

Complexity of Two-variable Logic on Finite Trees

SAGUY BENAÏM and MICHAEL BENEDIKT, University of Oxford
 WITOLD CHARATONIK and EMANUEL KIEROŃSKI, University of Wrocław
 RASTISLAV LENHARDT, University of Oxford
 FILIP MAZOWIECKI, University of Warsaw
 JAMES WORRELL, University of Oxford

Verification of properties expressed in the two-variable fragment of first-order logic FO^2 has been investigated in a number of contexts. The satisfiability problem for FO^2 over arbitrary structures is known to be NEXPTIME-complete, with satisfiable formulas having exponential-sized models. Over words, where FO^2 is known to have the same expressiveness as unary temporal logic, satisfiability is again NEXPTIME-complete. Over finite labelled ordered trees FO^2 has the same expressiveness as navigational XPath, a popular query language for XML documents. Prior work on XPath and FO^2 gives a 2EXPTIME bound for satisfiability of FO^2 over trees. This work contains a comprehensive analysis of the complexity of FO^2 on trees, and on the size and depth of models. We show that different techniques are required depending on the vocabulary used, whether the trees are ranked or unranked, and the encoding of labels on trees. We also look at a natural restriction of FO^2 , its guarded version, GF^2 . Our results depend on an analysis of types in models of FO^2 formulas, including techniques for controlling the number of distinct subtrees, the depth, and the size of a witness to satisfiability for FO^2 sentences over finite trees.

Categories and Subject Descriptors: F.4.1 [Mathematical Logic]: Finite Model Theory

General Terms: Theory

Additional Key Words and Phrases: trees, logic

1. INTRODUCTION

The complexity of verifying properties over a class of structures depends on both the specification language and the type of structure. Stockmeyer [Stockmeyer 1974] showed a non-elementary lower bound for satisfiability of full first-order logic (FO) even when applied to very restricted structures, such as words. The two-variable fragment, FO^2 , is known to have better complexity. Grädel *et al.* [Grädel *et al.* 1997] showed that satisfiability over arbitrary relational vocabularies is NEXPTIME-complete, with satisfiable sentences having exponential-sized models. Over words, Etessami *et al.* showed that satisfiability remains NEXPTIME-complete, with satisfiable formulas again having exponential-sized models [Etessami *et al.* 2002]. Moreover the complexity bounds over words extend to bounds on a host of related verification problems [Benedikt *et al.* 2012].

The NEXPTIME-completeness of FO^2 over both general structures and words raises the question of the impact of structural restrictions on the complexity of FO^2 . Surprisingly the complexity of satisfiability for FO^2 over trees has not been investigated in detail. By trees we will always mean by default *ordered node-labeled unranked finite trees*. By ordered, we mean that our trees are equipped with a sibling ordering that is a linear order on the children of any node, and by node-labelled we mean that each node is additionally assigned a subset of a collection of unary predicates. By unranked, we mean that there is no bound on the number of children.

Marx and De Rijke [Marx and de Rijke 2004] showed that FO^2 over such trees corresponds precisely to the navigational core of the XML query language XPath. From the work of Marx [Marx 2004] it follows that the satisfiability problem for XPath is complete for EXPTIME. Given that the translation from FO^2 to XPath in [Marx and de Rijke 2004] is exponential, this gives a 2EXPTIME bound on satisfiability for FO^2 over trees. In this work we will consider the satisfiability problem for FO^2 over finite

trees, and the corresponding question of the size and depth needed for witness models. In particular, we will consider:

- satisfiability in the presence of all navigational predicates: predicates for the parent/child relation, its transitive closure (the descendant relation), the previous/next sibling relation and its transitive closure;
- satisfiability when the signature is restricted to a subset of the predicates;
- satisfiability over trees where a node can satisfy multiple unary predicates, versus the case of trees in which each node must satisfy exactly one such predicate;
- satisfiability of the full logic, versus the restriction to the case where quantification of variables must be guarded;
- satisfiability over unranked trees, and satisfiability over ranked trees.

We will show that each of these variations affects the complexity of the problem. In the process, we will show that the tree case differs in a number of important ways from that of words. First, satisfiability is EXPSPACE-complete, unlike for general structures or words. Secondly, the basic technique for analyzing FO^2 on words [Etessami et al. 2002]—bounds on the number of quantifier-rank types that occur in a structure—is not useful for getting tight complexity bounds for FO^2 over trees. Instead we will use a combination of methods, including reductions to XPath, Scott-type normal forms, bounds on the number of atomic 1-types, and a quotient construction that is based not only on types, but on a set of distinguished witness nodes. These techniques allow us to distinguish situations where satisfiable FO^2 -formulas have models of (reasonably) small depth, and situations where they have models of small size.

1.1. Related work

As mentioned in the introduction, satisfiability of the modal language XPath has been heavily studied, beginning with the work of Marx [Marx 2004] and Marx and de Rijke [Marx and de Rijke 2004]. A survey of some later results can be found in [Benedikt and Koch 2008]. Given that the translation from two-variable logic to XPath is exponential, this work does not allow us to isolate the complexity of satisfiability for two-variable logic on trees.

The complexity of two-variable logic over ordinary trees is explicitly studied only in [Benedikt and Koch 2008]. Our results here show that the proof of the satisfiability result there (claiming NEXPTIME for full two-variable logic) is incorrect. The proof first argues that satisfiable formulas have satisfying models of exponential depth; we give a very similar argument in this work. [Benedikt and Koch 2008] next argues that satisfiable formulas must have a model in which there are only exponentially many distinct subtrees (and hence a model representable as an exponential sized DAG). The argument for the latter property (the second part of Lemma 5.4 in [Benedikt and Koch 2008]) uses a flawed induction.

Several papers deal with satisfiability problem that overlap with the one we study.

- FO^2 with the deterministic transitive closure of a single binary relation is shown to be EXPSPACE-complete in the later paper [Charatonik et al. 2014]. This logic allows one to express unranked trees (without the sibling relation, however), and in fact some techniques from the current paper are adapted there.
- Figueira [Figueira 2012b] proved decidability of two-variable logic with two successor relations, respectively corresponding to two different linear orders. Figueira’s results were generalized in [Charatonik and Witkowski 2013] where two-variable logic over structures that contain two forests of finite trees, even in the presence of additional binary predicates and counting quantifiers, is proved to be decidable in NEXPTIME. However, these results are not comparable with ours because the logic

in [Charatonik and Witkowski 2013] is restricted to ranked trees, it cannot express the sibling relation and does not allow using the transitive descendant relation.

Finally, several works deal with more general satisfiability problems.

- Two-variable logic over two transitive relations is shown undecidable in [Kieroński 2005]. On the other hand, the two-variable fragment with one transitive relation has been recently shown by Szwaast and Tendera to be decidable [Szwaast and Tendera 2013].
- Two-variable logic on *data trees*, in which nodes are associated with values in an infinite set of data, has been studied beginning with Bojańczyk et al. [Bojańczyk et al. 2009]. There the main result is decidability (in fact, a 3EXPTIME decision procedure) over the signature with data equality, the child relation, and the right sibling relation. Tan [Tan 2014] provides an alternative decision procedure via an automaton model equivalent in expressiveness to the logic of [Bojańczyk et al. 2009]. As the complexity bounds indicate, the existence of an infinite set of labels makes the satisfiability problem much more expensive compared to the setting of a fixed label set.

1.2. Organization

Section 2 gives preliminaries. Sections 3 and 4 constitute a complete analysis of the complexities of FO^2 with all possible navigational signatures over arbitrary unranked finite trees: Section 3 gives precise bounds for the satisfiability in the case of all four navigational predicates, while Section 4 considers the case where the descendant predicate is absent. In Section 5 we investigate the case of models in which each node satisfies precisely one unary predicate (we will say that such models obey Unary Alphabet Restriction, UAR). Section 6 considers restricting the logic to its guarded version in the case of both arbitrary and UAR models. Section 7 considers the case when the trees are ranked. Section 8 gives conclusions.

2. LOGICS AND MODELS

2.1. Trees

In this paper a *tree* is a structure containing:

- an edge relation \downarrow that must be a finite directed acyclic graph, where each element in the domain has at most one predecessor and exactly one single element has no predecessors. We let \downarrow_+ denote the transitive closure of \downarrow . Our definition of \downarrow imply that \downarrow_+ is a partial order with only one minimum element—the root of the tree. From \downarrow we can define the predecessor relation $u \downarrow v$, read ‘ v is a child of u ’ or ‘ u is a parent of v ’. We also read $u \downarrow_+ v$ as ‘ v is a descendant of u ’ or ‘ u is an ancestor of v ’.
- a *sibling* relation \rightarrow whose transitive closure \rightarrow^+ restricts to a linear order on the children of each node. We read $u \rightarrow v$ as ‘ v is the next-sibling of u ’ and ‘ u is the previous-sibling of v ’. We also read $u \rightarrow^+ v$ as ‘ u is a preceding-sibling of v ’ and ‘ v is a following-sibling of u ’.
- a finite family of unary predicates P_1, P_2, \dots on the vertices (for better readability we will sometimes use different names for them).

We further say that a tree satisfies the *unary alphabet restriction* (UAR) if exactly one predicate P_i holds on each vertex.

Given a tree t and vertex v , $\text{SubTree}(t, v)$ denotes the subtree of t rooted at v . A *path* in a tree by default is a path from the root of the tree to one of its leaves.

We also consider satisfiability for FO^2 over *k-ranked trees*, that is, ordered trees as above but where nodes have at most k children. Note that for k -ranked trees it is

natural to consider signatures that include the relation \downarrow_i , connecting a node to its i^{th} child for each $i \leq k$, either in place of or in addition to the predicates above. However we will not consider a separate signature for ranked trees, since it is easy to derive tight bounds for ranked trees for such signatures based on the techniques introduced here.

2.2. Logics

We consider first-order logic with equality over signatures of the form $\tau_{un} \cup \tau_{bin}$, where τ_{un} consists of unary predicates and $\tau_{bin} \subseteq \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$ is a subset of the *navigational relations*. Over such signatures we consider the two-variable subset of first-order logic FO^2 and its *guarded fragment* GF^2 (cf. [Andréka et al. 1998]). The former comprises those formulas built using only variables x and y , while in the latter quantifiers are additionally relativised to atoms, i.e., universally quantified formulas have the form $\forall y (\alpha(x, y) \Rightarrow \varphi(x, y))$ and existentially quantified formulas have the form $\exists y (\alpha(x, y) \wedge \varphi(x, y))$, where $\alpha(x, y)$ is atomic. Atom $\alpha(x, y)$ is called a *guard*. Equalities $x = x$ or $x = y$ are also allowed as guards.

We write $FO^2[\tau_{bin}]$ or $GF^2[\tau_{bin}]$ to indicate that the only binary symbols that are allowed in formulas are those from τ_{bin} and that formulas are interpreted over trees. Although we allow equality in our upper bounds, it will not play any role in the lower bounds. When interpreting formulas of the logic in a tree, we will always assume the navigational relations are given their natural interpretation: \downarrow as the child relation, \downarrow_+ the descendant relation, and so forth.

We consider the following problem: Given an FO^2 sentence φ , determine if there is some tree (resp. k -ranked, UAR tree) that satisfies it. In the case of k -ranked trees, we consider the combined complexity in the size of the binary representation of k and the formula.

2.3. Navigational XPath

Some of our results will go through XPath, a common language used for querying XML documents viewed as trees. The navigational core of XPath is a modal language, analogous to unary temporal logic on trees, denoted NavXP. NavXP is built on binary modalities, referred to as *axis relations*. We will focus on the following axes: self, parent, child, descendant, descendant-or-self, ancestor-or-self, next-sibling, following-sibling, preceding-sibling, previous-sibling. In a tree t , we associate each axis a with a set R_a^t of pairs of nodes. R_{child}^t denotes the set of pairs of nodes (x, y) in t where y is a child of x , and similarly for the other axes (see [Marx 2004]).

NavXP consists of path expressions, which denote binary relations between nodes in a tree, and filters, denoting unary relations. Below we give the syntax (from [Benedikt and Koch 2008]), using p to range over path expressions and q over filters. *AtomicNTest* ranges over symbols for atomic node tests. These can be either predicates, for general trees (e.g. node has predicate P_i) or label tests for UAR trees (node has label L).

$$\begin{aligned} p &::= \text{step} \mid p/p \mid p \cup p \\ \text{step} &::= \text{axis} \mid \text{step}[q] \\ q &::= p \mid \text{AtomicNTest} \mid q \wedge q \mid q \vee q \mid \neg q \end{aligned}$$

where axis relations are given above.

The semantics of NavXP path expressions relative to a tree t is given by:

- (1) $\llbracket \text{axis} \rrbracket_t = R_{\text{axis}}^t$
- (2) $\llbracket \text{step}[q] \rrbracket_t = \{(n, n') \in \llbracket \text{step} \rrbracket_t : n' \in \llbracket q \rrbracket_t\}$

- (3) $\llbracket p_1/p_2 \rrbracket_t = \{(n, n') : \exists w (n, w) \in \llbracket p_1 \rrbracket_t \wedge (w, n') \in \llbracket p_2 \rrbracket_t\}$
- (4) $\llbracket p_1 \cup p_2 \rrbracket_t = \llbracket p_1 \rrbracket_t \cup \llbracket p_2 \rrbracket_t$.

For filters we have:

- (1) $\llbracket \text{AtomicNTest} \rrbracket_t = \{n : n \text{ satisfies test } \text{AtomicNTest}\}$
- (2) $\llbracket p \rrbracket_t = \{n : \exists n' (n, n') \in \llbracket p \rrbracket_t\}$
- (3) $\llbracket q_1 \wedge q_2 \rrbracket_t = \llbracket q_1 \rrbracket_t \cap \llbracket q_2 \rrbracket_t$
- (4) $\llbracket \neg q \rrbracket_t = \{n : n \notin \llbracket q \rrbracket_t\}$.

A NavXP filter is said to hold in a tree t if it holds of the root under the above semantics.

As mentioned earlier, expressive equivalence of FO^2 and NavXP on trees, extending the translation to Unary Temporal Logic in the word case, is known:

PROPOSITION 2.1 ([MARX AND DE RIJKE 2004]). *There is an exponential translation from $\text{FO}^2[\downarrow_+]$ to NavXP with only the descendant and ancestor axes and from $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ to NavXP with all axes.*

In [Marx and de Rijke 2004] the statement holds when the target is a NavXP expression with node tests for the unary predicates in the FO^2 formula. In this paper we shall work with NavXP over trees with unique labels, as is standard for NavXP. Let φ be a $\text{FO}^2[\downarrow_+]$ or $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula and φ' its translation to NavXP. To overcome this problem we change the alphabet in the signature for φ' to the powerset of unary predicates τ_{un} . The intended meaning is that a new label is a set of unary predicates satisfied by the node. Now, we only need to adjust φ' to the new signature. It suffices to replace each test predicate test for U_i by a disjunction of exponential size of the form $\bigvee_{S: U_i \in S} S$. The formula obtained this way is still exponential in $|\varphi|$.

It is worth mentioning that the exponential blow-up in this translation is unavoidable, since it is necessary even on finite linear structures [Etessami et al. 2002].

From the fact that NavXP (over label tests) has an exponential time satisfiability problem [Marx 2004] and the above proposition, we get the following (implicit in [Marx and de Rijke 2004]):

COROLLARY 2.2. *The satisfiability problem for $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ is in 2EXPTIME.*

We will see in the next section that this bound can be improved.

2.4. Summary of our results

Having introduced all the required notions we are ready to present a summary of all the complexity results we obtain, which will help the reader to navigate through the paper.

Note that the UAR restriction can easily be described within FO^2 in any of our signatures — indeed, without any navigational predicates at all. Thus *all of the upper bounds for satisfiability over general trees carry over to UAR trees, and, analogously, all of the lower bounds for satisfiability over UAR trees carry over to general trees.*

Obviously, for any navigational signature τ_{bin} , $\text{GF}^2[\tau_{bin}]$ is a fragment of $\text{FO}^2[\tau_{bin}]$ and for any signatures $\tau_{bin} \subseteq \tau'_{bin}$, $\text{FO}^2[\tau_{bin}]$ is a fragment of $\text{FO}^2[\tau'_{bin}]$ and $\text{GF}^2[\tau_{bin}]$ is a fragment of $\text{GF}^2[\tau'_{bin}]$.

In Figure 1 we present our results in a form of a diagram in which a path from a variant A to a variant B means that the logic of A is a fragment of the logic of B, or that it reduces to it by expressing the UAR restriction as mentioned above. We use the superscript 'U' to denote that a particular variant assumes the UAR. For each variant which is minimal in one of the four relevant complexity regions of our diagram (PSPACE,

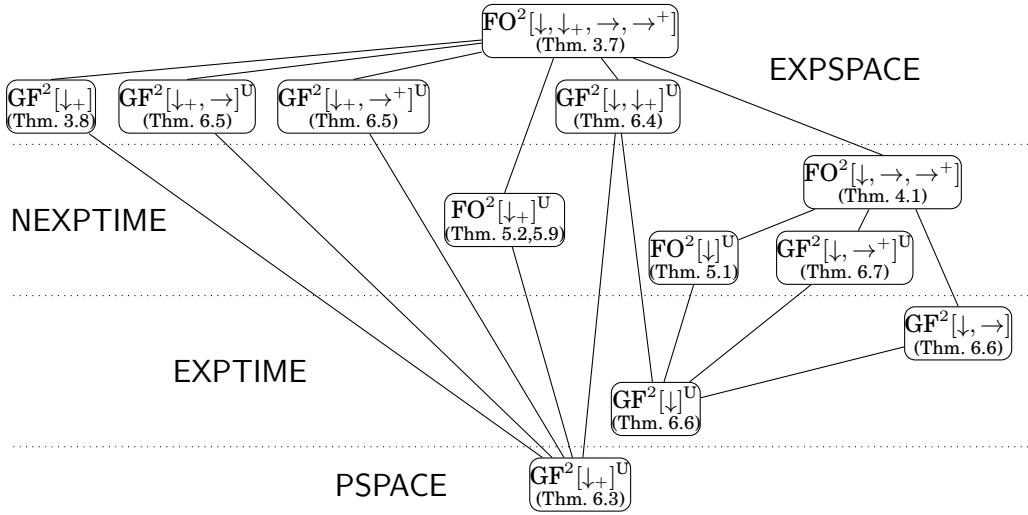


Fig. 1. List of complexities

EXPTIME, NEXPTIME, EXPSPACE) we have an appropriate lower bound and for each variant which is maximal in a region we have an appropriate upper bound.

