Decidability of Weak Logics with Deterministic Transitive Closure *

Witold Charatonik Emanuel Kieroński

Institute of Computer Science
University of Wrocław
Wrocław, Poland
{wch,kiero}@cs.uni.wroc.pl

Filip Mazowiecki

Institute of Informatics
University of Warsaw
Warsaw, Poland
f.mazowiecki@mimuw.edu.pl

Abstract

The deterministic transitive closure operator, added to languages containing even only two variables, allows to express many natural properties of a binary relation, including being a linear order, a tree, a forest or a partial function. This makes it a potentially attractive ingredient of computer science formalisms. In this paper we consider the extension of the two-variable fragment of first-order logic by the deterministic transitive closure of a single binary relation, and prove that the satisfiability and finite satisfiability problems for the obtained logic are decidable and ExpSpace-complete. This contrasts with the undecidability of two-variable logic with the deterministic transitive closures of several binary relations, known before. We also consider the class of universal first-order formulas in prenex form. Its various extensions by deterministic closure operations were earlier considered by other authors, leading to both decidability and undecidability results. We examine this scenario in more details.

Categories and Subject Descriptors F.4.1 [*Mathematical Logic*]: Computational logic, Model theory; F.4.3 [*Formal Languages*]: Decision problems

General Terms Theory

Keywords deterministic transitive closure, two-variable logic, satisfiability problem

1. Introduction

Deterministic transitive closure

The deterministic transitive closure operator plays an important role in descriptive complexity. In particular Immerman showed in [11] that on ordered structures first-order logic augmented with this operator captures LogSpace. The operator seems to be attractive also for applications in tasks related to verification of hardware and software, as it can express many natural properties, e.g., when added to languages containing even only two variables it allows to say that a binary relation is a linear order, a tree (with reversed edges), a forest, or a partial function. However, while there are quite a lot of related works on formalisms which assume that the admissible structures themselves are deterministic, i.e., the out-degree of their elements is at most one (let us mention here deterministic propositional dynamic logic [1] and separation logic [4, 28]; regarding separation logic see in particular [7] where its two-variable variant is investigated), there are not too many papers studying the deterministic transitive closure operator in languages interpreted over structures not so constrained. We are aware of two such papers. In the first, [10], Grädel, Otto and Rosen prove that the two-variable fragment of first-order logic, FO^2, becomes undecidable when augmented with the deterministic transitive closure operator. In the second, [12], Immerman et al. identify a decidable and NExpTime-complete logic, namely the class of universal first-order formulas in prenex form with unary relations, one binary relation E, and positive occurrences of the deterministic transitive closure of E. The latter paper contains also some contrasting undecidability results.

Our contribution

The above mentioned papers constitute a starting point for our investigations. Our main result is that the satisfiability and finite satisfiability problems for FO^2 with unary relations, one binary relation E and its deterministic transitive closure are decidable and ExpSpace-complete (Thm. 5). Note that this does not contradict the mentioned undecidability from [10], since to obtain that result more binary relations and their closures are needed (actually, in Thm. 19 we argue that already two binary relations and their closures cause undecidability). In comparison to the decidability result from [12] we drop the restriction of using deterministic transitive closures only positively, and we offer unrestricted nesting of existential and universal quantifiers. The cost is reducing the number of variables to two, but this is typical for many languages used in computer science.

We return also to the class of universal formulas in prenex form considered in [12]. We analyse the impact of the number of universally quantified variables on the decidability of the extension by the deterministic transitive closure of a single binary relation. We show that the variant with two variables has the finite model property, is decidable and NExpTime-complete (Thm. 17), and with four — undecidable (Thm. 21). The latter sharpens one of the results from [12]. The case of three variables is solved partially, by showing undecidability in the presence of an additional binary relation (Thm. 20).

* The authors acknowledge the support of the Polish National Science Center grants DEC-2011/03/B/ST6/00346, DEC-2013/09/B/ST6/01535, and DEC-2013/09/N/ST6/01170

CSL-LICS 2014, July 14–18, 2014, Vienna, Austria.

ACM 978-1-4503-2886-9/14/07…$15.00.
http://dx.doi.org/10.1145/10.1145/2603088.2603134

Related work.

Decidability and the finite model property of basic FO^2, over arbitrary relational signatures, was shown by Mortimer [24]. Its NEXPTIME-completeness follows from the results by Lewis [21] and Grädel, Kolaitis and Vardi [8]. Since then many natural extensions of FO^2 have been examined. Most of them turn out to have undecidable satisfiability problem (see, e.g., [10, 16]). On the positive side, FO^2 remains decidable when extended by counting quantifiers [9, 26, 27], counting quantifiers and two forests accessible by their successor relations [5] (finite satisfiability only), a linear order [25] (or, in the case of finite satisfiability, even two linear orders, subject to some further restrictions on the signatures, see [29]), one or two equivalence relations [18, 19], one transitive relation [31] (general satisfiability only, the case of finite models is open).

Regarding FO^2 with variations of transitive closure operators, the situation is clear in the case of the *equivalence* closures. In [20] it is shown that FO^2 with the equivalence closure of one binary relation is NEXPTIME-complete; allowing the closures of two binary relations lifts the complexity to 2-NEXPTIME, and adding the closure of a third binary relation leads to undecidability. If we consider the standard transitive closure operator, undecidability can be obtained if the closures of two relations are available — this follows from the undecidability of FO^2 with two transitive relations [13, 15]. The case in which the transitive closure of only one binary relation is available is open, but some partial positive results are obtained (see *Conclusion* in [17]).

Since in $FO^2 + DTC(E)$ it is possible to say that a model is a rooted tree, our work is closely related to the work from [2] where decidability of FO^2 over finite trees is considered. In particular, the results from Section 3.3 imply that satisfiability of FO^2 over infinite unordered trees (with successor and descendant relations) is EXPSPACE-complete.

The class of universal first-order formulas with constants and with the standard transitive closure operator was investigated in [17], where it was shown that in the case of two universally quantified variables the satisfiability problem is decidable if the transitive closure of one binary relation is available (it is in 2-NEXPTIME and 2-EXPTIME-hard), and undecidable if we have transitive closures of two relations. With three variables and the transitive closure of a single binary relation the problem is undecidable (even if there are no other non-unary symbols).

2. Preliminaries

2.1 Models

Let E be a binary relation. Let $E' \subseteq E$ be the relation containing precisely those pairs $(a,b) \in E$ for which $(a,c) \in E \rightarrow b = c$. The *deterministic transitive closure* of E is defined as the transitive closure of E'. Let $\mathcal{L}$ be a fragment of first-order logic. By $\mathcal{L} + DTC(E_1,\ldots,E_k)$ we denote the extension of $\mathcal{L}$ by the deterministic transitive closures of relations $E_1,\ldots,E_k$. Instead of employing explicitly a special closure operator, we consider signatures containing pairs of binary relation symbols E_i and $\overline{E_i}$, and we admit only structures in which each $\overline{E_i}$ is interpreted as the deterministic transitive closure of the interpretation of E_i.

We usually consider signatures containing arbitrary number of unary symbols and a single binary relation E together with its deterministic transitive closure $\overline{E}$ (equality symbol is also allowed). Let $\mathfrak{A}$ be a structure over such a signature. Note that the relation $E \cap \overline{E}$ is a partial function. If $\mathfrak{A} \models E[a,b] \wedge \overline{E}[a,b]$ then the edge (a,b) is called *deterministic*. A substructure $\mathfrak{T}$ of $\mathfrak{A}$ whose universe is a maximal set of elements such that each distinct pair of them is related by the symmetric closure of $\overline{E}$ is called a *tree*. (Note that such substructures indeed resemble trees, but their edges

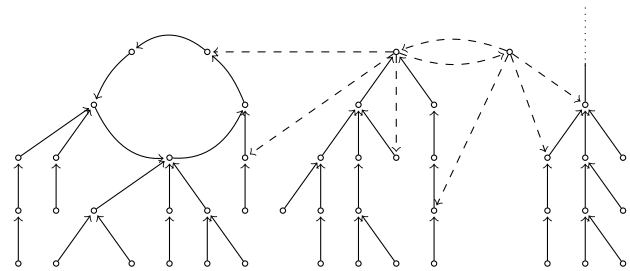

Figure 1. A structure with four trees. From left to right: a cyclically rooted tree, two rooted trees (the second of which consists of just a root), and a top-unbounded tree. Solid lines denote deterministic E-edges, broken lines — nondeterministic ones.

are reversed.) We employ a standard convention and for a structure $\mathfrak{A}$ denote by A its universe. Also if $B \subseteq A$ then we denote by $\mathfrak{B}$ the substructure induced by B. An element of A such that $\mathfrak{A} \models \neg\overline{E}[a,b]$ for all elements $b \in A$ is a *root*. Note that while each tree may contain at most one root there may be trees without a root. We distinguish three classes of trees: A tree $\mathfrak{T}$ is *rooted* if it contains a root; it is *cyclically-rooted* if it contains a set of elements $a_0,\ldots,a_{k-1}$ such that $\mathfrak{T} \models E[a_i, a_{(i+1) \bmod k}]$ for all $0 \le i < k$; and it is *top-unbounded* if it is neither rooted nor cyclically-rooted. Note that a top-unbounded tree $\mathfrak{T}$ contains an infinite sequence of distinct elements $a_0, a_1, \ldots$ such that $\mathfrak{T} \models E[a_i, a_{i+1}]$ for $i \ge 0$. A rooted or cyclically-rooted tree may be finite or infinite. See Fig.1 for an illustration of types of trees.

2.2 Two-variable logic

Our main contribution concerns the two-variable logic with the deterministic transitive closure of a single binary relation E, $FO^2 + DTC(E)$. In this language we can say, e.g., that E is a partial function: $\forall xy(E(x,y) \rightarrow \overline{E}(x,y))$, or that models are rooted trees (with reversed edges). To do the latter we say that there exists a unique root: $\exists x(R(x) \wedge \forall y(R(y) \rightarrow x = y) \wedge \forall y \neg E(x,y))$, and that all other elements have deterministic paths to the root: $\forall xy(R(x) \wedge \neg R(y) \rightarrow \overline{E}(y,x))$. We can also easily see that our logic does not have the finite model property, e.g., the formula $\forall x(\neg\overline{E}(x,x) \wedge \exists y \overline{E}(x,y))$ has only infinite models.

We are going to present our results for a slightly richer logic, $FO^2 + DTC(E) + [*]$. The main reason for introducing this extension is that it allows to present our proofs in a more structured and transparent way. Namely, it will allow us to work, to some extent, on two independent levels: a local level of individual trees and a global level of connections among different trees (in which the internal structure of trees is essentially irrelevant). Besides, we believe that its additional expressive power (in particular the predicate L) makes it even more attractive.

