Deciding definability in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ on trees

Thomas Place and Luc Segoufin
INRIA and ENS Cachan, France

Abstract

We prove that it is decidable whether a regular unranked tree language is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. *By* $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ *we refer to the two variable fragment of first order logic built from the descendant and following sibling predicates. In terms of expressive power it corresponds to a fragment of the navigational core of XPath that contains modalities for going up to some ancestor, down to some descendant, left to some preceding sibling, and right to some following sibling.*

We also investigate definability in some other fragments of XPath.

1 Introduction

This paper is part of a general program trying to understand the expressive power of first-order logic over trees. We say that a class of regular tree languages has a decidable characterization if the following problem is decidable: given as input a finite tree automaton, decide if the recognized language belongs to the class in question. Usually a decision algorithm requires a solid understanding of the expressive power of the corresponding class and is therefore useful in any context where a precise boundary of this expressive power is crucial. The main open problem in this area is to find a decidable characterization of the tree languages definable in $\mathrm{FO}(<_{\mathbf{v}})$, the first-order logic using a binary predicate $<_{\mathbf{v}}$ for the ancestor relation.

In this paper we work with unranked ordered trees and by $\mathrm{FO}(<_{\mathbf{h}}, <_{\mathbf{v}})$ we actually refer to the logic that has two binary predicates, one for the descendant relation, one for the following sibling relation.

We investigate an important fragment of $\mathrm{FO}(<_{\mathbf{h}}, <_{\mathbf{v}})$, its two variable restriction denoted $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. This is a robust formalism that, in term of expressive power, has an equivalent counterpart in temporal logic. This temporal counterpart can be seen as the fragment of the navigational core of XPath that does not use the successor axis [11]. More precisely, it corresponds to the temporal logic $\mathrm{EF}+\mathrm{F}^{-1}(\mathrm{F}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}^{-1})$ that navigates in the tree using two "vertical" modalities, one for going to some ancestor node (F^{-1}) and one for going to some descendant node (EF), and two "horizontal" modalities for going to some following sibling ($\mathrm{F}_{\mathbf{h}}$) or some preceding sibling ($\mathrm{F}_{\mathbf{h}}^{-1}$).

We provide a decidable characterization of $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$, or equivalently $\mathrm{EF}+\mathrm{F}^{-1}(\mathrm{F}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}^{-1})$, over unranked ordered trees. Note that for any k, $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ can express the fact that a tree has rank k, hence our result also apply for ranked trees.

Our characterizations are expressed in term of closure properties corresponding partly to identities in the syntactic forest algebra of the language as defined by Bojańczyk and Walukiewicz [8], and partly via closure under a saturation mechanism. A forest algebra is essentially a pair of monoids, the "horizontal" monoid for forest types and the "vertical" monoid for context types together with an action of contexts over types. It was introduced in [8] and was used successfully for obtaining decidable characterizations for the classes of tree languages definable in EF+EX [7], $\mathrm{EF}+\mathrm{F}^{-1}$ [3], BC-$\Sigma_1(<_{\mathbf{v}})$ [5], $\Delta_2(<_{\mathbf{v}})$ [4].

Over words, the induced logics: $\Delta_2(<)$, $\mathrm{FO}_2(<)$ and $\mathrm{EF}+\mathrm{F}^{-1}$, have exactly the same expressive power [10, 14]. But over trees this is no longer the case. For instance $\mathrm{EF}+\mathrm{F}^{-1}$ is closed under bisimulation while the other two are not. While decidable characterizations were obtained for $\Delta_2(<_{\mathbf{v}})$ and $\mathrm{EF}+\mathrm{F}^{-1}$ [3, 4], the important case of $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ was still missing and is solved in this paper.

Over words, a regular language is definable in $\mathrm{FO}_2(<)$ iff its syntactic monoid belongs to a variety of monoids known as **DA**, a decidable property [14]. Not surprisingly our first set of identities require that the horizontal and vertical monoids of the syntactic forest algebra belong to **DA**.

Our extra property is more complex and mixes at the same time the vertical and horizontal behavior of $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. We call it *closure under saturation* and we do not know yet whether it is implied by the previous identities or not.

It is immediate from the word case that being definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ implies that the vertical and horizontal monoids of the syntactic forest algebra belong to **DA**. That closure under saturation is also necessary is proved via a classical, but tedious, Ehrenfeucht-Fraïssé game argument. The main difficulty is to show that the closure conditions are

sufficient. In order to do so, as it is standard when dealing with $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ (see e.g. [3, 4, 14]), we introduce Green-like relations for comparing elements of the syntactic algebra. However, in our case, we parametrize these relations with a set of forbidden patterns: the contexts authorized for going from one type to another type cannot use any of the forbidden pattern. We are then able to perform an induction using this set of forbidden patterns, thus refining comparison relations more and more until they become trivial.

Our proof has many similarities with the one of Bojańczyk that provides a decidable characterization for the logic $\mathrm{EF}+\mathrm{F}^{-1}$ [3] and we reuse several ideas developed this paper. However it departs from it in many essential ways. First of all the closure under bisimulation of $\mathrm{EF}+\mathrm{F}^{-1}$ was used in an essential way in order to compute a subalgebra and perform inductions on the size of the algebra. Moreover, because $\mathrm{EF}+\mathrm{F}^{-1}$ does not have horizontal navigation, Bojańczyk was able to isolate certain labels and then perform an induction on the size of the alphabet. It is the combination of the induction on the size of the alphabet and on the size of the algebra that gave an elegant proof of the correctness of the identities for $\mathrm{EF}+\mathrm{F}^{-1}$ given in [3]. Our logic $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ is no longer closed under bisimulation and we were not able to perform an induction on the algebra. Moreover because our logic has horizontal navigation, it is no longer possible to isolate the label of a node from the labels of its siblings, hence it is no longer possible to perform an induction on the alphabet. In order to overcome these problems our proof replace the inductions used in [3] by an induction on the set of forbidden patterns. This make the two proofs technically fairly different.

It turns out that our proof technique applies for various horizontal modalities. In the final section of the paper we show how to adapt the characterization obtained for $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ in order to obtain decidable characterizations for $\mathrm{EF}+\mathrm{F}^{-1}(\mathrm{X}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}, \mathrm{X}_{\mathbf{h}}^{-1}, \mathrm{F}_{\mathbf{h}}^{-1})$, $\mathrm{EF}+\mathrm{F}^{-1}(S)$ and $\mathrm{EF}+\mathrm{F}^{-1}(S_+)$, where $\mathrm{X}_{\mathbf{h}}$, $\mathrm{X}_{\mathbf{h}}^{-1}$, S and S_+ are horizontal navigational modalities moving respectively to the next sibling, previous sibling, an arbitrary sibling or an arbitrary different sibling.

Related work Our characterization is essentially given using forest algebras. There exist several other formalisms that were used for providing characterizations of logical fragments of MSO (see e.g. [2, 13, 15, 9]). It is not clear however how to use these formalisms for $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

There exists decidable characterizations of EF+EX and $\Delta_2(<_{\mathbf{v}})$ over ranked trees [12]. But, as these logics cannot express the fact that a tree is binary, these characterizations are different over ranked trees than over unranked trees. As mentioned above, we don't have this problem with $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

Organization of the paper. We first provide the necessary definitions and state our characterization in Section 2. We give the proof that our characterization for $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ is sufficient in Section 3. In Section 4 we show how the proof can be adapted for handling several other horizontal navigation modalities. We only briefly discuss the complexity of our characterization and provide additional remarks in Section 5. Due to lack of space some proofs are omitted or only sketched.

2 Preliminaries

Trees and forests. We work with finite unranked ordered trees and forests whose nodes are labeled using a finite alphabet. More formally if A is a finite alphabet then our trees and forests are generated by the following rules: For all $a \in A$, a is a tree, furthermore if $a \in A$ and s is a forest then $a(s)$ is a tree, if $t_1, \cdots, t_k$ are trees then $t_1 + \cdots + t_k$ is a forest. A set of forests is called a forest language.

We use standard terminology for forests defining nodes, ancestors, descendant, following and preceding siblings. A context is a forest with a designated leaf that has no label and no sibling and which is called *the port* of the context. This definition is not standard as usually contexts are defined without the sibling restriction for ports but it is important here to work with this non-standard definition. A context c can be composed with another context c' or with a forest s in the obvious way. The corresponding notations are respectively cc' and cs.

If x is a node of a forest then the *subtree of* x is the tree rooted at x. The *subforest at* x is the forests consisting of all the subtrees of all the children of x.