One can readily verify that our analysis is complete, i.e., that any possible variant of FO^2/GF^2 logic with any admissible navigational signature and with or without the UAR lies between two variants: one for which we have a lower bound and one for which we have the matching upper bound.

References in the diagram lead to the appropriate theorems for the unranked case. The same complexity bounds apply also in the k -ranked case, under the assumption that k is coded in binary (but with new arguments required in some cases, see Section 7).

3. SATISFIABILITY FOR FULL FO^2 ON TREES

In the analysis of satisfiability of FO^2 for words of Etessami, Vardi, and Wilke [Etessami et al. 2002], a NEXPTIME bound is achieved by showing that any sentence with a finite model has a model of at most exponential size. The small-model property follows, roughly speaking, from the fact that *any* model realises only exponentially many “quantifier-rank types” – maximal consistent sets of formulas of a given quantifier rank – and the fact that two nodes with the same quantifier-rank type can be identified.

In the case of trees, this approach breaks down in several places. It is easy to see that one cannot always obtain an exponential-sized model, since a sentence can enforce binary branching and exponential depth of every branch. Because there are doubly exponentially many non-isomorphic small-depth subtrees, there can be doubly exponentially many quantifier-rank types realised even along a single path in a tree: so quantifier-rank types can not be used to show an exponential depth bound.

Instead, we will use some standard tools for FO^2 including Scott-style normal forms. An (atomic) 1-type over τ_{un} is a subset of τ_{un} . A τ_{bin} -order type is a maximal consistent set of atoms built out of navigational predicates and two distinct variables x and y , containing additionally $x \neq y$. One can view a τ_{bin} -order type as a specification of the relative position of two distinct elements in a tree. Let $\Theta_{\tau_{bin}}$ be the set of τ_{bin} -order types. We often identify a type with the conjunction of its members. Moreover, to simplify our notation, we omit negated atoms that are not needed to specify the type — negated

atoms that are implied by some positive atom in the type. The τ_{bin} -order type containing only negated atoms is abbreviated by $x \not\sim y$.

For $\tau_{bin} = \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$ there are nine members of $\Theta_{\tau_{bin}}: x \downarrow y, y \downarrow x, x \downarrow_+ y \wedge \neg(x \downarrow y), y \downarrow_+ x \wedge \neg(y \downarrow x), x \rightarrow y, y \rightarrow x, x \rightarrow^+ y \wedge \neg(x \rightarrow y), y \rightarrow^+ x \wedge \neg(y \rightarrow x), x \not\sim y$.

Normal form. Let us recall classical Scott-normal form for two-variable logic. We say that an FO^2 formula φ is in *Scott-normal form* if $\varphi = \forall xy \chi(x, y) \wedge \bigwedge_{i \in I} \forall x \exists y (x \neq y \wedge \psi_i(x, y))$, for some index set I , and quantifier-free formulas χ and ψ_i . The following lemma was proven in [Scott 1962], with the treatment here following [Grädel et al. 1997]. We sketch its proof as we will refer to it in Section 7.

LEMMA 3.1. *Let φ be an FO^2 sentence over a signature τ . There exists a polynomially computable FO^2 sentence φ' in Scott normal form over signature τ' extending τ by some fresh unary predicates, such that for structures having at least two elements*

- *any model $\mathfrak{A} \models \varphi$ can be expanded to a model of φ by interpreting the additional unary symbols, and*
- *any model $\mathfrak{A}' \models \varphi'$ restricted to τ is a model of φ .*

PROOF. Write φ using only existential quantifiers (eliminating the universal ones by De Morgan's laws). Let ψ be one of the most deeply nested subformulas starting with a quantifier. Assume that $\psi(x) = \exists y \psi'(x, y)$ (if x is the quantified variable then we proceed analogously). The variable x need not be used in ψ' , so ψ may have no free variables. Replace $\psi(x)$ by $p_\psi(x)$, where p_ψ is a fresh unary predicate, and add to φ normal form conjuncts enforcing over structures with at least two elements the equivalence $\forall x (p_\psi(x) \Leftrightarrow \exists y \psi'(x, y))$, namely $\forall x \exists y (p_\psi(x) \Rightarrow (x \neq y \wedge (\psi'(x, y) \vee \psi'(x, x))))$ and $\forall xy (\neg \psi'(x, y) \vee p_\psi(x))$. Note that if ψ has no free variables then p_ψ acts like a boolean formula. Repeat this process until the original formula is transformed into a formula with no quantifiers, and close it universally. Finally, reorganize all the $\forall \forall$ conjuncts merging them into a single conjunct, as required. \square

Now we refine Scott-normal form to make it more convenient to our purposes. We say that an $\text{FO}^2[\tau_{bin}]$ formula φ is in τ_{bin} -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 $a \in \tau_{un}$, $\psi_i(x, y)$ is a boolean combination of unary atomic formulas, and $\eta_i(x, y)$ is a τ_{bin} -order type. Note that in χ the navigational predicates may be used, e.g., we can enforce that a tree contains at most one a on each path: $\forall xy (x \downarrow_+ y \Rightarrow \neg a(x) \vee \neg a(y))$; χ may also use equalities or inequalities. The following lemma justifies the usefulness of τ_{bin} -normal forms.

LEMMA 3.2. *Let φ be an $\text{FO}^2[\tau_{bin}]$ formula over a signature $\tau = \tau_{un} \cup \tau_{bin}$. There exists a polynomially computable $\text{FO}[\tau_{bin}]$ τ_{bin} -normal form formula φ' over signature $\tau' = \tau'_{un} \cup \tau_{bin}$ consisting of τ and some additional unary symbols, such that for structures having at least two elements:*

- *any model $t \models \varphi$ can be expanded to a model of φ by interpreting the additional unary symbols, and*
- *any model $t' \models \varphi'$ restricted to τ is a model of φ .*

PROOF. Let φ'' be a Scott-normal form formula guaranteed by Lemma 3.1, $\varphi'' = \forall xy \chi(x, y) \wedge \bigwedge_{i \in I} \forall x \exists y (x \neq y \wedge \psi_i(x, y))$. Having computed φ'' , take fresh unary predicates $p_{i, \eta}$, for all $i \in I$ and all $\eta \in \Theta_{\tau_{bin}}$. Intuitively, $p_{i, \eta}$ asserts that there is witness y for the

conjunct with index i which has relation η to x . Let φ''' be the result of replacing each conjunct $\forall x \exists y (x \neq y \wedge \psi_i(x, y))$ in φ'' by

$$\bigwedge_{\eta(x, y) \in \Theta_{\tau_{bin}}} \forall x (p_{i, \eta}(x) \Rightarrow \exists y (\eta(x, y) \wedge \psi_i(x, y))),$$

adding the conjunct ensuring the reverse implication $\forall xy (\neg \eta(x, y) \vee \neg \psi_i(x, y) \vee p_{i, \eta})$, and the conjunct $\forall x (\bigvee_{\eta \in \Theta_{\tau_{bin}}} p_{i, \eta}(x))$, which guarantees that every node will have at least one witness. To obtain φ' it remains to eliminate navigational predicates from the subformulas $\psi_i(x, y)$. For each conjunct $\forall x (p_{i, \eta}(x) \Rightarrow \exists y (\eta(x, y) \wedge \psi_i(x, y)))$ replace in $\psi_i(x, y)$ every binary atom by \top or \perp , in accordance with $\eta(x, y)$.

It is readily verified that φ' has the properties required by the lemma. \square

In all of the arguments where these normal forms are applied, one-element structures can be dealt with using ad hoc methods. We will thus often drop the qualification that the equivalence holds only for structures with at least two elements.

3.1. Depth and branching bounds for $\text{FO}^2[\downarrow, \downarrow+, \rightarrow, \rightarrow^+]$

By $|\varphi|$ we denote the size of the formula φ , which is the number of all symbols used in φ . We first show that both the depth of models and the outdegree of their nodes can be bounded exponentially in $|\varphi|$. This observation will be then extensively used in various parts of this paper. Please note, that since it is obtained for our richest navigational signature, it is also true for any its subset. It works also in the case of the guarded restriction of the language, GF^2 , and in the case of UAR models.

THEOREM 3.3. *Every satisfiable $\text{FO}^2[\downarrow, \downarrow+, \rightarrow, \rightarrow^+]$ sentence φ has a model whose depth is bounded by $2^{\text{poly}(|\varphi|)}$. The outdegree of nodes can also be bounded by $2^{\text{poly}(|\varphi|)}$.*

We split the proof of Theorem 3.3 into two parts. In the first part we show the depth bound and in the second part we prove the branching bound. By Lemma 3.2 in both parts we can assume that φ is in normal form.

Depth bound in Theorem 3.3: We add some additional information to nodes, which will allow us to collapse the tree. Let $\text{DescTypes}(n)$ be the set of 1-types of descendants of n , $\text{AncTypes}(n)$ the set of 1-types of ancestors of n , and let $\text{IncompTypes}(n)$ be the set of 1-types of nodes that are neither descendants nor ancestors of n . Equivalently, the sets $\text{DescTypes}(n)$, $\text{AncTypes}(n)$ and $\text{IncompTypes}(n)$ are sets of nodes n' such that $n \rightarrow^+ n'$, $n' \rightarrow^+ n$ or $n \not\sim n'$ holds, respectively. We write $n_0 \equiv_{\text{Vert}} n_1$ if the nodes n_1 and n_0 have the same 1-type, and $\text{DescTypes}(n_1) = \text{DescTypes}(n_0)$, $\text{AncTypes}(n_1) = \text{AncTypes}(n_0)$, $\text{IncompTypes}(n_1) = \text{IncompTypes}(n_0)$.

Given a tree t and nodes n_0 and n_1 in t with n_1 not an ancestor of n_0 , the *overwrite* of n_0 by n_1 in t is the tree $t[n_1 \rightarrow n_0]$ formed by replacing the subtree of n_0 with the subtree of n_1 in t .

The following lemma is crucial for proving the depth bound.

LEMMA 3.4. *Let φ be a normal form formula, $t \models \varphi$ and n_0, n_1 nodes in t that n_1 is a descendant of n_0 and $n_0 \equiv_{\text{Vert}} n_1$. Then $t' = t[n_1 \rightarrow n_0]$ is a model of φ .*

PROOF. Assume $\varphi = \forall xy \chi(x, y) \wedge \bigwedge_{i \in I} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$. We will first argue that for each $i \in I$ the conjunct

$$\forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$$

is satisfied in t' .

Consider first the case where the variable x is bound to n_1 , and assume that $t', n_1 \models \lambda_i$ for some $i \in I$. Since the satisfaction of λ_i depends only on the 1-type of an element, thus also $t, n_1 \models \lambda_i$ and $t, n_0 \models \lambda_i$. It follows that both n_0 and n_1 have appropriate witnesses in t ; call these witnesses w_0, w_1 , respectively. Now, if $\eta_i(x, y)$ is one of the order types enforcing that y is a descendant of x , i.e., $x \downarrow y$ or $x \downarrow_+ y \wedge \neg x \downarrow y$ then the witness w_1 is located in the subtree of n_1 and is therefore preserved in t' . For any other η_i , the witness w_0 is connected by the navigational relations to n_1 in t' exactly as it is connected to n_0 in t , and it follows that w_0 is an appropriate witness for n_1 in t' .

Now consider the case where x is bound to an element m that is an ancestor of n_1 in t' (and thus an ancestor of both n_0 and n_1 in t). Let the witness for m in t be w . Consider the subcase where $i \in I$ is such that $\eta_i(x, y) = x \downarrow_+ y \wedge \neg x \downarrow y$. If the node w is not in $\text{SubTree}(t, n_0)$ then it is retained in t' and is still a witness for m there. If w is in $\text{SubTree}(t, n_1)$ and $w \neq n_1$ it is also retained in t' and is connected to m exactly as it is connected to it in t , hence remaining a valid witness. If $w = n_1$ then it is retained in t' , but the order type of the pair m, n_1 may change from $x \downarrow_+ y \wedge \neg x \downarrow y$ to $x \downarrow y$. In this case we take as a witness a descendant of n_1 having the same 1-type as n_1 . Such a descendant exists since $\text{DescTypes}(n_1) = \text{DescTypes}(n_0)$, and the 1-type of n_1 belongs to $\text{DescTypes}(n_0)$. Finally, if w belongs to $\text{SubTree}(t, n_0)$ but not to $\text{SubTree}(t, n_1)$ then it is removed in t' , but an appropriate witness can be found in $\text{SubTree}(t', n_1)$, again due to the argument involving $\text{DescTypes}(n_0)$, or (when $w = n_0$) due to the equality of the 1-types of n_0 and n_1 . In the subcases where $\eta_i(x, y)$ is one of the order types $y \downarrow_+ x, y \downarrow_+ x \wedge \neg y \downarrow x, x \not\downarrow y, x \rightarrow^+ y, x \rightarrow^+ y \wedge \neg x \rightarrow y, y \rightarrow^+ x, y \rightarrow^+ x \wedge \neg y \rightarrow x$, the witness w is retained in t' . And finally, if $\eta_i(x, y) = x \downarrow y$ then w is either retained in t' or, if $w = n_0, n_1$ is a valid witness.

The next case, where x is bound to an element m that is a descendant of n_1 in t' , can be treated in a very similar way. If $\eta_i(x, y)$ is one of the order types $x \downarrow y, y \downarrow x, x \downarrow_+ y \wedge \neg x \downarrow y, x \rightarrow^+ y, x \rightarrow^+ y \wedge \neg x \rightarrow y, y \rightarrow^+ x, y \rightarrow^+ x \wedge \neg y \rightarrow x$ then the witness for m from t is retained in t' . If $\eta_i(x, y) = y \downarrow_+ x \wedge \neg y \downarrow x$ or $\eta_i(x, y) = x \not\downarrow y$ then either an appropriate witness from t is retained in t' or its existence is guaranteed by the equalities $\text{AncTypes}(n_0) = \text{AncTypes}(n_1)$ or $\text{IncompTypes}(n_0) = \text{IncompTypes}(n_1)$, respectively.

In the case where m is a sibling of n_1 in t' (and thus a sibling of n_0 in t), if $\eta_i(x, y)$ is one of the order types $x \downarrow y, y \downarrow x, x \downarrow_+ y \wedge \neg x \downarrow y, y \downarrow_+ x \wedge \neg y \downarrow x$ then the witness from t is retained in t' . If $\eta_i(x, y)$ is one of $x \rightarrow^+ y, x \rightarrow^+ y \wedge \neg x \rightarrow y, y \rightarrow^+ x, y \rightarrow^+ x \wedge \neg y \rightarrow x$ then either the witness from t is retained or n_0 is a witness in t and in this case n_1 is a valid witness in t' . Finally, if $\eta_i(x, y) = x \not\downarrow y$ then a witness in t is retained or the existence of a witness is guaranteed by $\text{DescTypes}(n_0) = \text{DescTypes}(n_1)$.

The last case is where $t' \models n_1 \not\downarrow m$. For almost all subcases a witness from t must be retained in t' . The only exception may be if $\eta_i(x, y) = x \not\downarrow y$; in this subcase either a witness is retained from t , or its existence is guaranteed by $\text{DescTypes}(n_0) = \text{DescTypes}(n_1)$.

Consider now the conjunct $\forall xy \chi(x, y)$ of φ . Towards verifying it in t' , consider any elements m_0, m_1 of t' . Since χ is quantifier-free, it is enough to show that atomic formulas are preserved, and it suffices to look at navigational relations. For almost all m_0 and m_1 , they are connected by navigational relations in t' exactly as they are connected in t , from which we conclude $t', m_0, m_1 \models \chi$. The only exception is when one of them, say m_0 , is n_1 and the other is its parent or sibling. In this case however, m_0 and m_1 are connected in t' exactly as n_0 and m_1 were connected in t , and since the 1-types of n_0 and n_1 are identical it follows that $t', m_0, m_1 \models \chi$. \square

PROOF (OF THE DEPTH BOUND IN THEOREM 3.3). Apply Lemma 3.4 to any model $t \models \varphi$ as long as possible. We claim that the result of this process has exponentially bounded depth. To see this, note that the sets $\text{DescTypes}(n)$ either become smaller or stay the same as n varies down a path, and hence can only change exponentially often.

Similarly the sets $\text{AncTypes}(n)$ and $\text{IncompTypes}(n)$ grow bigger or stay the same, and thus can change only exponentially often. Finally, in intervals along a path where each of these sets are stable, the number of possibilities for the 1-type of a node is exponential. \square

Branching bound in Theorem 3.3: We now explain the variation of the argument for the exponential bound on branching in Theorem 3.3. Note that NavXP queries can already force exponential branching, and thus the result does not follow directly via translation to modal tree logics. In a nutshell, we use the same approach as in the proof of the exponential bound on depth, but shrinking horizontal rather than vertical paths.

Let t be a tree and n a node in t . We denote by $\text{PrecTypes}(n)$ the set of 1-types of the preceding-siblings of n , and by $\text{PrecNephTypes}(n)$ the set of 1-types of the descendants of the preceding-siblings of n . For the following-siblings we similarly define the sets $\text{FolTypes}(n)$, $\text{FolNephTypes}(n)$.

Consider the equivalence relation \equiv_{Hor} that relates two nodes n_0, n_1 if: n_0 and n_1 have the same 1-types; $\text{PrecTypes}(n_0) = \text{PrecTypes}(n_1)$ and $\text{FolTypes}(n_0) = \text{FolTypes}(n_1)$; and $\text{PrecNephTypes}(n_0) = \text{PrecNephTypes}(n_1)$ and $\text{FolNephTypes}(n_0) = \text{FolNephTypes}(n_1)$.

