order to make the technique applicable for more presentations.

Let Σ be an advice alphabet and Γ be an input alphabet. In the following we write $\hat{\Sigma}_m$ for $\Sigma \times \Gamma^m$

► **Definition 4.1.** Let $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R \in \tau})$ be a uniformly tree-automatic presentation of a class $\mathcal{C} \subseteq \text{Str}(\tau)$. An Ehrenfeucht-Fraïssé congruence (EF-congruence) for \mathfrak{c} is a collection of equivalence relations $(E_m^r)_{r, m \in \mathbb{N}}$, where $E_m^r \subseteq T_{\hat{\Sigma}_m} \times T_{\hat{\Sigma}_m}$ and for all $r, m \in \mathbb{N}$:

1. The relation E_m^r separates the trees in $T_{\hat{\Sigma}_m}$ that are a convolution of a tuple $(\alpha, t_1, \dots, t_m)$ such that (t_1, \dots, t_m) represents a tuple of elements in $\mathcal{S}(\mathfrak{c}[\alpha])$ from those trees in $T_{\hat{\Sigma}_m}$ that are not the convolution of such a tuple.
2. If $t_1, \dots, t_m \in \mathcal{S}(\mathfrak{c}[\alpha])$, $t'_1, \dots, t'_m \in \mathcal{S}(\mathfrak{c}[\beta])$, and $\langle \alpha, \bar{t} \rangle E_m^0 \langle \beta, \bar{t}' \rangle$ then (t_1, \dots, t_m) and (t'_1, \dots, t'_m) satisfy the same atomic formulas in $\mathcal{S}(\mathfrak{c}[\alpha])$ and $\mathcal{S}(\mathfrak{c}[\beta])$, respectively.
3. If $s E_m^{r+1} s'$ for some $s, s' \in T_{\hat{\Sigma}_m}$ then for all $t \in T_\Gamma$ there exists a $t' \in T_\Gamma$ such that $\langle s, t \rangle E_{m+1}^r \langle s', t' \rangle$.
4. The relation E_m^r respects contexts, i.e. if $t E_m^r t'$ for some $t, t' \in T_{\hat{\Sigma}_m}$ then for all $\hat{\Sigma}_m$ -contexts c the trees $c \circ t$ and $c \circ t'$ are also related by E_m^r .

For a function $f : \mathbb{N} \rightarrow \mathbb{N}$ we say that an EF-congruence $(E_m^r)_{r,m \in \mathbb{N}}$ is $f(r+m)$ bounded if the index of E_m^r is bounded by $f(r+m)$ for all $r, m \in \mathbb{N}$.

The EF-congruence $(E_m^r)_{r,m \in \mathbb{N}}$ for a presentation \mathfrak{c} refines the first-order indistinguishably relations $(\equiv_r)_{r \in \mathbb{N}}$ on the presented class \mathcal{C} (recall that \equiv_r means indistinguishable by formulas up to quantifier rank r). This can be shown using standard game theoretic arguments.

► **Lemma 4.2.** *Let \mathfrak{c} be a uniform tree-automatic presentation of a class \mathcal{C} and $(E_m^r)_{r,m \in \mathbb{N}}$ an EF-congruence with respect to \mathfrak{c} . Then for all $\alpha, \alpha' \in P^{\mathfrak{c}}$ and $t_1, t'_1, \dots, t_m, t'_m$ with $t_1, \dots, t_m \in \mathcal{S}(\mathfrak{c}[\alpha])$ and $t'_1, \dots, t'_m \in \mathcal{S}(\mathfrak{c}[\alpha'])$ the following is true:*

$$\langle \alpha, t_1, \dots, t_m \rangle E_m^r \langle \alpha', t'_1, \dots, t'_m \rangle \Rightarrow (\mathcal{S}(\mathfrak{c}[\alpha]), t_1, \dots, t_m) \equiv_r (\mathcal{S}(\mathfrak{c}[\alpha']), t'_1, \dots, t'_m).$$

As mentioned before, an EF-congruence with respect to some parametrized tree-automatic presentation connects the Myhill-Nerode-congruences of the languages involved in the presentation with the EF-relations on the presented class. We want to show that the runtime of Algorithm 1 largely depends on how well these relations play along with each other.

► **Theorem 4.3.** *Let $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R \in \tau})$ be a uniformly tree-automatic presentation of a class of τ -structures. Suppose there is an $f(r+m)$ bounded EF-congruence $(E_m^r)_{r,m \in \mathbb{N}}$ for \mathfrak{c} . Then for every $\varphi(x_1, \dots, x_m) \in \text{FO}$ of quantifier rank r Algorithm 2 computes the automaton \mathcal{A}_φ in time $\mathcal{O}(|\varphi|(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c)$ for some constant c .*

The proof is similar to [6]. We omit it here due to space constraints. In the following section we will be concerned with classes of finite structures that arise as the closure under direct products of a certain prime class. It is not hard to see that if a class \mathcal{C} is uniformly tree-automatic then the same holds for the closure of \mathcal{C} under direct products (see also [1]). We close this section by showing that also the EF-congruences can (with a certain blow up of the index) be lifted from the original presentation to a certain presentation of the direct product closure.

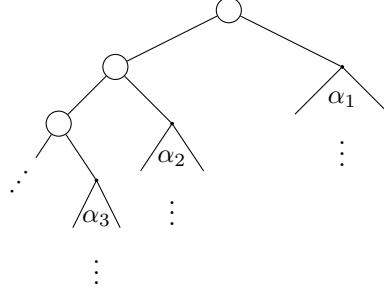
► **Definition 4.4.** Let \mathcal{C} be a class of τ -structures. Then \mathcal{C}^\times denotes the closure of \mathcal{C} under direct products. That is $\mathcal{C}^\times = \{\mathfrak{A}_1 \times \dots \times \mathfrak{A}_n \mid n \geq 1, \mathfrak{A}_1, \dots, \mathfrak{A}_n \in \mathcal{C}\}$.

► **Lemma 4.5** (Abu Zaid, Grädel, Reinhardt [1]). *Let \mathcal{C} be a uniformly tree-automatic class of structures. From a given tree-automatic presentation \mathfrak{c} of \mathcal{C} one can effectively construct tree-automatic presentations \mathfrak{c}^\times of \mathcal{C}^\times .*

Proof. Construction of $(P^\times, \mathfrak{c}^\times)$: Suppose \mathcal{C} is presented by the uniform tree-automatic presentation \mathfrak{c} over the advice set P . As the construction is rather straightforward we only give the parameter set for the presentation and the idea for the encoding. The parameter set consists of all trees where the right child of every node in the left-most branch induces a subtree which is in P . This is depicted in Figure 1. Such an advice presents the structure $\mathcal{S}(\mathfrak{c}[\alpha_1]) \times \mathcal{S}(\mathfrak{c}[\alpha_2]) \times \dots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$. Let t_1, \dots, t_n be elements of $\mathcal{S}(\mathfrak{c}[\alpha_1]), \dots, \mathcal{S}(\mathfrak{c}[\alpha_n])$, respectively. Then the element (t_1, \dots, t_n) is put together in the same way as the advices. ◀

In order to ease the process of analyzing the complexity of these presentations, we introduce some notations. Let Γ be an alphabet with $\# \notin \Gamma$. The n -context-tree $t_n^\#$ is the tree with domain $\text{dom}(t_n^\#) = \{0^k \mid k < n\} \cup \{0^k 1 \mid k + 1 < n\}$ and labeling

$$t_n^\#(w) = \begin{cases} \# & ; \text{if } w \in \{0\}^{<n} \\ c_{i+1} & ; \text{if } w = 0^i 1, \text{ with } 0 \leq i < n-1 \\ c_n & ; \text{if } w = 0^{n-1}. \end{cases}$$



■ **Figure 1** The Parameters for the class \mathcal{C}^\times .

With $T_\Gamma^{\#,n}$ we denote the set of all trees that are obtained from $t_n^\#$ by replacing all contexts with trees from T_Γ , that is $T_\Gamma^{\#,n} = \{t_n^\#[c_1/t_1, \dots, c_n/t_n] \mid t_1, \dots, t_n \in T_\Gamma\}$. Finally let $T_\Gamma^\#$ be the union of all sets $T_\Gamma^{\#,n}$ with $n \geq 1$.

► **Theorem 4.6.** *Let \mathfrak{c} be a uniformly tree-automatic presentation of a class \mathcal{C} with associated $f(r+m)$ bounded EF-congruences $(E_m^r)_{r,m \in \mathbb{N}}$. Then there is a uniformly tree-automatic presentation of \mathcal{C}^\times with associated $2^{\mathcal{O}((r+m)f(r+m)\log(f(r+m)))}$ bounded EF-congruences.*

Proof. Let \mathfrak{c}^\times be the presentation of \mathcal{C}^\times that is derived from \mathfrak{c} by the construction from Lemma 4.5. Recall that if P is the set of advice trees for the presentation \mathfrak{c} and $\alpha_1, \dots, \alpha_n \in P$, then the structure $\mathcal{S}(\mathfrak{c}[\alpha_1]) \times \dots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$ is presented by the advice $t_n^\#[c_1/\alpha_1, \dots, c_n/\alpha_n]$ and an element $(t_1, \dots, t_n) \in \mathcal{S}(\mathfrak{c}[\alpha_1]) \times \dots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$ is represented by the tree $t_n^\#[c_1/t_1, \dots, c_n/t_n]$, where $\#$ is a newly introduced letter.