Logic. Each forest is viewed as a relational structure whose domain is its set of nodes. The signature contains a unary predicate P_a for each symbol a of A plus a binary predicate for the ancestor relation $<_{\mathbf{v}}$ and a binary predicate for the following sibling relation $<_{\mathbf{h}}$. By $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ we denote the two variable restriction of first order logic over the relational signature as described above. In terms of expressive power, $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ is equivalent to the temporal logic $\mathrm{EF}+\mathrm{F}^{-1}(\mathrm{F}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}^{-1})$ that we describe below [11]. $\mathrm{EF}+\mathrm{F}^{-1}(\mathrm{F}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}^{-1})$ is essentially the restriction of the navigational core of XPath without the CHILD, PARENT, NEXT-SIBLING and PREVIOUS-SIBLING predicates. It is defined using the following grammar:

$$\varphi :: A \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathrm{EF}\varphi \mid \mathrm{F}^{-1}\varphi \mid \mathrm{F}_{\mathbf{h}}\varphi \mid \mathrm{F}_{\mathbf{h}}^{-1}\varphi$$

We use the classical semantics for this logic which defines when a formula holds at a node x of a forest s. In particular, $\mathrm{EF}\varphi$ holds at x if there is a strict descendant of x where φ holds, $\mathrm{F}^{-1}\varphi$ holds at x if there is a strict ancestor of x

where φ holds, $F_{\mathbf{h}}\varphi$ holds at x if φ holds at some strict following sibling of x, and so on... Each closed formula φ of EF+F^{-1}($F_{\mathbf{h}}, F_{\mathbf{h}}^{-1}$) or of $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ defines a forest language L_φ: Those forests s where φ holds at the root of the first tree of s. Note that $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ is expressive enough to test whether a forest is a tree. Hence any result concerning forest languages definable in $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ also applies for tree languages definable in $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

We aim at providing a decidable characterization of regular forest languages definable in $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. We shall mostly use formulas from $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. However, the EF+$F^{-1}$($F_{\mathbf{h}}, F_{\mathbf{h}}^{-1}$) point of view will be useful when considering other horizontal modalities as in Section 4 or when making comparisons with the decision algorithm obtained for EF+F^{-1} in [3].

Antichain Composition Principle. We shall make use of the following composition lemma, essentially taken from [3]. We reuse notations from [3].

A formula of $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ with one free variable is called *antichain* if in every forest, the set of nodes where it holds forms an antichain, i.e. a set (not necessarily maximal) of nodes pairwise incomparable with respect to the descendant relation. This is a semantic property, and may not be apparent just by looking at the syntax of the formula.

We fix (i) an antichain formula φ, (ii) disjoint tree languages $L_1, \cdots, L_n$ and (iii) leaves of label $a_1, \cdots, a_n$. Given a forest s, we define the forest $s[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n]$ as follows. For each node x of s such that $s, x \models \varphi(x)$, we determine the unique i such the tree language L_i contains the subtree of x. If such an i exists, we remove the subtree of x (including x), and replace x by a leaf labeled with a_i. Since φ is antichain, this can be done simultaneously for all x. Note that the formula φ may also depend on ancestors of x, while the languages L_i only talk about the subtree of x. A simple argument, similar to the one given in [3] for EF+F^{-1}, omitted in this abstract, shows:

Lemma 2.1 *[Antichain Composition Lemma] Let φ, $L_1, \cdots, L_n$ and $a_1, \cdots, a_n$ be as above. If $L_1, \cdots, L_n$ and K are languages definable in $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$, then so is $\{t \mid t[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n] \in K\}$.*

Forest algebras. *Forest algebras* were introduced by Bojańczyk and Walukiewicz as an algebraic formalism for studying regular forest languages [8]. We work with the following variant of forest algebra: the hole of each context has no sibling and we work with semigroups instead of monoids. These restrictions are necessary as, without them, the languages definable in $FO_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ would not form a variety.

We give a brief summary of the definition of forest algebras and of their important properties. More details can be found in [8]. A forest algebra consists of a pair (H, V) of finite semigroups, subject to some additional requirements, which we describe below. We write the operation in V multiplicatively and the operation in H additively, although H is not assumed to be commutative.

We require that V acts on the left of H. That is, there is a map $(h, v) \mapsto vh \in H$ such that $w(vh) = (wv)h$ for all $h \in H$ and $v, w \in V$. We further require that for every $g \in H$ and $v \in V$, V contains elements $(v + g)$ and $(g + v)$ such that $(v + g)h = vh + g, (g + v)h = g + vh$ for all $h \in H$. The *free forest algebra*, denoted by A^Δ, is the pair of semigroups (H_A, V_A) where H_A is the set of forests over the alphabet A and V_A the set of contexts, together with the natural actions.

A morphism $\alpha : (H_1, V_1) \to (H_2, V_2)$ of forest algebras is actually a pair (γ, δ) of semigroup morphisms $\gamma : H_1 \to H_2$, $\delta : V_1 \to V_2$ such that $\gamma(vh) = \delta(v)\gamma(h)$ for all $h \in H$, $v \in V$. However, we will abuse notation slightly and denote both component maps by α. We say that a forest algebra (H, V) *recognizes* a forest language L if there is a morphism $\alpha : A^\Delta \to (H, V)$ and a subset X of H such that $L = \alpha^{-1}(X)$. We also say that the morphism α recognizes L. It is easy to show that a forest language is regular if and only if it is recognized by a finite forest algebra. Moreover, given a tree automaton, a minimal forest algebra, also called *syntactic forest algebra*, recognizing the same language can be computed.

Given any finite semigroup S, there is (folklore) a number $\omega(S)$ (denoted by ω when S is understood from the context) such that for each element x of S, x^ω is an idempotent: $x^\omega = x^\omega x^\omega$. Therefore for any forest algebra (H, V) and any element $u \in V$ and $g \in H$ we will write u^ω and ωg for the corresponding idempotents.

Horizontal behavior. As mentioned in the introduction we will constantly be working with sequences of sibling nodes. For technical reasons, we also include the label of a subtree if this one is a leaf. We now make this notion more precise. We assume fixed a language L recognized by a forest algebra (H, V) via the morphism α.

A multicontext is defined as for context but has several ports. The *arity* of a multicontext is the number of its ports. A multicontext is said to be *shallow* if each of its trees is either a single leaf a, a single node with a port below $b(\Box)$ or, a tree of the form $b(a)$ where b is a node and a a leaf. Given a multicontext c of arity n and a sequence T of n forests, $c[T]$ denotes the forest obtained after placing each tree in T at the corresponding port of c. A multicontext c *occurs* in a forest t if $t = \Delta c[T]$ for some context Δ and sequence of forest T.

As expected we will only manipulate

shallow multicontexts modulo $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ definability. Intuitively, $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ treats a shallow multicontext as a string whose letters are either a, $b(a)$, or $b(\Box)$. For each positive integer k and any two shallow multicontexts p and p', we write $p \equiv_k p'$ the fact that p and p' agree on all sentences of $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ of quantifier rank k. We denote by k-MTypes the equivalence classes of this relation.

Let P be a set of k-MTypes - this set will play the role of forbidden patterns in our proof - a forest t is said to be *P-valid* if no element of P occur in t. Similarly we define the notion of P-valid multicontext.

Consider a set $X \subseteq H$, X will later be a parameter in our induction. We define a logic $\mathrm{FO}_2^X(<_{\mathbf{h}}, <_{\mathbf{v}})$ for denoting positions on shallow multicontexts. Intuitively, $\mathrm{FO}_2^X(<_{\mathbf{h}}, <_{\mathbf{v}})$ is like $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ on shallow multicontexts but it cannot distinguish the symbol $b(\Box)$ from the symbol $b(a)$ whenever $\alpha(a) \notin X$. More formally, when it tests a label, $\mathrm{FO}_2^X(<_{\mathbf{h}}, <_{\mathbf{v}})$ can use $P_{b(a)}$ when $\alpha(a) \in X$, or $P_{b(\Box)} \vee P_{b(a)}$ when $\alpha(a) \notin X$.