Formally, we work with signatures of the form $\tau = \tau_0 \cup \tau_s$, where τ_0 contains some unary symbols and the binary symbol E, all of which may be interpreted arbitrarily, and $\tau_s = \{\overline{E}, L, Q, R^*, R_0, R_1, R_2\}$ consists of symbols whose interpretation is fixed. As usually $\overline{E}$ is the deterministic transitive closure of E; L is the equivalence closure of $\overline{E}$ (in other words, $\mathfrak{A} \models L[a,b]$ iff a and b belong to the same tree); the unary symbol Q is true precisely for elements which receive an E-edge from the root of their tree; the unary symbols R^*, R_0, R_1, R_2 mark roots: a root is marked with R_0 if it has no outgoing E-edges to its own tree and has at least two outgoing E-edges to the other trees; a root is marked R_1 if it has precisely one outgoing E-edge to its own tree and has at least one outgoing E-edge to the other trees; a root is marked R_2 if it has at least two outgoing E-edges to its own tree and has an arbitrary number (possibly zero) of outgoing E-edges to the other trees. Finally, a root is marked R^* if it has no outgoing E-edges at all. We say that a τ-structure in which the interpretation of the symbols from τ_s meets the above listed requirements is *admissible*. We will write $\mathfrak{A} \models \varphi$ if $\mathfrak{A}$ is a model of φ in classical sense, and additionally, it is admissible.

We sometimes consider satisfaction of $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formulas in trees which are intended to become substructures of admissible structures. While a cyclically-rooted or top-unbounded tree which is a substructure of an admissible structure must be admissible itself, it is not necessarily the case for rooted trees, as, obviously, in a structure consisting of a single tree a root satisfying R_0 or R_1 have no other trees to send the required (respectively, two or one) E-edges. To deal with this subtlety we write $\mathfrak{T} \models^* \varphi$, for a tree $\mathfrak{T}$ and an $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula φ, if $\mathfrak{T}$ is a model of φ in classical sense and it is either admissible or it contains a root satisfying R_i ($i = 0, 1$) and would become admissible after emitting at least $2 - i$ E-edges (to some other trees) by this root.

2.3 Universal fragment of first-order logic

By $\exists^* \wedge \forall^*$ we denote the class of all formulas having the following shape for some natural numbers m, n:

$$(\bigwedge_{1 \le i \le m} \exists x \psi_i(x)) \wedge \forall x_1 \ldots x_n \psi_0(x_1, \ldots, x_n),$$

where ψ_i is quantifier-free. We also define in a natural way the restrictions with bounded number of universally quantified variables, and denote them $\exists^* \wedge \forall^n$, for $n \in \mathbb{N}$.

This class is closely related to the fragment considered in [12], i.e., the class of universal prenex formulas with constants. Even though we do not explicitly allow constants, their effect can be simulated by formulas of the form $\exists x P_c(x) \wedge \forall xy (P_c(x) \wedge P_c(y) \to x = y)$.

By $\exists^* \wedge \forall^n + \mathrm{DTC}(E_1, \ldots, E_k)$ we denote the extension of $\exists^* \wedge \forall^n$ by the deterministic transitive closures of binary relations $E_1, \ldots, E_k$. Technically, as in the case of FO^2, we employ pairs of binary relation symbols E_i, $\overline{E_i}$, and require $\overline{E_i}$ to be interpreted as the deterministic transitive closure of E_i. If not stated otherwise we assume that E_i and $\overline{E_i}$ are the only non-unary relations.

2.4 Atomic types

An (atomic) 1-*type* (over a given signature) is a maximal satisfiable set of atoms or negated atoms with free variable x. Similarly, an (atomic) 2-*type* is a maximal satisfiable set of atoms and negated atoms with free variables x, y. We assume that literals built using the symbols of fixed interpretation, i.e., the symbols from τ_s, are also members of atomic types. Note that the numbers of 1-types and 2-types are bounded exponentially in the size of the signature. We often identify a type with the conjunction of all its elements.

For a given τ-structure $\mathfrak{A}$, and $a \in A$ we say that *a realises* a 1-type α if α is the unique 1-type such that $\mathfrak{A} \models \alpha[a]$. We denote by $\mathrm{tp}_{\mathfrak{A}}(a)$ the 1-type realised by a. Similarly, for distinct $a, b \in A$, we denote by $\mathrm{tp}_{\mathfrak{A}}(a, b)$ the unique 2-type *realised* by the pair a, b, i.e., the type β such that $\mathfrak{A} \models \beta[a, b]$. We denote by $\boldsymbol{\alpha}[\mathfrak{A}]$ the set of all 1-types realised in $\mathfrak{A}$. For $S \subseteq A$, we denote by $\boldsymbol{\alpha}_{\mathfrak{A}}[S]$ the set of all 1-types realised in S. The set of all 1-types over a signature τ is denoted by $\boldsymbol{\alpha}[\tau]$.

3. Two-variable logic

3.1 Normal form and plan of the proof

We start this section with an adaptation of the well-known Scott normal form for two-variable logic [30]. Namely, we separate the conjuncts concerning the connections inside trees from those concerning the connections between elements from different trees.

We say that an $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula φ is in *normal form* if $\varphi = \varphi^l \wedge \varphi^g$ for some

$$\varphi^l = \forall xy (L(x, y) \to \varphi^l_0(x, y)) \wedge \bigwedge_{i=1}^{m^l} \forall x (\lambda^l_i(x) \to \exists y (L(x, y) \wedge \varphi^l_i(x, y))),$$

$$\varphi^g = \forall xy (\neg L(x, y) \to \varphi^g_0(x, y)) \wedge \bigwedge_{i=1}^{m^g} \forall x (\lambda^g_i(x) \to \exists y (\neg L(x, y) \wedge \varphi^g_i(x, y))),$$

where λ^l_i and λ^g_i are atomic formulas, φ^l_i and φ^g_i are quantifier-free and do not use the relation symbol L; additionally φ^g_i do not use $\overline{E}$ (but can use E). We say that φ^l, φ^g are, respectively the *local* and *global* parts of φ.

We employ the following standard terminology. Let $\mathfrak{A}$ be an admissible structure. We say that an element b is an ith *local* (*global*) *witness* for an element a if $\mathfrak{A} \models \lambda^l_i[a]$ ($\mathfrak{A} \models \lambda^g_i[a]$) and $\mathfrak{A} \models L[a, b] \wedge \varphi^l_i[a, b]$ ($\mathfrak{A} \models \neg L[a, b] \wedge \varphi^g_i[a, b]$).

The following lemma can be proved in a standard fashion.

Lemma 1. *Let φ be an $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula over a signature τ. Then there exists a polynomially computable normal form $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula φ' (over a signature τ' extending τ by some fresh unary predicates) such that φ is satisfiable iff φ' is. Moreover any model of φ can be expanded (by an appropriate interpretation of symbols from $\tau' \setminus \tau$) to a model of φ', and the restriction of any model of φ' to τ is a model of φ.*

To some extent, we will work on two independent levels, local and global. For a given normal form $\varphi = \varphi^l \wedge \varphi^g$, the task on the local level will be to check if there exist trees $\mathfrak{T} \models^* \varphi^l$ meeting some additional simple requirements; on the global level we will be interested in satisfaction of φ^g in models in which the internal structure of trees is essentially irrelevant.

We first announce a lemma whose purpose is to take care of the above mentioned local task.

Lemma 2. *Let $\boldsymbol{\alpha}$ be a set of 1-types over a signature τ. Let φ^l be a normal form $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula with empty global part. Then it can be verified in space $\mathfrak{f}(|\boldsymbol{\alpha}[\tau]|) + \mathfrak{g}(|\varphi^l|)$, for some fixed polynomial function $\mathfrak{f}$ and exponential function $\mathfrak{g}$, if there exists a (finite) tree $\mathfrak{T} \models^* \varphi^l$ such that $\boldsymbol{\alpha}[\mathfrak{T}] = \boldsymbol{\alpha}$.*

We postpone the proof of Lemma 2 until Section 3.3, but it is worth mentioning that if we would be interested only in finite rooted trees with no nondeterministic edges from the root to its nodes then we could use the results from [2], where decidability of FO^2 over finite trees (and even finite ordered trees) was shown to be in ExpSpace. In full generality we will need however to deal with both finite and infinite trees and also to consider cyclically rooted and top unbounded trees.

We concentrate now on developing some tools intended to deal with our global level tasks.

3.2 Global Level

The main work of this section is done in the following lemma which shows that a satisfiable formula has a model with small number of trees. Additionally, a small set of elements can be chosen, sufficient to provide the required global witnesses for all (potentially infinitely many) elements of the model, and which itself guarantees that the admissibility conditions related to R_i-predicates are satisfied.

Lemma 3. *Let $\mathfrak{A} \models \varphi$ be a τ-structure, a model of a normal form $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula φ. Then there exists a model $\mathfrak{A}' \models \varphi$*

over a signature τ' extending τ by polynomially many fresh unary predicates, and a set $B \subseteq A'$ of cardinality at most $\mathfrak{h}(|\boldsymbol{\alpha}[\tau]|)$, for fixed polynomial function $\mathfrak{h}$ such that:
(i) $\mathfrak{A}'$ consists of k trees $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$, for some $k \leq \mathfrak{h}(|\boldsymbol{\alpha}[\tau]|)$,
(ii) if $\mathfrak{A}' \models \lambda_i^g[a]$ then a has an ith global witness in B,
(iii) if for the root a of a tree $\mathfrak{T}_j \subseteq \mathfrak{A}'$ we have $\mathfrak{A}' \models R_i[a]$ $(i = 0, 1)$ then a has at least $2-i$ E-edges to elements in $B \smallsetminus T_j$,
(iv) if for the root a of a tree $\mathfrak{T}_j \subseteq \mathfrak{A}'$ we have $\mathfrak{A}' \models R_1[a]$ or $\mathfrak{A}' \models R_2[a]$ then a has, respectively, precisely one edge or at least two edges to elements in $T_j \cap B$,
(v) all elements from B belonging to the same tree have different 1-types.

Proof. Let us first distinguish subsets B_0, B_1, B_2 in A. This is done in Steps 0-2 below.
Step 0. For each 1-type α realised in $\mathfrak{A}$ mark 5 trees containing a realisation of α (or all such trees if there are less than 5 of them); in each of them mark 3 realisations of α (or all such realisations if there are less than three of them). Let B_0 be the set of all marked elements. Note that $|B_0| \leq 15|\boldsymbol{\alpha}[\tau]|$.
Step 1. For each $a \in B_0$ chose now a minimal set of elements B_a securing conditions (i) and (ii) for a, i.e. such that:

(a) a has all the required global witnesses in $B_0 \cup B_a$, and
(b) if a is a root of a tree $\mathfrak{T} \subseteq \mathfrak{A}$ such that $\mathfrak{A} \models R_i[a]$ for $i = 0, 1$ then a has at least $2 - i$ edges to $(B_0 \cup B_a) \smallsetminus T$.