For all $r, m \in \mathbb{N}$ we define a relation \sim_m^r on $T_{(\Sigma \cup \{\#\})} \times (\Gamma \cup \{\#\})^m$, where $t \sim_m^r t'$ if, and only if, one of the following conditions is true:

1. There are no n, n' such t and t' are the convolution of well-formed trees $\alpha \in T_\Sigma^{\#,n}, t_1, \dots, t_m \in T_\Gamma^{\#,n}$ and $\alpha' \in T_\Sigma^{\#,n'}, t'_1, \dots, t'_m \in T_\Gamma^{\#,n'}$, respectively.
2. There are n, n' such t and t' are the convolution of well-formed trees $\alpha \in T_\Sigma^{\#,n}, t_1, \dots, t_m \in T_\Gamma^{\#,n}$ and $\alpha' \in T_\Sigma^{\#,n'}, t'_1, \dots, t'_m \in T_\Gamma^{\#,n'}$, respectively. That is we can write $t = \langle t_n^\#[c_1/\alpha_1, \dots, c_n/\alpha_n], t_n^\#[c_1/t_{1,1}, \dots, c_n/t_{1,n}], \dots, t_n^\#[c_1/t_{m,1}, \dots, c_n/t_{m,n}] \rangle$ and also $t' = \langle t_{n'}^\#[c_1/\alpha'_1, \dots, c_{n'}/\alpha'_{n'}], t_{n'}^\#[c_1/t'_{1,1}, \dots, c_{n'}/t'_{1,n'}], \dots, t_{n'}^\#[c_1/t'_{m,1}, \dots, c_{n'}/t'_{m,n'}] \rangle$. Then $t \sim_m^r t'$ if for all E_m^r equivalence classes κ : $|\{i \mid 1 \leq i \leq n, [\langle \alpha_i, t_{1,i}, \dots, t_{m,i} \rangle]_{E_m^r} = \kappa\}| =_{f(r+m)^r} |\{i \mid 1 \leq i \leq n', [\langle \alpha'_i, t'_{1,i}, \dots, t'_{m,i} \rangle]_{E_m^r} = \kappa\}|$.

One easily checks that \sim_m^r is an equivalence relation with index bounded by

$$(f(r+m)^{r+m} + 1)^{f(r+m)} + 1 \in 2^{\mathcal{O}((r+m)f(r+m)\log f(r+m))}$$

for all $r, m \in \mathbb{N}$. What is left is to verify is that $(\sim_m^r)_{r,m \in \mathbb{N}}$ is indeed an EF-congruence of \mathfrak{c}^\times . Therefore we check that the collection $(\sim_m^r)_{r,m \in \mathbb{N}}$ has the Properties 1 - 4 described in Definition 4.1. This is done in the lemmata below.

► **Lemma 4.7.** *The relation \sim_m^r separates the trees that are the convolution of a tuple $(\alpha, t_1, \dots, t_n)$ such that (t_1, \dots, t_m) represents a tuple of elements in $\mathcal{S}(\mathfrak{c}^\times[\alpha])$ from those trees that are not the convolution of such a tuple.*

Proof. Suppose $t = \langle \alpha, t_1, \dots, t_m \rangle$ is a convolution of a tuple with $\alpha \in P^{\mathfrak{c}^\times}$ and $(t_1, \dots, t_m) \in \mathcal{S}(\mathfrak{c}^\times[\alpha])$ and suppose t' is not the convolution of such a tuple. If t' is not a convolution, then none of the two conditions holds for t and t' and they are not equivalent. Otherwise there are $n,$

$n' \geq 1$ with $t = \langle t_n^\# [c_1/\alpha_1, \dots, c_n/\alpha_n], t_n^\# [c_1/t_{1,1}, \dots, c_n/t_{1,n}], \dots, t_n^\# [c_1/t_{m,1}, \dots, c_n/t_{m,n}] \rangle$ and $t' = \langle t_{n'}^\# [c_1/\alpha'_1, \dots, c_{n'}/\alpha'_{n'}], t_{n'}^\# [c_1/t'_{1,1}, \dots, c_{n'}/t'_{1,n'}], \dots, t_{n'}^\# [c_1/t'_{m,1}, \dots, c_{n'}/t'_{m,n'}] \rangle$.

From our assumption about t and t' we know that $\alpha_i \in P^c$ and $t_{1,i}, \dots, t_{m,i} \in \mathcal{S}(c[\alpha_i])$ for all $1 \leq i \leq n$ and there is a $1 \leq j \leq n'$ with $\alpha'_j \notin P^c$ or $\alpha'_j \in P^c$ but $t'_{\ell,j} \notin \mathcal{S}(c[\alpha_i])$ for some $1 \leq \ell \leq m$.

But then $\langle \alpha_i, t_{1,i}, \dots, t_{m,i} \rangle E_m^r \langle \alpha'_j, t'_{1,j}, \dots, t'_{m,j} \rangle$, since the relation E_m^r fulfils Property 1 of Definition 4.1. Hence t and t' do not fulfil condition 2 and therefore $t \not\sim_m^r t'$. \blacktriangleleft

► **Lemma 4.8.** *If $t_1, \dots, t_m \in \mathcal{S}(c[\alpha])$, $t'_1, \dots, t'_m \in \mathcal{S}(c[\beta])$, and $\langle \alpha, \bar{t} \rangle \sim_m^0 \langle \beta, \bar{t}' \rangle$ then (t_1, \dots, t_m) and (t'_1, \dots, t'_m) satisfy the same atomic formulas in $\mathcal{S}(c[\alpha])$ and $\mathcal{S}(c[\beta])$, respectively.*

Proof. Suppose

- $\alpha = t_n^\# [c_1/\alpha_1, \dots, c_n/\alpha_n], \beta = t_k^\# [c_1/\beta_1, \dots, c_k/\beta_k] \in P^{c^\times}$,
- $t_i = t_n^\# [c_1/t_{i,1}, \dots, c_n/t_{i,n}] \in \mathcal{S}(c^\times[\alpha])$ for $i \in \{1, \dots, m\}$, and
- $t'_i = t_k^\# [c_1/t'_{i,1}, \dots, c_k/t'_{i,k}] \in \mathcal{S}(c^\times[\beta])$ for $i \in \{1, \dots, m\}$.

We show that if (t_1, \dots, t_m) and (t'_1, \dots, t'_m) do not fulfil the same atomic propositions in $\mathcal{S}(c^\times[\alpha])$ and $\mathcal{S}(c^\times[\beta])$, respectively, then they are not \sim_m^0 -equivalent. Consider an arbitrary atomic formula $Rx_{i_1} \dots x_{i_r}$ and suppose $\mathcal{S}(c^\times[\alpha]) \models Rt_{i_1} \dots t_{i_r}$ and $\mathcal{S}(c^\times[\beta]) \not\models Rt'_{i_1} \dots t'_{i_r}$. Then by definition $\mathcal{S}(c[\alpha_j]) \models Rt_{j,i_1} \dots t_{j,i_r}$ for all $1 \leq j \leq n$ but $\mathcal{S}(c[\beta_\ell]) \not\models Rt'_{\ell,i_1} \dots t'_{\ell,i_r}$ for some $1 \leq \ell \leq k$. Consequently $\langle \alpha_j, t_{j,1}, \dots, t_{j,m} \rangle E_m^0 \langle \beta_\ell, t_{\ell,1}, \dots, t_{\ell,m} \rangle$ for all $1 \leq j \leq n$ and therefore $\langle \alpha, t_1, \dots, t_m \rangle \not\sim_m^0 \langle \beta, t'_1, \dots, t'_m \rangle$. \blacktriangleleft

► **Lemma 4.9.** *If $s \sim_{m+1}^{r+1} s'$ then for all $t \in T_{(\Gamma \cup \{\#\})}$ there exists a $t' \in T_{(\Gamma \cup \{\#\})}$ such that $\langle s, t \rangle \sim_{m+1}^r \langle s', t' \rangle$.*

Proof. Let s, s' be two trees from $T_{(\Sigma \cup \{\#\}) \times (\Gamma \cup \{\#\})^m}$ such that $s \sim_m^{r+1} s'$ and $t \in T_{(\Gamma \cup \{\#\})}$. Then $s = \langle t_n^\# [c_1/\alpha_1, \dots, c_n/\alpha_n], t_n^\# [c_1/t_{1,1}, \dots, c_n/t_{1,n}], \dots, t_n^\# [c_1/t_{m,1}, \dots, c_n/t_{m,n}] \rangle$ and $s' = \langle t_k^\# [c_1/\alpha'_1, \dots, c_k/\alpha'_k], t_k^\# [c_1/t'_{1,1}, \dots, c_k/t'_{1,k}], \dots, t_k^\# [c_1/t'_{m,1}, \dots, c_k/t'_{m,k}] \rangle$ for some $n, k \geq 1$ and trees $\alpha_i, \alpha'_j \in T_\Sigma$ and $t_{i,j}, t'_{s,t} \in T_\Gamma$. Let t_{m+1} be an arbitrary tree from $T_{(\Gamma \cup \{\#\})}$. If $t_{m+1} \notin T_\Gamma^{\#,n}$ take some tree t'_{m+1} that is not in $T_\Gamma^{\#,k}$. Then $\langle s, t_{m+1} \rangle \sim_m^r \langle s', t'_{m+1} \rangle$ because of Condition 1. Otherwise $t_{m+1} = t_n^\# [c_1/t_{m+1,1}, \dots, c_n/t_{m+1,n}]$. For every E_m^r equivalence class κ let $\kappa(s) = \{i \in \{1, \dots, n\} \mid \langle \alpha_i, s_{i,1}, \dots, s_{i,m} \rangle E_m^r = \kappa\}$.

Let $X_1^\kappa, \dots, X_{\ell_\kappa}^\kappa$ be the partition of $\kappa(s)$ with respect to the E_{m+1}^r equivalence classes of $\{\langle \alpha, t_{1,i}, \dots, t_{m+1,i} \rangle \mid i \in \kappa(s)\}$. Because $s \sim_m^{r+1} s'$ it is ensured that $|\kappa(s)| = f(m+r+1)^{r+1} |\kappa(s')|$ and therefore we can find a partition $Y_1^\kappa, \dots, Y_{\ell_\kappa}^\kappa$ of $\kappa(s')$ with $|X_i^\kappa| = f(m+r+1)^r |Y_i^\kappa|$ (if $|\kappa(s)| < f(m+r+1)^{r+1}$ partition $\kappa(s')$ according to some bijection between $\kappa(s)$ and $\kappa(s')$). Otherwise, because $\ell_\kappa < f(m+r+1)$ there is at least one X_i^κ with $|X_i^\kappa| \geq f(m+r+1)$ which also ensures that we can find such a partition).

