DECIDING DEFINABILITY IN $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ ON TREES

THOMAS PLACE AND LUC SEGOUFIN

Bordeaux University, LaBRI

INRIA and ENS Cachan, LSV

ABSTRACT. We provide a decidable characterization of regular forest languages definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. By $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ we refer to the two variable fragment of first order logic built from the descendant relation and the following sibling relation. 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 show that our techniques can be applied to other two variable first-order logics having exactly the same vertical modalities as $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ but having different horizontal modalities.

1. Introduction

Logics for expressing properties of labeled trees and forests figure importantly in several different areas of Computer Science. This paper is about logics on finite unranked trees. The most prominent one is monadic second-order logic (MSO) as it can be captured by finite tree automata. All the logics we consider are less expressive than monadic second-order logic. Even with these restrictions, this encompasses a large body of important logics, such as variants of first-order logic, temporal logics including CTL* or CTL, as well as query languages used for XML data.

This paper is part of a research program devoted to understanding and comparing the expressive power of such logics.

We say that a logic has a decidable characterization if the following problem is decidable: given as input a finite tree automaton (or equivalently a formula of MSO), decide if the recognized language is definable by the logic in question. Usually a decidable characterization requires a solid understanding of the expressive power of the corresponding logic as witnessed by decades of research, especially for logics for strings. The main open problem in this research program is to find a decidable characterization of $\mathrm{FO}(<_{\mathbf{v}})$, 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 $\mathrm{FO}(<_{\mathbf{v}}, <_{\mathbf{h}})$ we refer to the logic that has two binary predicates, one for the descendant relation, one for the following

1998 ACM Subject Classification: F.4.1,F.4.3.

Key words and phrases: Tree Languages,Tree Automata, Two-Variables First-Order Logic, Characterization.

LOGICAL METHODS IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

sibling relation. We investigate an important fragment of $\mathrm{FO}(<_{\mathbf{v}},<_{\mathbf{h}})$, its two variable restriction denoted $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$. This is a robust formalism that, in terms 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 [Mar05]. More precisely, it corresponds to the temporal logic $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{F_h},\mathbf{F_h}^{-1})$ that navigates in the tree using two "vertical" modalities, one for going to some ancestor node ($\mathbf{F}^{-1}$) and one for going to some descendant node ($\mathbf{EF}$), and two "horizontal" modalities for going to some following sibling ($\mathbf{F_h}$) or some preceding sibling ($\mathbf{F_h}^{-1}$).

We provide a characterization of $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$, or equivalently $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{F_h},\mathbf{F_h}^{-1})$, over unranked ordered trees. We also show that this characterization is decidable. Since $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ can express the fact that a tree has rank k for any fix number k, our result also applies to ranked trees.

Our characterization is stated using closure properties expressed partly using identities that must be satisfied by the syntactic forest algebra of the input regular language, and partly via a mechanism that we call saturation.

Here, a forest algebra is essentially a pair of finite semigroups, the "horizontal" semigroup for forest types and the "vertical" semigroup for context types, together with an action of contexts over forests. It was introduced in [BW07], using monoids instead of semigroups, and is a formalism for recognizing forest languages whose expressive power is equivalent to definability in MSO. Given a formula of MSO, one can compute its syntactic forest algebra, which recognizes the set of forests satisfying the formula. Hence any characterization based on a finite set of identities over the syntactic forest algebra can be tested effectively when given a regular language as long as each identity can be effectively tested, which will always be the case in this paper.

The syntactic forest algebra was used successfully for obtaining decidable characterizations for the classes of tree languages definable in $\mathbf{EF}+\mathbf{EX}^{-1}$ [BW06], $\mathbf{EF}+\mathbf{F}^{-1}$ [Boj09], $\mathcal{B}\Sigma_1(<_{\mathbf{v}})$ [BSS12] and $\Delta_2(<_{\mathbf{v}})$ [BS10]. Here $\mathbf{EF}+\mathbf{EX}^{-1}$ is the class of languages definable in a temporal logic that navigates in trees using two vertical modalities, $\mathbf{EF}$, that we have already seen before, and $\mathbf{EX}$, which goes to a child of the current node. $\mathbf{EF}+\mathbf{F}^{-1}$ is the class of languages definable in $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{F_h},\mathbf{F_h}^{-1})$ without using the horizontal modalities. $\mathcal{B}\Sigma_1(<_{\mathbf{v}})$ stands for the class of languages definable by a Boolean combination of existential formulas of $\mathrm{FO}(<_{\mathbf{v}})$ and $\Delta_2(<_{\mathbf{v}})$ is the class of languages definable in $\mathrm{FO}(<_{\mathbf{v}})$ by both a formula of the form $\exists^*\forall^*$ and a formula of the form $\forall^*\exists^*$.

Over strings, the logics induced by $\Delta_2(<)$, $\mathrm{FO}^2(<)$ and $\mathrm{F}+\mathrm{F}^{-1}$, have exactly the same expressive power [EVW02, TW98]. But over trees this is not the case. For instance $\mathbf{EF}+\mathbf{F}^{-1}$ is closed under bisimulation while the other two are not. While decidable characterizations were obtained for $\mathbf{EF}+\mathbf{F}^{-1}$ and $\Delta_2(<_{\mathbf{v}})$ [Boj09, BS10], the important case of $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ was still missing and is solved in this paper.

Over strings, a regular language is definable in $\mathrm{FO}^2(<)$ iff its syntactic semigroup satisfies an identity that can be effectively tested [TW98]. Not surprisingly our first set of identities requires that the horizontal and vertical semigroups of the syntactic forest algebra both satisfy this identity. Our extra property is more complex and mixes at the same time the vertical and horizontal navigational power of $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$. We call it *closure under saturation.*

It is immediate from the string case that being definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ implies that the vertical and horizontal semigroups of the syntactic forest algebra satisfy the required identity. That closure under saturation is also necessary is proved via a classical, but tedious, Ehrenfeucht-Fraïssé game argument. As usual in this area, the difficulty is to show that the closure conditions are sufficient. In order to do so, as it is standard when dealing with $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ (see e.g. [Boj09, BS10, TW98]), 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 our 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 $\mathbf{EF} + \mathbf{F}^{-1}$ [Boj09] and we reuse several ideas developed in this paper. However it departs from it in many essential ways. First of all the closure under bisimulation of $\mathbf{EF} + \mathbf{F}^{-1}$ was used in [Boj09] in an essential way in order to compute a subalgebra and perform inductions on the size of the algebra. Moreover, because $\mathbf{EF} + \mathbf{F}^{-1}$ does not have horizontal navigation, Bojańczyk was able to isolate certain labels and then also perform inductions 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 $\mathbf{EF} + \mathbf{F}^{-1}$ given in [Boj09]. The logic $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ 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 size of the alphabet. In order to overcome these problems our proof replaces the inductions used in [Boj09] by an induction on the set of forbidden patterns. This makes the two proofs technically fairly different.