LEMMA 3.5. *Let φ be a normal form formula, $t \models \varphi$. Fix a horizontal path p containing all children of some node. Suppose that there are two nodes n_0, n_1 on the path p , such that $t \models n_0 \rightarrow^+ n_1$ and $n_0 \equiv_{\text{Hor}} n_1$. Let t' be the tree obtained from t by removing the subtrees rooted at n_0 and at all nodes from p strictly between n_0 and n_1 . Then t' is a model of φ .*

PROOF. The proof involves very similar mechanisms to those used in the proof of Lemma 3.4. Assume $\varphi = \forall xy \chi(x, y) \wedge \bigwedge_{i \in I} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$. Let us first argue that the nodes of t' have appropriate witnesses for all the conjuncts of φ indexed by I . As an example, let us consider the node n_1 and any $i \in I$. Assume that $t', n_1 \models \lambda_i$. We proceed similarly to the proof of Lemma 3.4. Since $t, n_1 \models \lambda_i$ and $t, n_0 \models \lambda_i$, it follows that both n_0 and n_1 have appropriate witnesses in t ; call these witnesses w_0, w_1 , respectively. Now, if $\eta_i(x, y)$ is one of the order types: $y \downarrow x, y \downarrow_+ x \wedge \neg y \downarrow x, x \downarrow y, x \downarrow_+ y \wedge \neg x \downarrow y, x \rightarrow y, x \rightarrow^+ y \wedge \neg x \rightarrow y$, then the witness w_1 is retained in t' . If $\eta_i(x, y)$ is either $y \rightarrow x$ or $y \rightarrow^+ x \wedge \neg y \rightarrow^+ x$ then the witness w_0 is connected by the navigational relations in t' to n_1 exactly as it is connected to n_0 in t , and it follows that it is a proper witness for n_1 in t' . Finally, if $\eta_i(x, y) = x \not\sim y$ then if w_1 is retained in t' then it still works as a witness for n_1 ; and if w_1 is one of the deleted nodes, then we take as a witness a descendant of one of the preceding-siblings of n_0 having the same 1-type as w_1 , whose existence is guaranteed by the equality of sets $\text{PrecNephTypes}(n_0) = \text{PrecNephTypes}(n_1)$.

As another example consider an element m being a following-sibling of n_1 in t' (and thus a following-sibling of both n_0 and n_1 in t) and the subcase of such $i \in I$ that $\eta_i(x, y) = y \rightarrow^+ x \wedge \neg y \rightarrow x$. Let the witness for m in t be w . If in t the node w is a preceding-sibling of n_0 , a following-sibling of n_1 or $w = n_1$ then it is retained in t' and is still a witness for m there. Otherwise w must be one of the deleted elements of p . In this case we take as a witness one of preceding-siblings of n_0 having the same 1-type as w , whose existence is guaranteed by the equality $\text{PrecTypes}(n_0) = \text{PrecTypes}(n_1)$.

All the remaining cases can be treated in a very similar way, employing, when necessary, the appropriate equalities between PrecTypes , FolTypes , PrecNephTypes and FolNephTypes .

Consider now the conjunct $\forall xy \chi(x, y)$ of φ , and take any elements m_0, m_1 of t' . In most cases m_0 and m_1 are connected by navigational relations in t' exactly as they are connected in t , thus $t', m_0, m_1 \models \chi$. The only exception is when one of them, say m_0 , is n_1 and the other is its preceding-sibling. In this case however, m_0 and m_1 are connected

in t' exactly as n_0 and m_1 were connected in t , and since the 1-types of n_0 and n_1 are identical it follows that $t', m_0, m_1 \models \chi$. \square

PROOF (OF THE BRANCHING BOUND IN THEOREM 3.3). We apply Lemma 3.5 to any model $t \models \varphi$ as long as possible. We claim that the result of this process has exponentially many nodes on each horizontal path. To see this, note that the sets $\text{PrecTypes}(n)$, $\text{FolTypes}(n)$, $\text{PrecNephTypes}(n)$ and $\text{FolNephTypes}(n)$ behave monotonically as n moves right a horizontal path, and hence can only change exponentially often. In an interval where each of these sets is stable the number of possible 1-types of a node is also bounded exponentially. \square

3.2. Upper Bound for $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$

We are now ready to give an upper bound for the full logic, $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$, which improves on the 2EXPTIME bound one obtains via translation to modal logic. It uses the depth bound from the previous subsection, and, as the second ingredient, the following upper bound on satisfiability of NavXP filters over trees of a given depth.

THEOREM 3.6. *The satisfiability of a NavXP filter φ over trees of depth bounded by b is in PSPACE (in b , $|\varphi|$ and the size of the alphabet).*

The result follows using Lemma 7.5 in [Benedikt et al. 2008], which states that the problem is in PSPACE when trees are restricted by a fixed-depth schema, and thus in particular when the depth is fixed.

THEOREM 3.7. *The satisfiability problem for $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ is in EXPSPACE.*

PROOF. Translate an $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ sentence φ into an NavXP filter φ' over atomic node tests for labels from the powerset alphabet. This can be done with a single exponential blow-up, using Proposition 2.1. Recall that by the comment below Proposition 2.1 we can assume that we work with the powerset alphabet and thus nodes have unique labels. By Theorem 3.3, the depth of a witness structure is bounded by an exponential in $|\varphi|$. The desired EXPSPACE bound follows from Theorem 3.6. \square

Notice that we did not use the branching bound from Theorem 3.3. This bound will be useful in the next sections.

We note that Theorem 3.7 could also be proven directly via translation to ranked tree automata. The states of the automata would be the equivalence classes of the equivalence relation used in the depth bound. The branching bound would be used to argue that we can use a ranked tree automata with exponential rank. However, the automata argument would require a careful analysis, using the exponential depth bound. Otherwise, one can get only a 2EXPTIME upper bound.

3.3. Lower bound

A matching lower bound can be obtained even for a much weaker logic, namely, the restriction of FO^2 to the guarded fragment, with descendant as the only navigational predicate.

THEOREM 3.8. *The satisfiability problem for FO^2 on trees is EXPSPACE-hard, with hardness holding even when formulas are restricted to be in $\text{GF}^2[\downarrow_+]$.*

This is proved by coding the acceptance problem for an alternating exponential time machine. A tree node can be associated with an n -bit address (here we exploit the fact that many propositions may simultaneously hold at the same node), a path corresponds to one thread of the alternating computation, and the tree structure is used to code alternation. The equality and successor relations between the addresses associated to

nodes x and y can be coded in $\text{GF}^2[\downarrow_+]$ using a standard argument—see [Kieroński 2002] for details, where it was shown that a restricted variant of the two-variable guarded fragment with some unary predicates and a single binary predicate that is interpreted as a transitive relation is EXPSPACE-hard. It is not hard to see that the proof presented there works fine (actually, it is even more natural) if we restrict the class of admissible structures to (finite) trees.

4. SATISFIABILITY WITHOUT DESCENDANT

As we have seen, the descendant predicate, \downarrow_+ , alone is sufficient to obtain an EXPSPACE-lower bound. In this section we show that its presence is crucial for this task, by showing that without it the satisfiability problem for FO^2 becomes NEXPTIME-complete.

4.1. Upper bound for $\text{FO}^2[\downarrow, \rightarrow, \rightarrow^+]$

The result is surprising, in that it is easy to write satisfiable $\text{FO}^2[\downarrow]$ sentences φ_n of polynomial size whose smallest tree model is of depth exponential in n , and whose size is doubly exponential. Indeed, such formulas can be obtained by coding a complete binary tree whose nodes are associated with n -bit numbers, increasing the number by 1 as we move from parent to either child.

THEOREM 4.1. *The satisfiability problem for $\text{FO}^2[\downarrow, \rightarrow, \rightarrow^+]$ is in NEXPTIME.*

PROOF. Let $\tau_{bin} = \{\downarrow, \rightarrow, \rightarrow^+\}$ and let us fix an $\text{FO}^2[\tau_{bin}]$ formula φ . By Lemma 3.2 we may assume that φ is τ_{bin} -normal form. Observe that there are seven τ_{bin} -order types: $x \rightarrow y$, $y \rightarrow x$, $x \rightarrow^+ y \wedge \neg x \rightarrow y$, $y \rightarrow^+ x \wedge \neg y \rightarrow x$, $x \downarrow y$, $y \downarrow x$, and $x \not\sim y$. Note also that in this proof the relation $\not\sim$ is now extended to include non-consecutive elements a, b belonging to the same vertical path.

Let us fix a model $t^* \models \varphi$. By Theorem 3.3 we may assume that both the depth of t^* and the outdegree of its nodes are bounded exponentially in φ . We will perform some surgery on the structure, while preserving certain global witnesses.

Recall that for a given node n the nodes related with n by \rightarrow^+ are called following-siblings or preceding-siblings. We define a set of “protected witnesses” W_1 . For each 1-type α realized in t^* , choose one of its realizations w_α . We add every w_α along with all its ancestors to the set W_1 and call them “basic witnesses”. We also add to W_1 all children, following-siblings and preceding-siblings of each basic witness and call them “child witnesses” and “sibling witnesses”. The number of 1-types is bounded exponentially thus by the bounds on the depth and outdegree of t^* the size of the set of “protected witnesses” W_1 is at most exponential.

We define the set of *incomparable global witnesses* W_2 . For every protected witness $w \in W_1$ and any conjunct of φ of the form $\forall x(\lambda(x) \rightarrow \exists y(x \not\sim y \wedge \psi_i(x, y)))$ if $t^*, w \models \lambda(x)$ then choose a witness w' such that $t^*, w, w' \models x \not\sim y \wedge \psi_i$ and add w' to W_2 , along with all its ancestors. The final set of witnesses is defined as $W = W_1 \cup W_2$. It is easy to see that the size of W_2 is bounded exponentially thus the size of W is also bounded exponentially.

Let τ_{un} be the set of unary symbols appearing in φ . We make use of a well-founded linear order $<$ on the set of (isomorphism types of) trees in which nodes have 1-types over τ_{un} , such that for every tree t and its nodes n, n'

- (1) $\text{SubTree}(t, n') < \text{SubTree}(t, n)$ implies n' is not an ancestor of n ;
- (2) for every tree C with a distinguished leaf, and for trees t_1, t_2 with $t_1 < t_2$, we have $C[t_1] < C[t_2]$, where $C[t_i]$ is the tree obtained by replacing the distinguished leaf of C with t_i .

There are many such orderings. Let us define one. Fix any linear order $<$ on the finite set of 1-types over τ_{un} . For a given tree t let $\text{depth}(t)$ denote its depth, and let S_t be

the sequence of trees that comes from t after removing its root, ordered according to the sibling order in t on their roots. Consider two trees t, t' . If $\text{depth}(t) < \text{depth}(t')$ then we define $t \prec t'$. If $\text{depth}(t) = \text{depth}(t')$ then we define the order recursively. If $\text{depth}(t) = \text{depth}(t') = 0$, that is t and t' consist only of their roots, r, r' , with 1-types α, α' respectively, define $t \prec t'$ iff $\alpha < \alpha'$. If $\text{depth}(t) = \text{depth}(t') > 0$ then let i be the smallest position such that the i -th elements of S_t and $S_{t'}$ differ. Let $t_i \in S_t$ and $t'_i \in S_{t'}$ be the different subtrees. By induction the order \prec is already defined on t_i, t'_i . Then $t \prec t'$ iff $t_i \prec t'_i$. If there is no such position i then $t \preceq t'$ iff $S_t \subseteq S_{t'}$. It is straightforward that this is a linear order with the desired properties.

LEMMA 4.2. *If there are two elements n, n' in t^* such that*

- (1) $n, n' \notin W$;
 - (2) n, n' have the same 1-types;
 - (3) $\text{SubTree}(t^*, n') \prec \text{SubTree}(t^*, n)$;
- then the tree $t^\dagger := t^*[n' \rightarrow n]$ is still a model of φ .*

Note that condition (3) implies that n' is not an ancestor of n . Note also that the same condition guarantees that $t^\dagger \prec t^*$.

PROOF (OF LEMMA 4.2). We assume that t^\dagger is built out of the elements of the initial tree t^* with the exception of the replaced subtree which is built out of fresh copies of the elements from $\text{SubTree}(t^*, n')$. Let $f : \text{SubTree}(t^*, n') \rightarrow t^\dagger$ be the function taking an element to its new copy.

Let us see that all elements of t^\dagger have appropriate witnesses. Consider first any conjunct of φ of the form $\forall x(\lambda_i(x) \Rightarrow \exists y(x \not\sim y \wedge \psi_i(x, y)))$ and take an element m such that $t^\dagger, m \models \lambda_i$. Note that due to condition (1) all elements of W are retained in t^\dagger . If $m \in W_1$ then by our construction an appropriate witness w such that $t^\dagger, m, w \models x \not\sim y \wedge \psi_i$ is retained in the set of incomparable global witnesses W_2 . If $m \notin W_1$ then observe that it is related by \sim to all basic witnesses in W_1 . Since the set of basic witnesses contains realizations of all 1-types realized in t^* one of them can be taken as w such that $t^\dagger, m, w \models x \not\sim y \wedge \psi_i(x, y)$.

Consider now a conjunct of φ of the form $\forall x(\lambda_i(x) \Rightarrow \exists y(\eta_i(x, y) \wedge \psi_i(x, y)))$ with η different from $x \not\sim y$. Thus, a desired witness must be a following-sibling, preceding-sibling, a child or the parent of a given element. Let $t^\dagger, m \models \lambda_i(x)$. If m is in the image of f and $m \neq f(n')$ then an appropriate witness is retained in the image of f . If $m = f(n')$ and $\eta = x \downarrow y$ then an appropriate witness is also retained in the image of f ; for any other η a desired witness must be a preceding-sibling, a following-sibling or the parent of m . Note that in t^\dagger the node $m = f(n')$ has the same siblings, the same parent and the same 1-type as n had in t^* . Thus if one of these siblings or the parent was an appropriate witness for n in t^* then it works also as a witness for $m = f(n')$ in t^\dagger , since the 1-type of $f(n')$ is equal to the 1-type of n . If m is not in the image of f and its witness in t^* was different than n then it is retained in t^\dagger ; otherwise m is a preceding-sibling, a following-sibling or the parent of $f(n')$ and, since $f(n')$ has the same 1-type as n it works as an appropriate witness.

Finally, consider the conjunct $\forall xy\chi(x, y)$ of φ . We want to see that for any pair of nodes m_0, m_1 of t^\dagger we have $t^\dagger, m_0, m_1 \models \chi$. Recall that χ may use navigational predicates. For our purposes it is sufficient to find elements $u_0, u_1 \in t^*$ having the same 1-types as m_0, m_1 , respectively, and connected by the navigational relations precisely as m_0, m_1 are connected in t^\dagger . If none of m_0, m_1 belongs to the image of f then we can simply take $u_0 := m_0, u_1 := m_1$. Similarly, if both m_0, m_1 are in the image of f we can take $u_0 := f^{-1}(m_0), u_1 := f^{-1}(m_1)$. It remains to analyse the case when m_0 is in the image of f and m_1 is not. If $m_0 = f(n')$ and m_1 is one of its siblings or the parent, then take $u_0 := n, u_1 := m_1$. Otherwise, it is the case that $t^\dagger, m_0, m_1 \models x \not\sim y$. Take as u_1 an element

from the set of basic witnesses whose 1-type equals the 1-type of m_1 (recall that all 1-types from t^\dagger are represented there). Take $u_0 := f^{-1}(m_0)$. Since $u_0 \notin W_1$ (because u_0 is a descendant of n' , $n' \notin W_1$ and W_1 is closed under ancestors) it follows that $t^*, u_0, u_1 \models x \not\sim y$ as required. This finishes the proof of Lemma 4.2. \square

We now iterate the update procedure from Lemma 4.2 as long as appropriate n, n' exist. Since $t^\dagger \prec t^*$, the process must terminate.

In the resulting tree, if two nodes are not in W and do not have isomorphic subtrees, they must disagree on their 1-types. There are exponentially many 1-types and this gives us an exponential bound on non-isomorphic subtrees with roots not in W . Given that W has size bounded by an exponential in these quantities as well, we see that the resulting tree will have a number of isomorphism types of subtrees that is bounded exponentially.

We can represent this tree as an “sibling-ordered DAG”: a set of nodes N and a function ChildList taking each node to an ordered list of nodes (possibly with repetition), such that the derived directed graph connecting m to n if $n \in \text{ChildList}(m)$ is acyclic. Such a structure represents a unique ordered tree via unfolding. To get to a sibling-ordered DAG representation from a tree, we simply take the set of nodes N to be each isomorphism type of subtree, and let $\text{ChildList}(m)$ be the list of isomorphism types of the children of m .

Thus we have shown that any satisfiable formula has an exponential-size DAG that unfolds into a model of the formula. Given such a DAG, we can check whether an FO^2 formula holds in polynomial time in the size of the DAG. Thus we have a NEXPTIME algorithm for checking satisfiability. \square

4.2. Lower bound

Recall that even on words with only the successor relation, the satisfiability problem for two-variable logic is NEXPTIME-hard [Etessami et al. 2002]. From this it is easy to see that the satisfiability for $\text{FO}^2[\downarrow]$ is NEXPTIME-hard.

THEOREM 4.3. *The satisfiability problem for $\text{FO}^2[\downarrow]$ is NEXPTIME-hard.*

The above lower bound finishes our investigations of the complexity of FO^2 over unranked trees without the UAR restriction. Recapitulating: the satisfiability problem is EXPSPACE-complete if \downarrow_+ is available and NEXPTIME-complete otherwise.

5. UAR RESTRICTION

In this section we consider the satisfiability problem for FO^2 over trees with the additional UAR restriction (as defined in Section 2). Recall from Subsection 2.4 that satisfiability with the UAR restriction polynomially reduces to satisfiability over arbitrary trees, and thus any upper bounds that hold without this restriction are inherited here.

We will start with simple arguments that the satisfiability problem remains NEXPTIME-hard even if only \downarrow or only \downarrow_+ is available. The first of these observations, coupled with the upper bound result in Theorem 4.1, isolates the complexity for signatures not containing \downarrow_+ .

Then we turn our attention to the most interesting case in the context of UAR trees, the case of $\text{FO}^2[\downarrow_+]$. Recall that Section 3 showed EXPSPACE-completeness for satisfiability of $\text{FO}^2[\downarrow_+]$. However the EXPSPACE-hardness argument for \downarrow_+ makes use of multiple predicates holding at a given node, to code the address of a tape cell of an alternating exponential-time Turing Machine. It thus does not apply to

satisfiability over UAR trees. We show that the complexity of satisfiability can be lowered to NEXPTIME.