By construction, $\langle \alpha, t_{1,i}, \dots, t_{m,i} \rangle E_m^{r+1} \langle \alpha', t'_{1,j}, \dots, t'_{m,j} \rangle$ whenever $i \in X_k^\kappa$ and $j \in Y_k^\kappa$. Thus $\langle \alpha, t_{1,i}, \dots, t_{m+1,i} \rangle E_{m+1}^r \langle \alpha', t'_{1,j}, \dots, t'_{m+1,j} \rangle$ for some appropriate $t'_{m+1,j}$. Now choose $t'_{m+1} = t_k^\# [c_1/t'_{m+1,1}, \dots, c_k/t'_{m+1,k}]$. By construction $\langle s, t_{m+1} \rangle \sim_{m+1}^r \langle s', t'_{m+1} \rangle$ due to Condition 2. \blacktriangleleft

In order to show that Property 4 is fulfilled, it is convenient to define a special kind of convolution for contexts. For $i \in \{1, \dots, n\}$ let c_i be an Γ_i -context such that $\text{dom}_{c_1} = \dots = \text{dom}_{c_n} =: \text{dom}$ and $c_1^{-1}(x) = \dots = c_n^{-1}(x) =: w_x$. Then $\langle c_1, \dots, c_n \rangle_c$ is the $(\Gamma_1 \times \dots \times \Gamma_n)$ -context with $\text{dom}(\langle c_1, \dots, c_n \rangle_c) = \text{dom}$ and

$$\langle c_1, \dots, c_n \rangle_c(w) = \begin{cases} (\gamma_1, \dots, \gamma_n) & \text{if } w \neq w_x, \\ x & \text{otherwise.} \end{cases}$$

10:10 Uniformly Automatic Classes of Finite Structures

► **Lemma 4.10.** *The relations $(\sim_m^r)_{r,m \in \mathbb{N}}$ respect contexts.*

Proof. Suppose $s \sim_m^r s'$ and let c be a $((\Sigma \cup \{\#\}) \times (\Gamma \cup \{\#\})^m)$ -context. We can assume that s and s' are equivalent due to Condition 2 and that

$$c = \langle t_n^\# [c_1/\alpha_1, \dots, c_n/\alpha_n], t_n^\# [c_1/t_{1,1}, \dots, c_n/t_{1,n}], \dots, t_n^\# [c_1/t_{m,1}, \dots, c_n/t_{m,n}] \rangle_c$$

for some $n \geq 1$ and $\langle \alpha_i, t_{1,i}, \dots, t_{m,i} \rangle_c$ is a $(\Sigma \times \Gamma^m)$ -context for exactly one $1 \leq i \leq n$ (because in any other case $c \circ t$ and $c \circ t'$ are equivalent by Condition 1). Fix this i and let $c' := \langle \alpha_i, t_{1,i}, \dots, t_{m,i} \rangle$. There two cases that we need to consider. First if s, s' are elements of $T_\Sigma^{\#,1} \otimes (T_\Gamma^{\#,1})^{\otimes m}$ ($= T_\Sigma \otimes (T_\Gamma)^{\otimes m}$). Then the requirement of Condition 2 reduces to $sE_m^r s'$. But then $c' \circ tE_m^r c' \circ s'$ and hence $c \circ s \sim_m^r c \circ s'$. Otherwise we can even assume that

$$c = \langle t_n^\# [c_1/\alpha_1, \dots, c_n/x], t_n^\# [c_1/t_{1,1}, \dots, c_n/x], \dots, t_n^\# [c_1/t_{m,1}, \dots, c_n/x] \rangle_c$$

(again otherwise we would get equivalence by Condition 1). But then

$$\begin{aligned} c \circ s &= \langle t_{n+k-1}^\# [c_1/\alpha_1, \dots, c_{n-1}/\alpha_{n-1}, c_n/\beta_1, \dots, c_{n+k-1}/\beta_k], \\ &\quad t_{n+k-1}^\# [c_1/t_{1,1}, \dots, c_{n-1}/t_{1,n-1}, c_n/s_{1,1}, \dots, c_{n+k-1}/s_{1,k}], \\ &\quad \vdots \\ &\quad t_{n+k-1}^\# [c_1/t_{m,1}, \dots, c_{n-1}/t_{m,n-1}, c_n/s_{m,1}, \dots, c_{n+k-1}/s_{m,k}] \rangle \end{aligned}$$

and

$$\begin{aligned} c \circ s' &= \langle t_{n+k-1}^\# [c_1/\alpha_1, \dots, c_{n-1}/\alpha_{n-1}, c_n/\beta'_1, \dots, c_{n+k-1}/\beta'_{k'}], \\ &\quad t_{n+k-1}^\# [c_1/t_{1,1}, \dots, c_{n-1}/t_{1,n-1}, c_n/s'_{1,1}, \dots, c_{n+k-1}/s'_{1,k'}], \\ &\quad \vdots \\ &\quad t_{n+k-1}^\# [c_1/t_{m,1}, \dots, c_{n-1}/t_{m,n-1}, c_n/s'_{m,1}, \dots, c_{n+k-1}/s'_{m,k'}] \rangle \end{aligned}$$

Using that s and s' are equivalent by Condition 2, it is easy to see that also $c \circ t$ and $c \circ t'$ are equivalent. ◀

The preceding lemmata show that $(\sim_m^r)_{r,m \in \mathbb{N}}$ is an EF-congruence for \mathbf{c}^\times , which completes the proof of Theorem 4.6. ◀

Another important class of operations under which uniform tree-automatic presentations are closed are parametrised first-order interpretations. Also in this case the complexity of the EF-congruence grows rather tamely under these operations.

► **Lemma 4.11.** *Let \mathbf{c} be a uniformly tree-automatic presentation of a class \mathcal{C} of τ -structures and \mathcal{I} be a parametrised τ -to- σ -interpretation of width ℓ that interprets for every $\mathfrak{A} \in \mathcal{C}$ a structure $\mathcal{I}(\mathfrak{A})$. Further let c be the maximal quantifier rank of any of the formulas in \mathcal{I} . If there is an $f(r+m)$ bounded EF-congruence for \mathbf{c} then there is a uniform tree-automatic presentation \mathcal{I}^c of the class $\mathcal{I}^c = \{\mathcal{I}^{\mathfrak{A}}(a) \mid \mathfrak{A} \in \mathcal{C}, a \in A\}$ with $g(r+m) := f((\ell+c)(r+m)+c)$ bounded EF-congruence.*

5 FPT Model Checking With Elementary Parameter Dependence

The runtime analysis from Section 3 not only enables us to show that first-order model checking is fixed parameter tractable on several classes of finite structures, but also gives us elementary bounds on the parameter dependence. In the following we write $\exp_k(x)$ for the k -fold tower of twos function applied to x , that is $\exp_0(x) = x$ and $\exp_{k+1}(x) = 2^{\exp_k(x)}$.

► **Theorem 5.1.** *Let \mathfrak{c} be a uniformly tree-automatic presentation such that Algorithm 2 computes in time $T(|\varphi|)$ from \mathfrak{c} the corresponding automaton \mathcal{A}_φ . Suppose for a class of finite structures \mathcal{C} there is a function $f : \text{code}(\mathcal{C}) \rightarrow \Gamma^*$ that computes in time $F(|w|)$ for every $w \in \text{code}(\mathfrak{A})$ with $\mathfrak{A} \in \mathcal{C}$ a tree α with $\mathfrak{A} \cong \mathcal{S}(\mathfrak{c}[\alpha])$. Then FO model checking on \mathcal{C} is decidable in time $\mathcal{O}(T(|\varphi|) \cdot |f(w)| + F(|w|))$.*

Proof. The runtime is achieved by the straight forward method of checking whether \mathcal{A}_φ accepts $f(w)$. ◀

5.1 Boolean Algebras

Our simplest application of Theorem 4.6 and Theorem 5.1 is for the class of all finite Boolean algebras. It is well known that every finite Boolean algebra is isomorphic to a finite direct power of the two element Boolean algebra. Especially, every finite Boolean algebra contains exactly 2^n elements for some $n \geq 1$ and every finite Boolean algebra is uniquely determined by the number of elements. Because of this simple structure it is natural to consider succinct encodings of Boolean algebras as inputs. In the following we will assume that a Boolean algebra is given by the number of atoms, encoded in unary. In other words, a finite Boolean algebra $\mathfrak{B} = (B, \cap, \cup, \bar{\cdot}, \mathbf{0}, \mathbf{1})$ is encoded by the string $1^{\log |B|}$.

► **Theorem 5.2.** *First-order model checking is fixed parameter tractable on the class of all finite Boolean algebras. Given a Boolean algebra \mathfrak{B} and an FO sentence φ one can decide in time $\exp_2(\text{poly}(|\varphi|)) \log |\mathfrak{B}|$ whether $\mathfrak{B} \models \varphi$.*

Proof. The class that contains just the Boolean algebra $\mathfrak{B}_2 = (\{\mathbf{0}, \mathbf{1}\}, \cap, \cup, \bar{\cdot}, \mathbf{0}, \mathbf{1})$ has the trivial automatic presentation \mathfrak{c} over the advice alphabet $\Sigma = \{a\}$ and the alphabet $\Gamma = \{0, 1\}$. The advice a (the tree of height 0 where the root is labeled with a) represents \mathfrak{B}_2 and the elements $\mathbf{0}$ and $\mathbf{1}$ are represented by 0 and 1, respectively. One checks that the relations $(E_m^r)_{r,m \in \mathbb{N}}$ where E_m^r is simply the identity relation on $T_{\Sigma_m}^r$ are an EF-congruence with respect to \mathfrak{c} and the index of E_m^r is bounded by $f(r+m) = 2^{r+m} + 2$ for all $r, m \in \mathbb{N}$.