It turns out that our proof technique applies to various horizontal modalities. In the final section of the paper we show how to adapt the characterization obtained for $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ in order to obtain characterizations for $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{X_h}, \mathbf{F_h}, \mathbf{X_h}^{-1}, \mathbf{F_h}^{-1})$, $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S})$ and $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S}^{\neq})$, where $\mathbf{X_h}$, $\mathbf{X_h}^{-1}$, $\mathbf{S}$ and $\mathbf{S}^{\neq}$ are horizontal navigational modalities moving respectively to the next sibling, previous sibling, an arbitrary sibling including the current node, or an arbitrary different sibling excluding the current node.

Other 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. [BS09, PS11, Wil96, ÉW05]). It is not clear however how to use these formalisms in order to provide a characterization of $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

There exist decidable characterizations of $\mathbf{EF} + \mathbf{F}^{-1}$ and $\Delta_2(<_{\mathbf{v}})$ over trees of bounded rank [Pla08]. But, as these logics cannot express the fact that a tree is binary, the unranked and bounded rank characterizations are different. As mentioned above, we don't have this problem with $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

Organization of the paper. We first provide the preliminary definitions in Section 2. The main definitions and their basic properties are described in Section 5. Our characterization is stated in Section 6. That our properties are necessary for being definable in $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ is proved in Section 7. We give the proof that our characterization for $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ is sufficient in Section 8. Decidability of closure under saturation is not immediate and Section 9 is devoted to this issue. In Section 10 we show how to adapt the arguments in order to characterize several other horizontal navigation modalities.

Note that Section 7, Section 8 and Section 9 can be read in an arbitrary order.

This paper is the journal version of [PS10]. From the conference version the statement of the characterization has been slightly changed and the proofs have been significantly modified in order to simplify the presentation.

2. Preliminaries

We work with finite unranked ordered trees and forests labeled over a finite alphabet $\mathbb{A}$. A finite alphabet is a pair $\mathbb{A} = (A, B)$ where A and B are finite sets of labels. We use A to label leaves and B to label inner nodes. Making the distinction between leaves and inner nodes labels makes our presentation slightly simpler without harming the generality of our results. Given a finite alphabet $\mathbb{A} = (A, B)$, trees and forests are defined inductively as follows: for any $a \in A$, a is a tree. If $t_1, \cdots, t_k$ is a finite non-empty sequence of trees then $t_1 + \cdots + t_k$ is a forest. If s is a forest and $b \in B$, then $b(s)$ is a tree. Notice that we do not consider empty trees nor empty forests. A set of trees (forests) over a finite alphabet $\mathbb{A}$ is called a tree language (forest language).

We use standard terminology for trees and forests defining nodes, ancestors, descendants, following and preceding siblings. We write $x <_{\mathbf{v}} y$ to say that x is a strict ancestor of y or, equivalently, that y is a strict descendant of x. We write $x <_{\mathbf{h}} y$ to say that x is a strict preceding sibling of y or, equivalently, that y is a strict following sibling of x.

A context is a forest over $(A \cup \{\Box\}, B)$ with a single leaf of label $\Box$ that cannot be a root and that has no sibling. This distinguished node is called *the port* of the context (see Figure 1). This definition is not standard as usually contexts are defined without the "no sibling" restriction but it is important for this paper to work with this non-standard definition. If c is a context, the path in c containing all the ancestors of its port is called the *backbone* of c.

A context c can be composed with another context c' or with a forest s in the obvious way by substituting c' or s in place of the port of c. This composition yields either the context cc' or the forest cs.

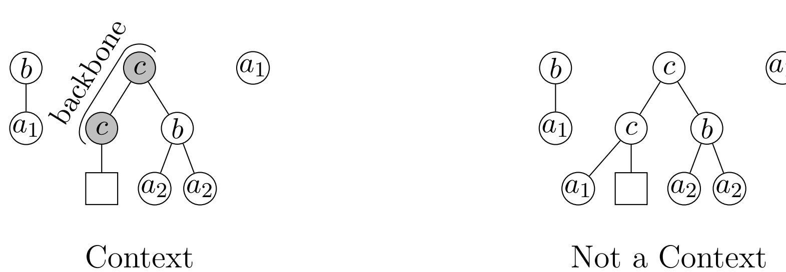

Figure 1: Illustration of the notion of context. The squared nodes represent ports. The right part is not a context because the port has a sibling.

If x is a node of a forest then the *subtree at* x is the tree rooted at x. The *subforest of* x is the forest consisting of all the subtrees that are rooted at siblings of x (including x). Finally, if x is not a leaf, the *subforest below* x is the forest consisting of all the subtrees that are rooted at children of x, see Figure 2. Notice that from the definitions if follows that s is a subforest of a forest t iff there exists a context c such that $t = cs$. In particular if we consider the forest $b(a_1 + a_2 + a_3)$, $a_1 + a_2 + a_3$ is a subforest but not $a_1 + a_2$.

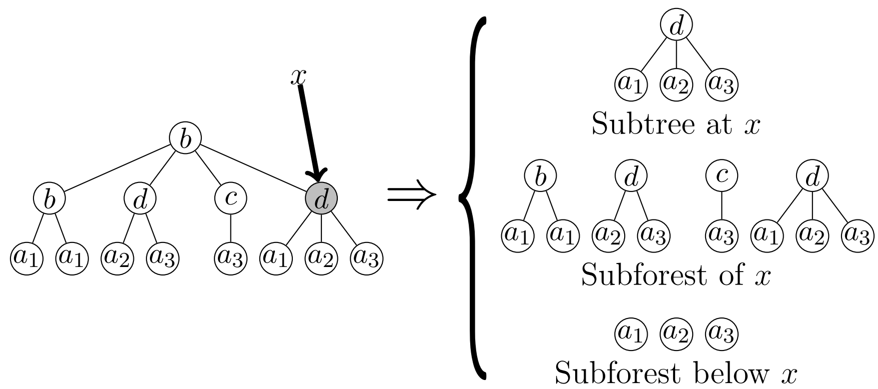

Figure 2: Illustration of the notion of subtrees and subforests

3. The Logic FO$^2(<_\mathbf{v}, <_\mathbf{h})$

3.1. **Definition.** A forest can be seen as a relational structure. The domain of the structure is the set of nodes. The signature contains a unary predicate P_a for each symbol $a \in \mathbb{A}$ plus the binary predicates $<_\mathbf{v}$ and $<_\mathbf{h}$. By MSO$(<_\mathbf{v}, <_\mathbf{h})$ we denote the monadic second order logic over this relational signature. We use the classical semantics for MSO$(<_\mathbf{v}, <_\mathbf{h})$ and write $s \models \phi(\bar{u})$ if the formula ϕ is true on s when interpreting its free variables with the corresponding nodes of $\bar{u}$. As usual, each sentence φ of MSO$(<_\mathbf{v}, <_\mathbf{h})$ defines a forest language $L_\varphi = \{s \mid s \models \varphi\}$. A language defined in MSO$(<_\mathbf{v}, <_\mathbf{h})$ is called a *regular language*. As usual regular languages form a robust class of languages and there is a matching notion of unranked ordered forests automata (see for instance [CDG$^+$, chapter 8]). We will see in Section 4 a corresponding notion of recognizability using forest algebras.

The logic of interest for this paper is FO$^2(<_\mathbf{v}, <_\mathbf{h})$, the two variable restriction of the first-order fragment of MSO$(<_\mathbf{v}, <_\mathbf{h})$.