Later it will turn out that augmenting $\text{FO}^2[\downarrow_+]$ with any of the remaining navigational predicates lifts the complexity again to EXPSPACE even under UAR. Since this can be obtained even for the two-variable guarded fragment, GF^2 , we postpone the proof of this fact until Section 6, which is dedicated to GF^2 .

5.1. Lower bounds

We first note that the lower bound in Theorem 4.3 can be obtained even under the UAR. This follows from the fact that FO^2 over UAR words is NEXPTIME-hard even if only the successor relation is available. (See Thm. 2.2.3 in [Weis 2011].)

THEOREM 5.1. *The satisfiability problem for $\text{FO}^2[\downarrow]$ over UAR trees is NEXPTIME-hard.*

Now we prove that also \downarrow_+ alone suffices for NEXPTIME-hardness.

THEOREM 5.2. *The satisfiability of $\text{FO}^2[\downarrow_+]$ with respect to UAR trees is NEXPTIME-hard.*

PROOF. We give a reduction from the satisfiability problem of FO^2 with only unary predicates, which is known to be NEXPTIME-complete (see e.g., [Etessami et al. 2002]). For a given FO^2 formula φ over a unary signature τ we construct an equisatisfiable $\text{FO}^2[\downarrow_+]$ formula $T(\varphi)$ over the signature $\tau \cup \{\downarrow_+, \text{elem}, b\}$ where elem, b are fresh unary predicates. Without loss of generality we may assume that φ is built from variables x, y , unary predicate symbols, boolean connectives \wedge, \neg and existential quantification.

Now we inductively define the translation $T(\varphi)$.

$$\begin{aligned} T(p(x)) &= \exists y x \downarrow_+ y \wedge p(y) \\ T(\neg \varphi) &= \neg T(\varphi) \\ T(\varphi_1 \wedge \varphi_2) &= T(\varphi) \wedge T(\varphi_2) \\ T(\exists x \psi) &= \exists x \text{elem}(x) \wedge T(\psi) \end{aligned}$$

Note that $T(\varphi)$ is a formula of length linear in $|\varphi|$. It remains to be shown that φ and $T(\varphi)$ are equisatisfiable.

For one direction, assume that \mathfrak{A} is a model of φ . Take a binary tree t_0 such that all elements of the universe of \mathfrak{A} are leaves of t_0 and are labeled elem and the inner nodes are labeled with b ; then construct a tree t by adding to each leaf e a binary subtree with as many leaves as there are predicates in τ that are true of e , and labeling each such leaf with a distinct predicate true of e in \mathfrak{A} , see Figure 2. It can be easily proved by induction on the structure of φ that t is a UAR tree model of $T(\varphi)$.

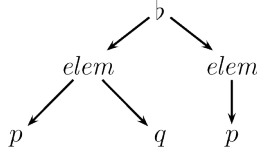


Fig. 2. Representation of a structure over the signature $\{p, q\}$. There are two elements in the universe; the first belongs to the relations p and q , the second to p .

For the other direction assume that t is a model of $T(\varphi)$. Construct a structure \mathfrak{A} such that the universe of \mathfrak{A} is the set of nodes labeled elem in t and for all elements e and all

predicates p , $p(e)$ is true in \mathfrak{A} if and only if there is a node e' labeled p that is below e in t . Again it is easy to prove by structural induction that φ is true in \mathfrak{A} . \square

The proof above does not rely on the fact that \mathfrak{T} is a *binary* tree. We add this requirement to be able to build on the argument while proving Theorem 7.2.

5.2. Upper bound for $\text{FO}^2[\downarrow_+]$ with the UAR

We will prove a matching NEXPTIME upper bound. To prove this bound, we extend an idea introduced in the thesis of Weis [Weis 2011], working in the context of $\text{FO}^2[<]$ on UAR words: polynomial bounds on the number of times a formula changes its truth value while keeping the same unary predicate along a given path.

First we introduce notation adapted from [Weis 2011].

We will revise $\not\sim$ to express incomparability in the relationships relevant to this proof. We now abbreviate the formula $\neg(x\downarrow_+y \vee x=y \vee x\uparrow^+y)$ by $x\not\sim y$, and let us consider $\not\sim$ as an atomic binary symbol. Thus there are four atomic formulas with two free variables: $x\downarrow_+y$, $x=y$, $x\uparrow^+y$, $x\not\sim y$. We call these *order formulas*.

A formula $\varphi \in \text{FO}^2[\downarrow_+]$ is in *existential negation normal form* (ENNF) if it does not contain any universal quantifiers, and negations only appear in front of unary predicates or existential quantifiers. Thus negations in front of order formulas are not allowed. It is easy to see that for any formula in $\text{FO}^2[\downarrow_+]$ we can find an equivalent formula in ENNF of linear size. The idea is to push negation inside the formula by induction and use equivalent formulas. For example the formula $\neg(x\downarrow_+y)$ can be translated to $x=y \vee x\uparrow^+y \vee x\not\sim y$.

Let $\delta(x, y) \in \text{FO}^2[\downarrow_+]$ be a formula in ENNF. We represent $\delta(x, y)$ in the following form

$$\delta(x, y) = \beta(x\downarrow_+y, x=y, x\uparrow^+y, x\not\sim y, \varphi_1, \dots, \varphi_r) \quad (\star)$$

where β is a positive boolean formula and φ_i are formulas in ENNF, each with at most one free variable. An ENNF formula can be assumed to be represented in this form, because the only atomic formulas with two free variables are order formulas. We split the formulas φ_i into two groups: $\xi_1(x), \dots, \xi_s(x)$ and $\zeta_1(y), \dots, \zeta_t(y)$ depending whether x or y is the free variable. It is immediate to see that every formula φ is equivalent to a disjunction of formulas $\varphi_{\downarrow_+}, \varphi_-, \varphi_{\uparrow^+}, \varphi_{\not\sim}$, which have the relation between x and y fixed. For example, using the above notation, $\varphi_{\downarrow_+} = \beta(\top, \perp, \perp, \perp, \varphi_1, \dots, \varphi_r)$, where \top and \perp stand for true and false.

Consider a tree t satisfying the UAR, and fix a root-to-leaf path p in t . Denote the length of p by $\max(p)$ and let $p = p_1 \dots p_{\max(p)}$. Let a be a unary predicate and consider $I \subseteq \{1, \dots, \max(p)\}$. We use the following notation (adapted from [Weis 2011]): $I^a = \{i \in I \mid p_i \models a(x)\}$, that is, I^a is the subset of I that contains only positions labeled with a . For every formula φ with one free variable we define the set $p(\varphi, a) = \{i \mid t, p_i \models \varphi(x) \wedge a(x)\}$, which is the set of positions labeled with predicate a , where the formula φ holds. We say that a non-empty subset S of I is an *interval* if for all $i, j \in S$ if $i < k < j$ then $k \in S$. The following is a generalization of Lemma 2.1.10 of Weis [Weis 2011].

THEOREM 5.3. *Consider an $\text{FO}^2[\downarrow_+]$ formula $\varphi(x)$ in ENNF, with at most one free variable x , and a tree t satisfying the UAR. Let a be a unary predicate, and let $p = p_1 \dots p_{\max(p)}$ be a root-to-leaf path in t . There exists a set S such that $p(\varphi, a) = S^a$ and S can be represented as a union of at most $|\varphi|^2$ intervals.*

PROOF OF THEOREM 5.3. We proceed by induction on the size of formula φ . The base step follows using the UAR. For a predicate $\varphi(x) = b(x)$ there are two cases. If $b \neq a$ then the set $p(\varphi, a)$ is empty; otherwise $a = b$ and then it suffices to take

$S = \{1, \dots, \max(p)\}$, which is a single interval. For $\neg\varphi(x)$ suppose the lemma holds for S . Then by complementing S we add at most 1 additional interval. The remaining cases for the boolean operations are routine.

The only interesting case is for existential quantification. We consider a formula $\varphi(x) = \exists y \delta(x, y)$, where

$$\delta(x, y) = \beta(x \downarrow_+ y, x = y, x \uparrow^+ y, x \not\sim y, \xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y))$$

where β is a positive boolean formula as in (\star) . For a boolean valuation σ of the ξ_i 's, and for a relation $R \in \{\downarrow_+, =, \uparrow^+, \not\sim\}$, we let $\delta(\sigma, R)(y)$ be the formula obtained from $\delta(x, y)$ by replacing all $\xi_i(x)$ with true or false according to σ , and replacing all R formulas with \top and the remaining order formulas with \perp . Notice that the variable x is no longer a free variable in the resulting formula.

Fix a number $\max(p)$ and a root-to-leaf path $p = p_1 \dots p_{\max(p)}$ in tree t , where p_1 is the root and $p_{\max(p)}$ is a leaf. For a given p_j let σ_j be the function giving the evaluation of all formulas ξ_i on p_j , i.e., defined by $\sigma_j(\xi_i) = \top \iff t, p_j \models \xi_i(x)$. For a given function σ let

— $U_{\downarrow_+}(\sigma)$ represent the largest i such that $\sigma = \sigma_i$ and

$$\exists n \in t : p_i \downarrow_+ n \wedge t, n \models \delta(\sigma_i, \downarrow_+)(y)$$

— $L_{\uparrow^+}(\sigma)$ represent the smallest i such that $\sigma = \sigma_i$ and

$$\exists n \in t : p_i \uparrow^+ n \wedge t, n \models \delta(\sigma_i, \uparrow^+)(y)$$

— $L_{\not\sim}(\sigma)$ represent the smallest i such that $\sigma = \sigma_i$ and

$$\exists n \in t : p_i \not\sim n \wedge t, n \models \delta(\sigma_i, \not\sim)(y)$$

By definition $U_{\downarrow_+}(\sigma), L_{\uparrow^+}(\sigma), L_{\not\sim}(\sigma) \in \{0, \dots, \max(p) + 1\}$, where 0 and $\max(p) + 1$ stand for the cases where proper witnesses do not exist. That is, $U_{\downarrow_+}(\sigma) = 0$ if all p_i , such that $\sigma = \sigma_i$, do not have a descendant witness; and similarly $L_{\uparrow^+}(\sigma) = \max(p) + 1$, $L_{\not\sim}(\sigma) = \max(p) + 1$ if all p_i , such that $\sigma = \sigma_i$, do not have respectively an ancestor and an incomparable witness.

Claim 5.4. A node p_j in the path p within t satisfies $\varphi(x)$ iff one of the following holds:

- $j \leq U_{\downarrow_+}(\sigma_j)$ (thus p_j has a witness to φ which is its descendant);
- $j \geq L_{\uparrow^+}(\sigma_j)$ (thus p_j has a witness to φ which is its ancestor);
- $j \geq L_{\not\sim}(\sigma_j)$ (thus p_j has a witness to φ that is incomparable to it);
- $t, p_j \models \varphi(x)$.

PROOF. Suppose $(t, p_j) \models \varphi$. Then there exists a q such that $(t, p_j, q) \models \delta(x, y)$. Suppose $p_j \downarrow_+ q$. Then by definition $t, q \models \delta(\sigma_j, \downarrow_+)(y)$ and thus $j \leq U_{\downarrow_+}(\sigma_j)$. Similarly for the cases when $p_j \uparrow^+ q$ or $p_j \not\sim q$. Otherwise $p_j = q$ and then of course $t, p_j \models \varphi(x)$.

For the other direction suppose $j \leq U_{\downarrow_+}(\sigma_j)$. Let $i = U_{\downarrow_+}(\sigma_j)$. By definition there exists a q such that $p_i \downarrow_+ q$ and $t, q \models \delta(\sigma_i, \downarrow_+)(y)$. Since $j \leq i$ then q is also a witness for p_j , i.e., $(t, p_j, q) \models \varphi$. For the other cases the proof is similar. \square

The proof of Claim 5.4 is essentially the same as the proof of Lemma 2.1.9 in [Weis 2011]. The main difference is that we have to deal with the $\not\sim$ relation. But it actually does not change a lot because $\not\sim$ is monotonic in the following sense: if $x \not\sim y$ and $y \downarrow_+ z$ then $x \not\sim z$.

We define $L(\sigma) = \min(L_{\uparrow^+}(\sigma), L_{\not\sim}(\sigma))$; and $U(\sigma) = U_{\downarrow_+}(\sigma)$. The points $L(\sigma)$ and $U(\sigma)$ correspond to the positions p and q from Lemma 2.1.9 in [Weis 2011]. We would like to point out the main difference in our approach. In [Weis 2011] the model is a word and the points p and q correspond to the most extremal witnesses: one could determine

whether a formula is true by establishing the relation of the free variable x with points p and q . Here the points $L(\sigma)$ and $U(\sigma)$ are the most extremal positions for which one can find a witness. While on words these two definitions amount to the same thing, on trees the distinction is important because some witnesses (descendants or incomparable elements) may lie on different paths and we want to work only on a single path.

The rest of the argument follows the approach of Lemma 2.1.10 of [Weis 2011], but we give a self-contained proof below. Recall that $\varphi(x) = \exists y \delta(x, y)$, where

$$\delta(x, y) = \beta(x \downarrow_+ y, x=y, x \uparrow^+ y, x \not\sim y, \xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y)).$$

By induction for every $u \leq s$ there exists a set S_u such that $p(\xi_u, a) = S_u^a$ and S_u is represented as a sum of at most $|\xi_u|^2$ intervals. Let F_u be the set of left boundaries of S_u , and let G_u be the set of right interval boundaries, where (by convention) we take the decomposition into intervals of the form $[c, d)$, i.e., such that the right (upper) boundary is not part of the interval but the left boundary is in the interval. If there is no proper boundary point from the right side then we use an additional point $\max(p) + 1$. Let $F = \bigcup_u F_u$ and $G = \bigcup_u G_u$ be the total sets of left and right interval boundaries, and let $H = F \cup G \cup \{1, \max(p) + 1\}$.

Recall that σ_j is the boolean valuation of ξ_i in p_j . Consider each interval I defined by two consecutive elements of H . The truth values of ξ_i are constant on such an interval, thus by Claim 5.4 the truth value of φ on position j labeled with a is determined by where j is relative to $L(\sigma_j)$ and $U(\sigma_j)$ and by the truth values of the formulas ζ_1, \dots, ζ_t .

LEMMA 5.5. *Let C be the set of all new interval boundaries that are positions from H or positions of the form $r + 1, q$ where $r = U(\sigma_j)$ and $q = L(\sigma_j)$ for all $j \in H$. Then $|C| \leq 4 + 2(\sum_{i=1}^s |\xi_i|)^2$.¹*

Lemma 5.5 is the most technical part of the proof. It is easy to show that $|C| \leq 4 + 6(\sum_{i=1}^s |\xi_i|)^2$. By the induction assumption for each subformula ξ_i we have $|H| \leq 2 \cdot (\sum_{i=1}^s |\xi_i|)^2$ (each interval has two endpoints). By definition of C for every endpoint we add at most two new endpoints defined by the functions L and U , which gives us $6 \cdot (\sum_{i=1}^s |\xi_i|)^2$ points. The 4 additional points arise when we analyze the extremal positions in path p (1 and $\max(p) + 1$). Unfortunately, the bound $4 + 6(\sum_{i=1}^s |\xi_i|)^2$ is not good enough to prove Theorem 5.3.

Before presenting the proof of Lemma 5.5, we show that Theorem 5.3 follows from Lemma 5.5. Recall that by Claim 5.4 the truth value of each position j labeled with a is determined by where j is relative to $L(\sigma_j)$ and $U(\sigma_j)$ and by the truth values of the formulas ζ_1, \dots, ζ_t . The truth value of the first part is determined by the interval boundaries from the set C , and by Lemma 5.5 $|C| \leq 4 + 2(\sum_{i=1}^s |\xi_i|)^2$. To determine the truth value of the formulas ζ_1, \dots, ζ_t by induction we can use a set of intervals of size at most $\sum_i |\zeta_i|^2$. Altogether, this gives us at most

$$4 + 2(\sum_i |\xi_i|)^2 + 2\sum_i |\zeta_i|^2 \leq 4 + 2(\sum_i |\xi_i| + \sum_i |\zeta_i|)^2 \leq 2 \cdot |\varphi|^2$$

of interval boundaries. Thus the number of intervals is bounded by $|\varphi|^2$. This completes the proof of Theorem 5.3, up to the missing proof of Lemma 5.5.

PROOF OF LEMMA 5.5. Consider each interval $I = [c, d)$ defined by two consecutive interval boundaries c and d from H . The truth values of the formulas ξ_1, \dots, ξ_s remain the same among all points from I^a . Let σ be the function giving this evaluation. We

¹In [Weis 2011] the corresponding positions r, q are denoted p, q . Here, we have changed p to r because p is already used to denote a path. Additionally, in [Weis 2011] the position q needed to be shifted to $q + 1$. In this proof the positions p, r are defined slightly different (using L and U), and instead we needed to shift the position r to $r + 1$.

assign the cost of a point $r + 1$, where $r = U(\sigma)$ (if it lies inside this interval) to the left interval boundary c , and the cost of a point q , where $q = L(\sigma)$ (if it lies inside this interval) to the right interval boundary d . From now on, we refer to these points as $r(c)$ and $q(d)$ respectively (in particular $r(c) = U(\sigma) + 1$). By definition $c < r(c)$ and $q(d) \leq d$. Thus any point $i \in H$ can contribute at most three new interval boundaries: the point itself, a point $r(i)$ if $r(i)$ is inside the interval that starts at i , and a point $q(i)$ if $q(i)$ is inside the interval that ends at i . Let $C(i)$ be the set of all new interval boundaries contributed by i , we additionally define:

$$\begin{aligned} R(i) &= \{r(i) \mid r(i) \in C \text{ and } r(i) < \min\{j \in H \mid j > i\}\} \\ Q(i) &= \{q(i) \mid q(i) \in C \text{ and } q(i) > \max\{j \in H \mid j < i\}\}. \end{aligned}$$

Thus $R(i)$ and $Q(i)$ respectively contain $r(i)$ and $q(i)$ only if they are used as new interval boundaries. Otherwise they are empty. Also notice that $C(i) \subseteq \{i\} \cup R(i) \cup Q(i)$.