As mentioned before, every finite Boolean algebra is a finite direct product of \mathfrak{B}_2 and hence \mathfrak{c}^\times is a uniform presentation of the class of all finite Boolean algebras. According to Theorem 4.6, \mathfrak{c}^\times has an EF-congruence bounded by $f'(r+m) \in 2^{\mathcal{O}((r+m+1)(2^{r+m}+2) \log(2^{r+m}+2))} \subseteq 2^{2^{\text{poly}(|\varphi|)}}$. Using Theorem 5.1, we conclude that for a sentence φ of quantifier-rank r Algorithm 1 constructs the corresponding automaton \mathcal{A}_φ in time $\mathcal{O}\left(|\varphi| \left(|\mathfrak{c}^\times|^{m+r} \cdot 2^{2^{\text{poly}(|\varphi|)}}\right)^c\right) \subseteq 2^{2^{\text{poly}(|\varphi|)}}$ (because $|\mathfrak{c}^\times|$ is constant). Note that the Boolean algebra with n atoms is represented by the tree $t_n^\# [c_1/a, \dots, c_n/a]$ in \mathfrak{c}^\times . We can therefore transform the encoding of the Boolean algebra into the tree-representation in linear time. Finally the claim follows from Theorem 5.1. ◀

With respect to the height of the tower of twos in the parameter dependence this result is probably optimal, as stated by the following theorem.

► **Theorem 5.3.** *Unless $\bigcup_{c \in \mathbb{N}} \text{STA}(*, 2^{c^n}, n) = \text{EXP}$ there is no algorithm that solves the model checking problem for finite Boolean algebras in time $2^{\text{poly}(|\varphi|)} \cdot \log |\mathfrak{B}|$.*

Proof. It is known that the theory of all finite Boolean algebras is complete for the complexity class $\bigcup_{c \in \mathbb{N}} \text{STA}(*, 2^{cn}, n)$. Further, using Lemma 4.2 and the computations of Theorem 5.2, we see that there is a constant c such that if \mathfrak{B} and \mathfrak{B}' are two Boolean algebras with at least 2^{r^c} many atoms then $\mathfrak{B} \equiv_r \mathfrak{B}'$. To check that a sentence φ of quantifier rank r belongs to the theory of finite Boolean algebras it is sufficient to check whether every finite Boolean algebra with at most 2^{r^c} many atoms models φ . If we could perform model-checking in time $\mathcal{O}(2^{\text{poly}(|\varphi|)} \cdot \log |\mathfrak{B}|)$ we could hence solve the theory of finite Boolean algebras in time $\mathcal{O}\left(2^{\text{poly}(|\varphi|)} \cdot \sum_{i=1}^{2^{r^c}} i\right) \subseteq 2^{\text{poly}(|\varphi|)}$, which implies $\bigcup_{c \in \mathbb{N}} \text{STA}(*, 2^{cn}, n) = \text{EXP}$. \blacktriangleleft

► **Remark.** Needless to say than an analogue of Theorem 5.2 also holds if the Boolean algebra is encoded traditionally by the multiplication tables of the operators. Obviously one can compute the succinct encoding from the traditional encoding efficiently by simply counting the number of atoms.

However, one could also argue that our encoding for the Boolean algebras is not optimal. Indeed a finite Boolean algebra \mathfrak{B} can be encoded by a word of length $\lceil \log \log |\mathfrak{B}| \rceil$ when we encode the number of atoms by its binary expansion. In this case our algorithm would not have a polynomial runtime in the size of the encoding of the structure because the advice would be of exponential size. However we could slot in a kernelisation procedure ahead. As we already explained in the proof of Theorem 5.3, there is a fixed polynomial p such that all finite Boolean algebras with at least $2^{p(k)}$ atoms are indistinguishable by a first-order Formula of quantifier rank at most k . In turn we can compute for a given finite Boolean Algebra \mathfrak{B} and a natural number k an advice α of size $\mathcal{O}\left(2^{2^{p(k)}}\right)$ such that $\mathcal{S}(\mathfrak{c}[\alpha]) \equiv_k \mathfrak{B}$ (where \mathfrak{c} is the presentation of the finite Boolean algebras constructed in Theorem 5.2). Because we are more interested in the application of automata based presentations than on encoding issues we will not work out the details here.

5.2 Finite Groups

Probably a bit more interesting is the class of all finite groups. In [10], Grohe posed the question on which classes of finite groups first-order model checking is fixed parameter tractable. In order to tackle this question we propose a structural parameter on finite groups. The Remak-Krull-Schmidt Theorem [16] states that a factorization of $\mathfrak{G} = \mathfrak{G}_1 \otimes \mathfrak{G}_2 \otimes \dots \otimes \mathfrak{G}_n$ into indecomposable subgroups \mathfrak{G}_i is unique up to permutation and isomorphism of the occurring subgroups for any finite group \mathfrak{G} . Therefore the size of the largest non-abelian subgroup in such a factorisation is uniquely determined. This leads to the following parameter.

► **Definition 5.4.** Let \mathfrak{G} be a finite group. The *non-abelian decomposition width* of \mathfrak{G} is $\text{dw}(\mathfrak{G}) = \max(\{|\mathfrak{G}'| \mid \mathfrak{G}' \text{ is non-abelian, indecomposable, and } \mathfrak{G} \cong \mathfrak{G}' \oplus \mathfrak{G}''\})$ the size of a maximal non-abelian indecomposable factor of \mathfrak{G} .

Note that the finite abelian groups are exactly the groups with non-abelian decomposition width one. As for the case of Boolean algebras, finite abelian groups have a quite simple structure. By the classification of finitely generated abelian groups every finite abelian group \mathfrak{G} is isomorphic to a finite sum of finite cyclic groups. That is $\mathfrak{G} \cong \mathbb{Z}_{n_1} \oplus \dots \oplus \mathbb{Z}_{n_k}$ for some $k \geq 1$ and $n_1, \dots, n_k \geq 1$. Hence, a finite abelian group can be encoded by a sequence of natural numbers (n_1, \dots, n_k) . Bova and Martin have independently shown in [2] that first-order model-checking is FPT on the class of all finite abelian groups. Their algorithm uses a quantifier elimination procedure. However, their analysis of the algorithm only yields a non-elementary parameter dependence. We will show that the automata based approach yields an algorithm with elementary parameter dependence.

Algorithm 2 Decomposing a Finite Abelian Group into Cyclic Factors.

Input: Finite abelian group \mathfrak{G} **Output:** String $\text{bin}(n_1)\# \cdots \# \text{bin}(n_k)$ such that $\mathfrak{G} \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_k}$

```

procedure DECOMPOSE( $\mathfrak{G}$ )
  Compute  $g$  with  $|g|$  maximal in  $\mathfrak{G}$ 
  if  $\langle g \rangle = \mathfrak{G}$  then
    return  $\text{bin}(|g|)$ 
  else
     $w \leftarrow \text{DECOMPOSE}(\mathfrak{G}/\langle g \rangle)$ 
    return  $\text{bin}(|g|)\#w$ 
  end if
end procedure

```

► **Theorem 5.5.** FO-model-checking is FPT on the class of all finite abelian groups. More precisely one can decide given a finite abelian group \mathfrak{G} and a formula $\varphi \in \text{FO}$ in time $\mathcal{O}(\exp_4(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{G}|)$ whether $\mathfrak{G} \models \varphi$.

Proof. Durand-Gasselin and Habermehl gave in an automatic presentation \mathfrak{d} of Presburger arithmetic and proved that there is a $f(m+r) = \exp_3(c(m+r))$ bounded EF-congruence with respect to \mathfrak{d} for some $c \in \mathbb{N}$ [6, Lemma 15].

We construct a uniform presentation of all finite cyclic groups from \mathfrak{d} by a parametrised first-order interpretation $\mathcal{I} = (\delta(n, x), \varphi_\circ(n, x, y, z))$ in Presburger Arithmetic. It is a well known fact that such an interpretation exists. Then $\mathcal{I}^{(\mathbb{N}, +)}(n) \cong \mathbb{Z}_n$ for all $n \in \mathbb{N}$ and therefore \mathcal{I}° is a uniform presentation of the class of all finite cyclic groups. By Lemma 4.11 there is a constant c' such that \mathcal{I}° has a $g(r+m) = \exp_3(c'(r+m))$ bounded EF-congruence. Further $(\mathcal{I}^\circ)^\times$ is a uniform presentation of the class of all finite abelian groups and Theorem 4.6 tells us that it has a $(g(r+m))^r \in \exp_4(\text{poly}(|\varphi|))$ bounded EF-congruence. Note that in $(\mathcal{I}^\circ)^\times$ a group $\mathfrak{G} \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_k}$ is represented by the tree $t_k^\# [c_1/\text{bin}^R(n_1), \dots, c_k/\text{bin}^R(n_k)]$ (The presentation in [6] uses binary encoding). Of course this tree can trivially be computed in linear time from the encoding (n_1, \dots, n_k) of G . By applying Theorem 5.1 we conclude that our algorithm solves the model-checking problem for finite abelian groups in time $\mathcal{O}(\exp_4(\text{poly}(|\varphi|)) \cdot \log |G|)$. ◀

► **Remark.** Although the encoding of an abelian group by the orders of its cyclic factors makes it trivial to compute the tree-presentation because it makes the relevant structural properties of the group explicit, it is still true that an analog of Theorem 5.5 holds if the group is encoded by its multiplication table. Indeed Algorithm 2 provides a simple procedure to compute the cyclic factors of the group in linear time. To see this, note that if g is an element of maximal order in a finite abelian group \mathfrak{G} then $\mathfrak{G} \cong \langle g \rangle \oplus \mathfrak{G}/\langle g \rangle$. The Algorithm 2 therefore computes a representant of a decomposition of \mathfrak{G} into cyclic factors. The computation of an element with maximal order can be done in time $\mathcal{O}(|\mathfrak{G}|^2)$ by computing the order of every element. The group $\mathfrak{G}/\langle g \rangle$ can also be computed in time $\mathcal{O}(|\mathfrak{G}|^2)$ by computing the multiplication table on the cosets of $\langle g \rangle$. Finally the procedure DECOMPOSE(\mathfrak{G}) is called at most $\log_2(|\mathfrak{G}|)$ times because $|\mathfrak{G}/\langle g \rangle| = |\mathfrak{G}|/|g|$. Together this gives a running time of $\mathcal{O}(|\mathfrak{G}|^2 \cdot \log(|\mathfrak{G}|))$, which is linear in the size of the multiplication table.