In terms of expressive power, FO$^2(<_\mathbf{v}, <_\mathbf{h})$ is equivalent to $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$, a temporal logic that we now describe. Essentially, $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ is 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:

$$\varphi :: \mathbb{A} \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbf{EF}\varphi \mid \mathbf{F}^{-1}\varphi \mid \mathbf{F_h}\varphi \mid \mathbf{F_h}^{-1}\varphi$$

We use the classical semantics for this logic which defines when a formula holds at a node x of a forest s. In particular, $\mathbf{EF}\varphi$ holds at x if there is a strict descendant of x where φ holds, $\mathbf{F}^{-1}\varphi$ holds at x if there is a strict ancestor of x where φ holds, $\mathbf{F_h}\varphi$ holds at x if φ holds at some strict following sibling of x, and $\mathbf{F_h}^{-1}\varphi$ holds at x if φ holds at some strict preceding sibling of x. We then say that a forest s satisfies a formula ϕ if ϕ holds at the root of the first tree of s. The following result is immediate from [Mar05]:

Theorem 3.1. *For any* FO$^2(<_\mathbf{v}, <_\mathbf{h})$ *formula* $\phi(x)$*, there exists a* $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ *formula φ holding true on the same set of nodes for every forests. In particular a forest language is definable in* FO$^2(<_\mathbf{v}, <_\mathbf{h})$ *iff it is definable in* $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$.

We aim at providing a decidable characterization of regular forest languages definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. This means finding an algorithm that decides whether or not a given regular forest language is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

Note that $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ is expressive enough to test whether a forest is a tree and, for each k whether it has rank k. Hence any result concerning forest languages definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ also applies to tree languages definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ and covers the ranked and unranked cases.

We shall mostly adopt the $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ point of view as it is useful when considering other horizontal modalities or when making comparisons with the decision algorithm obtained for $\mathbf{EF} + \mathbf{F}^{-1}$ in [Boj09].

3.2. **Ehrenfeucht-Fraïssé Games.** As usual definability in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ corresponds to winning strategies in a Ehrenfeucht-Fraïssé game that we briefly describe here. We adopt the $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ point of view as the corresponding game is slightly simpler. Its definition is standard.

There are two players, Duplicator and Spoiler. The board consists in two forests and the number k of rounds is fixed in advance. At any time during the game there is one pebble placed on a node of one forest and one pebble placed on a node of the other forest and both nodes have the same label. If the initial position is not specified, the game starts with the two pebbles placed on the root of the leftmost tree in each forest. Each round starts with Spoiler moving one of the pebbles inside its forest, 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 inside the other forest in the same direction to a node of the same label. If during a round Duplicator cannot move then Spoiler wins the game. If Duplicator was able to respond to all the moves of Spoiler then she wins the game. Winning strategies are defined as usual. If Duplicator has a winning strategy for the game played on the forests s, t then we say that s and t are k-equivalent.

The following results on games are classical and simple to prove.

Lemma 3.2 (Folklore)**.**

(1) For every k, k-equivalence is an equivalence relation of finite index.

(2) For every k, each class of the k-equivalence relation is definable by a sentence of $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ such that the nesting depth of its navigational modalities is bounded by k.

(3) For every k, if s and t are k-equivalent then they satisfy the same sentences of $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ such that the nesting depth of their navigational modalities is bounded by k.

When played on words instead of forests, the game is the same except that now Spoiler can move either to a previous or to a following position. The results are identical after replacing $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ with $\mathrm{FO}^2(<)$, the two variable first-order logic on strings, using the predicate $<$ for the following position relation.

3.3. **Antichain Composition Principle.** As mentioned in the introduction, we use induction to prove that if L satisfies the characterization then we can construct a $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ formula for L. At each step in this construction, we prove that L can be defined as the composition of simpler languages such that a formula for L can be constructed from formulas defining the simpler languages. This is what we do with the following simple composition lemma, essentially adapted from [Boj09] and using the same terminology.

A formula of $\mathrm{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ 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 that may not be apparent just by looking at the syntax of the formula. A typical antichain formula selects in a forest the set of nodes of label $b \in B$ that have no ancestor of label b.

Given (i) an antichain formula φ, (ii) disjoint forest languages $L_1, \cdots, L_n$ and (iii) labels $a_1, \cdots, a_n \in A$ and (iv) a forest s, we define the forest $s' = s[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n]$ as follows. For each node x of s such that $s \models \varphi(x)$, we determine the unique i such that the forest language L_i contains the *subforest below* x. If such an i exists, we remove the whole subforest below x, and replace it by a leaf of label 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 subforest below x.

The composition method that we will use is summarized in the the following lemma:

Lemma 3.3. *[Antichain Composition Lemma] Let φ , $L_1, \ldots, L_n$ and $a_1, \ldots, a_n$ be as above. If $L_1, \ldots, L_n$ and K are languages definable in* $\mathrm{FO}^2(<_\mathbf{v}, <_\mathbf{h})$*, then so is*

$$L = \{t \mid t[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n] \in K\}.$$

This lemma follows from a simple Ehrenfeucht-Fraïssé game argument. Using the $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h^{-1}})$ point of view we can also construct a formula defining L. The formula for L is obtained from the one for K by restricting all navigation to nodes that are not descendants of nodes satisfying φ and by replacing each test that a label is a_i by the formula for L_i where all navigations are now restricted to nodes that are descendants of a node that satisfies φ. The fact that φ is antichain makes this construction sound. The details are simple and are omitted here as they paraphrase those given in [Boj09] for $\mathbf{EF} + \mathbf{F}^{-1}$.

The inductive step of our proof consists in exhibiting $L_1, \ldots, L_n$ and K, together with an antichain formula φ such that $L = \{t \mid t[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n] \in K\}$ and $K, L_1, \ldots, L_n$ have smaller inductive parameters than L. In [Boj09] the antichain formula is of the form: "select the set of nodes of label $b \in B$ that have no ancestor of label b." Observe that such a formula allows us to use the size of B as an induction parameter as K does not contain the label b. In our case, we replace B by sibling patterns that we will define in Section 5.

4. Forest algebras

A key ingredient in our characterization is based on syntactic forest algebras. *Forest algebras* were introduced by Bojańczyk and Walukiewicz as an algebraic formalism for studying regular forest languages [BW07]. We work with the semigroup variant of forest algebras. Moreover we require that the port of each context has no sibling. These restrictions are necessary as, without them, the languages definable in $\mathrm{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ would not form a variety, i.e. would not be characterizable by its syntactic forest algebra only.

We give a brief summary of the definition of forest algebras and of their important properties. More details can be found in [BW07]. 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

$$(h, v) \in H \times V \mapsto vh \in H$$

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

$$(v + g)h \ = \ vh + g, \qquad\qquad (g + v)h \ = \ g + vh$$

for all $h \in H$.