First we add to B_a elements ensuring (a). Let us see that at most four elements of each 1-type suffice. Indeed, a fixed pair of 1-types may be extended to a 2-type in four possible ways, namely by $E(x,y) \wedge E(y,x)$, $E(x,y) \wedge \neg E(y,x)$, $\neg E(x,y) \wedge E(y,x)$ or $\neg E(x,y) \wedge \neg E(y,x)$. (Observe also that this maximal number of options concerns only the case when when both 1-types are root types, i.e., contain some R_i-predicates, since only roots are allowed to emit E-edges to other trees. If only one of the types is a root type then there are two possibilities, if none — just one possibility: $\neg E(x,y) \wedge \neg E(y,x)$.) Thus an element cannot require more than 4 global witnesses of the same 1-type. If after securing property (a) for a property (b) is not satisfied we simply add to B_a appropriately one or two elements to which a sends a global E-edge. Let $B_1 = \bigcup_{a \in B_0} B_a$. For each $a \in B_0$ we have $|B_a| \leq 4|\boldsymbol{\alpha}[\tau]|$ which implies that $|B_1| \leq |B_0| \cdot 4|\boldsymbol{\alpha}[\tau]|$.
Step 2. Repeat Step 1 with the role of B_0 played by B_1. (Note that in this step the sets B_a can be nonempty only for elements from $B_1 \smallsetminus B_0$.) Let B_2 be the set of elements chosen in this step. We have $|B_2| \leq |B_1 \smallsetminus B_0| \cdot 4|\boldsymbol{\alpha}[\tau]|$.

Let $B^* = B_0 \cup B_1 \cup B_2$. Note that $|B^*|$ is bounded by $15|\boldsymbol{\alpha}[\tau]| + 15|\boldsymbol{\alpha}[\tau]| \cdot 4|\boldsymbol{\alpha}[\tau]| + 15|\boldsymbol{\alpha}[\tau]| \cdot 4|\boldsymbol{\alpha}[\tau]| \cdot 4|\boldsymbol{\alpha}[\tau]| \leq 315|\boldsymbol{\alpha}[\tau]|^3$, polynomially in $|\boldsymbol{\alpha}[\tau]|$. We modify now some global connections between $A \smallsetminus (B_0 \cup B_1)$ and B_0 so that each element of $A \smallsetminus (B_0 \cup B_1)$ has all its required global witnesses in B_0, and each root from $A \smallsetminus (B_0 \cup B_1)$ marked by R_i (i=0, 1) has at least $2-i$ edges to elements in B_0 outside its tree. Again this is possible due to our strategy of choosing B_0: Consider an element $a \in A \smallsetminus (B_0 \cup B_1)$ of type α and think about its global requirements with respect to elements of type α'. Recall that if α' is realised in $\mathfrak{A}$ in at least five trees then we mark five of them. In this case, even if α=α', and a is in one of the marked trees it still has at least four elements of type α' in B_0 outside its tree, which, as explained above, is sufficient to provide all required global witnesses of type α' for a. It is readily verified that if necessary, in parallel with providing global witnesses, such four realisations of α' are enough to secure the global admissibility conditions related to R_i-predicates (i.e., to absorb one or two E-edges from a). If α' is realised in less than five trees in $\mathfrak{A}$ then a may require more than one element of type α' in some tree $\mathfrak{T}$ (in this case α' is not a root type, since a root type must be unique in its tree). In the worst case three realisations of α' in $\mathfrak{T}$ are required. This happens when both a connection containing $E(x,y)$ and a connection containing $\neg E(x,y)$ are necessary, α contains $R_0(x)$ and there is no way to realise the admissibility condition related to R_0 other than using another realisation of α' from $\mathfrak{T}$. Recall that in our marking strategy we take this possibility into account and, if it is possible, we take three realisations of α' to B_0 from each tree marked as containing realisations of α'.

Let us call the obtained structure $\mathfrak{A}^*$. Note that in $\mathfrak{A}^*$ the 1-types of elements and the internal structure of trees from $\mathfrak{A}$ are retained. Thus we clearly have $\mathfrak{A}^* \models \varphi^l$. Obviously after our surgeries we have $\mathfrak{A}^* \models \varphi^g$ and thus also $\mathfrak{A}^* \models \varphi$.

Let $\mathfrak{A}_0^*$ be the substructure of $\mathfrak{A}^*$ restricted to those trees which contain elements from B^*. Our construction guarantees that still $\mathfrak{A}_0^* \models \varphi$. The set B is obtained by adding to B^* some minimal number of elements so that (iv) is true for the trees from $\mathfrak{A}_0^*$. This can be done by choosing at most two new elements per tree. Finally, let $\mathfrak{A}'$ be obtained from $\mathfrak{A}_0^*$ by introducing fresh unary predicates (forming together with τ the signature τ') and making the 1-types of elements of B belonging to the same tree different in $\mathfrak{A}_0^*$. Clearly $\lceil \log_2 |B| \rceil$ (which is polynomial in $|\tau|$) new predicates suffice.

Let us see that $\mathfrak{A}'$ and B are as required. The passage from $\mathfrak{A}_0^*$ to $\mathfrak{A}'$ does not influence the satisfaction of φ. Thus $\mathfrak{A}' \models \varphi$. Property (i) follows from the given estimation on the size of B^* and from the observation that expanding B^* to B can at most treble the number of elements; (ii) follows from the fact that elements of B_0 have their witnesses in B_1, elements from B_1 — in B_2, and elements from B_2 and $A' \smallsetminus B$ — in B_0; (iii) is satisfied owing to our surgeries; for (iv) we take care in the passage from B^* to B, and (v) is guaranteed by colouring elements of B by means additional unary predicates. □

We will need one more technical definition. We say that a tree $\mathfrak{T}'$ *safely extends* a tree $\mathfrak{T}$ if $\boldsymbol{\alpha}[\mathfrak{T}'] = \boldsymbol{\alpha}[\mathfrak{T}]$ and if for any $\alpha \in \boldsymbol{\alpha}[\mathfrak{T}]$ the number of realisations of α in $\mathfrak{T}'$ is greater or equal to the number of realisations of α in $\mathfrak{T}$.

Lemma 4. *Let $\varphi = \varphi^g$ be an $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ normal form formula with empty local part. Let $\mathfrak{A} \models \varphi$ be a model consisting of trees $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$. Then for any set of trees $\mathfrak{T}_1', \ldots, \mathfrak{T}_k'$, such that $\mathfrak{T}_i'$ safely extends $\mathfrak{T}_i$ $(i = 1, \ldots, k)$ there exists a model $\mathfrak{A}' \models \varphi$ whose domain is the disjoint union of $T_1', \ldots, T_k'$, and $\mathfrak{A}' \restriction T_i' = \mathfrak{T}_i'$.*

Proof. Let $f : T_i' \to T_i$ be an onto function retaining 1-types. For each $a \in T_i'$, $b \in T_j'$, $i \neq j$, we set $\mathrm{tp}_{\mathfrak{A}'}(a,b) := \mathrm{tp}_{\mathfrak{A}}(f(a), f(b))$. All conjuncts of φ of type $\forall\forall$ are satisfied in $\mathfrak{A}'$ since the 2-types of pairs of elements from distinct trees are taken from $\mathfrak{A}$. It is also readily verified that all elements have enough global witnesses in $\mathfrak{A}'$ (this follows from the fact that f is onto) and thus that $\mathfrak{A}'$ is as required. □

Now we are ready to prove the main result of this paper.

Theorem 5. *The satisfiability (finite satisfiability) problems for $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ are in* ExpSpace.

Proof. Let φ be an $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula over a signature τ (we assume that τ consists of symbols used in φ; thus $|\tau| \leq |\varphi|$). By Lemma 1 we can assume that $\varphi = \varphi^l \wedge \varphi^g$ is in normal form. To verify if φ has a (finite) model we:
(i) guess k, $1 \leq k \leq \mathfrak{h}(|\boldsymbol{\alpha}[\tau]|)$, a signature $\tau' \supseteq \tau$ extending τ by some polynomially many fresh unary predicates, and a sequence $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ of sets of 1-types over τ';
(ii) for each i $(i = 1, \ldots, k)$ verify that there exists a tree (finite tree) $\mathfrak{T} \models^* \varphi^l$ such that $\boldsymbol{\alpha}[T] = \boldsymbol{\alpha}_i$;
(iii) guess a τ'-structure $\mathfrak{A}$, consisting of trees $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$, such that $\boldsymbol{\alpha}[T_i] = \boldsymbol{\alpha}_i$ and T_i contains precisely 1 element of each 1-type

from $\boldsymbol{\alpha}_i$ (the internal structure of $\mathfrak{T}_i$ is not relevant), and verify that $\mathfrak{A} \models \varphi^g$.

The outlined procedure can be implemented to work in nondeterministic exponential space: recall that $\mathfrak{h}$ is a polynomial function, and note that $|\boldsymbol{\alpha}[\tau]| \leq 2^{|\tau|}$, $|\tau'|$ is polynomial in $|\tau|$, $|\boldsymbol{\alpha}_i| \leq 2^{|\tau'|}$, and $|A| \leq k \cdot 2^{|\tau'|}$; thus all the guesses in (i) and (iii) are of exponential size. By Lemma 2, checking in (ii) if the appropriate trees exist can be done in space $\mathfrak{f}(\boldsymbol{\alpha}[\tau']) + \mathfrak{g}(|\varphi|)$, for polynomial function $\mathfrak{f}$ and exponential $\mathfrak{g}$, which is exponentially bounded in $|\varphi|$.

Finally, verification in (iii) if $\mathfrak{A} \models \varphi^g$ can be readily done in time polynomial in $|A|$ and $|\varphi^g|$.

We show now that the procedure has a successful run iff φ has a model (finite model). Assume that φ has a model (finite model). Take τ', a τ'-model (finite τ'-model) $\mathfrak{A}' \models \varphi$ and a subset of its domain B as guaranteed by Lemma 3. Choose the number k to be the number of trees in $\mathfrak{A}'$, and let $\mathfrak{T}'_1, \ldots, \mathfrak{T}'_k$ be those trees. Let $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}[T'_i]$ (for $i = 1, \ldots, k$). Clearly $\mathfrak{T}'_i \models^* \varphi^l$, which ensures that the verification in (ii) will be successful. Let $B' \supseteq B$ be such that for each $i = 1, \ldots, k$ the set $T'_i \cap B'$ contains precisely one realisation of each type realised in T'_i (this is possible, since by condition (v) of Lemma 3 the types realised in $T'_i \cap B$ are distinct). Let $\mathfrak{A}$ be a structure with domain B', retaining the 1-types of elements, and the 2-types of pairs of elements from distinct trees from $\mathfrak{A}'$, such that the universes of its trees are $T'_i \upharpoonright B'$ (the internal structure of trees is irrelevant, but note that the admissibility conditions on the roots marked with R_i predicates can be satisfied due to conditions (iii) and (iv) of Lemma 3). It is readily verified that $\mathfrak{A} \models \varphi^g$ as required in (iii).