Finally, we turn our attention to encoding issues. As it was the case for Boolean algebras, there is an encoding of finite abelian groups, which in some cases allows for a considerably more succinct presentation. More precisely an abelian group $\mathfrak{G} \cong (\mathbb{Z}_{n_1})^{k_1} \times \cdots \times (\mathbb{Z}_{n_\ell})^{k_\ell}$

can be encoded by the tuple of pairs $((n_1, k_1), \dots, (n_\ell, k_\ell))$. Again, using this encoding we would not directly obtain an FPT-algorithm from our method. However, using the same argument as for the Boolean algebras, for some fixed polynomial p we can truncate the second components of each pair to $\exp_3(p(r))$ in a preprocessing step, where r is the quantifier rank of the formula under consideration. Again we will leave the details of this approach to the reader.

We extend our ideas from abelian groups to groups of bounded non-abelian decomposition width.

► **Theorem 5.6.** *First-order model checking is FPT on the class of all finite groups with bounded non-abelian decomposition width. More precisely there exists a constant c such that we can decide in time $\mathcal{O}(\exp_4(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{G}| + |\mathfrak{G}|^c)$ whether $\mathfrak{G} \models \varphi$.*

5.3 Graphs of bounded Tree-Depth and MSO Model Checking

Algorithmic meta-theorems for MSO are particularly interesting because MSO is capable of defining many NP-complete problems such as 3-colourability. The most famous result of this kind is probably the theorem of Courcelle that every MSO-definable query can be decided in linear time on the class of all graphs with treewidth at most c for any given constant $c \in \mathbb{N}$ [5]. Because trees have treewidth one, it is immediately clear that the parameter dependence in Courcelle's Theorem must be non-elementary. Tree-depth is another parameter on graphs that has recently drawn quite some attention. Tree-depth is a more restrictive parameter than treewidth. Indeed, every class of graphs of bounded tree-depth has also bounded treewidth but there are classes of graphs of bounded treewidth that have unbounded tree-depth. It was shown by Gajarský and Hliněný that, in terms of the parameter dependence, MSO-model-checking can be performed significantly faster on graphs of bounded tree-depth [13]. Their algorithm relies on kernelisation to perform fast MSO-model-checking on trees of bounded depth. However, transferring their arguments into our framework reveals that no specialised algorithm is needed to achieve this runtime.

► **Definition 5.7.** The **tree-depth** of a graph $G = (V, E)$ is recursively defined as

$$\text{td}(G) := \begin{cases} 1, & \text{if } |V| = 1 \\ \min\{\text{td}(G \upharpoonright V \setminus \{v\}) \mid v \in V\} + 1 & \text{if } G \text{ is connected and } |V| > 1 \\ \max_{1 \leq i \leq n} \text{td}(G_i) & G \text{ has components } G_1, \dots, G_n \end{cases}$$

An equivalent characterisation is the minimal height of a rooted forest such that G is isomorphic to a subgraph of the symmetric closure of the ancestor-descendant graph of that forest.

Again a straight forward encoding yields for every $h > 0$ a uniformly automatic presentation of the class of all graphs of tree-depth at most h . Translating the ideas of [13] into our framework shows that our generic algorithm performs just as good as the best known specialized algorithms.

► **Theorem 5.8.** *The MSO model checking problem for graphs of tree-depth at most h is fixed parameter tractable. Given an MSO sentence φ and a graph \mathfrak{G} of tree-depth at most h one can decide in time $\mathcal{O}(\exp_{(h+2)}(\text{poly}(|\varphi|)) \cdot \text{poly}(|\mathfrak{G}|))$ whether $\mathfrak{G} \models \varphi$.*

References

- 1 Faried Abu Zaid, Erich Grädel, and Frederic Reinhardt. Advice Automatic Structures and Uniformly Automatic Classes. In *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, 2017. URL: <http://www.logic.rwth-aachen.de/pub/abuzaid/AbuGraRei17.pdf>.
- 2 Simone Bova and Barnaby Martin. First-Order Queries on Finite Abelian Groups. In Stephan Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, volume 41 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41–59, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.CSL.2015.41.
- 3 Thomas Colcombet and Christof Löding. Transforming structures by set interpretations. *CoRR*, abs/cs/0703039, 2007. arXiv:cs/0703039.
- 4 B. Courcelle, J. A. Makowsky, and U. Rotics. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory Comput. Systems*, 33(2):125–150, April 2000. doi:10.1007/s002249910009.
- 5 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, March 1990. doi:10.1016/0890-5401(90)90043-H.
- 6 Antoine Durand-Gasselín and Peter Habermehl. Ehrenfeucht-Fraïssé goes elementarily automatic for structures of bounded degree. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 242–253. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012. doi:10.4230/LIPIcs.STACS.2012.242.
- 7 Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49(2):129–141, 1961. URL: <https://eudml.org/doc/213582>.
- 8 Jochen Eisinger. Upper Bounds on the Automata Size for Integer and Mixed Real and Integer Linear Arithmetic (Extended Abstract). In Michael Kaminski and Simone Martini, editors, *Computer Science Logic*, number 5213 in *Lecture Notes in Computer Science*, pages 431–445. Springer Berlin Heidelberg, September 2008. DOI: 10.1007/978-3-540-87531-4_31. URL: http://link.springer.com/chapter/10.1007/978-3-540-87531-4_31.
- 9 Jeanne Ferrante and Charles W. Rackoff. *The Computational Complexity of Logical Theories*, volume 718 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1979. URL: <http://link.springer.com/10.1007/BFb0062837>.
- 10 JÖRG Flum, ERICH Grädel, and THOMAS Wolke, editors. *Logic and Automata: History and Perspectives*. Amsterdam University Press, 2008. URL: <http://www.jstor.org/stable/j.ctt46mv83>.
- 11 Roland Fraïssé. Sur l’extension aux relations de quelques propriétés des ordres. *Annales scientifiques de l’École Normale Supérieure*, 71(4):363–388, 1954. URL: <https://eudml.org/doc/81696>.
- 12 M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. In *17th Annual IEEE Symposium on Logic in Computer Science, 2002. Proceedings*, pages 215–224, 2002. doi:10.1109/LICS.2002.1029830.
- 13 Jakub Gajarský and Petr Hliněný. Faster Deciding MSO Properties of Trees of Fixed Height, and Some Consequences. In Deepak D’Souza, Telikeyalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 112–123, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.FSTTCS.2012.112.

- 14 Martin Grohe. Logic, Graphs, and Algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(091), 2007. URL: <http://eccc.hpi-web.de/eccc-reports/2007/TR07-091/index.html>.
- 15 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding First-order Properties of Nowhere Dense Graphs. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 89–98, New York, NY, USA, 2014. ACM. doi:10.1145/2591796.2591851.
- 16 Thomas W. Hungerford. *Algebra*, volume 73 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1980. URL: <http://link.springer.com/10.1007/978-1-4612-6101-8>.
- 17 Neeraj Kayal and Timur Nezhmetdinov. Factoring Groups Efficiently. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5555 of *Lecture Notes in Computer Science*. Springer Verlag, 2009. URL: <http://research.microsoft.com/apps/pubs/default.aspx?id=102435>.
- 18 Felix Klaedtke. Ehrenfeucht–Fraïssé goes automatic for real addition. *Information and Computation*, 208(11):1283–1295, November 2010. doi:10.1016/j.ic.2010.07.003.
- 19 Dietrich Kuske and Markus Lohrey. Automatic structures of bounded degree revisited. *J. Symbolic Logic*, 76(4):1352–1380, December 2011. doi:10.2178/jsl/1318338854.
- 20 Alexander Langer, Felix Reidl, Peter Rossmanith, and Somnath Sikdar. Evaluation of an MSO-Solver. In David A. Bader and Petra Mutzel, editors, *ALENEX*, pages 55–63. SIAM / Omnipress, 2012. URL: <http://dblp.uni-trier.de/db/conf/alenex/alenex2012.html#LangerRRS12>.
- 21 Derek C. Oppen. A $2^{2^{2^n}}$ upper bound on the complexity of Presburger Arithmetic. *Journal of Computer and System Sciences*, 16(3):323–332, June 1978. doi:10.1016/0022-0000(78)90021-1.
- 22 Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A Faster Parameterized Algorithm for Treedepth. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming*, number 8572 in *Lecture Notes in Computer Science*, pages 931–942. Springer Berlin Heidelberg, July 2014. DOI: 10.1007/978-3-662-43948-7_77. URL: http://link.springer.com/chapter/10.1007/978-3-662-43948-7_77.
- 23 Detlef Seese. Linear Time Computable Problems and First-Order Descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

A Proofs Omitted from Section 4

In order to analyse Algorithm 1 with respect to the given presentation, we observe the following runtime bounds for the subroutines. We omit the proofs here as they are easily obtainable by a straight forward analysis of the respective routines.