To bound the size of C we split the set $C(i)$ into two families of sets. For every $u \in [1, s]$ and all $i \in F_u$ we define:

$$\widehat{C}_F(u, i) = \begin{cases} \{i\} \cup R(i), & \text{if } i \in G. \\ C(i), & \text{otherwise.} \end{cases}$$

And similarly for $i \in G_u$:

$$\widehat{C}_G(u, i) = \begin{cases} \{i\} \cup Q(i), & \text{if } i \in F. \\ C(i), & \text{otherwise.} \end{cases}$$

We claim that

$$C \subseteq \{1\} \cup \{\max(p)+1\} \cup R(1) \cup Q(\max(p)+1) \cup \bigcup_{u \in [1, s]} \left(\bigcup_{i \in F_u} \widehat{C}_F(u, i) \cup \bigcup_{i \in G_u} \widehat{C}_G(u, i) \right). \quad (\heartsuit)$$

To prove this we need to show that for all points $i \in H$, $C(i)$ is contained in the set on the right. The only non-trivial cases are if: $i \in F \setminus G$; $i \in G \setminus F$; or $i \in F \cap G$. The first two cases follow easily from the definition of $\widehat{C}_F(u, i)$ and $\widehat{C}_G(u, i)$. In the last case notice that $\widehat{C}_F(u, i) \cup \widehat{C}_G(u, i) = \{i\} \cup R(i) \cup Q(i)$.

Let $F_{\bar{u}} = \bigcup_{v \neq u} F_v$ and let $G_{\bar{u}} = \bigcup_{v \neq u} G_v$.

Claim 5.6. Given c and d consecutive interval boundaries from $F_{\bar{u}}$, there is at most one $i \in F_u \cap [c, d)$ with $R(i) \neq \emptyset$.

PROOF. Suppose there is $i \in F_u \cap [c, d)$ with $R(i) \neq \emptyset$. By definition of $R(i)$ we know that $R(i) = \{r(i)\} = \{U(\sigma_i) + 1\}$ and $i < r(i)$. Therefore, by definition of $U(\sigma_i)$, there exists a witness w such that $t, p_l, w \models \delta(x, y)$ for every $i \leq l \leq U(\sigma_i)$ such that p_l is labeled with a .

Consider another $j \in F_u \cap [c, d)$ with $j < i$. We show that $R(j) = \emptyset$. Since the interval $[c, d)$ contains no left interval boundaries besides the ones from F_u , and since i and j are both in F_u , and hence are both in $p(\xi_u, a)$, we conclude that every $\xi_k : k \leq s$ that holds in the interval starting from i also holds at the interval starting from j . In other words $\sigma_i(\xi_k) = \top$ implies $\sigma_j(\xi_k) = \top$. Recall that $\delta(x, y)$ is represented as a positive boolean formula β . Since β is a positive formula then $t, p_l, w \models \delta(x, y)$ for every l that belongs to the interval starting with j and such that p_l is labeled with a . Thus if $r(j)$ is defined then $r(j) = U(\sigma_j) + 1 \geq U(\sigma_i) + 1 = r(i) > i$, and so $R(j) = \emptyset$. \square

Claim 5.7. Let c and d be two consecutive interval boundaries from $F_{\bar{u}}$, $i \in F_u \cap [c, d)$ and $Q(i) \neq \emptyset$ such that $Q(i) \subseteq \widehat{C}_F(u, i)$. Then we have $i \notin \widehat{C}_F(u, i)$.

PROOF. Fix c, d, i as in the claim. Since $Q(i) \neq \emptyset$ then $Q(i) = \{q(i)\} = \{L(\sigma_i)\}$ and $q(i) \leq i$. Therefore, by definition of $L(\sigma_i)$, there exists a witness w such that $t, p_l, w \models \delta(x, y)$ for every $L(\sigma_i) \leq l \leq i$ such that p_l is labeled with a .

By definition of $\widehat{C}_F(u, i)$ we have that $i \notin G$, i.e., i is not a right interval boundary. Therefore, the ξ_k that are true at the interval ending at i are also true at the interval starting at i . Since β is a positive formula then $t, p_l, w \models \delta(x, y)$ for every l that belongs to the interval starting with i and such that p_l is labeled with a . Thus i is not a new interval boundary and since $\widehat{C}_F(u, i) = C(i)$ then $i \notin \widehat{C}_F(u, i)$. \square

The rest of the argument follows that in [Weis 2011] precisely.

The above two claims imply that for every $i \in F_u \cap [c, d]$ except possibly one element, $\widehat{C}_F(u, i)$ is either empty, contains the single element $Q(i)$, or contains only i . At the one exceptional element $\widehat{C}_F(u, i)$ could consist of at most two elements, $R(i)$ and either $Q(i)$ or i (but not both, by the second claim).

Therefore, $\bigcup_{i \in F_u \cap [c, d]} \widehat{C}_F(u, i)$ has at most $|F_u \cap [c, d]| + 1$ elements. Summing over all intervals $[c, d]$ we get

$$\sum_{i \in F_u} |\widehat{C}_F(u, i)| \leq \sum_{c \in F_{\bar{u}}} (|F_u \cap [c, d]| + 1) = |F_{\bar{u}}| + |F_u|$$

Using again the fact that each $\widehat{C}_F(u, i)$ contains at most two elements, we also know $\sum_{i \in F_u} |\widehat{C}_F(u, i)| \leq 2 \cdot |F_u|$, and thus:

$$\sum_{i \in F_u} |\widehat{C}_F(u, i)| \leq |F_u| + \min\{|F_u|, |F_{\bar{u}}|\}$$

For each k , the number of intervals, and hence the number of left endpoints of intervals, is assumed to be at most $|\xi_k|^2$. Since the sum of squares is bounded by the square of a sum we get:

$$\begin{aligned} \sum_{i \in F_u} |\widehat{C}_F(u, i)| &\leq |\xi_u|^2 + \min\{|\xi_u|^2, (\sum_{v \neq u} |\xi_v|)^2\} \\ &\leq |\xi_u|^2 + |\xi_u| \cdot \min\{|\xi_u|, \sum_{v \neq u} |\xi_v|\} \\ &\leq |\xi_u|^2 + |\xi_u| \cdot \sum_{v \neq u} |\xi_v| \\ &= |\xi_u| \cdot \sum_v |\xi_v| \end{aligned}$$

By a symmetric argument we get

$$\sum_{i \in G_u} |\widehat{C}_G(u, i)| \leq |\xi_u| \cdot \sum_v |\xi_v|$$

Now the total number of boundary points for C is at most the four extremal points: $1, \max(p) + 1, R(1), Q(\max(p) + 1)$; and the union over i of $C(i)$ (see (\heartsuit)). Thus we have that the total number is at most:

$$4 + \sum_u (2 \cdot |\xi_u| \cdot \sum_i |\xi_i|) \leq 4 + 2 \cdot (\sum_i |\xi_i|)^2$$

This completes the proof of Lemma 5.5. \square

This proves Theorem 5.3. \square

From Theorem 5.3, we will show that $\text{FO}^2[\downarrow_+]$ sentences that are satisfiable over UAR trees always have polynomial-depth witnesses:

LEMMA 5.8. *If an $\text{FO}^2[\downarrow_+]$ formula φ is satisfied over a UAR tree, then it is satisfied by a model of depth bounded by a polynomial in $|\varphi|$.*

PROOF. First, we convert φ to ENNF. Our approach will slightly differ from the word case. In [Weis 2011] the idea was to take a model word w and divide it into polynomially

many intervals, where each interval required at most two witnesses. The final word was constructed by taking only these witnesses from w . On trees we could try the same thing. By Theorem 5.3 we could divide every path into polynomially many intervals, such that on every interval the subformulas would be constantly true or false. Then every interval requires at most two witnesses (they are provided by the functions U and L from the proof of Theorem 5.3). Unfortunately $\not\sim$ witnesses appear on different paths and we would have to control the number of witnesses on many paths at once. Because of this we shall shrink the paths. That is, instead of constructing a tree from witnesses, we shrink the intervals on every path.

Suppose that φ is satisfied over a UAR tree t . Let a be a unary predicate and let p be a fixed path in t . By Theorem 5.3 for every subformula ρ of φ with at most one free variable there is a set S_ρ such that $S_\rho^a = p(\rho, a)$ and S_ρ is represented as a union of at most $|\rho|^2$ intervals. Let $E_{\rho,a}$ be the set of endpoints of the intervals in this representation. Let $E_a = \bigcup_\rho E_{\rho,a} \cup \{1, \max(p)\}$. The size of E_a is still polynomial because the number of subformulas is polynomial.

The set E_a divides the path p into a polynomial number of intervals. Let I be one of these intervals. For every interval we define the set I_a of extremal positions labeled with a . Formally, if I does not contain a position labeled with a then $I_a = \emptyset$; otherwise I_a contains at most two positions i, j such that for every $k \in I$ if p_k is labeled with a then $i \leq k \leq j$ (if there is only one position labeled with a in I then $i = j$). We define the set of all extremal positions $W_a = \bigcup I_a$. Notice that W_a is of polynomial size, because there are polynomially many intervals I_a and each I_a is of size at most 2.

Finally, let $W = \bigcup_a W_a$. We note the following crucial property of W : Let $k \notin W$. There are positions $i, j \in W$ such that $i < k < j$ and p_k, p_i, p_j satisfy the same set of subformulas with one free variable. Formally, $p_k \models \rho(x)$ iff $p_i \models \rho(x)$ iff $p_j \models \rho(x)$ for every subformula with one free variable ρ . This property follows immediately from the definition of the set of extremal positions I_a , where a is the label of p_k .

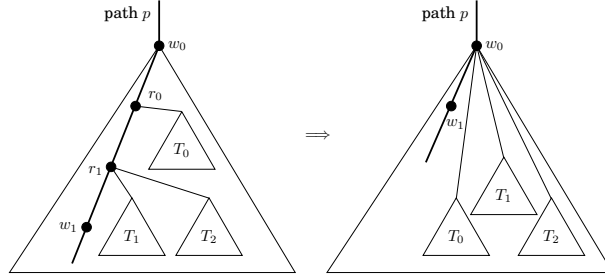


Fig. 3. Nodes $w_0, w_1 \in W$ are endpoints of some interval and r_0, r_1 are nodes between w_0 and w_1 . We remove r_0, r_1 but the subtrees of r_0 and r_1 are attached to w_0 .

The idea is now to remove all points from path p whose positions are not in W . This must be done in a slightly unusual way, by “promoting” subtrees that are off the path. Let r be a removed node from the path p . For every child c of r such that c does not lie on path p we attach $\text{SubTree}(t, c)$ to the closest node above r whose position is in W (see Figure 3). Let t' denote the tree obtained as a result of this surgery.

Let f be the partial function taking nodes from t to their copies in t' . By construction the image of f are all nodes in t' . We claim that t' still satisfies φ , and more generally that for every subformula with at most one free variable ρ of φ and for every node $f(m)$ in t' , we have $t', f(m) \models \rho$ iff $t, m \models \rho$. This is proved by induction on the size of ρ , with the base cases and the cases for boolean operators being straightforward. Consider the

only nontrivial case with the existential quantifier $\rho(x) = \exists y \delta(x, y)$. As in (\star) we can assume that

$$\delta(x, y) = \beta(x \downarrow_+ y, x=y, x \uparrow^+ y, x \nearrow y, \xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y)), \quad (1)$$

where β is a positive boolean formula and ξ_i, ζ_j are subformulas of ρ with at most one free variable.

First, suppose that $t', f(m) \models \rho$. Then there exists a node $f(w)$ such that $t', f(m), f(w) \models \delta$. By induction assumption both pairs of nodes $m, f(m)$ and $w, f(w)$ satisfy the same set of subformulas among $\xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y)$. The relation between m and w is the same as between $f(m)$ and $f(w)$ because the construction does not change the binary relations between nodes. Then by (1) $t, m, w \models \delta$, and therefore $t, m \models \rho$.

For the opposite direction assume that $t, m \models \rho$, such that f is defined on m . Then there exists a node m' such that $t, m, m' \models \delta$. If f is defined on m' then the proof follows the same steps as in the first case. The only nontrivial case is when m' is a node that was removed, i.e., $m' = p_k \notin W$. By the crucial property of W there exist p_i, p_j that satisfy the same set of subformulas as p_k and such that $i < k < j$. A simple case analysis shows that the binary relation between m and p_k is the same as between $f(m)$ and $f(w)$, where $w = p_i$ or $w = p_j$. For example if $m \sim p_k$ then $m \sim p_j$ and thus $f(m) \sim f(p_j)$. Then by (1) $t', f(m), f(w) \models \delta$, and therefore, $t', f(m) \models \rho$.

This process reduces the length of the chosen path p to the size of the set $|W|$, which is polynomial in $|\varphi|$. Since it does not increase the length of any other path, it is clear that iterating this process on all paths yields a tree of polynomial depth. \square

The proof of Lemma 5.8 relies on the fact that models are unranked trees. As seen in Figure 3 in this construction we need to attach new subtrees to some nodes. The problem when models are restricted to ranked trees is discussed in Section 7.

Note that the arguments in Theorem 3.3 and in Lemma 5.8 allow shrinking arbitrary models to obtain models with exponential branching and polynomial depth. Hence the two arguments can be combined to conclude that the logic $\text{FO}^2[\downarrow_+]$ has the exponential model property. We can therefore guess a tree as above in NEXPTIME and verify that it is a witness to satisfiability in polynomial time, giving the following bound:

THEOREM 5.9. *Satisfiability for $\text{FO}^2[\downarrow_+]$ formulas over UAR trees is in NEXPTIME, and hence is NEXPTIME-complete.*

The above theorem completes the analysis of FO^2 over UAR unranked trees, up to the advertised lower bounds for GF^2 which will be presented in the next section. Let us recapitulate: all fragments are at least NEXPTIME-hard; the maximal fragments in NEXPTIME are $\text{FO}^2[\downarrow_+]$ and $\text{FO}^2[\downarrow, \rightarrow, \rightarrow^+]$ (the latter because of the bound in Theorem 4.1 which holds even without UAR); the bigger fragments are EXPSPACE-complete.

6. TWO-VARIABLE GUARDED FRAGMENT

For $\text{FO}^2[\downarrow_+]$ over trees the complexity drop from EXPSPACE to NEXPTIME, resulting from restricting the class of models to those satisfying UAR, is slightly less significant than in the case of words, where an analogous restriction decreases the complexity from NEXPTIME to NP. However, to obtain NEXPTIME hardness, we need to speak about pairs x, y of incomparable elements, i.e., such elements that y is neither an ancestor nor descendant of x . It is natural to consider the situation where quantification is restricted to only pairs of elements that are connected by binary navigational relations. To capture this scenario we consider the restriction of FO^2 to the two-variable guarded fragment, GF^2 , in which all quantifiers have to be relativised by binary predicates. It is easy to

see that GF^2 on trees still embeds NavXP, while still being exponentially more succinct (see, for example, the comment and reference after Proposition 2.1).

We will show that, indeed, the satisfiability problem for $\text{GF}^2[\downarrow_+]$ over UAR trees is easier than for $\text{FO}^2[\downarrow_+]$, namely, it is PSPACE-complete. Further, we observe that adding to $\text{GF}^2[\downarrow_+]$ any of the remaining navigational predicates lifts the complexity to EXPSPACE. Finally, we establish the complexity of GF^2 over trees and UAR trees for signatures containing \downarrow but not containing \downarrow_+ .

We start with an adaptation of our normal form to the guarded case. We say that a $\text{GF}^2[\tau_{bin}]$ formula φ is in *guarded normal form* if

$$\varphi = \bigwedge_{i \in H} \exists x \psi_i(x) \wedge \bigwedge_{i \in I} \forall xy (\eta_i(x, y) \Rightarrow \psi'_i(x, y)) \wedge \bigwedge_{i \in J} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi''_i(x, y))),$$

for some disjoint index sets H, I and J , where η_i is a guard built from x, y and a navigational predicate from τ_{bin} or the equality symbol, $\lambda_i(x)$ is an atomic formula $a(x)$ for some unary symbol a , and each $\psi_i(x), \psi'_i(x, y), \psi''_i(x, y)$ are boolean combinations of unary atomic formulas. We will use guarded normal form in two cases: for $\tau_{bin} = \{\downarrow_+\}$ and $\tau_{bin} = \{\rightarrow, \downarrow\}$. One can show that for such τ_{bin} satisfiability of a $\text{GF}^2[\tau_{bin}]$ formula can be reduced to satisfiability of a normal form $\text{GF}^2[\tau_{bin}]$ formula nondeterministically.

LEMMA 6.1. *For $\tau_{bin} = \{\downarrow_+\}$ or $\tau_{bin} = \{\rightarrow, \downarrow\}$ there exists a polynomial nondeterministic procedure, taking as its input a $\text{GF}^2[\tau_{bin}]$ formula φ over a signature $\tau_{un} \cup \tau_{bin}$ and producing a guarded normal form $\text{GF}^2[\tau_{bin}]$ formula φ' over a signature $\tau'_{un} \cup \tau_{bin}$, for τ'_{un} consisting of τ_{un} and some fresh unary symbols, such that*

- φ is satisfiable iff the procedure has a run producing a satisfiable φ' ,
- φ is satisfiable over a UAR tree iff the procedure has a run producing φ' satisfiable over a tree which restricted to $\tau_{un} \cup \tau_{bin}$ is a UAR tree.

PROOF. The idea of the proof is similar to the proof of Lemmas 3.1 and 3.2. Some details concerning the guardedness are treated similarly to [Szwast and Tendera 2004]. Note that a normal form for GF appears also in [Grädel 1999] but the translation there increases the number of variables.