In the opposite direction, assume that there is a successful run of the procedure. Let $k, \tau', \boldsymbol{\alpha}_1, \ldots \boldsymbol{\alpha}_k, \mathfrak{A} \models \varphi^g$ and $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$ be as guessed during this run. Let $\mathfrak{T}'_1, \ldots, \mathfrak{T}'_k$ be trees (finite trees) such that $\mathfrak{T}'_i \models \varphi^l$ and $\boldsymbol{\alpha}[\mathfrak{T}_i] = \boldsymbol{\alpha}_i$ (for $i = 1, \ldots, k$), as guaranteed by step (ii). Clearly $\mathfrak{T}'_i$ safely extends $\mathfrak{T}_i$ (for $i = 1, \ldots, k$). Let $\mathfrak{A}' \models \varphi^g$ be a model built from the trees $\mathfrak{T}'_i$, as guaranteed by Lemma 4. As each of its trees satisfies φ^l we get that $\mathfrak{A}' \models \varphi$. This finishes the proof. □

A matching ExpSpace-lower bound follows from ExpSpace-hardness of FO^2 over (unordered) trees [2], which in turn is a consequence of the construction in [14]. Thus we get the following corollary.

Corollary 6. *The satisfiability and finite satisfiability problems for* $\mathrm{FO}^2 + \mathrm{DTC}(E)$ *and* $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ *are* ExpSpace-*complete.*

3.3 Local Level

In this section we are going to prove Lemma 2. The main result from [2] (see also [6] and [3]) is strong enough to check satisfiability of local formulas over finite rooted trees. The main contribution of this section is extending the technique we used in [6] in the case of finite rooted trees to handle infinite, cyclically-rooted or top-unbounded trees.

3.3.1 Preliminaries

To simplify our task we assume w.l.o.g. that if a tree $\mathfrak{T}$ has a root then the root has no outgoing E-edge. Recall that generally a rooted tree may have some edges from the root to its nodes (marked by Q), and moreover, in the case of roots marked with R_1 or R_2 it even must have such edges. These edges need not however be explicitly considered, since they can be simulated by employing Q and a fresh unary predicate V. We replace in φ^l every occurrence of $E(x, y)$ by $E(x, y) \vee (R_1(x) \vee R_2(x)) \wedge Q(y)$, and add to φ^l conjuncts which take care of the consistency between R_i root markers and the number of realisations of Q as follows.

$$\forall x (R_1(x) \to \exists y (L(x, y) \wedge Q(y) \wedge V(y)))$$
$$\forall x (R_2(x) \to \exists y (L(x, y) \wedge Q(y) \wedge V(y)))$$
$$\forall x (R_2(x) \to \exists y (L(x, y) \wedge Q(y) \wedge \neg V(y)))$$
$$\forall xy (L(x, y) \to (Q(x) \wedge Q(y) \wedge V(x) \wedge V(y) \to x = y))$$
$$\forall xy (L(x, y) \to (R_1(x) \wedge Q(y) \to V(y)))$$
$$\forall xy (L(x, y) \to (R^*(x) \vee R_0(x) \to \neg Q(y))).$$

Moreover, in an admissible model any pair of elements from a tree is connected by L, thus for our current purposes predicate L is irrelevant and we assume that it is not used (just replace each atom built of L by *true*). Further, we may add a conjunct stating that there is at most one element satisfying a root marker. The above assumptions allow us to consider only the following admissibility conditions in this section: $\overline{E}$ is the deterministic transitive closure of E, all edges are deterministic and a model is a single tree. For simplicity, we will write $\mathfrak{T} \models \varphi$ if φ is satisfied in a model $\mathfrak{T}$ admissible in the above sense (instead of technically correct $\mathfrak{T} \models^* \varphi$).

Now we adapt some notions from [6]. We use $x \nsim y$ to abbreviate the formula stating that x and y are in *free position*, i.e., the formula $x \neq y \wedge \neg\overline{E}(x, y) \wedge \neg\overline{E}(y, x)$. Let us call the formulas specifying the relative position of a pair of elements in a tree with respect to binary predicates *order formulas*. There are ten possible order formulas. Four of them specify a relative position of two distinct elements of the cycle in a cyclically-rooted tree: all these formulas are conjunctions of $\overline{E}(x, y) \wedge \overline{E}(y, x)$ and one of $E(x, y) \wedge E(y, x)$, $E(x, y) \wedge \neg E(y, x)$, $\neg E(x, y) \wedge E(y, x)$ or $\neg E(x, y) \wedge \neg E(y, x)$. We denote these formulas resp. by $\theta^C_{\uparrow\downarrow}, \theta^C_{\uparrow}, \theta^C_{\downarrow}$ and $\theta^C_{\Uparrow\Downarrow}$. The remaining formulas are $\overline{E}(x, y) \wedge \neg\overline{E}(y, x) \wedge E(x, y)$, $\overline{E}(y, x) \wedge \neg\overline{E}(x, y) \wedge E(y, x)$, $\overline{E}(x, y) \wedge \neg\overline{E}(y, x) \wedge \neg E(x, y)$, $\overline{E}(y, x) \wedge \neg\overline{E}(x, y) \wedge \neg E(y, x)$, $x \nsim y$, $x{=}y$. They are denoted, resp., as: $\theta_{\uparrow}$, $\theta_{\downarrow}$, $\theta_{\Uparrow}$, $\theta_{\Downarrow}$, $\theta_{\nsim}$, $\theta_{=}$. Let Θ be the set of these ten formulas.

A *full type* is a function $\bar{\alpha} : \Theta \to \mathcal{P}(\boldsymbol{\alpha}[\tau])$. We employ the following convention: for a given full type $\bar{\alpha}$ we denote by α the unique member of $\bar{\alpha}(\theta_{=})$. For a given tree $\mathfrak{T}$, and $v \in T$, we denote by $\mathrm{ftp}_{\mathfrak{T}}(v)$ the full type *realised* by v, i.e., the unique full type $\bar{\alpha}$, such that α is the 1-type of v, and for all $\theta \in \Theta$ we have that $\bar{\alpha}(\theta) = \{\mathrm{tp}_{\mathfrak{T}}(w) : \mathfrak{T} \models \theta[v, w]\}$.

A *reduced type* is a tuple $\langle \alpha, A, B, F \rangle$, where α is a 1-type and A, B, F are sets of 1-types. Reduced types are used to keep information recorded in full types in a (lossy) compressed form. We will use reduced types only in the context of elements that are not on a cycle, so in particular we ignore the components corresponding to the four θ^C formulas. Let v be a node in $\mathfrak{T}$ such that v is not on a cycle and $\mathrm{ftp}_{\mathfrak{T}}(v) = \bar{\alpha}$. By $\mathrm{rftp}_{\mathfrak{T}}(v)$ we denote the reduced type *realised* by v, i.e., the reduced type $\langle \alpha, A, B, F \rangle$, such that $A = \bar{\alpha}(\theta_{\uparrow}) \cup \bar{\alpha}(\theta_{\Uparrow})$, $B = \bar{\alpha}(\theta_{\downarrow}) \cup \bar{\alpha}(\theta_{\Downarrow})$ and $F = \bar{\alpha}(\theta_{\nsim})$. Note that α denotes the 1-type of v, and, informally speaking, A is the set of 1-types of elements realised *above* v, B is the set of 1-types of elements realised *below* v, and F is the set of 1-types of the elements realised in *free* position to v. We will refer to the sets A, B and F as the $\theta_{\uparrow^+}$, $\theta_{\downarrow^+}$ and $\theta_{\nsim}$ components of the reduced type.

We introduce a binary relation $\leq$ on reduced types as follows. $\langle \alpha, A, B, F \rangle \leq \langle \alpha', A', B', F' \rangle$ if $(B \subseteq B') \wedge (A' \subseteq A) \wedge (F' \subseteq F)$. Technically speaking, this relation on types is a preorder, but restricted to triples $\langle A, B, F \rangle$ this is (a finite and thus a well-founded) order. Note that chains in this order are of at most exponential length. Reduced types ρ and ρ' such that $\rho \leq \rho'$ and $\rho' \leq \rho$ are called *equivalent*.

A *pseudo root* in a cyclically-rooted tree $\mathfrak{T}$ is any node on the cycle in $\mathfrak{T}$; in a top-unbounded tree it is a node with a maximal wrt. ≤ reduced type. Note that in a top-unbounded tree each path contains only pseudo roots from some point on.

Local normal form We say that a local formula φ is in *local normal form* if

$$\varphi = \forall xy\chi(x,y) \wedge \bigwedge_{i\in I} \forall x(\lambda_i(x) \Rightarrow \exists y(\eta_i(x,y) \wedge \psi_i(x,y))),$$

for some index set I, where $\chi(x,y)$ is quantifier-free, $\lambda_i(x)$ is an atomic formula $a(x)$ for some unary symbol a, $\psi_i(x,y)$ is a boolean combination of unary atomic formulas, and $\eta_i(x,y)$ is an order formula. Note that in χ the equality symbol may be used, e.g., we can enforce that a model contains at most one node satisfying P: $\forall xy(P(x) \wedge P(y) \Rightarrow x{=}y)$. The local part φ^l of any $\mathrm{FO}^2 + \mathrm{DTC}(E) + [*]$ formula can be transformed into a formula in local normal form that is equivalent to φ^l on all trees in a standard way.

For a given formula φ in local normal form and a full type $\bar{\alpha}$, we say that $\bar{\alpha}$ is *φ-consistent* if an element realising $\bar{\alpha}$ cannot be a member of a pair violating the universal conjunct $\forall xy\chi(x,y)$ of φ, and has all witnesses required by φ. Formally, $\bar{\alpha}$ is φ-consistent if for every $\theta \in \Theta$, and every $\alpha' \in \bar{\alpha}(\theta)$ we have $\alpha(x) \wedge \alpha'(y) \wedge \theta(x,y) \models \chi(x,y) \wedge \chi(y,x)$, and for every conjunct $\forall x(\lambda_i(x) \Rightarrow \exists y(L(x,y) \wedge \eta_i(x,y) \wedge \psi_i(x,y)))$ of φ, such that $\alpha(x) \models \lambda_i(x)$, there exists a 1-type $\alpha' \in \bar{\alpha}(\eta_i)$ such that $\alpha(x), \alpha'(y) \models \psi_i(x,y)$. A proof of the following proposition is straightforward.

Proposition 7. *Let $\mathfrak{T}$ be a tree and let φ be a formula in local normal form. Then $\mathfrak{T} \models \varphi$ iff every full type realised in $\mathfrak{T}$ is φ-consistent.*

We say that a full type $\bar{\alpha}$ is *combined* of two full types $\bar{\alpha}_1$ and $\bar{\alpha}_2$ if $\alpha{=}\alpha_1{=}\alpha_2$ and for each $\theta \in \Theta$ we have $\bar{\alpha}(\theta) = \bar{\alpha}_1(\theta)$ or $\bar{\alpha}(\theta) = \bar{\alpha}_2(\theta)$. Also the following fact is immediate.