► **Lemma A.1.** *The procedure INTERSECT($\mathcal{A}_1, \mathcal{A}_2$) computes a tree-automaton \mathcal{A} with s states and $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ in time $\mathcal{O}(|\Gamma| \cdot s^2)$, where s is the number of states reachable from the initial state in the product automaton $\mathcal{A}_1 \times \mathcal{A}_2$.*

► **Lemma A.2.** *The procedure DETERMINIZE(\mathcal{A}) computes a deterministic tree-automaton \mathcal{A}' with s states and $L(\mathcal{A}') = L(\mathcal{A})$ in time $\mathcal{O}(|\Gamma| \cdot s^2)$, where s is the number of states reachable from the initial state in the power set automaton of \mathcal{A} .*

► **Lemma A.3.** *Let Γ be a ranked alphabet, \sim an equivalence relation on T_Γ , and $\mathcal{A}_1 = (Q_1, \Gamma, \delta_1, q_{01}, F_1)$, $\mathcal{A}_2 = (Q_2, \Gamma, \delta_2, q_{02}, F_2)$ tree-automata. Suppose $t \sim t'$ implies $\delta_i^*(t) = \delta_i^*(t')$ for all $i \in \{1, 2\}$ and for all $t, t' \in T_\Gamma$. Then the number of reachable states from the initial state in $\mathcal{A}_1 \times \mathcal{A}_2$ is bounded by the index of \sim .*

► **Theorem 4.3.** *Let $\mathbf{c} = (\mathcal{A}, (\mathcal{A}_R)_{R \in \tau})$ be a uniformly tree-automatic presentation of a class of τ -structures. Suppose there is an $f(r+m)$ bounded EF-congruence $(E_m^r)_{r,m \in \mathbb{N}}$ for \mathbf{c} . Then for every $\varphi(x_1, \dots, x_m) \in \text{FO}$ of quantifier rank r Algorithm 2 computes the automaton \mathcal{A}_φ in time $\mathcal{O}(|\varphi|(|\mathbf{c}|^{m+r} \cdot f(m+r))^c)$ for some constant c .*

Proof. We prove the claim by induction over the structure of φ . Actually we prove an extended claim, namely that the procedure computes the automaton \mathcal{A}_φ in the given time and \mathcal{A}_φ has the property $\delta_{\mathcal{A}_\varphi}^*(t) = \delta_{\mathcal{A}_\varphi}^*(t')$ for all $t, t' \in T_{\hat{\Sigma}_m}$ with $tE_m^r t'$.

Case $\varphi = R(x_{i_1}, \dots, x_{i_k})$: Obviously $|\mathcal{A}_\varphi| \leq |\mathbf{c}|^m$ and therefore there is a fixed polynomial p such that \mathcal{A}_φ is constructed in time $p(|\mathbf{c}|^m)$. Further, by construction, the automata \mathcal{A}'_R and \mathcal{A}_D , from which \mathcal{A}_φ is build up, are minimal. Let s, s' be two trees from $T_{\hat{\Sigma}_m}$ with $sE_m^0 s'$. Then by Property 4 also $(c \circ s)E_m^0 (c \circ s')$ for all $\hat{\Sigma}_m$ -contexts. If $c \circ s$ is not a convolution of a tuple (α, \bar{t}) with $\bar{t} \in \mathcal{S}(\mathbf{c}[\alpha])$ then because of the first property of E_m^0 the same holds for $c \circ s'$. Hence $c \circ s \notin L(\mathcal{A}_\varphi)$ and $c \circ s' \notin L(\mathcal{A}_\varphi)$. Otherwise $c \circ s = \langle \alpha, \bar{t} \rangle$ and $c \circ s' = \langle \beta, \bar{t}' \rangle$ and Property 2 yields $\langle \alpha, \bar{t} \rangle \in L(\mathcal{A}_\varphi) \Leftrightarrow \langle \beta, \bar{t}' \rangle \in L(\mathcal{A}_\varphi)$. We obtain from Myhill-Nerode Theorem for tree-languages that $\delta_{\mathcal{A}_\varphi}^*(s) = \delta_{\mathcal{A}_\varphi}^*(s')$.

Case $\varphi = \psi(x_1, \dots, x_m) \wedge \gamma(x_1, \dots, x_m)$: Let \mathcal{A}_ψ and \mathcal{A}_γ be the automata constructed by COMPOSE in the recursion step. By the induction hypothesis, we know that all pairs of tuples t, t' that are related by E_m^r the computation of \mathcal{A}_ψ and \mathcal{A}_γ reach the same state. Lemma A.3 tells us that the number of reachable states in $\mathcal{A}_\varphi \times \mathcal{A}_\psi$ is bounded by $f(m+r)$. The automata \mathcal{A}_ψ and \mathcal{A}_γ are computed in at most $d|\psi|(|\mathbf{c}|^{m+r} \cdot f(m+r))^c + d|\gamma|(|\mathbf{c}|^{m+r} \cdot f(m+r))^c$ many steps and, according to Lemma A.1, the computation of \mathcal{A}_φ takes at most $d'|\hat{\Sigma}_m|f(m+r)^2$. But $|\mathbf{c}|^{m+r}$ is an upper bound for $|\hat{\Sigma}_m|$. Hence the overall runtime is bounded by $d(|\psi| + |\gamma| + 1)(|\mathbf{c}|^{m+r} \cdot f(m+r))^c = d|\varphi|(|\mathbf{c}|^{m+r} \cdot f(m+r))^c$. The property $tE_m^r t' \Rightarrow \delta_{\mathcal{A}_\varphi}^*(t) = \delta_{\mathcal{A}_\varphi}^*(t')$ follows directly from the induction hypothesis and the fact that $\delta^*(t) = (\delta_{\mathcal{A}_\psi}^*(t), \delta_{\mathcal{A}_\gamma}^*(t))$.

Case $\varphi = \neg\psi(x_1, \dots, x_m)$: By the induction hypothesis the automaton \mathcal{A}_ψ is constructed in time $d|\psi|(|\mathbf{c}|^{m+r} \cdot f(m+r))^c$. The automaton \mathcal{A}_D is the minimal automaton that recognises exactly the words of the form $\langle \alpha, t_1, \dots, t_m \rangle$, where $\alpha \in P^c$ and t_1, \dots, t_m are elements of $\mathcal{S}(\mathbf{c}[\alpha])$. Using the properties 1 and 4 of Definition 4.1, we see that for all $t, t' \in T_{\hat{\Sigma}_m}$ with $tE_m^r t'$ and all $\hat{\Sigma}_m$ -contexts c it is the case that $c \circ t \in L(\mathcal{A}_D) \Leftrightarrow c \circ t' \in L(\mathcal{A}_D)$. Therefore we can once again apply the lemmata A.3 and A.1 to establish that also \mathcal{A}_φ is constructed in the right amount of time and has the proclaimed property (recall that \mathcal{A}_φ is the product automaton of $\overline{\mathcal{A}_\psi}$ and \mathcal{A}_D).

Case $\varphi = \exists x_{m+1} \psi(x_1, \dots, x_m, x_{m+1})$: Let \mathcal{A}_ψ be the automaton that is constructed in the recursion step. Then \mathcal{A}_φ is essentially the reachable part of the power-set automaton of the projection automaton derived from \mathcal{A}_ψ under the projection $(\sigma, \gamma_1, \dots, \gamma_{m+1}) \mapsto (\sigma, \gamma_1, \dots, \gamma_m)$. Now suppose $sE_m^{r+1} s'$ for some $s, s' \in T_{\hat{\Sigma}_m}$. Then $q \in \delta_{\mathcal{A}_\varphi}^*(s)$ if and only if there is a $t \in T_\Gamma$ such that $\delta_{\mathcal{A}_\psi}^*((s, t)) = q$. But then, by Property 3 of Definition 4.1, there is also a $t' \in T_\Gamma$ with $\langle s, t \rangle E_m^r \langle s', t' \rangle$. By the induction hypothesis $\delta_{\mathcal{A}_\psi}^*(s', t') = q$ and thus $q \in \delta_{\mathcal{A}_\varphi}^*(s')$. This shows that $sE_m^{r+1} s'$ implies $\delta_{\mathcal{A}_\varphi}^*(s) = \delta_{\mathcal{A}_\varphi}^*(s')$. Consequently the number of reachable states in the aforementioned power set automaton is bounded by $f(m+r)$. We can now apply the induction hypothesis and Lemma A.2 to conclude that the algorithm takes at most $d|\varphi|(|\mathbf{c}|^{m+r} f(m+r))^c$ many steps to compute \mathcal{A}_φ . ◀

B Proofs Omitted from Section 5

► **Theorem 5.6.** *First-order model checking is FPT on the class of all finite groups with bounded non-abelian decomposition width. More precisely there exists a constant c such that we can decide in time $\mathcal{O}(\exp_4(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{G}| + |\mathfrak{G}|^c)$ whether $\mathfrak{G} \models \varphi$.*

Proof. First we build a trivial presentation \mathfrak{d} for the groups of order at most d . Let $\mathfrak{G}_1, \dots, \mathfrak{G}_n$ be an enumeration of the non-abelian groups of size at most d (up to isomorphism). The advice alphabet is $\Sigma = \{g_1, \dots, g_n\}$. The input alphabet Γ is extended by new letters a_1, \dots, a_d . For every $1 \leq i \leq n$ we choose a bijection $\pi_i : \{a_1, \dots, a_{|\mathfrak{G}_i|}\} \rightarrow \mathfrak{G}_i$ and construct the automata that recognise the languages $\{\langle g_i, a_j \rangle \mid 1 \leq i \leq n, j \leq |\mathfrak{G}_i|\}$ and $\{\langle g_i, a_x, a_y, a_z \rangle \mid 1 \leq i \leq n, 1 \leq x, y, z \leq |\mathfrak{G}_i|, \pi_i(a_x) \circ_{\mathfrak{G}_i} \pi_i(a_y) = \pi_i(a_z)\}$. Note that the trivial EF-congruence for \mathfrak{d} is $g(m+r) = G(d)d^{r+m}$ bounded, where $G(d)$ is the number of groups of size at most d .

Let \mathfrak{c} be the uniform presentation of the cyclic groups as described previously. We build automata that recognize the alphabet-disjoint union of the languages in \mathfrak{d} and corresponding languages from \mathfrak{c} and obtain a presentation \mathfrak{e} of all cyclic groups and groups of order at most d . It is not hard to see that this presentation is also $\exp_3(\text{poly}(|\varphi|))$ bounded. Basically the union of the EF-congruences for \mathfrak{d} and \mathfrak{c} (where the “is not a tuple of the presentation” equivalence class of \mathfrak{d} is merged with the “is not a convolution” equivalence class of \mathfrak{c}) is an EF-congruence for \mathfrak{e} . Then \mathfrak{e}^\times is a presentation of the class of all finite groups with bounded abelian decomposition width at most d . By Theorem 4.6, \mathfrak{e}^\times is $\exp_4(\text{poly}(|\varphi|))$ bounded.