Let $\mathbb{A} = (A, B)$ be a finite alphabet. The *free forest algebra* on $\mathbb{A}$, denoted by $\mathbb{A}^{\triangle}$, is the pair of semigroups $(H_{\mathbb{A}}, V_{\mathbb{A}})$ where $H_{\mathbb{A}}$ is the set of forests over $\mathbb{A}$ equipped with the $+$ operation and $V_{\mathbb{A}}$ the set of contexts equipped with the composition operation, together with the natural action. One can verify that this action turns $\mathbb{A}^{\triangle}$ into a forest algebra.

A morphism $\alpha : (H_1, V_1) \to (H_2, V_2)$ of forest algebras is 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 $\alpha : \mathbb{A}^{\triangle} \to (H, V)$ 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.

Consider some forest language L over an alphabet $\mathbb{A}$. We define an equivalence relation $\sim_L$ over contexts and over forests. Given two forests t_1, t_2, we say that $t_1 \sim_L t_2$ iff for any two forests s, s' and any context c, $c(s + t_1 + s') \in L$ iff $c(s + t_2 + s') \in L$. Given two contexts c_1, c_2 we say that $c_1 \sim_L c_2$ iff for any forest s, $c_1 s \sim_L c_2 s$. This equivalence is a congruence of forest algebras that is of finite index iff L is regular. The quotient of $\mathbb{A}^{\triangle}$ by this congruence yields a forest algebra recognizing L which we call the *syntactic forest algebra* of L. The mapping sending a forest or a context to its equivalence class in the syntactic forest algebra, denoted α_L, is a morphism called the *syntactic morphism* of L.

It is also important to know that given an MSO($<_{\mathbf{v}}, <_{\mathbf{h}}$) sentence ϕ, the syntactic forest algebra of L_ϕ and the syntactic morphism α_ϕ can be computed from ϕ.

Idempotents. It follows from standard arguments of semigroup theory that given any finite semigroup S, there exists 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: $x^\omega = x^\omega x^\omega$. Given a forest algebra (H, V) we will denote by $\omega(H, V)$ (or just ω when (H, V) is understood from the context) the product of $\omega(H)$ and $\omega(V)$ and for any element $u \in V$ and $g \in H$ we will write u^ω and ωg for the corresponding idempotents.

Finally, given a semigroup S we will denote by S^1 the monoid formed from S by adding a neutral element.

Leaf Surjective Morphisms. Let $\mathbb{A} = (A, B)$ be a finite alphabet and let $\alpha : \mathbb{A}^\Delta \to (H, V)$ be a morphism into a finite forest algebra (H, V). We say that α is *leaf surjective* iff for any $h \in H$, there exists $a \in A$ such that $\alpha(a) = h$.

Observe that given any morphism $\alpha : (A, B)^\Delta \to (H, V)$, one can construct a leaf surjective one $\beta : (A \cup H, B)^\Delta \to (H, V)$ by extending α in the obvious way. We call β the *leaf completion* of α.

Lemma 4.1. *Let $\alpha : \mathbb{A}^\Delta \to (H, V)$ be a morphism into a finite forest algebra, β be its leaf completion and $h \in H$ be such that $\beta^{-1}(h)$ is $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ definable. Then $\alpha^{-1}(h)$ is $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ definable.*

Proof. A forest in $\alpha^{-1}(h)$ is a forest in $\beta^{-1}(h)$ that contains no leaf with label in H. Therefore, one can construct an $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ formula for $\alpha^{-1}(h)$ from a formula for $\beta^{-1}(h)$. □

Working with leaf surjective morphisms will be convenient for us. Typically when applying the Antichain Composition Lemma we will construct K from L by replacing some subforests of L by leaf nodes with the same forest type. It is therefore important that such nodes exist.

The string case. A reason for using syntactic forest algebras is that the same problem for strings was solved using syntactic semigroups. In the string case there is only one linear order and the corresponding logic is denoted by $\mathrm{FO}^2(<)$. Recall that the syntactic semigroup (monoid) of a regular string language is the transition semigroup (monoid) of its minimal deterministic automata. It is therefore computable from any reasonable presentation of the regular string language. The following characterization was actually stated using syntactic monoids, but it is equivalent to this statement[1].

Theorem 4.2 ([TW98])**.** *A regular string language is definable in $\mathrm{FO}^2(<)$ iff its syntactic semigroup S satisfies for all $u, v \in S$:*

$$(uv)^\omega v (uv)^\omega = (uv)^\omega$$

Unfortunately in the case of forest languages we were not able to state our characterization using only the syntactic forest algebra of the input regular language. We will need an extra ingredient that we call *saturation*.

5. Shallow multicontexts

In this section, we define shallow multicontexts which represent sequences of siblings. We will often manipulate shallow multicontexts modulo $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ definability. This is captured by an equivalence relation on shallow multicontexts that we also define in this section.

This notion of shallow multicontext is central for this paper as we will use it not only as a parameter in the inductive argument but also to define the notion of *saturation* that we use in our characterization of $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$.

Set $\mathbb{A} = (A, B)$ as a finite alphabet. All definitions are parametrized by a morphism $\alpha : \mathbb{A}^\Delta \to (H, V)$. Note that while the definitions make sense for any morphism α, they are designed to be used with a leaf surjective one. Given a forest s (a context p) we refer to its image under α as the *forest type* of s (the *context type* of p).

[1] We are actually not using the identity of [TW98]. Ours can easily seen to be equivalent to it. This is a folklore result. A proof can be found in [Kuf06].

5.1. **Shallow multicontexts.** A multicontext is defined in the same way as a context but with several ports, possibly none. The *arity* of a multicontext is the number of its ports, possibly 0. Note that as our forests are ordered, each multicontext implicitly defines a linear order on its ports. A multicontext is said to be *shallow* if each of its trees is either a single node $a \in A$, a single inner node with a port below, $b(\square)$, or a tree of the form $b(a)$ where $b \in B$ and $a \in A$ (see Figure 3).

For technical reasons we do not consider forests with a single tree of the form $a \in A$ as a shallow multicontext. Observe that in our definition of shallow multicontext we include trees of the form $b(a)$. This is because, as mentioned earlier, we will often replace a subforest by a node having the same type and therefore it is convenient to immediately have access to this type by looking at the label of that node.

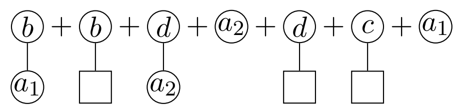

Figure 3: Illustration of a shallow multicontext of arity 3

Let x be a node of a forest s. Let $t = t_1 + \cdots + t_\ell$ be the subforest of x, composed of ℓ trees. The *shallow multicontext of x in s* is the sequence $p_1 + \cdots + p_\ell$ such that $p_i := a$ if $t_i = a \in A$, $p_i := b(a)$ if $t_i = b(a)$, and $p_i := b(\square)$ if $t_i = b(s')$, where $a \in A, b \in B$ and $s' \notin A$. A shallow multicontext p *occurs* in a forest s iff there exists a node x of s such that p is the shallow multicontext of x in s. In the rest of the paper, a node x will almost always be considered together with the shallow multicontext p occurring at x. For this reason we will write "let (p, x) be a node of a tree t" when x is a node of t and p is the shallow multicontext at x. Similarly, if P is a set of shallow multicontexts, we will write "let $(p, x) \in P$" when $p \in P$ and x is a node of p.