Take a $\text{GF}^2[\tau_{bin}]$ formula φ (for an appropriate τ_{bin}). We begin with an innermost subformula ψ of φ starting with a quantifier. If ψ contains a free variable, i.e., it is of the form $\exists y (\eta(x, y) \wedge \psi'(x, y))$, for some binary guard η , then we replace it by $p_\psi(x)$ and add two guarded normal form conjuncts $\forall x (p_\psi(x) \Rightarrow \exists y (\eta(x, y) \wedge \psi'(x, y)))$ and $\forall xy (\eta(x, y) \Rightarrow \neg \psi'(x, y) \vee p_\psi(x))$ axiomatising p_ψ . If ψ is a subsentence, i.e., it is of the form $\exists y (\lambda(y) \wedge \psi'(y))$ or $\exists xy (\eta(x, y) \wedge \psi'(x, y))$ then we nondeterministically guess its truth value, replace it by \top or \perp according to this guess and add ψ or, respectively, $\neg \psi$ as a conjunct of our new formula. Moving up the original formula φ we repeat this procedure for subformulas that are not innermost, and so forth. The formula obtained in this process has, up to trivial logical transformations, the required format

$$\bigwedge_{i \in H} \exists x \psi_i(x) \wedge \bigwedge_{i \in I} \forall xy (\eta_i(x, y) \Rightarrow \psi'_i(x, y)) \wedge \bigwedge_{i \in J} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi''_i(x, y))).$$

However, its subformulas of the form $\eta_i(x, y) \Rightarrow \psi'_i(x, y)$ or $\eta_i(x, y) \wedge \psi''_i(x, y)$ may contain in $\psi'_i(x, y)$ or $\psi''_i(x, y)$ some atoms involving binary navigational predicates. We replace each such atom by \top if it equals the guard $\eta_i(x, y)$ and by \perp otherwise. E.g., having a conjunct $\forall xy (x \rightarrow y \Rightarrow y \downarrow x \vee y \rightarrow x \vee p(x) \wedge x \rightarrow y)$ we obtain $\forall xy (x \rightarrow y \Rightarrow \perp \vee \perp \vee p(x) \wedge \top)$ or equivalently, $\forall xy (x \rightarrow y \Rightarrow p(x))$. The last transformation is sound, because out of the atoms built from x, y and a navigational predicate, at most one can be satisfied

by an ordered pair of elements, in the case of both relevant navigational signatures, $\tau_{bin} = \{\downarrow_+\}$ or $\tau_{bin} = \{\rightarrow, \downarrow\}$. \square

Since our upper bounds will be above NP, the above lemma allows us to restrict attention to normal form GF^2 formulas.

6.1. Signatures containing descendant

Let us now show that in an arbitrary (not necessarily UAR) tree t satisfying a guarded normal form $\text{GF}^2[\downarrow_+]$ formula φ we can find a smaller tree in which the degree of nodes is bounded polynomially in $|\varphi|$ and the depth of t . Since by Lemma 5.8 we are able to shorten paths in UAR trees to length polynomial in $|\varphi|$, this will lead to a polynomial bound on the degree of nodes in UAR models of $\text{GF}^2[\downarrow_+]$ formulas (which, as we have seen, contrasts with the case of $\text{FO}^2[\downarrow_+]$).

LEMMA 6.2. *Let φ be a $\text{GF}^2[\downarrow_+]$ guarded normal form formula and let t be its model. Then, there exists a tree t' , obtained by removing some subtrees from t , still satisfying φ , such that the degree of nodes in t' is bounded polynomially in $|\varphi|$ and the depth of t .*

PROOF. Let r be the root of t . For every conjunct $\exists x \psi_i(x)$ of φ pick a witness w satisfying ψ_i and mark it together with all elements on the path from r to w . Similarly for every conjunct of φ of the form $\forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i''(x, y)))$, with $\eta_i(x, y) = x \downarrow_+ y$, if $t, r \models \lambda_i(x)$ then pick a witness w such that $t, r, w \models \eta_i \wedge \psi_i''$, and mark it together with elements on the path from r to w . Remove all the subtrees rooted at the children of r containing no marked elements. Repeat this procedure of choosing witnesses for conjuncts of φ of the form $\forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i''(x, y)))$ and removing unnecessary subtrees for all the remaining nodes from t , say, in the depth-first manner. Note that the structure obtained after each step is a model of φ , since we explicitly take care of providing lower witnesses, and the upper witnesses are retained automatically as every element which is not removed from the model is kept together with the whole path from the root of the original model t . Let t' be the tree obtained after the final step of the above procedure. Observe that the number of marked children of a node n of depth l is bounded by $(l + 1) \cdot |\varphi|$. Indeed, n may require at most $|\varphi|$ witnesses in its subtree, and has l ancestors, each of which may require also at most $|\varphi|$ witnesses in the subtree of n . Thus, denoting the depth of t by d , the degree of each node of t' is bounded by $d \cdot |\varphi|$ as required. \square

THEOREM 6.3. *The satisfiability problem for $\text{GF}^2[\downarrow_+]$ over UAR trees is PSPACE-complete.*

PROOF. For the upper bound we design an alternating polynomial time procedure solving the problem. Let φ be a $\text{GF}^2[\downarrow_+]$ input formula over signature $\tau_{un} \cup \{\downarrow_+\}$. We first nondeterministically compute a guarded normal form formula φ' as guaranteed by Lemma 6.1. It remains to test satisfiability of φ' . The procedure builds a path in a model together with information about all the children of its nodes. The information about a node n consists of its atomic 1-type, i.e., the set of predicates from τ'_{un} which are true at n , and a polynomially bounded set of atomic 1-types of the *promised types of descendants* of n . The procedure starts by guessing information about the root and then moves down the tree in the following way: when inspecting a node n it guesses information about all its children (polynomially many) and then proceeds universally to each of them. During the execution the following conditions are checked:

- (1) Every guessed atomic type contains precisely one predicate from τ_{un} .

- (2) The set of promised types of descendants of the current node n is sufficient to provide necessary witnesses for n for conjuncts of φ' of the form $\forall x(\lambda_i(x) \Rightarrow \exists y(x \downarrow_+ y \wedge \psi_i(x, y)))$; additionally the promised types of descendants of the root ensure the satisfaction of the conjuncts of φ' of the form $\exists x\psi_i(x)$.
- (3) The current node has the required witnesses for the conjuncts of the form $\forall x(\lambda_i(x) \Rightarrow \exists y(y \downarrow_+ x \wedge \psi_i(x, y)))$ among its ancestors.
- (4) The universal part $\forall\forall$ of φ is not violated by a pair consisting of the current node n and any of its ancestors.
- (5) Every promised type of a descendant of the inspected node n is either realised or promised by one of its children.

The procedure accepts when it reaches a node with no promised descendants (without violating the above conditions) in at most polynomially many steps.

It should be clear that existence of an accepting run of the procedure implies the existence of a model of φ' . In the opposite direction, if φ' has a model, then by combining Lemma 5.8 and Lemma 6.2, we know that it also has a model t with depth and degree of nodes bounded polynomially in the size of φ' . Our procedure thus has an accepting run in which all guesses are made in accordance with t . Note in particular that polynomially bounded sets of promised types of descendants are sufficient, since every node requires at most linearly many witnesses and we restrict attention to paths with at most polynomially many nodes.

The lower bound follows from PSPACE-hardness of NavXP with only the descendant axis (Proposition 6.10 in [Figueira 2012a]), since the standard translation of NavXP to first-order logic is contained in GF^2 . Here we show an alternative proof which involves an alphabet of constant size, in contrast to Figueira's proof, which requires an unbounded alphabet. We proceed by a reduction from the satisfiability problem for quantified boolean formulas, QBF. Let ψ be an instance of QBF problem. Without loss of generality we may assume that ψ is of the form

$$\exists v_k \dots \exists v_2 \forall v_1 \psi'$$

where the number of all quantifiers, k , is even, all even-numbered variables are existentially quantified, all odd-numbered variables are universally quantified and ψ' is a propositional formula over the variables v_1, \dots, v_k .

We now translate the formula ψ to a formula over the signature

$$\tau = \{\text{root}, \text{leaf}, \text{true}, \text{false}, \downarrow_+\}$$

such that ψ is true if and only if its translation is satisfiable over UAR trees.

First, for $i \in \{0, \dots, k\}$ we define auxiliary formulas depth_i and height_i . Let $\text{depth}_0(x) = \text{root}(x)$ and for $i \geq 1$ let $\text{depth}_i(x) = \exists y \ x \downarrow_+ y \wedge \text{depth}_{i-1}(y)$. Intuitively, the formula $\text{depth}_i(x)$ expresses that the node x occurs at distance at least i from the root. Let $\text{height}_0(x) = \text{leaf}(x)$, $\text{height}_1(x) = \text{depth}_k(x)$ and let $\text{height}_i(x) = \text{depth}_{k+1-i}(x) \wedge \neg \text{depth}_{k+2-i}(x)$ for $i > 1$. For $i > 0$ the formula $\text{height}_i(x)$ expresses that x is a node at depth exactly $k+1-i$; in the construction below, for $i \geq 0$, the formula $\text{height}_i(x)$ will mean that the subtree rooted at x has height i . Note that $\text{height}_i(x)$ is a guarded formula of length linear in i .

In the following construction, a model of the translation of ψ is a tree that describes a set of valuations justifying that ψ is true. Intuitively, it is a binary tree of depth $k+1$ where every path describes a valuation of variables v_1, \dots, v_k . Formally the tree might contain more than one node labeled *root*, but each such node provides enough evidence that ψ is true. Every node at height i for $i \leq k$ is labeled either *true* or *false*, which corresponds to a value of the variable v_i under a given valuation. Every non-leaf node at even height i has two successors corresponding to the universally quantified variable

v_{i-1} ; every node at odd height i (here $i \leq k+1$) has one successor corresponding to the existentially quantified variable v_{i-1} . If $k > 0$ then let $tree_k$ be the conjunction of

$$\exists x \text{ root}(x), \quad (2)$$

$$\forall x \text{ root}(x) \Rightarrow (\exists y x \downarrow_+ y \wedge \text{height}_k(y) \wedge (\text{true}(y) \vee \text{false}(y))), \quad (3)$$

$$\begin{aligned} \forall x \text{ true}(x) \Rightarrow & (\text{height}_i(x) \Rightarrow \\ & ((\exists y x \downarrow_+ y \wedge \text{height}_{i-1}(y) \wedge \text{true}(y)) \\ & \wedge (\exists y x \downarrow_+ y \wedge \text{height}_{i-1}(y) \wedge \text{false}(y)))) \\ & \text{for all odd numbers } 3 \leq i \leq k, \end{aligned} \quad (4)$$

$$\begin{aligned} \forall x \text{ false}(x) \Rightarrow & (\text{height}_i(x) \Rightarrow \\ & ((\exists y x \downarrow_+ y \wedge \text{height}_{i-1}(y) \wedge \text{true}(y)) \\ & \wedge (\exists y x \downarrow_+ y \wedge \text{height}_{i-1}(y) \wedge \text{false}(y)))) \\ & \text{for all odd numbers } 3 \leq i \leq k, \end{aligned} \quad (5)$$

$$\begin{aligned} \forall x \text{ true}(x) \Rightarrow & (\text{height}_i(x) \Rightarrow \\ & \exists y x \downarrow_+ y \wedge \text{height}_{i-1}(y) \wedge (\text{true}(y) \vee \text{false}(y))) \\ & \text{for all even numbers } 2 \leq i \leq k, \end{aligned} \quad (6)$$

$$\begin{aligned} \forall x \text{ false}(x) \Rightarrow & (\text{height}_i(x) \Rightarrow \\ & \exists y x \downarrow_+ y \wedge \text{height}_{i-1}(y) \wedge (\text{true}(y) \vee \text{false}(y))) \\ & \text{for all even numbers } 2 \leq i \leq k, \end{aligned} \quad (7)$$

$$\begin{aligned} \forall x \text{ true}(x) \Rightarrow & (\text{height}_1(x) \Rightarrow \\ & (\exists y x \downarrow_+ y \wedge \text{leaf}(y))), \end{aligned} \quad (8)$$

$$\begin{aligned} \forall x \text{ false}(x) \Rightarrow & (\text{height}_1(x) \Rightarrow \\ & (\exists y x \downarrow_+ y \wedge \text{leaf}(y))). \end{aligned} \quad (9)$$

In the case $k = 0$ the formula $tree_0$ reduces to $\exists x \text{ root}(x) \wedge \forall x \text{ root}(x) \Rightarrow (\exists y x \downarrow_+ y \wedge \text{leaf}(y))$. Note that $tree_k$ is a guarded formula of length polynomial in k . Now we inductively define the translation $T(\psi')$ of the quantifier-free formula ψ' .

$$\begin{aligned} T(\text{true}) &= \text{true} \\ T(\text{false}) &= \text{false} \\ T(v_i) &= \exists y y \downarrow_+ x \wedge \text{height}_i(y) \wedge \text{true}(y) \\ T(\neg\varphi) &= \neg T(\varphi) \\ T(\varphi_1 \wedge \varphi_2) &= T(\varphi) \wedge T(\varphi_2) \\ T(\varphi_1 \vee \varphi_2) &= T(\varphi) \vee T(\varphi_2) \end{aligned}$$

Note that $T(\psi')$ is a guarded formula of length polynomial in $(|\psi'| + k)$. It is not difficult to prove by induction on k (and by nested structural induction on propositional formulas with free variables v_1, \dots, v_k) that ψ is true if and only if $tree_k \wedge \forall x \text{ leaf}(x) \Rightarrow T(\psi')$ has a UAR tree model. Each node labeled *leaf* in such a model uniquely determines a path to a node labeled *root* and such a path corresponds to a valuation of the variables v_1, \dots, v_k that makes the formula ψ' true. \square

Now we show that augmenting $\text{GF}^2[\downarrow_+]$ with any of the remaining binary navigational predicates leads to an EXPSPACE lower bound over UAR trees. (Recall that this completes also Section 5.) This is done in the two following theorems.

THEOREM 6.4. *The satisfiability problem for $\text{GF}^2[\downarrow_+, \downarrow]$ over UAR trees is EXPSPACE-hard.*

PROOF. We give a reduction from $\text{GF}^2[\downarrow_+]$ over arbitrary trees, which is EXPSPACE-hard by Theorem 3.8. The idea of the encoding is similar to Theorem 5.2: a node e in a tree is modeled by a UAR node labeled $elem$ with immediate successors encoding predicates true in e . The binary predicate \downarrow_+ is used to preserve the structure of the tree, the additional \downarrow predicate gives the access to nodes modeling unary predicates. In the following reduction, for a given $\text{GF}^2[\downarrow_+]$ formula φ over a signature $\tau = \tau_0 \cup \{\downarrow_+\}$ we construct a $\text{GF}^2[\downarrow_+, \downarrow]$ formula over the signature $\tau \cup \{\downarrow, elem\}$ that is satisfiable over UAR trees if and only if φ is satisfiable over arbitrary trees.

Let us start with a formula ensuring that the underlying structure is an encoding of a tree. The formula $tree$ is defined as the conjunction of

$$\bigwedge_{p \in \tau_0 \cup \{elem\}} \forall x p(x) \Rightarrow \forall y y \downarrow_+ x \Rightarrow elem(y)$$

with

$$\forall x elem(x) \Rightarrow \forall y x \downarrow_+ y \Rightarrow \bigvee_{p \in \tau_0 \cup \{elem\}} p(y).$$

It ensures that (unless the tree is trivial, i.e., no node is labeled at all) each node is labeled with some predicate symbol, all internal nodes are labeled $elem$ and only leaves may be labeled with predicates from τ_0 . Note that $tree$ is a guarded formula of length linear in $|\tau|$.

Without loss of generality we may assume that the formula φ is built from unary atoms, boolean connectives \wedge, \neg and guarded existential quantification. The translation $T(\varphi)$ of a formula φ is defined inductively as follows.

$$\begin{aligned} T(p(x)) &= \exists y x \downarrow y \wedge p(y) \\ T(\neg \varphi) &= \neg T(\varphi) \\ T(\varphi_1 \wedge \varphi_2) &= T(\varphi_1) \wedge T(\varphi_2) \\ T(\exists x p(x) \wedge \psi(x)) &= \exists x elem(x) \wedge T(p(x)) \wedge T(\psi(x)) \\ T(\exists y x \downarrow_+ y \wedge \psi(x, y)) &= \exists y x \downarrow_+ y \wedge elem(y) \wedge T(\psi(x, y)) \\ T(\exists y y \downarrow_+ x \wedge \psi(x, y)) &= \exists y y \downarrow_+ x \wedge elem(y) \wedge T(\psi(x, y)) \end{aligned}$$

Note that $T(\varphi)$ is a guarded formula of length linear in $|\varphi|$. Again a simple inductive argument shows that φ is satisfiable if and only if $tree \wedge T(\varphi)$ has a UAR tree model. \square

THEOREM 6.5. *The satisfiability problems for $\text{GF}^2[\downarrow_+, \rightarrow]$ and $\text{GF}^2[\downarrow_+, \rightarrow^+]$ over UAR trees are EXPSPACE-hard.*

PROOF. We follow the construction from [Kieroński 2002] and sketch a generic reduction from AEXPTIME. Consider an alternating Turing machine M working in exponential time. Without loss of generality we may assume that M works in time 2^n and that every non-final configuration of M has exactly two successor configurations. Let w be an input word of size n . Following [Kieroński 2002] we construct a formula whose models encode accepting configuration trees of machine M on input w .

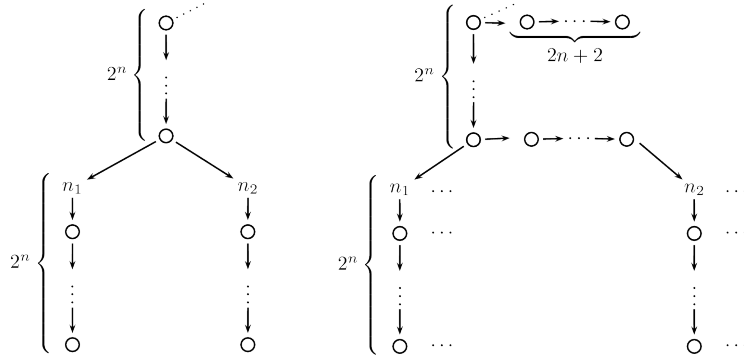


Fig. 4. Left: frame of a configuration tree in [Kieroński 2002]; nodes n_1 and n_2 are siblings. Right: frame of a configuration tree in our encoding; nodes n_1 and n_2 are not siblings.