A decomposition of $\mathfrak{G} = \mathfrak{G}_1 \oplus \dots \oplus \mathfrak{G}_k \oplus \mathbb{Z}_{n_1} \oplus \dots \oplus \mathbb{Z}_{n_\ell}$ with non-abelian indecomposable factors $\mathfrak{G}_1, \dots, \mathfrak{G}_k$ can be computed in polynomial time [17]. From the decomposition we can compute in linear time an advice that represents \mathfrak{G} . Note that such an advice has logarithmic size in $|\mathfrak{G}|$. Applying Theorem 5.1 completes the proof. ◀

In order to handle MSO model-checking on graphs of bounded tree-depth we need to enrich the structure by the powerset of the universe in order to simulate quantification over sets.

► **Definition B.1** ([3]). Let $\mathfrak{A} = (A, R_1, \dots, R_n)$ be a τ -structure. The **power set structure** $\mathcal{P}(\mathfrak{A})$ is the $(\tau \uplus \{\subseteq\})$ -structure $(\mathcal{P}(A), R_1^{\mathcal{P}(\mathfrak{A})}, \dots, R_n^{\mathcal{P}(\mathfrak{A})}, \subseteq)$, where $(\mathcal{P}(A), \subseteq)$ is the powerset lattice on A and $R_i^{\mathcal{P}(\mathfrak{A})} = \{(\{a_1\}, \dots, \{a_{r_i}\}) \in \mathcal{P}(A)^{r_i} \mid (a_1, \dots, a_{r_i}) \in R_i\}$ for all $i \in \{1, \dots, n\}$.

Clearly the MSO-theory of \mathfrak{A} is reducible to the FO-theory of $\mathcal{P}(\mathfrak{A})$ and vice versa. In the following we also need to make a distinction between trees that serve as an input to a tree automaton and an unordered rooted tree in the graph theoretic sense. A **finite unordered labeled tree-structure** \mathfrak{T} is a tuple $(V, E, P_1, \dots, P_n, r)$ where

- V is a finite set of nodes,
- $E \subseteq \binom{V}{2}$ such that (V, E) is connected and acyclic,
- $P_i \subseteq V$ for all $1 \leq i \leq n$, and
- $r \in V$ is the root of the tree.

There are standard techniques to encode a finite unordered tree-structures of unbounded degree by trees of bounded degree.

► **Definition B.2.** For a finite unordered tree-structure $\mathfrak{T} = (V, E, P_1, \dots, P_n, r)$ the set of **tilts** of \mathfrak{T} , $\text{tilt}(\mathfrak{T}) \subseteq T_{\mathcal{P}(\{1, \dots, n\})}$, is inductively defined by the following rules.

- if $\mathfrak{T} = (\{v\}, P_1, \dots, P_n, v)$ then $\text{tilt}(\mathfrak{T}) = \{t\}$, where $\text{dom}(t) = \{\epsilon\}$ and $t(\epsilon) = \{i \mid v \in P_i\}$

- if $\mathfrak{T} = (V, P_1, \dots, P_n, r)$ is of depth $h > 1$ then $t \in \text{tilt}(\mathfrak{T})$ if, and only if, there is an enumeration $\mathfrak{T}_0, \dots, \mathfrak{T}_k$ of the subtrees induced by the children of the root r of \mathfrak{T} such that there are trees t_0, \dots, t_k with
 - t_i is a tilt of \mathfrak{T}_i ,
 - $\text{dom}(t) = \bigcup_{0 \leq i \leq k} \{1^i 0\} \text{dom}(t_i)$,
 - $t(w) = \begin{cases} \{i \mid w \in P_i\} & w = \epsilon \\ X & w \in \{1\}^i, 1 \leq i \leq k \\ t_i(w') & w = 1^i 0 w', 0 \leq i \leq k \end{cases}$

Note that if t is a tilt of a tree-structure \mathfrak{T} and $v \in \text{dom}(t)$ with $t(v) \neq X$ then v corresponds to a node of depth $|v|_0 + 1$ in \mathfrak{T} .

► **Lemma B.3.** *Let $h \in \mathbb{N}$ be some fixed number. Then the class \mathcal{C}_h of all power set structures of graphs of tree-depth at most h is uniformly tree-automatic.*

Proof. The advice set consists of all tilts of tree-structures $(V, E, P_1, \dots, P_{h-1})$ of depth at most $h + 1$ such that every node of depth ℓ appears only in sets P_i with $i + 1 < \ell$. This is obviously a regular set. Such a tree α presents (the isomorphism type of) the graph $\mathfrak{G} = (V, E)$ with $V = \text{dom}(\alpha) \cap (\{0, 1\}^* \{0\})$ and $E = \{\{v, w\} \mid v \preceq w \text{ and } |v|_0 \in \alpha(w)\}$. If α is a tilt of an optimal decomposition of \mathfrak{G} then the subtrees induced by the nodes in $\text{dom}_\alpha \cap \{1\}^* \{0\}$ correspond to the connected components of \mathfrak{G} . Building a uniformly tree-automatic presentation $\mathfrak{c} = (\mathcal{A}, \mathcal{A}_E, \mathcal{A}_\subseteq)$ is then straight forward. The automaton \mathcal{A} is chosen such that $L(\mathcal{A}[\alpha]) = \{t \in T_{\{0,1,X\}} \mid \text{dom}(t) = \text{dom}(\alpha) \wedge \forall w \in \text{dom}(\alpha) : \alpha(w) = X \rightarrow t(w) = X\}$. A tree $t \in L(\mathcal{A}[\alpha])$ represents the set $\{v \in \text{dom}(\alpha) \mid t(v) = 1\}$. Then the relation \subseteq is trivially regular and the relation E can also be recognised with the advice α , because the prefix relation is regular on the domain of a tree and $|w|_0 \leq h$ for every $w \in \text{dom}(t)$ and every $t \in L(\mathcal{A}[\alpha])$, so an automaton can check whether w is the first ancestor with $|w|_0 = i$ of a node v with $i \in t(v)$. ◀

For a tree $t \in T_\Sigma$, $w \in \text{dom}_t$, and $a \in \Sigma$ we write $t[w \rightarrow a]$ for the tree that is obtained by replacing the label of the node w by a .

► **Theorem 5.8.** *The MSO model checking problem for graphs of tree-depth at most h is fixed parameter tractable. Given an MSO sentence φ and a graph \mathfrak{G} of tree-depth at most h one can decide in time $\mathcal{O}(\exp_{(h+2)}(\text{poly}(|\varphi|)) \cdot \text{poly}(|\mathfrak{G}|))$ whether $\mathfrak{G} \models \varphi$.*

Proof. We define the EF-congruence on the basis of equivalence relations $(\sim_{r,k}^h)_{r,k \in \mathbb{N}}$ on (P_1, \dots, P_k) -labeled tree-structures of depth h :

- For tree-structures $\mathfrak{S}, \mathfrak{T}$ of depth 1 we define $\mathfrak{S} \sim_{r,k}^1 \mathfrak{T} :\Leftrightarrow \mathfrak{S} \cong \mathfrak{T}$.
- Let $\mathfrak{S}, \mathfrak{T}$ be trees of depth $h + 1$ and let $\mathfrak{S}_1^i, \dots, \mathfrak{S}_{n_i}^i$ be the subtrees of depth i rooted in a child node of the root in \mathfrak{S} for all $i \leq h$ and let $\mathfrak{T}_1^i, \dots, \mathfrak{T}_{n'_i}^i$ be the corresponding trees with respect to \mathfrak{T} . Then \mathfrak{S} and \mathfrak{T} are $\sim_{r,k}^{h+1}$ -equivalent if, and only if, the roots of \mathfrak{T} and \mathfrak{S} share the same labels and for all $i < h$ and all $\sim_{r,k}^i$ -equivalence classes κ

$$|\{j \in \mathbb{N} \mid j \leq n_i, \mathfrak{S}_j^i \in \kappa\}| =_{\text{index}(\sim_{r,k}^i)_{r+1}} |\{j \in \mathbb{N} \mid j \leq n'_i, \mathfrak{T}_j^i \in \kappa\}|$$

The proof of [13, Theorem 3.1] can be easily adapted to show that no FO-formula with r quantifiers can distinguish between two power set structures of two $\sim_{r,(r+k)}^h$ -equivalent (P_1, \dots, P_k) -labeled tree-structures of depth h .

Moreover, a straightforward induction shows that whenever two such tree-structures $\mathfrak{S}, \mathfrak{T}$ of depth h are $\sim_{0,k}^h$ -equivalent then the following two observations hold for every path $v_0 v_1 \dots v_n$ in \mathfrak{S} starting from the root:

1. There is a path $w_0 w_1 \dots w_n$ in \mathfrak{T} starting from the root of \mathfrak{T} such that for all $0 \leq i \leq n$ the nodes v_i and w_i share the same labels, that is $v_i \in P_j^{\mathfrak{S}} \Leftrightarrow w_i \in P_j^{\mathfrak{T}}$ for all $1 \leq j \leq k$.
2. If for some subsets $I \subseteq \{1, \dots, n\}$, $J \subseteq \{1, \dots, k\}$ the nodes v_i with $i \in I$ are unique in the sense that for every path $v'_0 v'_1 \dots v'_n$ with $v_i \in P_j^{\mathfrak{S}} \Leftrightarrow v'_i \in P_j^{\mathfrak{S}}$ for all $i \in I$ and $j \in J$ implies $v_i = v'_i$ for all $1 \leq i \leq n$ then there is also a unique path $w_0 w_1 \dots w_n$ in \mathfrak{T} with $w_i \in P_j^{\mathfrak{T}} \Leftrightarrow v_i \in P_j^{\mathfrak{S}}$ for all $i \in I, j \in J$.

Let $\sim_{r,k}^{\leq h} := \bigcup_{1 \leq i \leq h} \sim_{r,k}^i$. We define an EF-congruence for the presentation in Lemma B.3 from $\sim_{r,k}^{\leq h}$. Let h be fixed.