Given a shallow multicontext p of arity n and a sequence T of n forests, $p[T]$ denotes the forest obtained after placing the i^{th} forest of T at the i^{th} port of p. Moreover, given a node x of p whose unique child is a port (i.e. x is the root of a tree of the form $b(\square)$ within p) and a sequence T of $n-1$ forests, $p[T, x]$ denotes the context obtained as above but leaving the subtree at x unchanged.

***P*-Valid Forests and Contexts.** Let P be a set of shallow multicontexts. Later on P will be a key parameter for the induction. We say that a forest t is *P-valid* iff it has more than one node and all shallow multicontexts occurring in t are in P. Similarly we define the notion of a P-valid context. Note that we distinguish forests with one node in the definition. This is a technical restriction that will be convenient without harming the generality of the argument as the omitted forests are definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. We extend the notion of P-validity to elements of H and V. We say $h \in H$ is P-valid iff there exists a P-valid forest t such that $h = \alpha(t)$. Similarly for $v \in V$.

***P*-Reachability.** The logic $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ can be seen as a two-way logic navigating up and down within forests. Over strings, this two-way behavior is reflected by two partial orders over the syntactic semigroup capturing respectively the current knowledge when reading the string from left to right and from right to left, and correspond to the Green's relations L and R.

Over forests it turns out that the relevant bottom-up order is a partial order on forest types while the relevant top-down order is a partial order on context types [Boj09]. In our case those are even parametrized by a set P of shallow multicontexts and are called *P-reachability*. The index within these orders will be another parameter in our induction.

Let h, h' be two P-valid forest types, h is said to be P-reachable from the forest type h' if there exists a P-valid context type v such that $h = vh'$. Two forest types are P-equivalent if they are mutually P-reachable.

The definition is similar for context types and is defined for any $v, v' \in V$, not just P-valid ones. Given two contexts $v, v' \in V$ we say that v is *P-reachable* from v' whenever there is a P-valid context type u such that $v = v'u$.

Notice that for both partial orders, if $P \subseteq P'$ then P-reachability implies P'-reachability.

We will reduce the case when all shallow multicontexts of P have arity 1 or less to the string case. On the other hand, when P contains at least one shallow multicontext of arity at least 2 we will make use of the following property:

Claim 5.1. *If P contains a shallow multicontext of arity at least 2 then among P-equivalence classes of P-valid forest types there exists a unique maximal one with respect to P-reachability.*

Proof. Take $p \in P$ of arity $n \geq 2$. Given $h, h' \in H$ that are P-valid, consider t and t' two P-valid forests such that $\alpha(t) = h$ and $\alpha(t') = h'$. Consider the sequence 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 $\alpha(p[T])$ is P-reachable from both h and h'. The result follows. $\square$

In the cases when Claim 5.1 applies we say that P is *branching* and we denote by H_P the maximal class given by Claim 5.1. Finally, we say that a branching set of shallow multicontexts P is *reduced* if all P-valid forest types are mutually reachable, i.e. H_p is the whole set of P-valid forest types.

5.2. $\mathbf{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$-Equivalence for Shallow multicontexts.

It will often be necessary to manipulate shallow multicontexts modulo definability in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$. Typically, when applying the Antichain Composition Lemma with a formula of the form "select all nodes whose shallow multicontext is in P but have no ancestor with that property", it will be necessary that P is definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$.

Definable set of shallow multicontexts. Intuitively, $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ treats a shallow multicontext as a string whose letters are a, $b(a)$, or $b(\square)$. More formally, we define $\mathbb{A}_s$ as the alphabet containing the letters a, $b(\square)$ and $b(a)$ for all $a \in A$ and $b \in B$. We say that b is the *inner-label* of $b(\square)$ or $b(a)$. We see a shallow multicontext p as a string over the alphabet $\mathbb{A}_s$.

For each positive integer k and any two shallow multicontexts p and p', we write $p \equiv_k p'$ for the fact that Spoiler has a winning strategy in the game played on p and p', seen as strings over $\mathbb{A}_s$. In particular, we say that a set P of shallow multicontexts is k-definable iff it is a union of classes of under $\equiv_k$. As the name suggests a set P of k-definable shallow multicontexts is definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$. In particular we get the following claim which is an immediate consequence of Lemma 3.2.

Claim 5.2. *For any k and any k-definable set P of shallow multicontexts, the language of P-valid forests is definable in* $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$.

Definable nodes in shallow multicontexts. It will also sometimes be necessary to refer to an explicit node within a shallow multicontext. Typically, when applying the Antichain Composition Lemma with a formula of the form "select all nodes (p, x) such that ... and having no ancestor with that property". We will of course need to treat (p, x) modulo definability. It would be tempting to use a notion of definability similar to the one used for shallow multicontexts in the previous paragraph. However the notion of *saturation* building from this would give a necessary but not sufficient characterization for languages definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. In particular, the induction in our suffiency proof would not terminate (see Lemma 8.12). Our notion of definability is based on an Ehrenfeucht-Fraïssé game relaxing the rules defined in Section 3.2 in order to grant more power to Duplicator. Moreover it will be useful to parametrize the game by a set $X \subseteq H$. In the sequel X will be another parameter of the induction denoting those forest types for which we are still looking for a formula defining the set of forests for that type. When applying the Antichain Composition Lemma, any forest of type $h \notin X$ can be safely replaced by a node with the appropriate label as the corresponding language is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

Given X, we distinguish between three kinds of nodes within shallow multicontexts: *port-nodes* are nodes that are roots of trees of the form $b(\square)$ with $b \in B$, X*-nodes* are nodes that are roots of trees of the form $b(a)$ with $a \in A, b \in B$ and $\alpha(a) \in X$, finally $\overline{X}$*-nodes* are the remaining nodes, i.e. nodes with label $a \in A$ or roots of trees of the form $b(a)$ with $a \in A, b \in B$ and $\alpha(a) \notin X$.

Let p and p' be two shallow multicontexts, seen as strings over $\mathbb{A}_s$. The k-round X*-relaxed game* between p and p' is defined as in Section 3.2 but tests on labels in this new game are relaxed, making the game easier for Duplicator. Since this alone makes the equivalence too permissive (this yields a non-necessary characterization), we compensate by giving Spoiler a third "safety" move in addition to the usual left and right sibling moves. Spoiler can use this third move only under specific conditions depending on the labels of the nodes the pebbles are currently on. These two modifications achieve the right balance of expressive power.

At the start of each round Spoiler can move one of the two pebbles inside its shallow multicontext to some left or right sibling x. Duplicator must respond by moving the other pebble inside the other shallow multicontext in the same direction to a node y, if x is an $\overline{X}$-node then y must have the same label as x. Otherwise, if x is a port- or X-node, y must be a port- or X-node with the same inner-label. Hence, at any point in the game the pebbles may lie on nodes with labels $b(c), b(c')$ with $c \neq c'$ and $c, c' \in A \cup \{\square\}$. In that particular case Spoiler can use a 'safety' move: he selects one of the two pebbles but does not move it, by hypothesis this pebble is on a node of label $b(\square)$ or $b(a)$ with $\alpha(a) \in X$. Duplicator must then place the other pebble on a node of label $b(\square)$.