Let x be a node of a tree t. Let $x_1, \cdots, x_l$ be the sequence of siblings of x, including x. Let $t_1, \cdots, t_l$ be the subtrees of t rooted at those nodes. The *shallow multicontext of t at x* is the sequence $p_1, \cdots, p_l$ such that $p_i := a$ if $t_i = a$, $p_i := b(a)$ if $t_i = b(a)$, $p_i := b(a)$ if $t_i = b(s)$ with $\alpha(s) = \alpha(a) \in X$ and, $p_i := b(\Box)$ otherwise. Given two nodes x and x' of t we write $x \cong_{k,X} x'$ if the shallow multicontext of x and the shallow multicontext of x' satisfy the same formulas of $\mathrm{FO}_2^X(<_{\mathbf{h}}, <_{\mathbf{v}})$ of quantifier depth at most k, with one free variable denoting respectively the position x and x'. We denote by (X, k)-PTypes the equivalence classes of this relation and we only consider (X, k)-PTypes such that $P_{b(\Box)}(x)$ holds for some b.

Given a (X, k)-PType δ and a k-MType τ, we say that δ is *compatible with* τ if all shallow multicontexts $p \in \tau$ contain a position $x \in \delta$.

Saturation. As before, P denotes a set of k-MTypes for some k. A type $h \in H$ is said to be P-reachable from the type h' if there exists a P-valid context u such that $h = \alpha(u)h'$. Two types are P-equivalent if they are mutually P-reachable.

In the case where all P-valid shallow multicontexts have arity 1 we will see that we are in a setting similar to the word case and we use a specific argument. In the case where there is at least one P-valid shallow multicontext of arity two we have the following important property: P-reachability contains a unique maximal P-equivalence class (see Claim 3.2 below). We then denote by H_P the unique maximal P-equivalence class and by $\bar{H}_P$ the types of H not in H_P. In this case we say that P is *good*.

Finally, we are able to define the notion of *saturation* which is part of our characterization. Intuitively a context is saturated if it is P-valid and contains one representative for each k-MType $\tau \notin P$ and compatible $(\bar{H}_P, k)$-PType. More formally, let P be a good set of k-MTypes. A context Δ is said to be *P-saturated* if (i) it is P-valid and (ii) for each P-valid k-MType τ, and each each compatible $(\bar{H}_P, k)$-PType δ, there exists a node x occurring in Δ on the path from the root of Δ to its port such that $x \in \delta$ and the shallow multicontext of Δ at x is in τ.

We say that a tree language L is *closed under k-saturation* if for all good set P of k-MTypes, for all context Δ that is P-saturated, for all P-valid tree t, for all P-valid shallow multicontext p, for all position x of p and for all sequence T of P-valid forests whose types are in H_P, we have:

$$\alpha(\Delta)^\omega \alpha(t) = \alpha(\Delta)^\omega \alpha(p[T, x])\alpha(\Delta)^\omega \alpha(t) \qquad (1)$$

were $p[T, x]$ is the context formed from p by placing the forests of T at the corresponding holes of p except for the hole at position x. A language is closed under saturation if it is closed under k-saturation for some k.

The main result.

Theorem 2.2 *A regular forest language L recognized by the forest algebra (H, V) is definable in* $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ *iff*

a) H verifies the equation

$$\omega(f + g + h) + g + \omega(f + g + h) = \omega(f + g + h) \qquad (2)$$

b) V verifies the equation

$$(uvw)^\omega v(uvw)^\omega = (uvw)^\omega \qquad (3)$$

c) L is closed under saturation.

It turns out that (2) and (3) above are exactly the identities characterizing membership in the variety of semigroups known as **DA** [14]. Hence (2) and (3) could be equivalently rephrased as $H \in \mathbf{DA}$ and $V \in \mathbf{DA}$.

Recall that $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ can express the fact that a forest is a tree and, for each k, that a tree has rank k, hence Theorem 2.2 also apply for regular ranked tree languages.

It is simple to see that Equations (2) and (3) are necessary. That saturation is necessary is proved using a classical, but tedious, Ehrenfeucht-Fraïssé argument whose proof is omitted in this abstract:

Lemma 2.3 *A forest language definable in* $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ *is closed under saturation.*

The most difficult part of the proof of Theorem 2.2 is to show that the conditions imply definability in the corresponding logic. Section 3 is devoted to the proof of this implication. In Section 4 we discuss how the argument can be modified in order to cope with other horizontal modalities.

3 Sufficientness of the properties

In all this section we fix a regular forest language L that is recognized by the forest algebra (H, V) via the morphism α. We assume that L is closed under k'-saturation and that H and V verify Equations (2) and (3). We will show that L is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

Let k'' be the number such that whenever p and p' have the same k''-MType then for all forest s we have $\alpha(p[\bar{s}]) = \alpha(p'[\bar{s}])$, where $p[\bar{s}]$ is the forest constructed from p by placing s at each hole of p. Such a k'' exists because we are essentially in the string case and (2) guarantees definability in $\mathrm{FO}_2(<)$ in the string case as proved in [14], and taking k'' as the quantifier rank of the resulting formula yields the desired result. We omit the details in this abstract.

We now take k as the maximum between k' and k''. Notice that L remains closed under k-saturation.

Given a forest s, its *type* is its image in H by α. We assume that for each type $h \in H$ there is a tree consisting of a single leaf node that has h for type via α. This simplifies the notations in the proof with no harm in the generality of the result.

The proof of Theorem 2.2 is done by induction using an inductive hypothesis that is stated in the proposition below. One of the parameters is a subset X of H. The following definition is taken from [3]. A forest s is said to be *X-trimmed* if the only nodes of s that are of type in X are leaves. We say that a forest language L is *definable modulo* X if there is a definable forest language L' that agrees with L over X-trimmed forests. For each $h \in H$ and $v \in V$, let $L^P_{v,h} = \{t \mid v \cdot \alpha(t) = h \text{ and } t \text{ is } P\text{-valid}\}$.

Our goal in this section is to show that:

Proposition 3.1 *$\forall h \in H, v \in V$ and $X \subseteq H$, and P a set of k-MTypes, $L^P_{v,h}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X.*

We can then complete the proof of Theorem 2.2 by applying Proposition 3.1 for all $h \in \alpha(L)$ with v the empty context, and P, X empty sets.

In the rest of this section we only care about P-valid forests and hence we implicitly ignore the types $h \in H$ such that $\alpha^{-1}(h)$ contains no P-valid forests.

Recall the notion of P-reachability for two types f and g of H. Similarly given two contexts $u, v \in V$ we say that v is *P-reachable* from u whenever there is a context c which is P-valid such that $v = u \cdot \alpha(c)$. The *P-depth* of v is then the distance relative to P-reachability between v and the empty context.

We now define an order on sets of k-MTypes. For each k-MType τ, its *X-number* is the number of (X, k)-PTypes compatible with τ. For each set P of k-MTypes the *n-index of P* is the number of k-MTypes of P of X-number n. The *index of P* is then the sequence of its n-indexes ordered by decreasing n. We write $P_1 < P_2$ if the index of P_1 is strictly smaller than the index of P_2 (notice that the notion of index depends on X).

In the rest of this section we prove Proposition 3.1 by induction on the three following parameters, given below in their order of importance:

- $|X|$
- the index of P
- the P-depth of v

We consider three main cases: In the first case we suppose that all shallow multicontexts that are not in P have arity 0 or 1. In this case we show that we can treat our forests as words and Proposition 3.1 follows from known results over words. The reason why we distinguish this case is that when we have at least one shallow multicontext of arity at least 2 outside of P then P-reachability for forests contains a unique maximal class as the following claim shows:

Claim 3.2 *If there is a shallow multicontext of arity at least 2 outside of P then there is a unique maximal class regarding P-reachability.*

Proof. Take p outside of P and of arity $n \geq 2$. Given $h, h' \in H$, consider t and t' be two P-valid trees such that $\alpha(t) = h$ and $\alpha(t') = h'$. Consider the ordered set T of n P-valid forests containing copies of t and t', with at least one copy of t and one copy of t'. Now $\alpha(pT)$ is P-reachable from both h and h'. □

Therefore as soon as we are not in the first case we denote by H_P the unique maximal class relative to P-reachability as guaranteed by Claim 3.2. Our second case assumes that there exists a P-valid forest whose type is neither in X nor in H_P. In this case we can conclude by induction by either adding types in X or a forbidden pattern in P, hence increasing its index. In the remaining case, $H \setminus X$ is reduced to H_P on P-valid forests. We then show that we can increase the index of P, or increase the P-depth of v or make use of closure under saturation of L to show that v must be constant and hence $L^P_{v,h}$ is trivially definable.