In a first step, we partition the set of all $(\mathcal{P}(\{1, \dots, h+m-1\}) \uplus \{X\})$ -labeled trees into $2h+1$ classes $T_1^m, \dots, T_h^m, Q_1^m, \dots, Q_h^m, F$.

- A tree t is in T_i^m if, and only if, t is a tilt of a tree of depth i .
- A tree t is in Q_i^m if, and only if, t is not a tilt of a tree of depth i but $t[\epsilon \rightarrow \emptyset]$ is a tilt of a tree of depth i (this is exactly the case if $t = t'[\epsilon \rightarrow X]$ for some tilt t' of a tree of depth i).
- All other trees are in F .

The EF-congruence is then defined by

$$\begin{aligned}
 t E_m^r t' &: \Leftrightarrow \exists 0 \leq i \leq h : (t \in T_i^m \wedge t' \in T_i^m \wedge \\
 &\quad \exists \mathfrak{S}, \mathfrak{S}' : t \in \text{tilt}(\mathfrak{S}) \wedge t' \in \text{tilt}(\mathfrak{S}') \wedge \mathfrak{S} \sim_{r, (r+m+k)}^i \mathfrak{S}') \\
 &\vee \exists 0 \leq i \leq h : (t \in Q_i^m \wedge t' \in Q_i^m \wedge \\
 &\quad \exists \mathfrak{S}, \mathfrak{S}' : t[\epsilon \rightarrow \emptyset] \in \text{tilt}(\mathfrak{S}) \wedge t'[\epsilon \rightarrow \emptyset] \in \text{tilt}(\mathfrak{S}') \wedge \mathfrak{S} \sim_{r, (r+m+k)}^i \mathfrak{S}') \\
 &\vee t, t' \in F
 \end{aligned}$$

For Property 1 let us consider under which circumstances a tree t does not present a graph of tree-depth at most h . First of all t might not be a tilt of a tree-structure of depth at most $h+1$. In this case $t \in F$ or $t \in Q_i$ for some $i \leq h+1$. In this case E_m^r separates t from all trees that represent a graph from \mathcal{C}_h . Otherwise t might be the tilt of a tree-structure \mathfrak{T} of depth at most $h+1$ but there is a node $v \in \mathfrak{T}$ of depth i with $v \in P_i^{\mathfrak{T}}$ and $i+1 \geq j$. But then by Observation 1 every E_m^r -equivalent tree-structure contains also a node of depth j which is contained in P_i and therefore does also not present a graph from \mathcal{C}_h .

We use Observation 2 to show that Property 2 is fulfilled. Let s and t be $(\mathcal{P}(\{1, \dots, h+m-1\}) \cup \{X\})$ -labeled trees that present Structures in \mathfrak{c} with $s E_m^r t$. Let $\mathfrak{S}, \mathfrak{T}$ be the tree-structures with $s \in \text{tilt}(\mathfrak{S})$ and $t \in \text{tilt}(\mathfrak{T})$, let $(\mathfrak{S}_s, V_1, \dots, V_m)$ be the tuple presented by s , and $(\mathfrak{S}_t, W_1, \dots, W_m)$ be the tuple presented by t . If $\mathfrak{S}_s \models E(V_i, V_j)$ for some $i, j \leq m$ then V_i and V_j are singletons and therefore there are unique nodes v_i, v_j with $v_i \in P_{h+i-1}^{\mathfrak{S}}$ and $v_j \in P_{h+j-1}^{\mathfrak{S}}$. Further v_i and v_j are ordered by the ancestor-relationship. Without loss generality assume that v_i is an ancestor of v_j and let d be the depth of v_i in \mathfrak{S} . Then $v_j \in P_{d-1}^{\mathfrak{S}}$. By Observation 2 there must be unique nodes w_i, w_j with $w_i \in P_{h+i-1}^{\mathfrak{T}}$ and $w_j \in P_{h+j-1}^{\mathfrak{T}}$. Further w_i has depth d , is an ancestor of w_j , and $w_j \in P_{d-1}^{\mathfrak{T}}$. Hence $\mathfrak{S}_t \models E(W_i, W_j)$. If $\mathfrak{S}_s \not\models V_i \subseteq V_j$ then there is node $v \in \text{dom}_s$ such that $i \in s(v)$ but $j \notin s(v)$. Using similar arguments as in the previous case we can follow that there is also a $w \in \text{dom}_t$ with $i \in s(v)$ and $j \notin s(v)$. Hence $\mathfrak{S}_t \not\models W_i \subseteq W_j$. The case of $\mathfrak{S}_s \not\models V_i = V_j$ is analogous.

In order to establish Property 3 suppose $s E_m^{r+1} t$. Let s' be any tree that can be derived from s by adding the label $(h+m)$ to some nodes $w \in \text{dom}(s) \cap \{0, 1\}^* \{0\}$. We distinguish three cases.

- Case $s, t \in F$:** then $t' \in F$ and we can extend the labeling of t in an arbitrary way to obtain an E_{m+1}^r -equivalent t' .
- Case $s, t \in T_i^m$ for some $1 \leq i \leq h$:** then there is a (P_1, \dots, P_{h+m-1}) -labeled tree-structures $\mathfrak{S}, \mathfrak{T}$ of depth i with $s \in \text{tilt}(\mathfrak{S})$ and $t \in \text{tilt}(\mathfrak{T})$. Further there is a set $X_{\mathfrak{S}} \subseteq \mathfrak{S}$ such that s' is a tilt of $(\mathfrak{S}, X_{\mathfrak{S}})$. Because $\mathfrak{S} \sim_{(r+1), ((r+1)+h+m)}^h \mathfrak{T}$ there must be a set $X_{\mathfrak{T}} \subseteq \mathfrak{T}$ such that $(\mathfrak{S}, X_{\mathfrak{S}}) \sim_{r, (r+h+(m+1))}^h (\mathfrak{T}, X_{\mathfrak{T}})$. Finally choose the extension t' of the labeling of t such that $t' \in \text{tilt}((\mathfrak{T}, X_{\mathfrak{T}}))$. Then $t' E_{m+1}^r s'$.
- Case $s, t \in Q_i^m$ for some $1 \leq i \leq h$:** the case follows analogously to the previous one by considering $s[\epsilon \rightarrow \emptyset]$ and $t[\epsilon \rightarrow \emptyset]$.

At last, we see that Property 4 holds. Indeed, if $t \in F$ then $(c \circ t) \in F$ for every context c . For the case $s, t \in T_i^m$ for some $1 \leq i \leq h$ one can distinguish two cases based on the structure of the context c .

- Case $c^{-1}(x) \in \{0, 1\}^* \{1\} \cup (\{1\}^* \{0\})^{>(h-i)}$:** then $(c \circ s)$ and $(c \circ t)$ do not present trees of depth at most h and hence $s, t \in F$.
- Case $c^{-1}(x) \in (\{1\}^* \{0\})^{\leq(h-i)}$:** there are three subcases that might occur.
- It might be that $(c \circ t) \in F$ and $(c \circ s) \in F$ (because c is a “template” of a tree of depth larger than h or c contains an inconsistent labeling). in this case equivalence is guaranteed by definition.
 - It is also possible that $(c \circ t) \in T_j^m$ and $(c \circ s) \in T_j^m$ for some $i \leq j \leq h$. Then let $\mathfrak{S}, \mathfrak{T}$ be trees of depth j such that $(c \circ t) \in \text{tilt}(\mathfrak{S})$ and $(c \circ s) \in \text{tilt}(\mathfrak{T})$. By induction over $j - i$ one shows that $\mathfrak{S} \sim_{r, r+h+m}^j \mathfrak{T}$. For $j - i = 0$ this is the case by definition. For $j - i = k + 1$ let $\mathfrak{S}_1, \dots, \mathfrak{S}_\ell$ and $\mathfrak{T}_1, \dots, \mathfrak{T}_\ell$ be the subtrees of \mathfrak{S} and \mathfrak{T} that are rooted in the children of the roots \mathfrak{S} and \mathfrak{T} , respectively. Without loss of generality assume that \mathfrak{S}_1 and \mathfrak{T}_1 are the subtrees which resulted from adding s and t into the context c . Then by the induction hypothesis $\mathfrak{S}_1 \sim_{r, r+h+m}^h \mathfrak{T}_1$ and also $\mathfrak{S}_n \cong \mathfrak{T}_n$ for all $1 < n \leq \ell$. But then for all $n < j$ and all $\sim_{r, r+h+m}^n$ -equivalence classes τ the number of τ -children of the root in \mathfrak{S} is equal to the number in \mathfrak{T} , hence $\mathfrak{S} \sim_{r, r+h+m}^j \mathfrak{T}$ and therefore $(c \circ s) E_m^r (c \circ t)$.
 - The last case that might happen is $(c \circ t) \in Q_j^m$ and $(c \circ s) \in Q_j^m$ for some $i \leq j \leq h$. In this case we might again argue analogously to the previous cases by considering $(c \circ t)[\epsilon \rightarrow \emptyset]$ and $(c \circ s)[\epsilon \rightarrow \emptyset]$.

Next, let us estimate the index of E_m^r . By the definition of E_m^r , $\text{index}(E_m^r) \leq 1 + 2 \sum_{i=0}^{h+1} \text{index}(\sim_{r, r+m+h+1}^i)$. An inductive analysis of $\text{index}(\sim_{r, r+m+h+1}^i)$ (see [13, Lemma 3.1 c)) shows $\text{index}(\sim_{r, r+m+h+1}^i) \in \exp_{(i+1)}(\text{poly}(r + m + h + 1))$. Applying this to the above estimation yields $\text{index}(E_m^r) \in \exp_{(h+2)}(\text{poly}(r + m))$.

In order to fulfil the prerequisites of Theorem 5.1 we can apply textbook methods to compute the decomposition of a graph of fixed tree-depth (see for instance [22]). From the decomposition the construction of an advice for the presentation in Lemma B.3 can be performed efficiently. \blacktriangleleft