Given two nodes (p, x) and (p', x'), we write $(p, x) \cong_k^X (p, x')$ if *i*) p and p' contain the same set of labels in $\mathbb{A}_s$, *ii*) x, x' have the same label and *iii*) Duplicator wins the k-rounds X-relaxed game between p and p' starting at positions (p, x) and (p', x'). It will also be convenient to define $\cong_k^X$ on shallow multicontexts only. We write $p \cong_k^X p'$ iff $(p, x) \cong_k^X (p', x')$ with x, x' the leftmost positions in p, p'. The following claim is immediate from the definitions:

Claim 5.3. *Assume $p \cong_{k+2}^X p'$, then for any port-node $x \in p$ (resp. $x' \in p'$) there exists a port-node $x' \in p'$ (resp $x \in p$) such that $(p, x) \cong_k^X (p', x')$.*

Proof. Set y, y' as the leftmost positions in p, p'. In the $(k+2)$-rounds X-relaxed game between (p, y) and (p', y'), Spoiler can use the two initial rounds to move the pebble to x' in p' (resp. to x in p) and then use (if necessary) a safety move. Duplicator's strategy yields the desired port-node x in p (resp. $x' \in p'$). □

It is not immediate from the definitions that $\cong_k^X$ is an equivalence relation (transitivity is not obvious). We prove it in the next lemma.

Lemma 5.4. *For all $X \subseteq H$ and all k, $\cong_k^X$ is an equivalence relation.*

Proof. It is clear from the definitions that the relation is reflexive and symmetric. We now prove transitivity. Assume $(p, x) \cong_k^X (p', x')$ and $(p', x') \cong_k^X (p'', x'')$. We want to show that $(p, x) \cong_k^X (p'', x'')$. It is clear that p and p'' contain the same set of labels and that x and x' have the same label. We need to prove that Duplicator has a winning strategy in the k-rounds X-relaxed game between (p, x) and (p'', x'').

By hypothesis, Duplicator has winning strategies in the k-rounds X-relaxed games between (p, x) and (p', x') and between (p', x') and (p'', x''). We combine these strategies in the obvious way. Assume that Spoiler makes a non safety move on p, then Duplicator obtains an answer in p' from her strategy in the game on p and p', plays that answer as a move for Spoiler in the game on p' and p'' which yields an answer in p'' from her strategy on that game. This is her answer to Spoiler's move, and it is immediate to check that this is a correct answer. By symmetry she can answer a similar move of Spoiler on p''.

Assume now that Spoiler makes a safety move on p. Observe that this cannot happen in the first-round since x and x'' have the same label. Therefore after this round there will be at most $k-2$ rounds left to play. Let b be the inner label of the pebble on p. Since a safety move was used, the pebbles in p and p'' must have different labels. Hence at least one of these labels is different from the label of the pebble in p'. We distinguish two cases depending on which one it is.

If the pebbles on p' and p'' have different labels, then Duplicator can use a Safety move in the game between p' and p'' to get y'' in p'' with label $b(\square)$ from where she can continue to play the game. If the pebbles on p' and p have different labels, then Duplicator can use a Safety move in the game between p' and p to get y' in p' with label $b(\square)$. We can then use Claim 5.3 to get y'' with label $b(\square)$ and such that $(p'', y'') \cong_{k-2}^X (p', y')$, this is Duplicator's answer. It is correct since the number of rounds left to play is less than $k-2$. □

As for Lemma 3.2 is it easy to show that for all k, $\cong_k^X$ has finite index. Finally, the following claim is a simple variant of Claim 5.2:

Claim 5.5. *Let $X \subseteq H$ and (p, x) be a node. There is a $FO^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ formula $\psi_{p,x}$ having one free variable and such that for any forest s, $\psi_{p,x}$ holds exactly at all nodes (p', x') such that $(p, x) \cong_k^X (p', x')$.*

Proof. It is immediate that $(p, x) \equiv_k (p', x') \Rightarrow (p, x) \cong_k^X (p', x')$ (following the strategy provided by $(p, x) \equiv_k (p', x')$ prevents Spoiler from using any safety move in the X-relaxed game). Hence any $\cong_k^X$-class is a union of $\equiv_k$-classes and the result follows. □

6. Decidable Characterization of $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$

In this section we present our decidable characterization for $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. It involves three properties of the syntactic morphism of the language. As we explained, the first two are simple identities on H and V, the third one, *saturation* is a new notion that is parametrized by a set P of shallow multicontexts and the associated equivalences. We first define saturation and then state the characterization.

6.1. **Saturation.** Let $\alpha : \mathbb{A}^{\Delta} \to (H, V)$ be a morphism into a finite forest algebra (H, V). Note that as for shallow multicontexts while saturation makes sense for any morphism, it is designed to be used with leaf surjective ones. In particular, in the characterization, we state saturation on the leaf completion of the syntactic morphism of the language. Consider some *branching* and *reduced* set P of shallow multicontexts (note that we do *not* ask P to be definable). Recall that P will be the set of allowed patterns. Let H_P be the unique maximal class given by Claim 5.1. Note that since P is reduced, H_P is also the set of P-valid forest types.

Set $k \in \mathbb{N}$. We say that a context Δ is *(P, k)-saturated* if (i) it is P-valid and, (ii) for each port-node $(p, x) \in P$ there exists a port-node (p', x') on the backbone of Δ such that $(p, x) \cong_k^{H_P} (p', x')$. We say that α is *closed under k-saturation* if for all branching and reduced sets P of shallow multicontexts, for all contexts Δ that are (P, k)-saturated and all $h_1, h_2 \in H_P$, we have:

$$\alpha(\Delta)^{\omega} h_1 = \alpha(\Delta)^{\omega} h_2 \tag{6.1}$$

We say that α is *closed under saturation* if it is closed under k-saturation for some k. We will need the following simple observation.

Lemma 6.1. *Let $\alpha : \mathbb{A}^{\Delta} \to (H, V)$ be a morphism into a finite forest algebra (H, V) and k, k' two integers such that $k' > k$. If α is closed under k-saturation then α is closed under k'-saturation.*

Proof. This is immediate since by definition, any Δ that is (P, k')-saturated is (P, k)-saturated as well. □

6.2. **Characterization of $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.** We are now ready to state the main result of this paper.

Theorem 6.2. *A regular forest language L is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ iff its syntactic morphism $\alpha : \mathbb{A}^{\Delta} \to (H, V)$ satisfies the following properties:*
a) H satisfies the equation

$$\omega(h + g) + g + \omega(h + g) = \omega(h + g) \tag{6.2}$$

b) V satisfies the equation

$$(uv)^{\omega} v (uv)^{\omega} = (uv)^{\omega} \tag{6.3}$$

c) the leaf completion of α is closed under saturation.

Notice that (6.2) and (6.3) above are exactly the identities characterizing, over strings, definability in $\mathrm{FO}^2(<)$ (recall Theorem 4.2) and are therefore necessary for being definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. It is easy to see that they are not sufficient to characterize $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. To see this, consider the language of trees that corresponds to Boolean expressions, with AND and OR inner nodes and 0 or 1 leaves, that evaluates to 1. One can verify that the syntactic forest algebra of this language satisfies (6.2) and (6.3). However it is not definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$, actually not even in $\mathrm{FO}(<_{\mathbf{v}}, <_{\mathbf{h}})$ [Pot94].