3.1 Case 1: All k-MTypes outside of P have arity 0 or 1

We show that in this case we can treat our forests as words and use the known results on words. Any P-valid forest t that is not a collection of trees is of the form:

$$c_1 \cdots c_k s$$

where the $c_1, \cdots, c_k$ are P-valid shallow multicontexts of arity 1 and s a P-valid shallow multicontext of arity 0. For each $u \in V$ and $g \in H$, consider the languages:

$$M_{u,g} = \{t \mid t = c_1 \cdots c_k s \text{ is } P\text{-valid}, \alpha(c_1 ... c_k) = u, \text{ and } \alpha(s) = g\}$$

Notice that $L^P_{v,h}$ is the union of those languages where $vug = h$. We show that for any u and g, $M_{u,g}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X. This will conclude this case.

Let $\{\tau_1, ..., \tau_n\}$ be the set of k-MTypes not in P of arity 1. As H is in **DA**, all contexts of type τ_i have the same image in V by α. Let $\{v_1, ..., v_n\}$ be those types. Let $\Gamma = \{d_1, ..., d_n\}$ be a word alphabet and define a morphism $\beta : \Gamma^* \to V$ by $\beta(d_i) = v_i$.

Since V is also in **DA**, for each $v \in V$ there is a $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ formula φ_v such that the words of Γ^* satisfying φ_v are the words of type v under β [14, 10]. From each such formula φ_v we construct an $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ formula Ψ_v by replacing each symbol d_i with a formula that tests if the k-MType at the current position is τ_i (recall that this is expressible in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$). Since we can also easily express in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ that $\alpha(s) = g$, by putting all this together we get that $M_{u,g}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X.

This proves Proposition 3.1 for this case. In the rest of this section we assume the existence of a k-MType of arity 2 outside of P and therefore, by Claim 3.2, the existence of a unique maximal P-reachable class H_P.

3.2 Case 2: There exists a P-valid forest whose type is neither in X nor in H_P.

Let t be such a P-valid tree. Fix G as a class of mutually P-reachable types such that the type of t is reachable from any type of G, $G \not\subseteq X$, and G is P-minimal with the previous two properties. In other words G is just above X according to P-reachability, and is not in H_P by hypothesis.

Our agenda for this case is as follows. First, we show that being a forest whose type is in G can be detected in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ as it only depends on the presence or absence of certain k-MTypes. Note that our hypothesis then guarantees that there exists at least one k-MType whose presence forces that the corresponding forest has a type outside G.

Then, intuitively, we can add G in X and use our induction hypothesis in order to get an $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ formula describing the part of the tree which is above all subtrees of type in G. We can also add to P the k-MTypes that are forbidden for having a type in G and use again our induction hypothesis in order to get an $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ formula giving the precise type in G of a forest in G. We then conclude using the antichain composition principle, see Figure 1.

We first show that membership in G can be detected in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

Lemma 3.3 *There is a formula $\varphi(x) \in \mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ such that for any P-valid and X-trimmed tree t the set of nodes x such that the subtree of x or the subforest at x has type in G is exactly the set of nodes at which φ holds.*

Proof. This lemma is proved using membership of H and V in **DA**. We show that a subforest has a type in G iff it does not contain certain k-MTypes. Since we can detect those forbidden k-MTypes using a formula of $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$, the result will follow. The proof relies on the following claim:

Claim 3.4 *Take a shallow multicontext p of arity n and take two sequences T and T' of n P-valid forests of type in G. We have:*

$$\alpha(p[T]) \in G \;\Leftrightarrow\; \alpha(p[T']) \in G$$

Proof. We use Equation (3) to prove this claim. We write $T = \{t_1, ..., t_n\}$ and $T' = \{t'_1, ..., t'_n\}$. For $i \in [1, n]$ we write c_i the context obtained from $p[T']$ by replacing t'_i by a hole and t'_j by t_j for $j > i$. Notice that by hypothesis on p, T and T', c_i is P-valid. We write $u_i = \alpha(c_i)$, and show that:

$$u_i\alpha(t_i) \in G \;\Leftrightarrow\; u_i\alpha(t'_i) \in G \tag{4}$$

Let $g = \alpha(t_i)$ and $g' = \alpha(t'_i)$ and suppose that $u_i g \in G$, we show that $u_i g' \in G$. By symmetry this will prove (4). By definition we always have $u_i g'$ P-reachable from g'. Therefore it remains to show that g' is P-reachable from $u_i g'$. From $u_i g \in G$ we get that g' is P-reachable from $u_i g$. As g and g' are both in G they are mutually P-reachable. Therefore we have two P-valid contexts c and c' such that $g' = \alpha(c)u_i g$ and $g = \alpha(c')g'$. A little bit of algebra and Equation (3) yields:

$$\begin{aligned}
&g' = \alpha(c)u_i\alpha(c')g' \\
&g' = (\alpha(c)u_i\alpha(c'))^\omega g' \\
&g' = (\alpha(c)u_i\alpha(c'))^\omega u_i(\alpha(c)u_i\alpha(c'))^\omega g' \quad \text{using Equation (3)} \\
&g' = (\alpha(c)u_i\alpha(c'))^\omega u_i g' \\
&g' = (\alpha(cc_ic'))^\omega u_i g'
\end{aligned}$$

as cc_ic' is P-valid, g' is P-reachable from $u_i g'$ and (4) is proved.

For concluding the proof of the claim, notice that by construction $\alpha(p[T]) = u_1\alpha(t_1)$. From Equation (4) we obtain $u_i\alpha(t_i) \in G$ iff $u_i\alpha(t'_i) \in G$. Notice that $u_i\alpha(t'_i) = u_{i+1}\alpha(t'_{i+1})$. Now, again from Equation (4), we have $u_{i+1}\alpha(t_{i+1}) \in G$ iff $u_{i+1}\alpha(t'_{i+1}) \in G$. Altogether this gives $u_i\alpha(t_i) \in G$ iff $u_{i+1}\alpha(t_{i+1}) \in G$. Finally by construction we also have $u_n\alpha(t'_n) = \alpha(p[T'])$. By putting all this together we obtain $\alpha(p[T]) \in G$ iff $\alpha(p[T']) \in G$ as desired. □

A shallow multicontext p of arity n is said to be H-good if for some sequence T of n P-valid forests of type in G we have $\alpha(p[T]) \in G$. From the previous claim we know that this definition does not depend on the choice of T. A shallow multicontext p that is not H-good is said to be H-bad. It turns out that this distinction between good and bad shallow multicontexts characterizes membership in G.

Claim 3.5 *Let t be a P-valid X-trimmed forest. Then we have $\alpha(t) \in G$ iff t contains only H-good shallow multicontexts.*

Proof. Suppose that $\alpha(t) \notin G$, we show that t contains an H-bad shallow multicontext. Let s be a subforest of t such that $\alpha(s) \notin G$ and $s = p[T]$ where p is a shallow multicontext and T a sequence of forests of type in G (possibly empty if p is of arity 0). The existence of such a subforest s is ensured by the fact that $\alpha(t) \notin G$, that G is P-minimal and that t is X-trimmed. By Claim 3.4 p is an H-bad shallow multicontext and it is contained in t. □

It follows from this claim that in order to check whether a subforest is of type in G, it is sufficient to check whether it contains an H-bad shallow multicontexts or not. It remains to show that this can be expressed in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. For this we show that the set of H-good shallow multicontexts is a union of k-MTypes.

Claim 3.6 *Let p and p' be two shallow multicontexts of the same k-MType. Then we have p is H-good iff p' is H-good.*

Proof. Suppose that p is H-good and of arity n. We show that p' is H-good. Let n' be the arity of p', by Claim 3.4 it is sufficient to prove that there exists a sequence of n' forests T' of type in G such that $\alpha(p'[T']) \in G$. Let t be a forest such that $\alpha(t) \in G$ and T be the sequence of n copies of t and T' the sequence of n' copies of t. As $p \equiv_k p'$, because $k \geq k''$, we get $\alpha(p'[T']) = \alpha(p[T])$. Since p is H-good, $\alpha(p[T]) \in G$, therefore $\alpha(p'[T']) \in G$. □

This last claim concluded the proof of Lemma 3.3. □

We now aim at applying Lemma 2.1, the antichain formula being essentially the one given by Lemma 3.3. The next two lemmas show that the appropriate languages are definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

Lemma 3.7 *$L^P_{v,h}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo $X \cup G$.*

Proof. By induction on $|X|$ in Proposition 3.1 we get that $L^{\emptyset}_{v,h}$ is definable modulo $X \cup G$. But as the language of P-valid forests is definable modulo X it is also definable modulo $X \cup G$. By combining the two we get that $L^P_{v,h}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo $X \cup G$. □

Lemma 3.8 *For any $g \in G$, $L^P_{v,g}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X.*

Proof. Let P' be the set of H-bad k-MTypes described in the proof of Lemma 3.3. Because G is not H_P, there exists at least a H-bad k-MType and hence P' is not empty. We also know from the proof of Lemma 3.3 that forests that have a type in G do not contain any k-MTypes in P'. Therefore for any $g \in G$, $L^P_{v,g} = L^{(P \cup P')}_{v,g}$. Notice that the index of $P \cup P'$ is strictly higher than the index of P. Hence, by induction on the index of P in Proposition 3.1, $L^{(P \cup P')}_{v,g}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. □

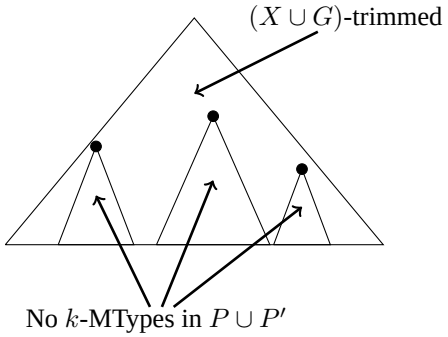


Figure 1. Illustration of the Antichain Composition Lemma for Case 2. The marked nodes are the topmost nodes in G.