In [Kieroński 2002] each configuration is represented by 2^n elements of a tree, each of which represents a single cell of the tape of M (see left part of Figure 4). Each such node is then labeled with unary predicate symbols from the set $\{c_0, \dots, c_{n-1}, p_0, \dots, p_{n-1}\}$ to encode the index of a configuration (i.e., the depth of the configuration in the computation tree) it belongs to and its position (i.e., the number of a cell) in this configuration: $c_i(x)$ is true if the i -th bit of the configuration index is 1 and $p_i(x)$ is true if the i -th bit of the position number is 1. Additional predicate symbols are used to encode the tape symbol and the state of the machine (if it is necessary, i.e., if the head of the machine is scanning the cell under consideration). Here, to encode the numbers, we use additional $2n$ elements that are siblings of the node representing a cell, as shown in the right part of Figure 4. Each of these elements stores information about a single bit using one of two unary predicates *zero* or *one*. Then the atomic formulas $c_i(x)$ and $p_i(x)$ are simulated by formulas

$$\exists y x \rightarrow^+ y \wedge Path_i(y) \wedge one(y) \quad \text{and respectively} \quad \exists y x \rightarrow^+ y \wedge Path_{n+i}(y) \wedge one(y)$$

where the subformula $Path_i(y)$ is defined recursively as follows. For the logic $GF^2[\downarrow_+, \rightarrow]$ we define

$$Path_0(y) = \neg \exists x x \rightarrow y$$

$$Path_{i+1}(y) = \exists x x \rightarrow y \wedge Path_i(x)$$

and for the logic $GF^2[\downarrow_+, \rightarrow^+]$ we define

$$Path_{\geq 0}(y) = \neg \exists x x \rightarrow^+ y$$

$$Path_{\geq i+1}(y) = \exists x x \rightarrow^+ y \wedge Path_{\geq i}(x)$$

$$Path_i(y) = Path_{\geq i}(y) \wedge \neg Path_{\geq i+1}(y).$$

Note that in both cases the formula $Path_i$ is guarded and has polynomial length. The negated atomic formulas $\neg c_i(x)$ and $\neg p_i(x)$ are simulated using predicate *zero* instead of *one*.

Now, having the ability to count, we may encode tape symbols and states of the machine by simply using more siblings, and we may follow along the lines of the construction in [Kieroński 2002] to encode the computation of M . The only remaining subtle point is that in [Kieroński 2002] the two successor configurations are siblings in a computation tree, while here they must not be siblings in order not to mess up the

encoding of numbers. This can be handled simply by rooting the two configurations at different nodes as shown in Figure 4. \square

This completes the picture for the case of signatures containing \downarrow_+ (since Theorem 3.8 shows that without the UAR restriction already $\text{GF}^2[\downarrow_+]$ is EXPSPACE-hard).

6.2. Signatures without descendant

We now consider the case of signatures containing \downarrow but not containing \downarrow_+ .

The first observation is that satisfiability of $\text{GF}^2[\downarrow, \rightarrow]$ is EXPTIME-complete. Our proof of this fact will be similar in spirit to the proof of the EXPTIME-upper bound for the guarded fragment with a bounded number of variables [Grädel 1999]. In fact, the latter uses the tree-like model property of GF. The decision procedure there employs an alternating Turing machine with polynomially bounded space. In the case of two variables, it stores a single node, universally chooses a $\forall\exists$ -conjunct of the given normal form formula, guesses an appropriate witness (a child of the considered node), and proceeds to this witness. In comparison with our task, the difference is that in [Grädel 1999] witnesses of a given node are not connected by any relations, and in our case \downarrow -witnesses of a node are related by \rightarrow . Moreover, a node may be forced to have exponentially many children: this can be done by organizing a standard counter by means of n unary predicates, and requiring that (i) there is a child storing the counter value 0, (ii) each node storing a value smaller than $2^n - 1$ is related by \rightarrow to a node storing a value greater by one. Thus we cannot afford to guess all of them at once. We deal with this problem by guessing only the children which are \downarrow -witnesses of the node and then verifying by a universal subroutine that they can be appropriately linked by \rightarrow . This is sufficient, since GF^2 subformulas can only speak about guarded pairs elements, i.e., pairs connected by a binary relation (\rightarrow or \downarrow in our case). This result contrasts with the case of FO^2 , where even without any navigational predicates NEXPTIME-hardness can be shown.

Let us see some more details now.

THEOREM 6.6. *The satisfiability problem for $\text{GF}^2[\downarrow, \rightarrow]$ over trees is in EXPTIME. The satisfiability problem for $\text{GF}^2[\downarrow]$ is EXPTIME-hard, even under the UAR assumption.*

PROOF. For the upper bound we design an alternating procedure working in polynomial space. By Lemma 6.1 we can assume that the input formula φ is in guarded normal form. At each moment during its execution the procedure stores information about a node of the tree and at most polynomially many of its children. For each node n this information is similar to that used in the proof of Theorem 6.3: namely, it consists of the atomic 1-type of n and a polynomially bounded set of atomic 1-types, called the *promised types of descendants* of n . Note that this information is of polynomial size. Recall that in the proof of Theorem 6.3 the promised types of descendants were used to ensure the existence of global witnesses: i.e., elements satisfying the conjuncts of φ of the form $\exists x\psi_i(x)$, and the existence of witnesses of n related to it by \downarrow_+ . Since in the current variant we do not have \downarrow_+ in the signature, we will use the promised types only for the former purpose this time.

The procedure first guesses information about the root r and makes sure that the set of promised types of its descendants guarantees satisfaction of all conjuncts of the form $\exists x\psi_i(x)$. For each conjunct of the form $\forall x(\lambda_i(x) \Rightarrow \exists y(x\downarrow y \wedge \psi_i''(x, y)))$ we guess information about a child of r witnessing this formula. Also, for each promised type of a descendant of r we add a child realising or promising this type. Additionally, we guess information about the leftmost and the rightmost child of r . Note that in this way we guess polynomially many children. As remarked above these are not necessarily all of

the children of r (as we may enforce exponentially many children). We verify that none of the pairs consisting of r and one of its guessed children violates $\forall\forall$ conjuncts of φ with $\eta_i(x, y)$ of the form $x \downarrow y$ or $y \downarrow x$, and that r is an appropriate upper witness for each of its children. If only one child was guessed the procedure moves to this child, verifies that it requires no horizontal witnesses and proceeds analogously as in the case of the root.

If the number of guessed children is greater than one, our task is to verify that we indeed can build a horizontal chain of elements including the desired ones, without violating any constraints imposed by φ . To this end we guess the order in which all the guessed children appear on the list of all children of r . The procedure universally chooses a pair of them, n_1, n_2 such that n_1 precedes n_2 in the guessed order, and, starting from n_1 , tries to reach n_2 by running an alternating polynomial space subroutine that verifies the existence of the appropriate \rightarrow -chain of elements between the two. We omit the details of this subroutine, but note some of the tasks that are needed when we guess an element b as a successor of an element a in such a chain. We check if b is an appropriate right-sibling witness for a , if a is an appropriate left-sibling witness for b , if r is an appropriate upper witness for b , and if the pairs a, b and b, r do not violate $\forall\forall$ conjuncts of φ . Additionally the procedure makes a universal choice between continuing the horizontal path to n_2 or going down the tree (and proceeding analogously as with the root).

The procedure accepts if after making the choice to move down it turns out that the current node requires no \downarrow -witnesses and its list of promised types of descendants is empty.

Obviously the procedure works in polynomial space. From its accepting run one can construct a model of φ in a natural way. Conversely, if a model of φ exists then there also exists a model t of depth and degree exponentially bounded in the size of φ , as guaranteed by Theorem 3.3. An accepting run of our procedure can be constructed by making all the guesses in accordance with t .

The matching lower bound for $\text{GF}^2[\downarrow]$ can be shown by an encoding of an alternating Turing machine working in polynomial space. As we wish to show the bound in the presence of the UAR restriction, we represent a configuration of the machine by $p(n)$ elements, for an appropriate polynomial p , each of which encodes a single tape cell, rather than representing it using just a single element. The intended shape of models is illustrated in Fig. 5.

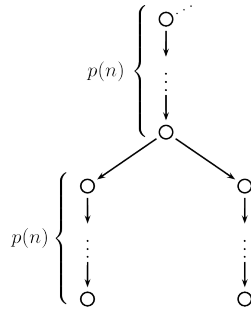


Fig. 5. A configuration and its two successor configurations.

The details of the encoding are routine. Note in particular that the distance between an element representing a tape cell in a configuration and the elements representing the same cell in the consecutive configurations is precisely $p(n)$. Therefore it is easy

to write a formula of polynomial size that, by reusing variables x and y , describes the relationship of elements $p(n)$ steps apart and thus describes dependencies between cells in consecutive configurations. \square

Finally, let us note that equipping the logic with \rightarrow^+ allows us to lift the lower bound, even assuming UAR:

THEOREM 6.7. *The satisfiability problem for $\text{GF}^2[\downarrow, \rightarrow^+]$ over UAR trees is NEXPTIME-hard.*

PROOF. Recall that without the UAR assumption the satisfiability problem for FO^2 is NEXPTIME-hard even if only unary relations are allowed in the signature [Etesami et al. 2002]. This can be easily translated to our setting. We use the children of the root to encode the elements in a model of such a unary formula. Then the relation \rightarrow^+ may be used as a guard, allowing to refer to any pair of these. The combination of unary predicates holding at a given position can be simulated by means of the \downarrow -successors. We omit the details of this reduction, as it is similar to the construction from the proof of Theorem 5.2. \square

Recall that an upper bound matching the lower bound from Theorem 6.7 holds for $\text{FO}^2[\downarrow, \rightarrow, \rightarrow^+]$ even without the UAR assumption (Theorem 4.1). Thus our analysis of the guarded case is complete.

7. SATISFIABILITY FOR RANKED TREES

In the previous sections we worked with unranked trees. We now consider the case of k -ranked trees. Recall from Section 2 that for k -ranked trees, the number k is part of the input and written in binary form. It turns out that *all the complexity results obtained in the previous sections in the case of unranked trees transfer to this new scenario.*

Concerning the lower bounds, note that satisfiability over unranked trees reduces to satisfiability over k -ranked trees, using the exponential degree property shared by all our logics (cf. Thm. 3.3): for a given formula φ whose satisfiability over unranked trees we want to test, we just ask about its satisfiability over k_φ -ranked trees, with k_φ being exponentially bounded $|\varphi|$. Please note that this argument would not work if k were coded in unary.

It turns out that all of the upper bounds from the unranked case can be also transferred to the k -ranked case. This can be done without major problems, except in the case of $\text{FO}^2[\downarrow_+]$ over UAR trees (Theorem 5.9). This requires a new approach and is discussed in detail below. Our decision procedures in the unranked case rely on the fact that satisfiable formulas have models of small depth and degree. Recall that the proofs of these bounds worked by taking an arbitrary model of a given formula and performing some “surgery” on it. It is easy to verify that, with the exception of the construction used in the proof of Lemma 5.8, our surgery does not increase the degree of nodes, and thus applies also in the ranked case. For example, in Theorem 3.3, to bound the depth of the trees we use $t[n_1 \rightarrow n_0]$, where n_0 is an ancestor of n_1 . By replacing the subtree of n_0 with the subtree of n_1 all nodes in $t[n_1 \rightarrow n_0]$ have the same outdegree as their corresponding nodes in t .

The problematic Lemma 5.8 says that in UAR models of $\text{FO}^2[\downarrow_+]$ formulas all the paths can be shortened to polynomial length. This lemma is used in two of our upper bounds in the case of unranked trees: the above mentioned Theorem 5.9 claiming that $\text{FO}^2[\downarrow_+]$ is in NEXPTIME and Theorem 6.3, concerning PSPACE bound for $\text{GF}^2[\downarrow_+]$. For $\text{FO}^2[\downarrow_+]$ Lemma 5.8 is not true in the ranked case and this is why this case requires a new analysis. However, when we restrict the language to $\text{GF}^2[\downarrow_+]$ the polynomial paths

property over ranked UAR trees can be saved. Recall that in the case of $\text{FO}^2[\downarrow_+]$ when we wanted to delete a fragment of a path, we deleted only the elements from this path but we protected their subtrees by “promoting” them in the tree. This was the step in which the degree of a node could increase. The elements from such promoted subtrees could, however, be necessary only as $\not\sim$ witnesses, which are not relevant in the case of GF^2 . Thus in this case we can simplify the proof of Lemma 5.8: when we delete a fragment of a path, we can delete it together with its subtrees. This new simplified proof works for the ranked case.

Before turning to the remaining result, Theorem 5.9, we describe an additional modification needed for the ranked case in several arguments. Recall that in several of the arguments for the unranked case, after showing a small model property we define procedures that guess models, e.g., in the proof of Theorem 3.6. For ranked trees these procedures must additionally ensure that in the guessed model nodes have bounded outdegree. In Theorem 3.6 this can be done by keeping an additional counter that keeps track of the node degree. In Theorem 5.9 we guess the whole model at once so we can verify the degree of nodes in polynomial time. In the remaining cases this additional verification is also routine.

It remains to analyse the above-mentioned interesting case of $\text{FO}^2[\downarrow_+]$ under the UAR assumption.

7.1. NEXPTIME-completeness of $\text{FO}^2[\downarrow_+]$ over UAR k -ranked trees

The proof of Theorem 5.9 used the collapse argument in Lemma 5.8, which relied heavily on the fact that trees were unranked. Indeed, we can show that over ranked trees, a $\text{FO}^2[\downarrow_+]$ formula satisfiable over UAR trees need *not* have a model of polynomial depth:

PROPOSITION 7.1. *There are $\text{FO}^2[\downarrow_+]$ formulas φ_n of polynomial size in n that are satisfiable over UAR binary trees, where the minimum depth of satisfying UAR binary trees grows as 2^n .*

PROOF. Let τ_{un} consist of $\{b, s\} \cup \{a_i : i \leq n\}$.

We consider trees in which:

- the root is labelled b ;
- nodes labelled b are always comparable via descendant;
- nodes labelled s are never comparable via descendant;
- every ancestor of a b -labelled node is labelled b ;
- every ancestor of an s -labelled node is labelled b ;
- descendants of s -labelled nodes can be labelled with any of the a_i (but not with b).

These conditions can easily be enforced by an $\text{FO}^2[\downarrow_+]$ formula.

In such trees the b -labelled nodes must go down a single branch, with s -labelled nodes splitting off on a separate branch. See Figure 6. We now let $\psi_i : i \leq n$ be the formula that holds at an s -labelled node if it has a descendant a_i . Note that any combination of the ψ_i are consistent, and the set of ψ_i that hold at an s -labelled node can thus be considered an n -bit address for the s -node. We can write a formula φ_n that asserts that:

- (1) the constraint on the shape of the tree above holds;
- (2) there is an s -node with address 0^n ;
- (3) for every s -labelled node with address a not equal to 1^n , there is an s -labelled node whose bit address is the successor of a .

A binary tree satisfying φ_n must have exponential depth. See Figure 6 for an example. \square

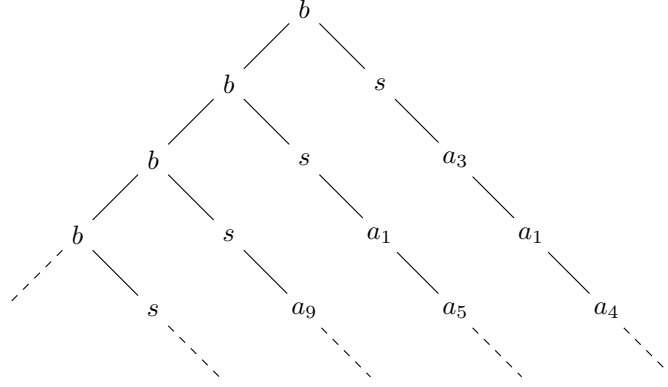


Fig. 6. An example model of exponential depth for $\text{FO}^2[\downarrow_+]$ formula in the ranked case

Nevertheless, we can still obtain a NEXPTIME upper bound for UAR trees of a given rank, and, since the proof of the matching lower bound in Theorem 5.2 works already for binary trees we will have:

THEOREM 7.2. *The satisfiability problem for $\text{FO}^2[\downarrow_+]$ over k -ranked UAR trees is NEXPTIME-complete.*

This proof will also serve as an alternative proof of Theorem 5.9. The idea of the proof is similar to the proof of Theorem 4.1. Namely, we are going to represent a model as an exponential-sized DAG. In our approach we are going to employ τ_{bin} -normal form (for $\tau_{bin} = \{\downarrow_+\}$), introduced before Lemma 3.2. This may seem problematic at first glance, since for a given formula φ and its model t , the corresponding τ_{bin} -normal form formula φ' and model t' from Lemma 3.2 make use of an extended signature; in particular t' need not obey the UAR. It turns out, however, that it will be possible to adapt the argument with the help of the following observation.

LEMMA 7.3. *Let φ be an $\text{FO}^2[\downarrow_+]$ formula, and let $t \models \varphi$ be some k -ranked UAR model of φ . Let φ' be a τ_{bin} -normal form formula computed as in the proof of Lemma 3.2, and t' an expansion of t guaranteed by the same lemma, obtained by interpreting appropriately the fresh unary predicates. Then the number of atomic 1-types realised on any root-to-leaf path in t' is bounded polynomially in $|\varphi|$.*

PROOF. Let τ_{un} be the set of unary symbols appearing in φ , let φ^* be the intermediate Scott-normal form formula computed as in Lemma 3.1, τ_{un}^* the set of its unary symbols and $t^* \models \varphi^*$ the appropriate expansion of t , and let τ'_{un} be the set of unary symbols of φ' . Fix any root-to-leaf path p in t' . It is also a path in t and t^* . By the construction from the proof of Lemma 3.1 any unary symbol p_ψ corresponds to a subformula of φ of the form $\exists y \psi'(x, y)$. That is, we have $t^*, m \models p_\psi$ iff $t, m \models \exists y \psi'$. Thus, by Theorem 5.3, for any letter $a \in \tau_{un}$, when we move from the root of p to its leaf visiting only positions carrying a , the truth value of p_ψ may change at most polynomially many times. It follows that the number of 1-types computed over τ_{bin}^* on the path p is bounded polynomially.