Proposition 8. *Let φ be a formula in local normal form, and let $\bar{\alpha}$ be a full type combined of two φ-consistent full types $\bar{\alpha}_1, \bar{\alpha}_2$. Then $\bar{\alpha}$ is φ-consistent.*

3.3.2 Short cycles and small degree

In the following we will argue that we can bound the length of cycles in cyclically-rooted trees and the degree of nodes in all trees that are models of local formulas. We start by showing how to shorten a path in a model — we will use it later in the construction of gadgets, but we show it here because it is conceptually simpler then shortening cycles.

Lemma 9. *Let φ be a formula in local normal form, $\mathfrak{T}$ its model, and $v, w \in T$ two nodes of $\mathfrak{T}$, such that $\mathfrak{T} \models \overline{E}(w,v)$. If u, v are not elements of a cycle in a cyclically-rooted tree and $\mathrm{rftp}_{\mathfrak{T}}(v) = \mathrm{rftp}_{\mathfrak{T}}(w)$, then the tree $\mathfrak{T}'$, obtained from $\mathfrak{T}$ by replacing the subtree rooted at v by the subtree rooted at w, is a model of φ.*

Proof. It can be verified that for every $u \in T'$, if $u{\not\approx}w$ then $\mathrm{ftp}_{\mathfrak{T}'}(u) = \mathrm{ftp}_{\mathfrak{T}}(u)$, and that $\mathrm{ftp}_{\mathfrak{T}'}(w)$ is combined of $\mathrm{ftp}_{\mathfrak{T}}(v)$ and $\mathrm{ftp}_{\mathfrak{T}}(w)$. Thus, by Propositions 7 and 8, all types realised in $\mathfrak{T}'$ are φ-consistent, and $\mathfrak{T}' \models \varphi$. □

Shortening a cycle requires a bit more care than shortening a path, but it can be done in a very similar way.

Lemma 10. *Let φ be a formula in local normal form, $\mathfrak{T}$ its cyclically-rooted model. Then there exists a cyclically-rooted tree $\mathfrak{T}'$ such that $\mathfrak{T}' \models \varphi$ and the cycle in $\mathfrak{T}'$ has at most exponential length. Moreover $\mathfrak{T}'$ can be obtained by removing from $\mathfrak{T}$ some number of elements (together with the subtrees rooted at them).*

Proof. Let C (the cycle) be the set of all pseudo roots in $\mathfrak{T}$ and let $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\mathfrak{T}}[C]$ be the set of 1-types of elements in C. We are going to mark some important elements of C and then remove some paths on C connecting elements of the same 1-type and not containing marked elements, thus producing a model $\mathfrak{T}' \models \varphi$. For all $c \in C$ let $\mathfrak{T}_c$ be the subtree of $\mathfrak{T}$ obtained by first removing from $\mathfrak{T}$ all edges connecting elements from C and then taking the subtree rooted at c. For $\alpha \in \boldsymbol{\alpha}$ let C_α be the set of pseudo roots in C of 1-type α and let C'_α be the set of pseudo roots in C such that tree $\mathfrak{T}_c$ contains a non-root element of 1-type α.

For each α mark $\min(2, |C'_\alpha|)$ elements in C'_α. Additionally mark some elements in C such that for each $\alpha \in \boldsymbol{\alpha}$ there is at least one marked element of 1-type α. There are now at most $3|\boldsymbol{\alpha}[\tau]|$ marked elements.

If there exist two non-marked elements $v, w \in C$ such that (i) $\mathrm{tp}_{\mathfrak{T}}(v) = \mathrm{tp}_{\mathfrak{T}}(w)$, (ii) the distance from w and v to the closest marked element is at least 2, and (iii) the path π from v to w contains no marked elements, then path π can be shortened (i.e., all nodes c between v and w, including v and excluding w, together with subtrees $\mathfrak{T}_c$, can be removed from $\mathfrak{T}$). Let $\mathfrak{T}''$ be the structure obtained from $\mathfrak{T}$ by this shortening.

Observe that $\mathrm{ftp}_{\mathfrak{T}''}(w)$ is combined of $\mathrm{ftp}_{\mathfrak{T}}(v)$ and $\mathrm{ftp}_{\mathfrak{T}}(w)$ (this is obvious for $\theta^C_\uparrow$ and $\theta^C_\downarrow$ and the six components not related to the cycle; $\theta^C_{\uparrow\downarrow}$ component is empty in both types and $\theta^C_{\not\uparrow\not\downarrow}$ component equals $\boldsymbol{\alpha}$ in both types), so it is φ-consistent. It can be verified that for every $u \in T''$, if $u{\not\approx}w$ then $\mathrm{ftp}_{\mathfrak{T}''}(u) = \mathrm{ftp}_{\mathfrak{T}}(u)$: this is quite obvious for all θ components of full types for $\theta \in \Theta \setminus \{\theta_{\not\approx}\}$; in the case of $\theta_{\not\approx}$ component, the non-root nodes of 1-type α below elements in C'_α guarantee existence of all needed free witnesses, which implies the equality of $\theta_{\not\approx}$ components of full types. Thus, by Propositions 7 and 8, all types realised in $\mathfrak{T}''$ are φ-consistent, and $\mathfrak{T}'' \models \varphi$. By repeating this shortening we obtain a cyclically-rooted tree $\mathfrak{T}'$ with a cycle of at most exponential length. □

Our final observation in this subsection is that to provide all witnesses for $\forall\exists$ conjuncts of φ we only need nodes with at most exponential degree.

Lemma 11. *Let φ be a formula in local normal form and let $\mathfrak{T} \models \varphi$. Then there exists a model $\mathfrak{T}' \models \varphi$ in which the number of predecessors of each node is bounded by $4|\boldsymbol{\alpha}[\mathfrak{T}]|$. Moreover $\mathfrak{T}'$ can be obtained by removing from $\mathfrak{T}$ some number of elements (together with the subtrees rooted at them).*

Proof. As in the previous lemma we are going to mark some important elements and remove trees rooted at unmarked elements. First, in the case of cyclically-rooted trees and top-unbounded trees we choose some pseudo root r and we mark all nodes on the path starting in r. Then we repeat the following procedure.

We choose a (marked) node v in the structure and the set of its predecessors V. Let V_α be the set of elements from V of 1-type α and let V'_α be the set of elements $e \in V$ such that the tree $\mathfrak{T}_e$ rooted at e contains a non-root element of 1-type α. For each α we mark $\min(2, |V_\alpha|)$ elements in V_α and $\min(2, |V'_\alpha|)$ elements in V'_α, and remove from $\mathfrak{T}$ the subtrees rooted at all unmarked elements in V.

We repeat this procedure inductively infinitely many times, starting from the chosen pseudo root and following to all predecessors of the processed node. In the case of top-unbounded trees we also follow in the inductive step the chosen path to the successors of processed nodes. Then $\mathfrak{T}'$ is the structure obtained in the limit (it is the intersection of all intermediate structures processed in all iterations of the procedure). Obviously $\mathfrak{T}'$ is nonempty (it contains the initial pseudo root). It can be verified that for every $u \in T'$ the types $\mathrm{ftp}_{\mathfrak{T}'}(u)$ and $\mathrm{ftp}_{\mathfrak{T}}(u)$ are equal, so by Proposition 7 all types realised in $\mathfrak{T}'$ are φ-consistent, and $\mathfrak{T}' \models \varphi$. □

3.3.3 Gadgets

Let G be a finite pointed graph (a graph with a distinguished node called root). By unfolding of G we mean a tree $\mathfrak{T}_G$ obtained from G by the following procedure.

- The root of $\mathfrak{T}_G$ is a node corresponding to the root of G,
- if v' is a node in $\mathfrak{T}_G$ corresponding to a node v in G and v has immediate predecessors $v_1, \dots, v_n$ in G then v' has n predecessors $v'_1, \dots, v'_n$ corresponding respectively to $v_1, \dots, v_n$.

Figure 2 shows an example of a graph (on the right) and its unfolding (on the left). Note that the procedure above does not unfold the successors of the root — this is done on purpose, to correctly handle gadgets for cyclically-rooted trees. However, to correctly handle top-unbounded trees we will also need gadgets where the successors of the root are unfolded, too. We will refer to such gadgets explicitly as *top-unbounded gadgets*.

Definition 12 (Gadget)**.** *Let ρ be a reduced type. A ρ-gadget is a finite pointed graph G such that replacing in any model the subtree rooted at any element of type ρ by the unfolding of G results in a model.*

Note that unfoldings of gadgets may be used only at nodes that are roots of subtrees. They cannot be used to replace substructures of cyclically-rooted trees at cycle nodes.

Lemma 13. *Let φ be a formula in local normal form and let $\mathfrak{T} \models \varphi$. If ρ is a reduced type realised in $\mathfrak{T}$ then there exists a ρ-gadget.*

Proof. Our construction of the gadget employs well-founded induction wrt. $\preceq$. Note that if a node in $\mathfrak{T}$ is a leaf then the $\theta_{\downarrow_+}$ component of its reduced type is empty, so when $\mathfrak{T}$ has leaves, the induction starts in a leaf with maximal components $\theta_{\uparrow^+}$ and $\theta_{\not\updownarrow}$ in the reduced type. But $\mathfrak{T}$ does not have to have a leaf.

For the induction step (which in the case of well-founded induction includes the induction basis) take any element e of arbitrary reduced type $\rho = \langle \alpha, A, B, F \rangle$ and assume that for all types ρ' that are strictly smaller than ρ (according to $\preceq$) the ρ'-gadgets are already defined.

Since e has a reduced type, it is not an element of a cycle and there exists the subtree $\mathfrak{T}_e$ of $\mathfrak{T}$ rooted at e. We start by finding important nodes in $\mathfrak{T}_e$. Let $I = \{\alpha_1, \dots, \alpha_k\}$ be the set of 1-types such that for all $i \in \{1, \dots, k\}$ the types $\langle \alpha_i, A, B, F \rangle$ are realised in $\mathfrak{T}_e$ strictly below the root e. For each of these types choose an element e_i in $\mathfrak{T}_e$ such that $e_i \neq e$ and $\mathrm{rftp}_{\mathfrak{T}}(e_i) = \langle \alpha_i, A, B, F \rangle$. Let $B = \{\alpha_1, \dots, \alpha_l\}$ for some $l \geq k$. For all $i \in \{1, \dots, k\}$ and $j \in \{1, \dots, l\}$ we find an element $e_{i,j}$ that is below e_i and has 1-type α_j.

Let $\mathfrak{S}$ be the least subtree of $\mathfrak{T}$ that

- contains e,
- contains all nodes $e_{i,j}$ for $i \in \{1, \dots, k\}$ and $j \in \{1, \dots, l\}$, and
- contains all immediate predecessors (in $\mathfrak{T}$) of all nodes that are not leaves of $\mathfrak{S}$.