We are now ready to give the final argument which is depicted in Figure 1. Let φ be the formula which holds at a node x of a tree t iff x is in L_G and there is no node between the root of t and x in L_G. From Lemma 3.3, φ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ and by definition it is an antichain formula. By Lemma 3.7, there exists a language K definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ that agrees with $L^P_{v,h}$ on $(X \cup G)$-trimmed forests. Assume $G = \{g_1, \cdots, g_l\}$. For any $i \leq l$, let a_i be a leaf node such that $\alpha(a_i) = g_i$. By Lemma 3.8 for any $i \leq l$, there exists a language L_i definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ that agrees with L^P_{v,g_i} over X-trimmed forests. Hence from the Antichain Composition Lemma, Lemma 2.1, we have that $K' = \{t \mid t[(L_1, \varphi) \to a_1, \cdots, (L_k, \varphi) \to a_k] \in K\}$ is also definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$. By definition of K and the L_i, K' agrees with $L^P_{v,h}$ on X-trimmed and hence $L^P_{v,h}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X. This concludes the proof of Proposition 3.1 for this case.

3.3 Case 3: $H \setminus X$ is reduced to H_P on P-valid forests

An element $u \in V$ *P-preserves* v if v is P-reachable from vu. A context c *P-preserves* v if $\alpha(c)$ preserves v.

We then distinguish two subcases. In the first subcase we assume that there is a k-MType τ not in P and a compatible (X, k)-PType δ such that no matter what forests we place in the shallow multicontexts of τ, leaving a hole at a position in δ, the resulting context does not *P-preserve* v. In this subcase, we conclude using the composition principle lemma, after increasing the index of P for showing definability of one piece and increasing the P-depth of v for showing definability of the other pieces.

In the remaining subcase, we will use closure under saturation to conclude that $L^P_{v,h}$ is trivial.

Formally, we say that a k-MType τ is *P-bad for v* if $\tau \notin P$ and there exist a compatible (X,k)-PType δ such that for any shallow multicontext $p \in \tau$ and any position x of p in δ, all the contexts $p[T,x]$ do not P-preserve v.

We distinguish two subcases.

Subcase 1: There exists a k-MType τ which is P-bad for v.

We fix a $\tau \notin P$ of maximal X-number that is P-bad for a (X,k)-PType δ. Let $p \in \tau$ and x a position in p of type δ.

The following lemma is immediate from the definitions.

Lemma 3.9 *There is a formula $\varphi(y) \in \mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ that holds on any P-valid X-trimmed tree t at exactly the nodes y of (X,k)-PType δ and such that the shallow multicontext of t at y is in τ.*

Given two elements h and h' of H, we say that h is v^+-equivalent to h' if for all u P-reachable from v such that v is not P-reachable from u (hence the P-depth of u is strictly higher that the P-depth of v) we have $uh = uh'$.

Lemma 3.10 *Each v^+-equivalence class is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X.*

Proof. This is immediate by induction on the P-depth of v in Proposition 3.1. □

Intuitively, we want to approximate the subtree below a v-bad position by its v^+-equivalence class. When doing this we may reintroduce shallow multicontexts that were forbidden by P. But fortunately the index of P will increase when doing so.

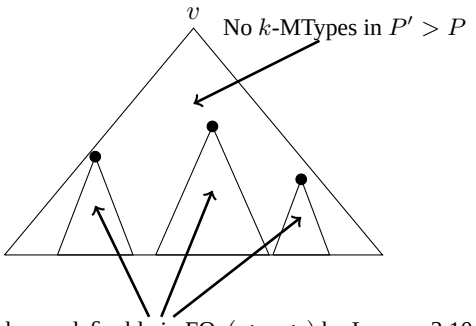
Let $p \in \tau$. Let $x_1, \cdots, x_l$ be all the positions of p of (X,k)-PType δ. Let $b(\Box)$ be the label of all the x_i in p. Let P^+ be the set of all the shallow multicontexts constructed from p by replacing at all the positions x_i, $b(\Box)$ by $b(a_i)$, for some arbitrary choice of $a_i \notin X$. Let Δ be the set of k-MTypes τ' of all the shallow multicontexts in P^+. Let P' be $(P \cup \{\tau\}) \setminus \Delta$.

Lemma 3.11 *The set $L^{P'}_{v,h}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ modulo X.*

Proof. We show that $P' > P$, the result follows by induction on the index of P in Proposition 3.1.

More precisely, we show that any $\tau' \in \Delta$ is of X-number strictly smaller than the X-number of τ. This gives the desired result.

By definition of Δ there exists $p \in \tau$ and $p' \in \tau'$ such that p' can be obtained from p by replacing symbols some $b(\Box)$ with subtrees of the form $b(a)$ with $\alpha(a) \notin X$. Consider a position x' of p' of label $b'(\Box)$. By construction the corresponding position x of p has the same label. By the definition of the logic used for defining (X,k)-PTypes, x and x' must have the same (X,k)-PType. Hence any (X,k)-PType compatible with τ' is also compatible with τ. Moreover, by construction of Δ, δ is no longer compatible with τ'. Has τ had a maximal X-number, $P' > P$. □



v^+-equivalence classes definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ by Lemma 3.10

Figure 2. Illustration of the Antichain Composition Lemma for Subcase 1. The marked nodes are the topmost nodes of type τ.

Based on the above lemmas, we conclude this case of Proposition 3.1 as follows. Consider the property that holds at a node y of a tree t if the k-MType of the shallow multicontext at y is in τ and its (X,k)-PType in δ and there is no node between the root of t and y satisfying this property. By Lemma 3.9 this property is expressible by a formula $\varphi(y)$ of $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ and it is antichain by definition. We also know that each such position y has the same label, say b.

Let $\gamma_1, \cdots, \gamma_k$ be all the equivalence classes of the v-equivalence relation. For each such class γ_i, consider the set of trees $\{b \cdot t \mid t \in \gamma_i\}$. Thanks to Lemma 3.10, for each such set there exists L_i definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ that agrees with it on X-trimmed trees. For any $i = 1, \cdots, k$, let h_i be an arbitrarily chosen forest type in the class γ_i, and let a_i be a leaf label whose type is h_i.

By Lemma 3.11, there exists K definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$ that agrees with $L^{P'}_{v,h}$ on X-trimmed trees. Hence we can apply the Antichain Composition Lemma (see Figure 2) and have that $\{t \mid t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k] \in K\}$ is definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

We conclude by showing that $L^P_{v,h} = \{t \mid t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k] \in K\}$ over X-trimmed trees. It follows that $L^P_{v,h}$ is definable modulo X. This is a simple consequence of the following two lemmas.

Lemma 3.12 *For any P-valid X-trimmed tree t, $t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k]$ is P'-valid.*

Proof. This follows from the construction of P' and the definition of φ. □

Lemma 3.13 *For any X-trimmed tree t, $v\alpha(t) = v\alpha(t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k])$.*

Proof. The proof goes by induction on the number of occurrences of τ in t and the number of nodes y of (X,k)-PType δ in each occurrence p of τ. If there is no occurrence of τ, this is immediate as the substitution does nothing.