Recall that $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ can express the fact that a forest is a tree and, for each k, that a tree has rank k, hence Theorem 6.2 also apply for regular tree languages and regular ranked tree languages.

In Section 7 we will prove that the properties listed in the statement of Theorem 6.2 are necessary for having definability in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ using a simple but tedious Ehrenfeucht-Fraïssé argument. In Section 8 we prove the difficult direction of Theorem 6.2, i.e. that the properties imply definability in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

Finally in Section 9 we show that the properties listed in Theorem 6.2 can be effectively tested. This is simple for (6.2) and (6.3) but will require an intricate pumping argument for saturation. Altogether Theorem 6.2 achieves our goal and provides a decidable characterization of regular forest languages definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

7. Correctness of the properties

In this section we prove that the properties stated in Theorem 6.2 are necessary for being definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. We prove that any language L definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ is closed under saturation and its syntactic forest algebra satisfies the identities (6.2) and (6.3).

If L is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ then it is simple to see that its syntactic forest algebra must satisfy the identities (6.2) and (6.3). This is because Identity (6.2) is essentially concerned by sequences of forests with the + operation. Therefore each such sequence can be treated as a string over $<_{\mathbf{h}}$ and Theorem 4.2 can be applied to show that the identity is necessary. Similarly Identity (6.3) concerns only sequences of contexts that can also be treated as strings over $<_{\mathbf{v}}$.

The necessary part of Theorem 6.2 then follows from the following proposition.

Proposition 7.1. *Let L be a forest language that is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. Then the leaf completion of the syntactic morphism of L is closed under saturation.*

The rest of this section is devoted to the proof of Proposition 7.1. It is a classical but tedious Ehrenfeucht-Fraïssé argument.

Assume L is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. It follows from Theorem 3.1 that L is definable in $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h^{-1}})$. Let $\alpha : \mathbb{A}^{\vartriangle} \to (H, V)$ be the leaf completion of its syntactic morphism. Let k be the nesting depth of the navigational modalities used in the formula recognizing L, we prove that α is closed under k-saturation.

Let P be a branching and reduced set of shallow multicontexts. Let $X = H_P$ be the associated class of P-valid forest types. Let Δ be a (P, k)-saturated context. Let $u = \alpha(\Delta)$ and h_1, h_2 be two forest types in H_P. We need to show that $u^\omega h_1 = u^\omega h_2$.

For this we exhibit two forests S_1 and S_2 over $\mathbb{A}$ such that $\alpha(S_1) = u^\omega h_1$ and $\alpha(S_2) = u^\omega h_2$ and such that Duplicator has a winning strategy for the k-move game described in Section 3.2 when playing on S_1 and S_2. Therefore it follows from Lemma 3.2 that no formula of $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{F_h}, \mathbf{F_h}^{-1})$ whose nesting depth of its navigational modalities is less than k can distinguish between the two forests. This implies $u^\omega h_1 = u^\omega h_2$ as desired.

Our agenda is now as follows. In Section 7.1 we define the two forests S_1 and S_2 on which we will play. Then in Section 7.3 we give the winning strategy for Duplicator in the k-move game on S_1 and S_2.

We start with some definitions that will play the key role in the proof. The *root* of a forest is the root of the leftmost tree of that forest. Recall that the *backbone* of a context is the path containing all the ancestors of the port of that context. The *skeleton* of a context is the set of nodes composed of the backbone together with their siblings. Both notions are illustrated in Figure 4.

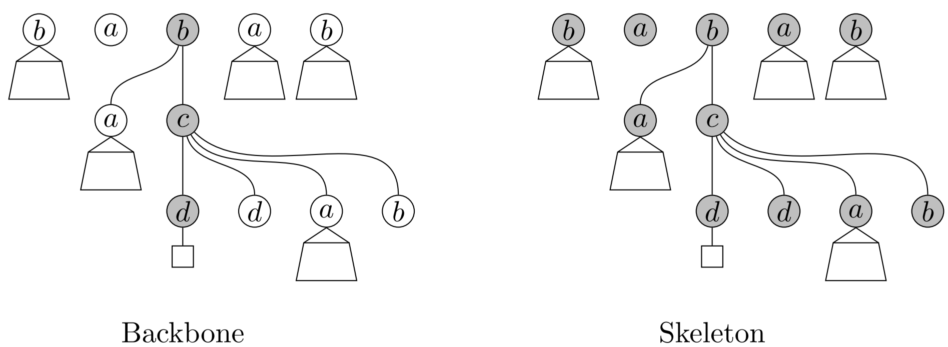

Figure 4: Illustration of the notions of backbone and skeleton.

7.1. **Definition of the forests S_1 and S_2.** Let $X = H_P = \{h_1, \cdots, h_\ell\}$. Recall that since P is reduced, X is also the set of P-valid types. In particular all subforests of a P-valid forest or context that are not leaves have a type in X. We fix a set $\{s_1, \ldots, s_\ell\}$ of P-valid forests such that for all i, $s_i \notin A$ and $\alpha(s_i) = h_i$. This is without loss of generality as for each i, h_i is in H_P and therefore reachable from any type and therefore there is a forest of arbitrary depth of that type.

Given a P-valid context C and ℓ forests $t_1, \cdots, t_\ell$, we say that C' *is the context obtained from C by replacing all subforests of type h_j by t_j* if C' is constructed by considering all the nodes x that are on the skeleton of C but not on the backbone and, if the subforest below x is s where $\alpha(s) = h_j$ with $j \leq \ell$, we replace it with t_j. By construction, C and C' have the same skeleton. Notice that since P is reduced and C is assumed to be P-valid, the construction replaces the subforests below all ports and X-nodes on the skeleton of C that are not on the backbone and leaves the $\overline{X}$-nodes unchanged. Since we assumed that all s_i are not in A, this means that all X-nodes on the skeleton of C become port nodes within C'. Therefore, C' may contain on its backbone shallow multicontexts that are not in P and saturation may not be preserved. We will show how to deal with this fact later.

Since P is branching, there exists a shallow multicontext $q_0 \in P$ of arity greater than 1. For $i \leq \ell$, we denote by V_i the context obtained from q_0 by placing the forest s_i into all the ports of q_0 except for the rightmost one.

By maximality of H_P relative to P-reachability, for all $i \leq \ell$ there exists a P-valid context U_i' such that $h_i = \alpha(U_i')\alpha(V_\ell \cdots V_1)u^\omega h_1$. For $i \leq \ell$, we write U_i for the context

$U'_i V_\ell \cdots V_0$. For all $i \leq \ell$ we write $u_i = \alpha(U_i)$. By definition, for $i \leq \ell$, the contexts U_i have the following properties:

- U_i is P-valid,
- $u_i u^\omega h_1 = h_i$,
- the context U_i contains for all $j \leq \ell$ a subforest of type h_j such that all nodes on the path to this copy are port-nodes (namely s_j within V_j).