The ρ-gadget is obtained by removing leaves from $\mathfrak{S}$ according to the following rules.

- Leaves of types strictly smaller than ρ are replaced with their respective gadgets, and
- leaves of type equivalent to ρ (that is, of type $\langle \alpha_j, A, B, F \rangle$) are removed and the arrows leading from them are replaced with arrows from e_j.

To see that the constructed graph G is a ρ-gadget we have to show that replacing in any model the subtree rooted at any element of type ρ by the unfolding of G results in a model. By Propositions 7 and 8 it is enough to show that the construction preserves reduced types of corresponding nodes in both structures. This can be proved by a simple induction on the structure of the subtree $\mathfrak{S}$. $\square$

Example 14. *The left part of Figure 2 shows an example of an infinite tree with root of reduced type $\langle \alpha_1, \varnothing, \{\alpha_1, \alpha_2, \alpha_3\}, \varnothing \rangle$. The most interesting nodes in this tree are the black ones, let us call them n_1 and n_2, respectively of type $\rho_1 = \langle \alpha_2, \{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_2, \alpha_3\}, \{\alpha_1, \alpha_2, \alpha_3\} \rangle$ and $\rho_2 = \langle \alpha_2, \{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_1, \alpha_2, \alpha_3\} \rangle$. After constructing the gadgets for leaves (which are graphs consisting of single nodes of type α_1), the type ρ_1 is a minimal according to $\preceq$ unprocessed reduced type; note that all nodes below n_1 have the type equivalent to ρ_1. We choose n_1 as a representative of all these nodes (the element e in Lemma 13) and construct a ρ_1-gadget, which is the subgraph (of the graph on the right of Figure 2) rooted at the black node. Similarly we choose n_2 as a representative of types equivalent to ρ_2 (which is a minimal type of non-leaf nodes in the right branch of the tree) and construct a respective ρ_2-gadget. Then by processing nodes above n_1 and n_2 we construct a gadget for the whole structure shown on the right.*

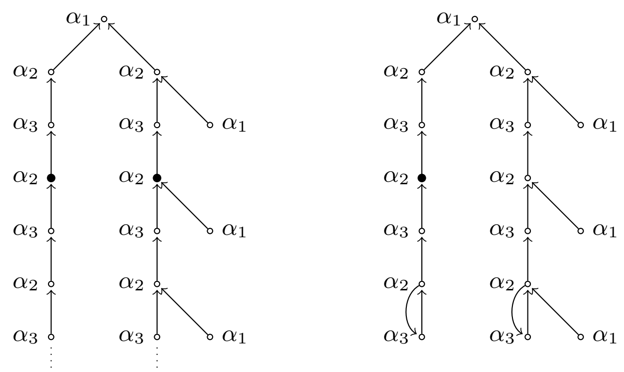

Figure 2. An infinite model $\mathfrak{T}$ and a gadget.

Let G be a $\langle \alpha, A, B, F \rangle$-gadget. We define a skeleton of G as the set of nodes equivalent (according to $\preceq$) to the root of G. By shortening long paths in a gadget in the same way as we did in Lemma 9 (more precisely: nodes e, e_i and $e_{i,j}$ are important, branching nodes on paths between important nodes are important; any path between important nodes may be required to consist of nodes of different reduced types) we obtain the following lemma.

Lemma 15. *Let φ be a formula in local normal form and let $\mathfrak{T} \models \varphi$. If ρ is a reduced type realised in $\mathfrak{T}$ then there exists a ρ-gadget G such that each sub-gadget of G has the skeleton of size at most exponential in the size of the formula φ.*

3.3.4 Graph representations of trees

A *graph representation* of a rooted tree with root of reduced type ρ is a ρ-gadget. A graph representation of a cyclically-rooted tree $\mathfrak{T}$ is a graph obtained from $\mathfrak{T}$ by replacing all nodes that are immediately below the cycle but not on the cycle by their corresponding gadgets. In the case of top-unbounded trees the construction of graph representation is a bit more complicated, as the following lemma shows.

Lemma 16. *Let φ be a formula in local normal form and let $\mathfrak{T}$ be a top-unbounded tree such that $\mathfrak{T} \models \varphi$. Then there exists a top-unbounded gadget G whose unfolding is a model of φ. Moreover, each sub-gadget of G has the skeleton of size at most exponential in the size of the formula φ.*

Proof. Let e be a pseudo root in $\mathfrak{T}$ and let $\mathrm{rftp}_{\mathfrak{T}}(e) = \langle \alpha, A, B, F \rangle$ be the reduced type of e. Let $A = \{\alpha_1, \ldots, \alpha_n\}$ and $F = \{\beta_1, \ldots, \beta_m\}$. Let π be the unique infinite path going upwards and starting in e. Choose nodes $e_1, \ldots, e_n$ on path π, of 1-types $\alpha_1, \ldots, \alpha_n$, respectively. Choose nodes $f_1, \ldots, f_m$ in free position to e, of 1-types $\beta_1, \ldots, \beta_m$, respectively. Let $e'_1, \ldots, e'_m$ be the first nodes on path π that are above $f_1, \ldots, f_m$, respectively. Let e' be a node on path π of 1-type the same as 1-type of e and above all these nodes. The nodes $e, e_1, \ldots, e_n, e'_1, \ldots, e'_m, e'$ are important and there are at most exponentially many of them. Now remove long paths between important nodes and replace an edge leading to e' by an edge leading to e, obtaining a cycle of exponential size, as in the case of cyclically-rooted trees. Now add gadgets for the predecessors of the cycle as in a graph representation of cyclically-rooted trees, with a little bit more care not to remove the nodes $f_1, \ldots, f_m$ from the gadgets. The top-unbounded unfolding of the obtained graph is a model of φ: the nodes below the pseudo-root behave like in the case of trees; for the nodes on the unfolding of the cycle, going upwards to a node corresponding to e_i we obtain an above-witness of type α_i and going first upwards to a node corresponding to e'_j and then downwards to a node corresponding to f_j we obtain a free-witness of type β_j. Below-witnesses can be found below the pseudo-root e. □

Now we are ready to prove Lemma 2. To check whether a local formula φ is satisfiable we check if there exists a graph representation G of its model. We start by guessing the type of the model (rooted, cyclically-rooted or top-unbounded). In the case of cyclically-rooted and top-unbounded trees we then guess the cycle together with all its predecessors; in the case of rooted trees we just guess a root. Every time when we guess a new node we also guess its full type and we check that it is φ-consistent and locally consistent with full types of its immediate neighbours. Then for each node that is not on a cycle, we guess its reduced type ρ and check that there exists a ρ-gadget. Checking existence of a ρ-gadget can be done by guessing a skeleton of exponential size and in each leaf universally checking that there exist respective sub-gadgets. Note that by Lemma 11 there are at most exponentially many leaves, the sub-gadgets have types strictly smaller according to $\preceq$ and all chains in $\preceq$ are exponentially bounded. Taking into account that the skeletons are of exponential size and while checking sub-gadgets we have to remember only the skeleton of the currently processed gadget, the whole procedure runs in alternating exponential time. Since AEXPTIME=EXPSPACE, checking satisfiability of local formulas is in EXPSPACE. Let $\mathfrak{T}$ be the structure obtained by unfolding of G. To ensure that $\boldsymbol{\alpha}[\mathfrak{T}] = \boldsymbol{\alpha}$ it is enough to check that the types occurring on the cycle (if $\mathfrak{T}$ is cyclically-rooted) are members of $\boldsymbol{\alpha}$ and that for every reduced type $\langle \alpha, A, B, F \rangle$ occurring in any gadget the union $\{\alpha\} \cup A \cup B \cup F$ equals $\boldsymbol{\alpha}$. For finite satisfiability it is enough to put additional restriction on gadgets (not to be confused with graph representations) that they must be acyclic.

4. Universal fragment of first-order logic

In [12] it is shown that the logic $\exists^* \wedge \forall^*$ extended by positive occurrences of the deterministic transitive closure of E has the finite model property and that its satisfiability problem is NEXPTIME-complete. It is also argued that allowing negative occurrences of DTC of E leads to undecidability. In this section we give another look at this logic. We will be interested in its fragments with bounded number of universally quantified variables, $\exists^* \wedge \forall^k + \mathrm{DTC}(E)$. In subsection 4.1 we observe that $\exists^* \wedge \forall^2 + \mathrm{DTC}(E)$ retains the finite model property and NEXPTIME-completeness of the satisfiability problem, even if the closures of E are used both positively and negatively. (As this logic is a fragment of $\mathrm{FO}^2 + \mathrm{DTC}(E)$ decidability follows from Section 3. Recall however that full $\mathrm{FO}^2 + \mathrm{DTC}(E)$ does not have the finite model property and that its satisfiability problem is EXPSPACE-complete.) In subsection 4.2 we discuss related undecidability results.

4.1 Finite model property for $\exists^* \wedge \forall^2 + \mathrm{DTC}(E)$

Theorem 17. *$\exists^* \wedge \forall^2 + \mathrm{DTC}(E)$ has an exponential model property. Thus its satisfiability and finite satisfiability problems are NEXPTIME-complete.*

Proof. Let $\varphi = (\bigwedge_{1 \le i \le m} \exists x \psi_i(x)) \wedge \forall xy \psi_0(x, y)$ be a satisfiable $\exists^* \wedge \forall^2 + \mathrm{DTC}(E)$ sentence over a signature τ. Since φ belongs to $\mathrm{FO}^2 + \mathrm{DTC}(E)$ we may convert it to normal form from Section 3.1. Let φ' be the resulting formula over an extended signature τ'. Let $\mathfrak{A}' \models \varphi'$ and $B \subseteq A'$ be as guaranteed by Lemma 3. Let $\mathfrak{A}$ be the restriction of $\mathfrak{A}'$ to τ. By Lemma 1 we have $\mathfrak{A} \models \varphi$. Let us recall the properties ensured by Lemma 3 (namely by parts (i), (iii) and (iv) of this lemma) important in our current scenario:

(i) the number of trees in $\mathfrak{A}$ and the size of B are exponentially bounded in $|\varphi|$,

(ii) if an element of $\mathfrak{A}$ emits at least two edges then it emits at least two edges to elements in B.

Let c_i ($1 \le i \le m$) be an element such that $\mathfrak{A} \models \psi_i[c_i]$, and let $C = \bigcup_{1 \le i \le m} \{c_i\}$. For any $a, b \in B \cup C$ let P_{ab} be the set of elements on the deterministic path from a to b if such a path exists in $\mathfrak{A}$, and the set $\{a, b\}$ otherwise. Let $B' = \bigcup_{a,b \in B \cup C} P_{ab}$. Clearly $B \subseteq B'$. Let $\mathfrak{B}' = \mathfrak{A} \restriction B'$ (we assume that when taking this restriction, E and $\overline{E}$ are treated as usual relational symbols; generally in such restrictions it is possible that $\overline{E}$ is not the deterministic transitive closure of E; as we will explain in a moment, this is not the case this time). Note that $\mathfrak{B}'$ is a finite structure.