Consider a node y of a shallow multicontext p such that $p \in \tau$ and y is in δ and no node above y satisfies that property. Let s be the subforest below y in t and i such that $\alpha(s) \in \gamma_i$. Let c be the context formed from t by placing a hole at y. Let d the context formed from c by removing all the strict ancestors of y. By choice of y, αd does not P-preserve v. We write t' the tree constructed from t by replacing the subforest under y with the leaf a_i. By construction $t'[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k]$ is $ca_i[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k]$. By induction hypothesis we have that $v\alpha(t') = v\alpha(t'[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k])$. Therefore it remains to show that $v\alpha(t') = v\alpha(cs)$. We first claim that v is not P-reachable from $v\alpha(cd)$. This is a consequence of Equation (3), suppose that v is P-reachable from $v\alpha(cd)$, then there exists a P-valid u such that $v = v\alpha(cd)u$. From there we get the following sequence of equalities:

$$\begin{aligned} v &= v\alpha(cd)u \\ v &= v(\alpha(c)\alpha(d)u)^\omega \\ v &= v(\alpha(c)\alpha(d)u)^\omega\alpha(d)(\alpha(c)\alpha(d)u)^\omega && \text{using (3)} \\ v &= v\alpha(d)(\alpha(c)\alpha(d)u)^\omega \end{aligned}$$

This implies that $\alpha(d)$ P-preserves v, which we know to be false. Let then $u = v\alpha(cd)$. From the above v is not P-reachable from u but, as t is P-valid, u is P-reachable from v. Hence $u\alpha(s) = u\alpha(a_i)$ by definition of v^+-equivalence. This implies the desired result. □

Subcase 2: There is no k-MType τ which is P-bad for v.

Using closure under saturation, we show that in this case, v is P-preserved by a context that is constant over P-valid trees. This implies that $L^P_{v,h}$ contains no P-valid trees or all of them and is therefore definable in $\mathrm{FO}_2(<_{\mathbf{h}}, <_{\mathbf{v}})$.

By hypothesis, for each $\tau \notin P$ and each compatible (X,k)-PType δ, there exists a shallow multicontext $p \in \tau$ and a position $x \in \delta$ of p such that there exists a sequence T of P-valid X-trimmed forests such that the context $p[T,x]$, P-preserves v. For each pair (τ,δ), we fix such a context $p[T,x]$. Let Δ be the context defined as the concatenation of all those contexts. By construction, Δ^ω is P-valid and P-preserves v. By construction Δ^ω is also saturated.

Using closure under saturation we show that Δ^ω is constant on P-valid trees. Let h_1 and h_2 be two elements of H_P. We want to show that $\alpha(\Delta)^\omega h_1 = \alpha(\Delta)^\omega h_2$.

Consider a P-valid shallow multicontext p of arity at least 2, and two positions x, y of p and a sequence T of P-valid forests. Let T be an arbitrary sequence of P-valid forests with types in H_P. Let $p[T,x,y]$ be the multicontext of arity 2 constructed from p by placing the two holes in x and y and placing the forests of T for the other holes. Let $p^+[T,x]$ be the context constructed from $p[T,x,y]$ by placing $\Delta^\omega h_2$ at the hole denoted by y. Let $p^+[T,y]$ be the context constructed from $p[T,x,y]$ by placing $\Delta^\omega h_1$ at the hole denoted by x.

Then we have:

$$\begin{aligned} \alpha(\Delta)^\omega h_1 &= \alpha(\Delta)^\omega\alpha(p^+[T,x])\alpha(\Delta)^\omega h_1 && \text{using (1)} \\ &= \alpha(\Delta)^\omega\alpha(p^+[T,y])\alpha(\Delta)^\omega h_2 \\ &= \alpha(\Delta)^\omega h_2 && \text{using (1)} \end{aligned}$$

And we are done with the last case.

4 Other logics

Using the same proof structure we can obtain the decidability of several other logics that differ with EF+$\mathrm{F}^{-1}(\mathrm{F}_{\mathbf{h}}, \mathrm{F}^{-1}_{\mathbf{h}})$ only in the horizontal modalities.

We illustrate this with the predicates S_+, S, $\mathrm{X}_{\mathbf{h}}$ and $\mathrm{X}^{-1}_{\mathbf{h}}$ but we believe that other modalities could be considered, assuming the induced logic over words has a decidable characterization.

The predicate $\mathrm{S}_+\varphi$ holds at x if φ holds at some sibling of x (it is a shorthand for $\mathrm{F}_{\mathbf{h}}\varphi \vee \mathrm{F}^{-1}_{\mathbf{h}}\varphi$), and the predicate $\mathrm{S}\varphi$ as a shorthand for $\varphi \vee S\varphi$. The predicates $\mathrm{X}_{\mathbf{h}}$ and $\mathrm{X}^{-1}_{\mathbf{h}}$ are the usual next sibling and previous sibling modalities.

In the sequel, $\mathcal{O}$ is either $\{\mathrm{S}\}$, $\{\mathrm{S}_+\}$ or $\{\mathrm{X}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}, \mathrm{X}^{-1}_{\mathbf{h}}, \mathrm{F}^{-1}_{\mathbf{h}}\}$ and we denote by EF+$\mathrm{F}^{-1}(\mathcal{O})$ the corresponding logics over forests. When considering only their horizontal behavior, these logics correspond over words to a fragment of LTL denoted by LTL($\mathcal{O}$).

We first recall the known characterizations over words, the first two being folklore while the last one is taken from [14]. A regular language L is definable in LTL(S_+) iff its syntactic monoid satisfies $3h = 2h$ and $f + g = g + f$. It is definable in LTL(S) iff its syntactic monoid satisfies $2h = h$ and $f + g = g + f$. It is definable in LTL($\mathrm{X}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}, \mathrm{X}^{-1}_{\mathbf{h}}, \mathrm{F}^{-1}_{\mathbf{h}}$) iff its syntactic monoid is in a variety known as **DA*D**, a decidable property as shown in [1].

The characterizations of EF+$\mathrm{F}^{-1}(S)$, EF+$\mathrm{F}^{-1}(S_+)$ and EF+$\mathrm{F}^{-1}(\mathrm{X}_{\mathbf{h}}, \mathrm{F}_{\mathbf{h}}, \mathrm{X}^{-1}_{\mathbf{h}}, \mathrm{F}^{-1}_{\mathbf{h}})$ require that V is in **DA** as before, that H satisfy the known characterization of the fragment of LTL induced by the horizontal modalities, together with a notion of saturation modified in order to use a notion k-MType and (X,k)-PType appropriate to the new horizontal expressive power.

For instance in the case of EF+F^{-1}(S) a k-MType is now completely specified by the presence or absence of certain

trees in the shallow multicontexts up to threshold 2. In particular it does not depend on k. Similarly, in the case of EF+F^{-1}($\mathrm{X_h}, \mathrm{F_h}, \mathrm{X_h^{-1}}, \mathrm{F_h^{-1}}$), k-MTypes correspond to definability in LTL($\mathrm{X_h}, \mathrm{F_h}, \mathrm{X_h^{-1}}, \mathrm{F_h^{-1}}$).

For a given set of horizontal axis $\mathcal{O}$, we then say that L is closed under saturation relative to $\mathcal{O}$ if it is closed under saturation as defined in Section 2 using a specification of k-MTypes and of (X, k)-PTypes based on LTL($\mathcal{O}$).

Theorem 4.1 *Let $\mathcal{O}$ be either S, S_+ or $\{\mathrm{X_h}, \mathrm{F_h}, \mathrm{X_h^{-1}}, \mathrm{F_h^{-1}}\}$. A regular language L is definable in* EF+F^{-1}($\mathcal{O}$) *iff*

1. *H satisfies $3h = 2h$ and $f + g = g + f$, in the case of $\mathcal{O} = S_+$*

1'. *H satisfies $2h = h$ and $f + g = g + f$, in the case of $\mathcal{O} = S$*

1". *H is in* **DA*D**, *in the case of $\mathcal{O} = \{\mathrm{X_h}, \mathrm{F_h}, \mathrm{X_h^{-1}}, \mathrm{F_h^{-1}}\}$*

2. *V is in* **DA**

3. *L is closed under saturation relative to $\mathcal{O}$.*

The proof of Theorem 4.1 follows the same outline as the proof of Theorem 2.2. Note also that besides for $\mathcal{O} = \{S\}$, EF+F^{-1}($\mathcal{O}$) is equivalent in expressive power to FO$_2$($<_v, \mathcal{O}$). The details are omitted in this abstract.

5 Discussion

Recall that the syntactic forest algebra (H, V) of a regular language L can be computed from any automaton recognizing L. Then, by testing all possible combinations, it is decidable whether H and V satisfy (2) and (3). When k is fixed, given a tree automaton recognizing L, it is not too hard to see that it is decidable whether L is closed under k-saturation. This is because L is regular and hence the pumping lemma shows that it is enough to consider only finitely many forests. Then, a brute force approach testing all possibilities yields the decidability.