Consider now any predicate $p_{i,\eta}$ from τ' . Recall that by the proof of Lemma 3.2 it corresponds to a subformula of φ^* of the form $\exists y(\eta(x, y) \wedge \psi_i(x, y))$. Fix any 1-type α^* (computed over τ_{un}^*) realised on p and observe that for any admissible η — that is $\eta \in \{x \downarrow_+ y, y \downarrow_+ x, x \not\downarrow_+ y\}$ — the truth value of $p_{i,\eta}$ may change at most once when we move from the root of p to its leaf visiting only the positions of type α^* . Indeed, if,

e.g., $p_{i,x \not\sim y}$ is true at some position m of type α then it must be also true at any of its descendant n of type α , since for any witness w satisfying $t', m, w \models x \not\sim y \wedge \psi_i$ we also have $t', n, w \models x \not\sim y \wedge \psi_i$. Similar arguments work for $p_{i,x \downarrow y}$ and $p_{i,y \downarrow x}$. Since the number of $p_{i,\eta}$ predicates is bounded linearly it follows that the total number of 1-types computed over τ'_{un} , realized on p is still bounded polynomially in φ . \square

For now let us ignore the UAR assumption; we have need of it only to use Lemma 7.3. Let us consider any normal form $\text{FO}^2[\downarrow_+]$ formula φ and its model t of exponential depth such that on each of its root-to-leaf paths at most polynomially many atomic 1-types are realised. We are going to perform some surgery on t preserving φ , involving collapsing some paths and replacing some subtrees by some other subtrees. To identify “similar” nodes we can not afford to abstract a node by the set of *all* the 1-types realised below it, since within the tree as a whole there can be doubly-exponentially many such sets. Instead we will approximate the information about descendants and in addition we employ some “global information” about the tree, in the form of a set of “protected witnesses”, which we denote W . We define W in two steps, first defining the sets W_1 and W_2 .

We say that an atomic 1-type α is *linear* in t if all of its realizations are contained in a single path of t . Call the deepest element of a linear type *royal* and denote it by w_α^0 . For each non-linear 1-type α choose two of its realizations with maximal depth and call them w_α^1 and w_α^2 . We include all w_α^i and all of their ancestors in a set W_1 , and call these *basic global witnesses*.

For all $m \in W_1$, and any conjunct of φ of the form $\forall x (\lambda_i(x) \Rightarrow \exists y (x \not\sim y \wedge \psi_i(x, y)))$, if $t, m \models \lambda_i$ choose a witness w satisfying $t, m, w \models m \not\sim w \wedge \psi_i$ and add it to W_2 , along with all its ancestors — these are the *incomparable global witnesses*, as in Theorem 4.1.

We define $W = W_1 \cup W_2$. The sets W_1, W_2 are bounded exponentially because there are exponentially many 1-types and the length of the paths in the model t are also bounded exponentially. Thus the set W is of exponential size.

We need one more definition. Let m be a node of t . For every 1-type α realised by some ancestor m' of m , and for every conjunct $\forall x (\lambda_i(x) \Rightarrow \exists y (x \downarrow y \wedge \psi_i(x, y)))$ of φ if $t, m' \models \lambda_i$ and there is a descendant w of m such that $t, m', w \models \psi_i$, choose one such witness w and let $\text{SelectedDescTypes}(m)$ include the 1-type of that witness. Note that the same witness will suffice for every ancestor m' of m realising α .

Recall that in Section 4 we defined an order \prec with the following properties:

- (1) $\text{SubTree}(t, n') \prec \text{SubTree}(t, n)$ implies n' is not an ancestor of n ;
- (2) for every tree C with a distinguished leaf, and for trees t_1, t_2 with $t_1 \prec t_2$, we have $C[t_1] \prec C[t_2]$, where $C[t_i]$ is the tree obtained by replacing the distinguished leaf of C with t_i .

LEMMA 7.4. *Let φ be a normal form $\text{FO}^2[\downarrow_+]$ formula and t its model of exponential depth such that on each of its root-to-leaf paths at most polynomially many atomic 1-types are realised. Let $W, W_1, W_2, \text{AncTypes}(), \text{SelectedDescTypes}()$ are computed as above. Let n, n' be nodes of t such that*

- $n, n' \notin W$;
- n and n' have the same 1-types
- $\text{AncTypes}(n) = \text{AncTypes}(n')$;
- $\text{SelectedDescTypes}(n) = \text{SelectedDescTypes}(n')$;
- for any royal node m we have that m is an ancestor of n iff m is an ancestor of n' ;
- $\text{SubTree}(t, n') \prec \text{SubTree}(t, n)$ (which implies that n' cannot be an ancestor of n).

Then $t^* := t[n' \rightarrow n]$ is a model of φ . Note that $t^* \prec t$.

PROOF. Assume $\varphi = \forall xy \chi(x, y) \wedge \bigwedge_{i \in I} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$. Let $f : \text{SubTree}(t, n') \rightarrow t^*$ be the function taking an element to its new copy. We first show that the nodes in t^* have appropriate witnesses for all the conjuncts of φ indexed by I .

We start with the observation that nodes in t that have incomparable witnesses must have at least one appropriate witness in the set W . Consider a conjunct $\forall x (\lambda_i(x) \Rightarrow \exists y (x \not\downarrow y \wedge \psi_i(x, y)))$, and let $m \in t$ be a node such that $t, m \models \lambda_i$. Let w be a witness such that $t, m, w \models m \not\downarrow w \wedge \psi_i$. Let α be the 1-type of w , and let w_α be a basic global witness of 1-type α and maximal depth. If $m \not\downarrow w_\alpha$ then we are done. If $m \downarrow_+ w_\alpha$ or $m = w_\alpha$ then by definition $m \in W_1$ and there is an incomparable witness in W_2 . Otherwise $w_\alpha \downarrow_+ m$. Notice that α cannot be linear because if it were we would have $w \downarrow_+ m$. Since α is not linear there exists a second basic global witness $w_\alpha^2 \in W_1$ of type α of maximal depth. By definition $w_\alpha \not\downarrow w_\alpha^2$ and thus $m \not\downarrow w_\alpha^2$.

Now fix a conjunct $\forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$. We prove that this conjunct is true in t^* by case analysis depending on η_i . Let m be a node in t^* such that $t^*, m \models \lambda_i$.

- Suppose $\eta_i(x, y)$ is $x \not\downarrow y$. If m is not one of the nodes in the image of f then by the previous observation we can assume it had a witness $w \in W$ in t . Thus w is kept in t^* and it is straightforward that m and w are incomparable in t^* . Otherwise $m = f(m')$ for some $m' \in \text{SubTree}(t, n')$. Then there exists w such that $t, m, w \models x \not\downarrow y \wedge \psi_i(x, y)$. We show that m' has an incomparable witness in t^* . Let α be the 1-type of w and let w_α be a basic global witness of 1-type α . If $m \not\downarrow w_\alpha$ then we are done. Since $m \notin W$ then the case where $m \downarrow_+ w_\alpha$ is not possible. Finally, suppose that $w_\alpha \downarrow_+ m$. If α is not a linear 1-type then there is a second deepest node w_α^2 of type α , and for this node we have $m \not\downarrow w_\alpha^2$. Otherwise if α is linear then w_α is royal and thus $w, w_\alpha \in W$. Recall that the set of royal ancestors of n and n' is the same. Therefore $w_\alpha \downarrow_+ m'$ and thus $w \downarrow_+ m'$, which is a contradiction because w was chosen as an incomparable witness for m' .
- Suppose $\eta_i(x, y)$ is $y \downarrow_+ x$. If m is not a node from the image of f then its ancestors are the same as its ancestors in t and we can use the witness for m in t . Otherwise if m is in the image of f then its witness is either on the path from m to $f(n')$ or is guaranteed by $\text{AncTypes}(n) = \text{AncTypes}(n')$.
- Suppose $\eta_i(x, y)$ is $x \downarrow_+ y$. First, assume that $m = f(m')$ for some $m' \in \text{SubTree}(t, n')$. Note that an appropriate witness for m' in t exists. Call it w . Observe that w is in the image of f and that $f(w)$ can be taken as a witness for m in t^* . Now assume that m is not in the image of f . Then m is also a node in t . Let w be its descendant witness in t . If w was not removed in t^* then it is still a witness for m in t^* . If $w = n$ then, recalling that n and $f(n')$ have the same 1-types, observe that by the construction $m \downarrow_+ f(n')$, and thus $f(n')$ can serve as a witness. The remaining case is when w is a proper descendant of n . In this case we can find a new witness using the equality $\text{SelectedDescTypes}(n) = \text{SelectedDescTypes}(n')$.

Finally, consider a conjunct of the form $\forall xy \chi(x, y)$ of φ . We want to show that for all nodes $m_0, m_1 \in t^*$ we have $t^*, m_0, m_1 \models \chi(x, y)$. It suffices to show that for every pair $m_0, m_1 \in t^*$ there exists a pair $u_0, u_1 \in t$ such that w_i and u_i have the same 1-types for $i = 1, 2$, and the navigational relation between m_0 and m_1 in t^* is the same as of u_0 and u_1 in t . If m_0, m_1 are both not in the image of f then it suffices to take $u_0 = m_0, u_1 = m_1$. If they are both in the image of f then we can take $u_0 = f^{-1}(m_0)$ and $u_1 = f^{-1}(m_1)$. For the remaining case suppose that $m_0 = f(m)$ for some m and that m_1 is not in the image of f . We define $u_0 = m$ and we show that how to find u_1 . Notice that m_1 cannot be a descendant of m_0 , since then it would be in the image of f . Thus m_1 is either an ancestor of m_0 or is incomparable to m_0 . In the case where m_1 is an ancestor of m_0 we can easily find u_1 because $\text{AncTypes}(n) = \text{AncTypes}(n')$. It remains to analyze the case in

which m_0 and m_1 are incomparable. Let α be the 1-type of m_1 . Suppose α is linear. Then we define $u_1 = w_\alpha^0$, i.e., to be the royal node. Since $w_\alpha \in W$ then $w_\alpha \notin \text{SubTree}(t, n')$. We need to show that it is not possible that $w_\alpha \downarrow_+ m$. Since n and n' have the same royal nodes among their ancestors, this would imply $w_\alpha \downarrow_+ f(m)$ and thus $m_1 \downarrow_+ f(m)$, which is a contradiction. Now, suppose α is not linear. Then there are two deepest nodes $w_\alpha^1, w_\alpha^2 \in W$ of 1-type α and by assumption $w_\alpha^1, w_\alpha^2 \notin \text{SubTree}(t, n')$. Since $w_\alpha^1 \not\sim w_\alpha^2$, at least one of them is not an ancestor of m and thus is incomparable with m . \square

PROOF OF THEOREM 7.2. Let φ be any $\text{FO}^2[\downarrow_+]$ formula and t its model that is a UAR tree with exponentially bounded depth guaranteed by the proof of Theorem 3.3. Let φ' be a normal form formula and t' its model guaranteed by Lemma 3.2. By Lemma 7.3 t' realizes at most polynomially many atomic 1-types on each path thus for every node $m \in t'$ the sets $\text{SelectedDescTypes}(m)$, $\text{AncTypes}(m)$ and the set of royal nodes among ancestors of m are all of polynomial size.

We iterate the procedure in Lemma 7.4 as long as possible. This procedure terminates because $t^* \prec t$. In the resulting tree, if two nodes are not in W and do not have isomorphic subtrees, they must disagree on one of the following: their 1-types; one of the sets $\text{SelectedDescTypes}()$, $\text{AncTypes}()$; or the set of royal nodes among its ancestors. There are exponentially many 1-types and exponentially many polynomial size sets of 1-types. This gives us an exponential bound on non-isomorphic subtrees with roots not in W . Given that W has size bounded by an exponential in these quantities as well, we see that the resulting tree will have a number of isomorphism types of subtrees that is bounded exponentially. By Lemma 3.2 if we remove the auxiliary unary symbols then we get a UAR tree that is a model for φ .

Thus we have shown that if φ is satisfiable then it has an exponential-size DAG that unfolds into a model of the formula. Given such a DAG, we can check whether φ holds in polynomial time in the size of the DAG. Thus we have a NEXPTIME algorithm for checking satisfiability. \square

8. CONCLUSIONS AND ACKNOWLEDGEMENTS

The main result of the paper is that the satisfiability problem for FO^2 over finite trees, with four navigational predicates: $\downarrow, \downarrow_+, \rightarrow, \rightarrow^+$, is EXPSPACE-complete. We also consider an additional semantic restriction that at a single node precisely one unary predicate holds (UAR). Under UAR the full logic remains EXPSPACE-complete, but for some of its weakened variants this assumption makes difference. Namely, $\text{FO}^2[\downarrow_+]$ becomes NEXPTIME-complete and $\text{GF}^2[\downarrow_+]$ is even PSPACE-complete under UAR, even though both logics are still EXPSPACE-complete without UAR.

We go on to establish the precise complexity bounds for all logics $\text{GF}^2[\tau_{bin}]$ and $\text{FO}^2[\tau_{bin}]$, with $\tau_{bin} \subseteq \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$ containing at least \downarrow or \downarrow_+ , for arbitrary finite trees or under UAR.

We also consider the case of ranked trees and observe that most of the techniques applied in the unranked case transfers to the ranked case without major difficulties. In particular all the complexity results are retained. We considered binary encodings of ranks, but we believe most of our lower bounds in the ranked case are tight also for unary encodings. We were able to identify one exception: $\text{GF}^2[\downarrow, \rightarrow, \rightarrow^+]$ which is in EXPTIME under unary encodings and NEXPTIME-hard under binary encodings. However we do not discuss this issue in detail, leaving it for future work.

Another direction of future research is to extend the analysis to infinite trees. Some of the analysis already exists in [Charatonik et al. 2014]. From that work EXPSPACE-completeness of $\text{FO}^2[\downarrow_+, \downarrow]$ over infinite trees follows.

Acknowledgements. This paper is a merger of two independently-developed works, [Benaïm et al. 2013] and [Charatonik et al. 2013]. Its preliminary version appeared in *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP)*, Riga, Latvia, 2013. This work includes all the results of the conference paper, but in some cases changing the proofs to improve presentation or to correct errors in the conference version. In particular, the use of normal forms in Lemma 3.4 is a crucial correction from the conference version. The equivalence relation used in Lemma 7.4, along with the corresponding proof, have also been corrected from the conference version.

We thank the anonymous reviewers of ICALP for many helpful remarks on both works. We also thank the reviewers of this journal version for their patient readings of the submission and numerous valuable comments. Benedikt’s work was supported by the Engineering and Physical Science Research Council projects “Enforcement of Constraints on XML Streams” EP/G004021/1, and “DBOnto: Bridging Ontologies and Databases” EP/L012138/1. Charatonik was supported by Polish National Science Centre grant DEC-2011/03/B/ST6/00346. Kieroński was supported by Polish National Science Centre grant DEC-2013/09/B/ST6/01535. Mazowiecki was supported by Polish National Science Centre grants DEC-2011/03/B/ST6/00346 and DEC-2013/09/N/ST6/01170.

References

- Hajnal Andréka, Johan van Benthem, and Istvan Németi. 1998. Modal Languages and Bounded Fragments of Predicate Logic. *J. Phil. Logic* 27 (1998), 217–274.
- Saguy Benaïm, Michael Benedikt, Rastislav Lenhardt, and James Worrell. 2013. Controlling the depth, size, and number of subtrees in two variable logic over trees. *CoRR* abs/1304.6925 (2013).
- Michael Benedikt, Wenfei Fan, and Floris Geerts. 2008. XPath satisfiability in the presence of DTDs. *J. ACM* 55, 2 (2008), 8:1–8:79.
- Michael Benedikt and Christoph Koch. 2008. XPath Leashed. *ACM Comput. Surv.* 41, 1 (2008), 3:1–3:54.
- Michael Benedikt, Rastislav Lenhardt, and James Worrell. 2012. Verification of Two-Variable Logic Revisited. In *QEST*. IEEE, 114–123.
- Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. 2009. Two-variable logic on data trees and XML reasoning. *J. ACM* 56, 3 (2009).
- Witold Charatonik, Emanuel Kieroński, and Filip Mazowiecki. 2013. Satisfiability of the Two-Variable Fragment of First-Order Logic over Trees. *CoRR* abs/1304.7204 (2013).
- Witold Charatonik, Emanuel Kieroński, and Filip Mazowiecki. 2014. Decidability of Weak Logics with Deterministic Transitive Closure. In *CSL-LICS*. 29:1–29:10.
- Witold Charatonik and Piotr Witkowski. 2013. Two-variable Logic with Counting and Trees. In *LICS*. 73–82.
- Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. 2002. First-Order Logic with Two Variables and Unary Temporal Logic. *Inf. Comput.* 179, 2 (2002), 279–295.
- Diego Figueira. 2012a. Decidability of Downward XPath. *ACM Trans. Comput. Log.* 13, 4 (2012), 34.
- Diego Figueira. 2012b. Satisfiability for two-variable logic with two successor relations on finite linear orders. *CoRR* abs/1204.2495 (2012).
- Erich Grädel. 1999. On The Restraining Power of Guards. *J. Symb. Log.* 64, 4 (1999), 1719–1742.
- Erich Grädel, Phokion Kolaitis, and Moshe Vardi. 1997. On the Decision Problem for Two-Variable First-Order Logic. *Bulletin of Symbolic Logic* 3, 1 (1997), 53–69.
- Emanuel Kieroński. 2002. EXPSPACE-Complete Variant of Guarded Fragment with Transitivity. In *STACS*. 608–619.
- Emanuel Kieroński. 2005. Results on the Guarded Fragment with Equivalence or Transitive Relations. In *CSL*. 309–324.
- Maarten Marx. 2004. XPath with Conditional Axis Relations. In *EDBT*. 477–494.
- Maarten Marx and Maarten de Rijke. 2004. Semantic Characterization of Navigational XPath. In *TDM*. 73–79.
- Dana S. Scott. 1962. A Decision Method for Validity of Sentences in Two Variables. *Journal Symbolic Logic* 27 (1962), 477.

- Larry J. Stockmeyer. 1974. *The Complexity of Decision Problems in Automata Theory and Logic*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- Wiesław Szwał and Lidia Tendera. 2004. The guarded fragment with transitive guards. *Ann. Pure Appl. Logic* 128, 1-3 (2004), 227–276.
- Wiesław Szwał and Lidia Tendera. 2013. FO^2 with one transitive relation is decidable. In *STACS (LIPIcs)*, Vol. 20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 317–328.
- Tony Tan. 2014. Extending two-variable logic on data trees with order on data values and its automata. *ACM Trans. Comput. Log.* 15, 1 (2014), 8.
- Philipp Weis. 2011. *Expressiveness and Succinctness of First-Order Logic on Finite Words*. Ph.D. Dissertation. University of Massachusetts.

Received Month 2014; revised Month Year; accepted Month Year