We claim that $\mathfrak{B}' \models \varphi$. Indeed, the existential statements are satisfied due to the elements from C, and $\mathfrak{B}' \models \forall xy \psi_0(x, y)$, since $\mathfrak{B}$ is a substructure of $\mathfrak{A}$. It remains to see that $\overline{E}$ is the deterministic transitive closure of E in $\mathfrak{B}'$. The crucial observation is that

($\star$) an edge (a, b) is deterministic in $\mathfrak{B}'$ iff it is deterministic in $\mathfrak{A}$.

This statement follows from the fact that if an element emits at most one E-edge in $\mathfrak{A}$ then it cannot emit more in its substructure, and that if an element emits at least two E-edges in $\mathfrak{A}$ then it still emits at least two in $\mathfrak{B}'$ by (ii). Take arbitrary $a, b \in B'$. If $\mathrm{tp}_{\mathfrak{B}'}(a, b)$ contains $\overline{E}(x, y)$ then there is a deterministic path from a to b in $\mathfrak{A}$. This path must be a fragment of one of the paths $P_{a'b'}$ added to B' (or a whole such path), thus there is a deterministic path from a to b in $\mathfrak{B}'$. If $\mathrm{tp}_{\mathfrak{B}'}(a, b)$ does not contain $\overline{E}(x, y)$ then there cannot be a deterministic path from a to b in $\mathfrak{B}'$, since by ($\star$) such a path would be also deterministic in $\mathfrak{A}$.

$\mathfrak{B}'$ may still be very large, however it can be now simply decreased to the required size. This can be done as in the proof of Theorem 4 from [12]. Let us call elements of $B \cup C$ and the elements in which some paths P_{ab} meet *important*. Note that the number of important elements is exponentially bounded in $|\varphi|$. A fragment of a path with no important elements can be shortened in such a way that it contains at most one realisation of each 1-type. This leads to an exponentially bounded model.

Our construction implies the NEXPTIME-upper bound on the satisfiability (= finite satisfiability problem). A matching lower bound follows from the proof of Thm. 5 in [12]. □

4.2 Limits of decidability

A natural question appears if the decidability result from Thm. 17 can be extended by allowing more variables or more binary relations whose DTC can be used. This section is devoted to a discussion on this topic. Some related undecidability results were obtained already in [12]. They are gathered in the following theorem. The superscripts $^{+,-}$ are used with DTC to denote that only positive, respectively negative occurrences of DTC are allowed.

Theorem 18. *The satisfiability and finite satisfiability problems are undecidable for:*

(a) $\exists^* \wedge \forall^4 + \mathrm{DTC}^+(E_1, E_2)$ *(Thm. 7 in [12])*
(b) $\exists^* \wedge \forall^4 + \mathrm{DTC}^+(E)$ *with one additional binary predicate* B *(Thm. 8 in [12])*
(c) $\exists^* \wedge \forall^5 + \mathrm{DTC}^-(E)$ *(Thm. 13 in [12]).*

Moreover $\exists^* \wedge \forall^3 + \mathrm{DTC}^-(E)$ *does not have the finite model property (Prop. 11 in [12]).*

Additionally, Cor. 10 in [12] suggests a stronger variant of Thm. 18(a), namely the undecidability of $\exists^* \wedge \forall^2 + \mathrm{DTC}^+(E_1, E_2)$. However, the statement of that corollary is unclear, and no formal proof is given. It seems to us that to get the desired effect one needs to take DTC not over one additional formula γ but over at least several additional formulas γ_i. This translates to using DTC of several binary relations. Nevertheless, if we do not insist on using DTC only positively the undecidability of $\exists^* \wedge \forall^2$ with DTC of two relations can be obtained in a slightly different way. In [22] (see also [23], p. 67) it is shown by a reduction from the Post Correspondence Problem that FO^2 with unary relations, two linear orders and their induced successor relations is undecidable. We can naturally translate this proof to our scenario, with E_1, E_2 corresponding to the successor relations, and their deterministic transitive closures corresponding to the order relations. (To enforce that $\overline{E}_i$ is a finite linear order it suffices to say that there is an element a with in-degree 0, there is an element b with out-degree 0, and that any other element is accessible by a deterministic path from a and can access b by a deterministic path.) Actually, the proof in [23] uses some nested existential quantifiers, but it is not difficult to get rid of them. A closer inspection shows that indeed all the required properties can be expressed by using a $\exists^* \wedge \forall^2 + \mathrm{DTC}(E_1, E_2)$ formula. Thus we get:

Theorem 19. *The satisfiability and finite satisfiability problems for* $\exists^* \wedge \forall^2 + \mathrm{DTC}(E_1, E_2)$ *are undecidable.*

Thm. 18(b) can be easily improved by reducing the number of variables to 3. The proof from [12] defines a snake-like deterministic E-path through a finite grid, and then uses four variables to enforce the remaining edges of the grid, represented by an additional relation B, by *closing squares* formulas. If we add to our grid diagonal B-connections, then a similar effect may be obtained with employing only three variables and *closing triangles* formulas. This allows to state:

Theorem 20. *The satisfiability and finite satisfiability problems for* $\exists^* \wedge \forall^3 + \mathrm{DTC}^+(E)$ *with one additional binary symbol* B *are undecidable.*

It is worth commenting that the proof of Thm. 17 can be easily adapted to handle other binary symbols besides E (but not their deterministic transitive closures), which contrasts with the above theorem. Finally, let us sharpen Thm. 18(c).

Theorem 21. *The satisfiability and the finite satisfiability problems for* $\exists^* \wedge \forall^4 + \mathrm{DTC}^-(E)$ *are undecidable.*

Proof. Our proof incorporates the idea of enforcing an infinite path by means of negative occurrences of DTC from the proof of Thm. 18(c) given in [12]. However, our idea slightly differs from the ideas there and is a bit more straightforward, in particular we avoid using special *gadgets* defined in [12].

We consider first the general satisfiability problem. We describe a reduction from the undecidable octant tiling problem. The desired infinite octant grid structure is illustrated in Fig. 3. We use unary predicates R_i and C_i, $0 \le i \le 3$, intended to mark respectively rows and columns $4k + i$, for $k \in \mathbb{N}$. The bottom row is additionally indicated by the predicate L. The upmost elements in the columns are marked by D. Moreover, predicates P_0, P_1, P_2 distinguish the three initial elements, which will be referred to as c_0, c_1, c_2. To define this structure we write a formula

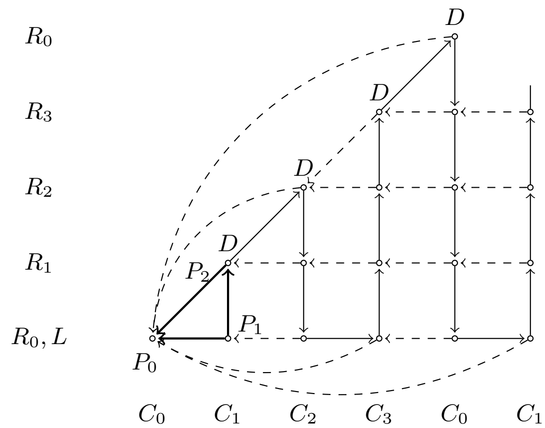

Figure 3. A fragment of the octant grid used in the proof of Thm. 21. Thick lines denote connections among the three initial nodes. Broken lines are those enforced by (iii) and (iv). Solid lines are enforced by (ii).

$$\Phi = \bigwedge_{i=0,1,2} \exists x P_i(x) \wedge \forall xyzt\, \varphi(x, y, z, t)$$

where φ states:
(i) Each P_i is realised precisely once. The elements c_i satisfying P_i are appropriately marked by the remaining unary predicates and connected by E (as in Fig. 3, bold edges).
(ii) There are no deterministic edges:

$$E(x, y) \to \neg\overline{E}(x, y).$$

(iii) Triangles and squares from the picture are appropriately closed (by horizontal broken edges). E.g.:

$$(D(x) \wedge R_3(x) \wedge C_3(x) \wedge D(y) \wedge R_0(y) \wedge C_0(y) \wedge R_3(z) \wedge C_0(z) \wedge E(x, y) \wedge E(y, z)) \to E(z, x),$$
$$(R_2(x) \wedge C_3(x) \wedge R_3(y) \wedge C_3(y) \wedge R_3(z) \wedge C_0(z) \wedge R_2(t) \wedge C_0(t) \wedge E(x, y) \wedge E(z, y) \wedge E(z, t)) \to E(t, x).$$

(Some formulas of this type take into account also the bottom row marker L.)
(iv) Appropriate elements from the diagonal and from the bottom row are connected to P_0 (bent edges):

$$D(x) \wedge (C_0(x) \vee C_2(x)) \wedge P_0(y) \to E(x, y),$$
$$L(x) \wedge (C_1(x) \vee C_3(x)) \wedge P_0(y) \to E(x, y).$$

(v) Predicates D and L are appropriately propagated (from left to right). E.g.:

$$E(x, y) \wedge R_1(x) \wedge C_1(x) \wedge D(x) \wedge R_2(y) \wedge C_2(y) \to D(y),$$
$$E(y, x) \wedge C_1(x) \wedge L(x) \wedge C_2(y) \to L(y),$$

D is allowed only on the diagonal:

$$E(x, y) \wedge ((C_0(x) \wedge C_0(y)) \vee (C_2(x) \wedge C_2(y)) \to \neg D(y),$$
$$E(y, x) \wedge ((C_1(x) \wedge C_1(y)) \vee (C_3(x) \wedge C_3(y)) \to \neg D(y),$$

and analogously L is allowed only in the bottom row.
(vi) E-connections are only allowed between elements of specific 1-types, as in the picture (in particular only elements from D and L may send edges to c_0):

$$E(x,y) \to (R_2(x) \wedge C_0(x) \wedge R_1(y) \wedge C_0(y)) \vee \\ (R_3(x) \wedge C_3(x) \wedge D(x) \wedge R_2(y) \wedge C_2(y) \wedge D(y)) \vee \\ (C_3(x) \wedge L(x) \wedge P_0(y)) \vee (...)).$$

(vii) For each combination of i, j, each element other than c_0 receives an edge from at most one element satisfying R_i, C_j. Taking formula (vi) into account this implies that each element (except c_0) can have at most two incoming edges.
(viii) For each combination of i, j, each element sends an edge to at most one element satisfying R_i, C_j.