By using the regularity of L it is also possible to show (details omitted in this abstract) that L is closed under k-saturation for some k iff L is closed under k-saturation for a k computable from any tree automaton recognizing L.

Altogether, we get the following corollary of Theorem 2.2 and Theorem 4.1:

Corollary 5.1 *It is decidable, given an automaton for L, whether L is definable in* FO$_2$($<_\mathbf{h}, <_\mathbf{v}$), EF+F^{-1}($S$), EF+F^{-1}($S_+$), EF+F^{-1}($\mathrm{X_h}, \mathrm{F_h}, \mathrm{X_h^{-1}}, \mathrm{F_h^{-1}}$).

Note that the pumping argument combined with the brute force algorithm described above yields an awful complexity with several nested exponential for the decision problem. We don't know yet whether this can be improved.

It would be interesting to incorporate the vertical successor and obtain a decidable characterization for the navigational core of XPath or, equivalently FO$_2$($<_\mathbf{h}, <_\mathbf{v}, +_\mathbf{h}1, +_\mathbf{v}1$), over trees. But this seems to require new ideas.

It would also be interesting to obtain an equivalent decidable characterization of FO$_2$($<_\mathbf{h}, <_\mathbf{v}$) without using the cumbersome notion of saturation. For instance it is not clear whether the notion of confusion introduced in [6] can be used as a replacement. We leave this as an open problem.

Our proof technique requires that the logic can at least express the fact that two nodes are siblings. In particular it does not apply to FO$_2$($<_\mathbf{v}$). We leave as an open problem to find a decidable characterization for FO$_2$($<_\mathbf{v}$).

References

[1] J. Almeida. A syntactical proof of locality of **DA**. *International Journal of Algebra and Computation*, 6(2):165–177, 1996.

[2] M. Benedikt and L. Segoufin. Regular languages definable in FO and FOmod. *ACM Trans. Of Computational Logic*, 11(1), 2010.

[3] M. Bojańczyk. Two-way unary temporal logic over trees. In *IEEE Symposium on Logic in Computer Science (LICS)*, pages 121–130, 2007.

[4] M. Bojańczyk and L. Segoufin. Tree languages defined in first-order logic with one quantifier alternation. In *Intl. Coll. on Automata, Languages and Programming (ICALP)*, 2008.

[5] M. Bojańczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. In *LICS*, 2008.

[6] M. Bojańczyk, H. Straubing, and I. Walukiewicz. Wreath products of forest algebras, with applications to tree logics. In *LICS*, 2009.

[7] M. Bojańczyk and I. Walukiewicz. Characterizing EF and EX tree logics. *Theoretical Computer Science*, 358, 2006.

[8] M. Bojańczyk and I. Walukiewicz. Forest algebras. In *Automata and Logic: History and Perspectives*, pages 107 – 132. Amsterdam University Press, 2007.

[9] Z. Esik and P. Weil. Algebraic characterization of regular tree languages. *Theoretical Computer Science*, 340:291–321, 2005.

[10] K. Etessami, M. Y. Vardi, and T. Wilke. First-order logic with two variables and unary temporal logic. *Inf. Comput.*, 179(2):279–295, 2002.

[11] M. Marx. First order paths in ordered trees. In *Intl. Conf. in Database Theory (ICDT)*, pages 114–128, 2005.

[12] T. Place. Characterization of logics over ranked tree languages. In *Conference on Computer Science Logic (CSL)*, pages 401–415, 2008.

[13] T. Place and L. Segoufin. A decidable characterization of locally testable tree languages. In *ICALP (2)*, pages 285–296, 2009.

[14] D. Thérien and T. Wilke. Over words, two variables are as powerful as one quantifier alternation. In *Proc. ACM Symp. on the Theory of Computing(STOC)*, pages 234–240, 1998.

[15] T. Wilke. An algebraic characterization of frontier testable tree languages. *Theoretical Computer Science*, 154(1):85–106, 1996.

A Necessity of saturation: Game argument

The goal of this section is to prove Lemma 2.3, i.e. that saturation is a necessary condition.

Assume L is definable in EF+F^{-1}(F$_{\mathbf{h}}$, F$_{\mathbf{h}}^{-1}$) and is recognized by the tree algebra (H, V) via some morphism α. Let k be the quantifier rank of a formula recognizing L, we show that L is closed under k-saturation.

The proof is an Ehrenfeucht Fraïssé argument and we adopt the EF+F^{-1}(F$_{\mathbf{h}}$, F$_{\mathbf{h}}^{-1}$) point of view instead of FO$_2$($<_{\mathbf{h}}$, $<_{\mathbf{v}}$) as the corresponding game is slightly simpler. The definition of the game corresponding to EF+F^{-1}(F$_{\mathbf{h}}$, F$_{\mathbf{h}}^{-1}$) is standard. There are two players, Duplicator and Spoiler, the board consists in two forests and both players agree on the number of moves in advance. At any time there is one pebble placed on a node of each of the two forests and the corresponding nodes have the same label. At the beginning of the game the two pebbles are placed on the root of the leftmost tree of each forests. At each step Spoiler moves one of the pebble, either to some ancestor of its current position, or to some descendant or to some left or right sibling. Duplicator must respond by moving the other pebble in the same direction to a node of the same label. If Duplicator cannot move then Spoiler wins.

Given P, Δ, p, x, t and T as in the definition of k-saturation, let $u = \alpha(\Delta)$, $h = \alpha(t)$ and $v = \alpha(p[T, x])$.

We exhibit two forests T and T' such that $\alpha(T) = u^\omega h$ and $\alpha(T') = u^\omega v u^\omega h$ and such that the Duplicator has a winning strategy for the k-move game above.

A classical argument then shows that this implies that no formula of EF+F^{-1}(F$_{\mathbf{h}}$, F$_{\mathbf{h}}^{-1}$) of quantifier depth k can distinguish the two forests. This implies that $u^\omega h = u^\omega v u^\omega h$ as desired.

Our agenda is as follows. In Section A.1 we define the two trees on which we will play. Finally in Section A.2 we give the winning strategy for Duplicator for the k-move game on the two trees.

A.1 Definition of the trees.

Let $H_P = \{h_1, \cdots, h_l\}$ be the maximal P-equivalence class. Let $h = \alpha(t) \in H_p$ and $h' = u^\omega$. Because mutual P-reachability has only one equivalence class, for each i there exists a P-valid context U_i such that $h_i = \alpha(U_i)h'$. Recall that by definition of k-saturation, for each simple multicontext p occurring in U_i there exists a p' occurring in Δ such that $(p, x) \cong_k (p', x')$ where x and x' mark the position in p and p' of the skeletons of the corresponding context.

We now construct by induction on j contexts Δ_j and $U_{i,j}$ for all i, such that $\alpha(\Delta_j) = u$ and $\alpha(U_{i,j}) = u_i$.

Set $U_{i,0} := U_i$ and $\Delta_0 := \Delta$. Let $m := 2^k$. For $j > 0$ $U_{i,j}$ and Δ_j are formed from the $U_{i,j-1}$ and Δ_{j-1} by replacing each maximal subforest of type h_i by $U_{i,j-1}\Delta_j^m t$.

The following claim will be useful later. It follows by a simple adaptation of the winning strategy of the n-move game over the strings $\mathbf{1}^d$ and $\mathbf{1}^{d'}$ with $d, d' \geq n$.

Claim A.1 *For all n and all $d, d' \geq n$, Duplicator has a winning strategy in the n-move game on Δ_d and $\Delta_{d'}$.*

Let P_m be the context formed from p by replacing all subforests of type h_i by $U_{i,m}\Delta_m^m t$ and T_m be the sequence of forests constructed from T by replacing each forests of type h_i by $U_{i,m}\Delta_m^m t$.

Finally let:

$$\begin{aligned} T &:= \Delta_m^m \, P_m[T_m, x] \, \Delta_m^m \, t \\ T' &:= \Delta_m^m \, \Delta_m^m \, t \end{aligned} \tag{5}$$

The following claim then conclude the proof of Lemma 2.3.

Claim A.2 *Duplicator has a winning strategy for the k-move game between T and T'.*

A.2 The winning strategy

Proof sketch of Claim A.2. We give a winning strategy for Duplicator in this game. In order to be able to formulate this strategy we need further definitions.

Given two nodes x and y and a number n we denote by $x \equiv_n^H y$ the fact that Duplicator has a winning strategy in the n-move game played on the sequence of siblings of x and y, starting from positions x, y.