This finishes the description of Φ. Consider now an arbitrary model of Φ. By (i) it contains the initial bold triangle. The edge from c_2 to c_0 is deterministic in this triangle, thus by (ii) we must add an edge from c_2 to another element, a, which by (vi) and (viii) satisfies R_2, C_2 and by (v) it satisfies D. By (iv) it must send an edge to c_0. To satisfy (ii) and make this edge nondeterministic there must be another edge to an element b satisfying, again by (vi) and (viii), R_1, C_2. By one of *completing triangles* formulas from (iii) there must be an edge from b to c_2. A further fresh element must be added to make this edge nondeterministic. This way an infinite path (solid edges) through the grid is formed. At each point, we indeed must add a fresh element, since an attempt of reusing an earlier one of appropriate 1-type would violate (vii), as all previously introduced elements have at this moment two incoming edges (except c_0 but it can be used only by elements in L or D from the appropriate columns). Thus every model contains an infinite octant grid of shape depicted in Fig. 3. The remaining details of the encoding of an instance of the octant tiling problem are routine.

To deal with the finite satisfiability problem a modification is needed. We introduce a predicate P_3 which is intended to be realised precisely once and to mark the rightmost, bottom element, and we allow this element to be deterministic, i.e., to send only a single edge. Any finite model must then contain an element satisfying P_3, and the path starting from c_0 must eventually end in this element. Thus a model must contain a finite portion of our octant, which allows to encode finite tilings (or Turing machine finite computations of potentially unbounded length). ☐

To our surprise reducing the number of variables in the above result to three is problematic. Actually, we suspect that $\exists^* \wedge \forall^3 +$ DTC(E), without additional binary symbols is decidable. However, for this moment we leave this problem open.

References

[1] M. Ben-Ari, J. Y. Halpern, and A. Pnueli. Deterministic propositional dynamic logic: Finite models, complexity, and completeness. *Journal of Computer and System Sciences*, 25(3):402 – 417, 1982.

[2] S. Benaim, M. Benedikt, W. Charatonik, E. Kieroński, R. Lenhardt, F. Mazowiecki, and J. Worrell. Complexity of two-variable logic on finite trees. In *ICALP (2)*, pages 74–88, 2013.

[3] S. Benaim, M. Benedikt, R. Lenhardt, and J. Worrell. Controlling the depth, size, and number of subtrees for two-variable logic on trees. *CoRR*, abs/1304.6925, 2013.

[4] R. Brochenin, S. Demri, and É. Lozes. On the almighty wand. *Information and Computation*, 211:106–137, 2012.

[5] W. Charatonik and P. Witkowski. Two-variable logic with counting and trees. In *LICS*, pages 73–82, 2013.

[6] W. Charatonik, E. Kieronski, and F. Mazowiecki. Satisfiability of the two-variable fragment of first-order logic over trees. *CoRR*, abs/1304.7204, 2013.

[7] S. Demri and M. Deters. Two-variable separation logic and its inner circle. *Unpublished*, 2014.

[8] E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[9] E. Grädel, M. Otto, and E. Rosen. Two-variable logic with counting is decidable. In *LICS*, pages 306–317, 1997.

[10] E. Grädel, M. Otto, and E. Rosen. Undecidability results on two-variable logics. *Archiv für Mathematiche Logik und Grundlagenforschung*, 38(4-5):313–354, 1999.

[11] N. Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16:760–778, 1987.

[12] N. Immerman, A. Rabinovich, T. Reps, S. Sagiv, and G. Yorsh. The boundary between decidability and undecidability for transitive-closure logics. In *CSL*, volume 3210 of *LNCS*, pages 160–174, 2004.

[13] Y. Kazakov. *Saturation-based decision procedures for extensions of the guarded fragment*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2006.

[14] E. Kieroński. EXPSPACE-complete variant of guarded fragment with transitivity. In *STACS*, volume 2285 of *LNCS*, pages 608–619, 2002.

[15] E. Kieroński. Results on the guarded fragment with equivalence or transitive relations. In *CSL*, volume 3634 of *LNCS*, pages 309–324, 2005.

[16] E. Kieroński. Decidability issues for two-variable logics with several linear orders. In *CSL*, volume 12 of *LIPIcs*, pages 337–351, 2011.

[17] E. Kieronski and J. Michaliszyn. Two-variable universal logic with transitive closure. In *CSL*, volume 16 of *LIPIcs*, pages 396–410, 2012.

[18] E. Kieroński and M. Otto. Small substructures and decidability issues for first-order logic with two variables. *Journal of Symbolic Logic*, 77: 729–765, 2012.

[19] E. Kieroński and L. Tendera. On finite satisfiability of two-variable first-order logic with equivalence relations. In *LICS*, pages 123–132, 2009.

[20] E. Kieroński, J. Michaliszyn, I. Pratt-Hartmann, and L. Tendera. Two-variable first-order logic with equivalence closure. In *LICS*, pages 431–440, 2012.

[21] H. R. Lewis. Complexity results for classes of quantificational formulas. *Journal of Computer and System Sciences*, 21(3):317 – 353, 1980.

[22] A. Manuel. Two variables and two successors. In *MFCS*, volume 6281 of *LNCS*, pages 513–524, 2010.

[23] A. Manuel. *Counter automata and classical logics for data words*. PhD thesis, Homi Bhabha National Institute, India, 2012.

[24] M. Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

[25] M. Otto. Two-variable first-order logic over ordered domains. *Journal of Symbolic Logic*, 66:685–702, 2001.

[26] L. Pacholski, W. Szwast, and L. Tendera. Complexity of two-variable logic with counting. In *LICS*, pages 318–327, 1997.

[27] I. Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395, 2005.

[28] J. C. Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS*, pages 55–74, 2002.

[29] T. Schwentick and T. Zeume. Two-variable logic with two order relations. *Logical Methods in Computer Science*, 8(1), 2012.

[30] D. Scott. A decision method for validity of sentences in two variables. *Journal Symbolic Logic*, 27:477, 1962.

[31] W. Szwast and L. Tendera. FO^2 with one transitive relation is decidable. In *STACS*, volume 20 of *LIPIcs*, pages 317–328, 2013.

.1 Proof of Thm. 19

Proof. In [22] (see also [23], p. 67) it is shown by a reduction from the Post Correspondence Problem that FO^2 with unary relations and two linear orders and their induced successor relations is undecidable. We can naturally translate this proof to our scenario, with E_1, E_2 corresponding to the successor relations, and their deterministic transitive closures to the order relations. Let us review some details. The construction strictly follows the lines of the proof from [23], we give it here just to show that all the required properties can indeed by expressed by using a $\exists^* \wedge \forall^2 + \mathrm{DTC}(E_1) + \mathrm{DTC}(E_2)$ formula.

Let $\Sigma = \{l_1, \dots, l_s\}$ be a finite alphabet and let $I = \{(u_i, v_i) : 1 \le i \le n, u_i, v_i \in \Sigma^+\}$ be an instance of the Post Correspondence Problem. For each $l \in \Sigma$ we introduce a unary symbol L_l, for each position j, $1 \le j \le |u_i|$ of each u_i we introduce a unary symbol U_i^j, and analogously, for each position j, $1 \le j \le |v_i|$ of each v_i we introduce a unary symbol V_i^j. We use also unary predicates U, V, P_0, P_1.

We construct a formula Φ such that every element in its model lies on a deterministic E_1-path and a deterministic E_2-path from the element c_0 satisfying P_0 to the element c_1 satisfying P_1. Every element satisfies precisely one of L_l predicates, and thus encodes one of the elements from Σ. Every element is also marked by one of the predicates U, V. Each element marked by U satisfies precisely one of U_i^j-s. Similarly each element marked by V satisfies precisely one of V_i^j-s. Thus, every element says to which letter of one of the words u_i or v_i it corresponds.

The letters from Σ, encoded in elements of a model read along the E_1 path from P_0 to P_1 form a word from the language $(u_1 v_1 + \dots + u_n v_n)^+$, while read along the E_2-path form a word from the language $(l_1 l_1 + \dots l_s l_s)^+$. Moreover, when read along the E_2-path the predicates of type U and V alternate. Finally, if elements a, b are both marked with predicates of type U or both are marked with predicates of type V, and a is earlier than b on the E_1-path then a is also earlier than b on the E_2-path. It is not difficult to see that the above properties guarantee that a model of Φ exists iff I has a solution.

Let us see how to write Φ in details. It is of the form

$$\Phi = \exists x P_0(x) \wedge \exists x P_1(x) \wedge \forall x \forall y \varphi(x, y),$$

where φ enforces the desired properties.

P_0 and P_1 are realized precisely once their realizations have, respectively, no predecessors and no successors:

$$\bigwedge_{i=0,1} (P_i(x) \wedge P_i(y) \rightarrow x = y) \tag{1}$$

$$E_1(x, y) \vee E_2(x, y) \rightarrow \neg P_0(y) \wedge \neg P_1(x) \tag{2}$$

Every element lies on the appropriate deterministic paths:

$$P_0(y) \rightarrow x = y \vee \bigwedge_{i=1,2} \overline{E_i}(y, x) \tag{3}$$

$$P_1(y) \rightarrow x = y \vee \bigwedge_{i=1,2} \overline{E_i}(x, y) \tag{4}$$

Elements are marked with appropriate unary symbols:

$$\dot{\bigvee}_{l \in \Sigma} L_l(x) \tag{5}$$

$$U(x) \dot{\vee} V(x) \tag{6}$$

$$U(x) \rightarrow \dot{\bigvee} U_i^j(x) \tag{7}$$

$$V(x) \rightarrow \dot{\bigvee} V_i^j(x) \tag{8}$$

For all i, j, if the j-th letter of U_i is l and the j-th letter of V_j^i is l' then we write

$$(U_i^j(x) \rightarrow L_l(x)) \wedge (V_i^j(x) \rightarrow L_{l'}(x)) \tag{9}$$

The element satisfying P_0 encodes the first letter of one of the words u_i, the element satisfying P_1 encodes the last letter of one of the words v_i:

$$P_0(x) \rightarrow \bigvee_i U_i^1 \tag{10}$$

$$P_1(x) \rightarrow \bigvee_i V_i^{|v_i|} \tag{11}$$

U and V alternate along the E_2-path:

$$E_2(x, y) \rightarrow (U(x) \leftrightarrow V(y)) \tag{12}$$

The word read along the E_1-path is as required:

$$E_1(x, y) \rightarrow \psi(x, y), \tag{13}$$

where $\psi(x, y)$ says, using predicates U_i^j and V_i^j, that:

- if x encodes a non-final letter of some u_i then y encodes the next letter of u_i,
- if x encodes the last letter of some u_i then y encodes the first letter of v_i,
- if x encodes a non-final letter of some v_i then y encodes the next letter of v_i,
- if x encodes the last letter of some v_i then y encodes the first letter of some u_j.

The word read along the E_2-path is as required:

$$E_2(x, y) \wedge U(x) \rightarrow \bigwedge_{l \in \Sigma} (L_l(x) \leftrightarrow L_l(y)) \tag{14}$$

The order of U elements (V-elements) is identical on both paths:

$$(U(x) \wedge U(y)) \vee (V(x) \wedge V(y)) \wedge \overline{E_1}(x, y) \rightarrow \overline{E_2}(x, y) \tag{15}$$

□