Given a node x, the *subforest of* x is the forest formed by all the subtrees of all the siblings of x.

The *nesting level* of a node x of T or T' is the minimal number l such x belongs to a context Δ_l or $U_{i,l}$.

The *skeleton* of T (or of T') is the longest path of T containing all the nodes of nesting level m. i.e. the path that goes from the root of T to the port of each of the Δ_m.

The *upward level* of a node $x \in T$ (or $x \in T'$) is the number of occurrences of Δ_m above x in the skeleton.

Notice that given a node x of nesting level l, either x has a child of nesting level l or x has exactly one sibling with such a child. This sibling is denoted as the *l-sibling of* x.

The *downward level* of a node $x \in T$ (or $x \in T'$) is the number of copies of Δ_l that are below the l-sibling of x, where l is the level of x.

Given a node x of nesting level l, for all $l' > l$, its *key ancestor at level* l' is the first (starting from x) ancestor y of x that has nesting level l'.

Given an integer n we say that x has a *n-ancestor* if it has a key ancestor of downward level smaller than n. If this is the case the *n-ancestor of* x is the node of maximal nesting level satisfying this property.

We now state a property $\mathcal{P}(n)$ that depends on an integer n, two nodes $x \in T$ and $y \in T'$ and possibly two nodes

$\hat{x} \in T$ and $\hat{y} \in T'$ that are ancestors of respectively x and y.

We then show that when $\mathcal{P}(n)$ holds on a game starting at x, y, then Duplicator can play one move while enforcing $\mathcal{P}(n-1)$. As it is easy to see that $\mathcal{P}(k)$ holds for the roots of T and T', this will conclude the proof of Claim A.2.

$\mathcal{P}(n)$ states that $\hat{x}$ is defined iff $\hat{y}$ is defined and, whenever they are defined, both have nesting level greater than n, upward level greater than n and no n-ancestor.

Moreover it requires the disjunction of the following three cases:

1. $\hat{x}$ and $\hat{y}$ are defined. In this case Duplicator has a winning strategy in the n-move game played on the subforest of $\hat{x}$ and the subforest of $\hat{y}$, and starting at positions x and y.

2. $\hat{x}$ and $\hat{y}$ are undefined and the upward level of x is smaller than n. In this case x and y are at the same position in the tree (recall that by construction T and T' are isomorphic up to m copies of Δ_m).

3. $\hat{x}$ and $\hat{y}$ are undefined, the upward level of x is greater than n. In this case x and y have no n-ancestor and are of nesting level greater than n, moreover $x \equiv_n^H y$.

Assume we are in a situation where $\mathcal{P}(n+1)$ holds. We sketch how Duplicator can play while enforcing $\mathcal{P}(n)$. The strategy depends on why $\mathcal{P}(n+1)$ holds.

Case 1: $\hat{x}$, and therefore also $\hat{y}$, are defined.

If Spoiler moves to a node below $\hat{x}$ then Duplicator simply use the strategy provided by item one of $\mathcal{P}(n+1)$. $\hat{x}$ and $\hat{y}$ remain unchanged.

We now assume that Spoiler moves to a node x' above $\hat{x}$.

If the upward level of x' is less than n, then Duplicator can easily answer while satisfying item two of $\mathcal{P}(n)$. In this case $\hat{x}$ and $\hat{y}$ now become undefined.

If the upward level of x' is $> n$. By saturation of Δ_m, there is a node z in Δ_m such that $x' \equiv_n^H z$. By hypothesis the upward level of y is larger than the upward level of $\hat{y}$ which is larger than $n+1$. Hence we can find above y an occurrence of Δ_m of upward level larger than n. Duplicator answer by the copy of z in this occurrence of Δ_m and item three of $\mathcal{P}(n)$ is satisfied. In this case $\hat{x}$ and $\hat{y}$ now become undefined.

Case 2: The upward level of x is smaller than $n+1$.

If Spoiler moves up or horizontally, Duplicator simply copy Spoiler's move and item two of $\mathcal{P}(n)$ is true if we end up with an upward level $\leq n$ otherwise item three trivially hold. Assume now that Spoiler moves to some descendant x' of x.

If x' has a key ancestor of nesting level n that has no n-ancestor. Then we set $\hat{x}$ to this key ancestor. As the nesting level of y must be greater than $n+1$ (by $\mathcal{P}(n+1)$ the nesting level of y is equal to the nesting level of x), the subtree at y contains a copy of all the subforests occurring at nesting level n. Hence we can find a descendant $\hat{y}$ of y whose subforest is isomorphic to the subforest of $\hat{x}$. Duplicator then pick the copy of x' in the subforest of $\hat{y}$ and item one of $\mathcal{P}(n)$ is satisfied.

If x' has a n-ancestor of nesting level greater than n. Then we set $\hat{x}$ to this n-ancestor and let s be the subforest of $\hat{x}$. As the nesting level of y must be greater than $n+1$ (by $\mathcal{P}(n+1)$ the nesting level of y is equql to the nesting level of x), there is below y an occurrence of Δ_n. We set $\hat{y}$ to the copy of $\hat{x}$ in the skeleton of this occurrence of Δ_n. Let s' be the subforest of $\hat{y}$. Notice that s and s' only differ by their nesting imbrication but those are bigger than n. Hence by Claim A.1 Duplicator has a winning strategy when playing n-moves on s and s'. Item one of $\mathcal{P}(n)$ is satisfied.

If x' has nesting level greater than n and no n-ancestor. By saturation of Δ_m, there is a node z in Δ_m such that $x' \equiv_n^H z$. By hypothesis the nesting level of y is larger than $n+1$. Hence we can find below y an occurrence of Δ_j with $j \geq n$. Duplicator answer by the copy of z in this occurrence of Δ_j and item three of $\mathcal{P}(n)$ is satisfied. In this case $\hat{x}$ and $\hat{y}$ remain undefined.

Case 3: The upward level of x is greater than $n+1$ and x does not have a $(n+1)$-ancestor.

• If Spoiler moves horizontally, Duplicator moves according to its winning strategy given by $\equiv_{(n+1)}^H$.

• If Spoiler moves up to some node x'.

If the upward level of x' is less than n, then Duplicator can easily answer while satisfying item two of $\mathcal{P}(n)$. In this case $\hat{x}$ and $\hat{y}$ remain undefined.

If the upward level of x' is $> n$. By saturation of Δ_m, there is a node z in Δ_m such that $x' \equiv_n^H z$. By hypothesis the upward level of y is larger than $n+1$. Hence we can find above y an occurrence of Δ_m of upward level larger than n. Duplicator answer by the copy of z in this occurrence of Δ_m and item three of $\mathcal{P}(n)$ is satisfied. In this case $\hat{x}$ and $\hat{y}$ remain undefined.

• If Spoiler moves down to some node x'.

If x' has a key ancestor of nesting level n that has no n-ancestor. Then we set $\hat{x}$ to this key ancestor. As the nesting level of y must be greater than $n+1$ because of $\mathcal{P}(n+1)$, the subtree at y contains a copy of all the subforests occurring at nesting depth n. Hence we can find a descendant $\hat{y}$ of y whose subforest is isomorphic to the subforest of $\hat{x}$. Duplicator then pick the copy of x' in the subforest of $\hat{y}$ and item one of $\mathcal{P}(n)$ is satisfied.

If x' has a n-ancestor of nesting level greater than n. Then we set $\hat{x}$ to this n-ancestor and let s be the subforest

of $\hat{x}$. As the nesting level of y must be greater than $n+1$ because of $\mathcal{P}(n+1)$, hence below y there is an occurrence of Δ_n. We set $\hat{y}$ to the copy of $\hat{x}$ in the skeleton of this occurrence of Δ_n. Let s' be the subforest of $\hat{y}$. Notice that s and s' only differ by their nesting imbrication but those are bigger than n. Hence by Claim A.1 Duplicator has a winning strategy when playing n-moves on s and s'. Item one of $\mathcal{P}(n)$ is satisfied.

If x' has nesting level greater than n and no n-ancestor. By saturation of Δ_m, there is a node z in Δ_m such that $x' \equiv_n^H z$. By hypothesis the nesting level of y is larger than $n+1$. Hence we can find below y an occurrence of Δ_j with $j \geq n$. Duplicator answer by the copy of z in this occurrence of Δ_j and item three of $\mathcal{P}(n)$ is satisfied. In this case $\hat{x}$ and $\hat{y}$ remain undefined.

This conclude the proof of Claim A.2 and the proof of Lemma 2.3.