We now construct by induction on j contexts Δ_j and $U_{i,j}$, and forests $T_{i,j}$ such that for all $i \leq \ell$, we have $\alpha(\Delta_j) = u$, $\alpha(U_{i,j}) = u_i$ and $\alpha(T_{i,j}) = h_i$. We initialize the process by setting for all $i \leq \ell$:

- $U_{i,0}$ is formed from U_i by replacing all subforests of type h_j by s_j,
- Δ_0 is obtained from Δ by replacing all subforests of type h_j with s_j,
- $T_{i,0} := U_{i,0} \cdot (\Delta_0)^\omega \cdot s_1$.

By construction we have $\alpha(\Delta_0) = u$, $\alpha(U_{i,0}) = u_i$ and $\alpha(T_{i,0}) = u_i u^\omega h_1 = h_i$ as desired. When $j > 0$, the inductive step of the construction is done as follows for all $i \leq \ell$:

- $U_{i,j}$ is formed from U_i by replacing each subforest of type $h_{i'}$ by $T_{i',j-1}$ (see Figure 5),
- Δ_j is formed from Δ by replacing each subforest of type $h_{i'}$ by $T_{i',j-1}$,
- $T_{i,j} = U_{i,j} \cdot (\Delta_j)^\omega \cdots (\Delta_0)^\omega \cdot s_1$.

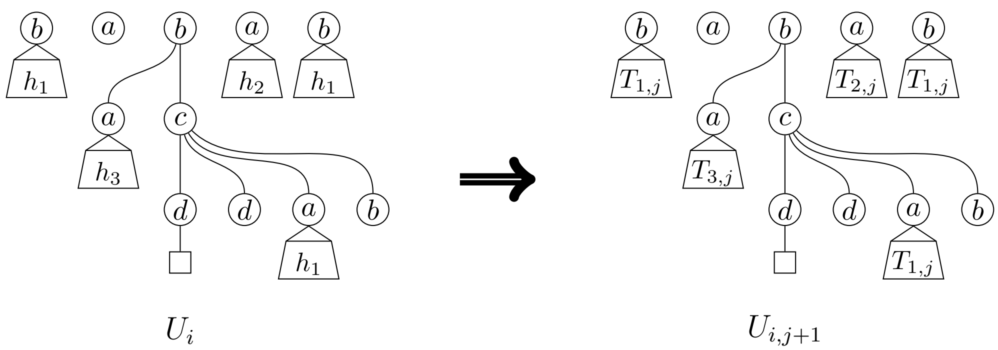

Figure 5: Illustration of the construction of $U_{i,j}$ from U_i: each subforest of type $h_{i'}$ in U_i is replaced by $T_{i',j-1}$.

By induction we have for all $j \leq \ell$, $\alpha(U_{i,j}) = \alpha(U_i) = u_i$, $\alpha(\Delta_j) = \alpha(\Delta) = u$ and $\alpha(T_{i,j}) = u_i u^\omega h_1 = h_i$ as required. Notice that each $U_{i,j}$ contains a copy of $T_{i',j-1}$ for all $i' \leq \ell$. Let $m = 2^k$ and let:

$$
\begin{aligned}
S_1 &:= (\Delta_m)^{(m+1)\omega} \cdot T_{1,m} \\
S_2 &:= (\Delta_m)^{(m+1)\omega} \cdot T_{2,m}
\end{aligned}
\tag{7.1}
$$

Note that by definition $\alpha(S_1) = u^\omega h_1$ and $\alpha(S_2) = u^\omega h_2$. Therefore the following lemma concludes the proof of Proposition 7.1.

Lemma 7.2. *Duplicator has a winning strategy for the k-move game between S_1 and S_2.*

7.2. **Basic Properties.** Before proving Lemma 7.2 we state some basic properties of the construction.

Recall that the operation constructing C' from a P-valid C by replacing all subforests of type h_j by t_j does not preserve saturation. The issue is that all X-nodes become port-nodes and the resulting shallow multicontext may no longer be equivalent to one occurring in the backbone of C. In order to cope with this problem, we remember what the initial situation was. If x is a port-node on the skeleton of C' (recall that C and C' have the same skeleton) we say that:

- x is an *ex-port-node* if x, as a node of C, was a port-node.
- x is an *ex-X-node* if x, as a node of C, was a X-node.

Observe that, by construction, for any port-node x of C', the subtree at x in C' is $b(t_j)$ for some j and some $b \in B$. For ex-X-nodes, this is by construction. For ex-port-nodes, this is because P is reduced and C was P-valid, hence all subforets of C but leaves had type in H_P. Also notice that all remaining nodes of C' are $\overline{X}$-nodes and are unchanged with respect to C.

Let x'_1 (resp. x'_2) be a node on the skeleton of a context C'_1 (resp. C'_2) constructed from a P-valid context C_1 (resp. C_2) by replacing all subforests of type h_i by T_{i,j_1} for some fixed j_1 (resp. T_{i,j_2} for some fixed j_2). Let x_1, x_2 be the corresponding nodes on the skeletons of C_1, C_2. We write $x'_1 \sim_n^X x'_2$ when $x_1 \cong_n^X x_2$. Because $\cong_n^X$ is an equivalence, it is straightforward to verify that $\sim_n^X$ is also an equivalence relation.

There is a game definition of $\sim_n^X$, called pseudo-X-relaxed game. In this game Spoiler can now use the safety move when one pebble is on a ex-port-node and the other on a ex-X-node or when both pebbles are on ex-X-nodes with subtrees $b(T_{i_1,j_1}), b(T_{i_2,j_2})$ such that $i_1 \neq i_2$. In this case Duplicator must answer by placing the other pebble on a ex-port-node.

By construction the following property is an immediate consequence of the (P,k)-saturation of Δ: Assume C is P-valid and that C' is constructed from C by replacing all subforests of type h_i by $T_{i,j}$ for some fixed j. Then for any n and any ex-port-node x of C' there exists a y in the **backbone** of Δ_n such that $x \sim_k^X y$. We call this property, the *pseudo-saturation* of Δ_n.

Notice that when using *pseudo-saturation* in her strategy, Duplicator may end up in a situation where the pebbles are above subtrees T_{i,j_1} and T_{i,j_2} with $j_1 \neq j_2$. Note that T_{i,j_1} and T_{i,j_2} are essentially the same tree, only with different nesting. In this situation, Duplicator may have to play a subgame within the trees T_{i,j_1} and T_{i,j_2}. The following lemma states that this is possible as soon as j_1, j_2 are large enough.

Lemma 7.3. *Given integers i, n, j_1, j_2 such that $2(n-1)+1 \leq j_1, j_2$, Duplicator has a winning strategy for the n-move game between T_{i,j_1} and T_{i,j_2}.*

Proof. This can be proved by a simple induction on n. Essentially this is a straightforward generalization of a classical argument used to prove that the words w^{j_1} and w^{j_2} cannot be distinguished by a first-order formula of fixed quantifier rank provided that j_1, j_2 are large enough (see [Str94] for example). □

7.3. **The winning strategy: Proof of Lemma 7.2.** We give a winning strategy for Duplicator in the k-move game between S_1 and S_2. In order to be able to formulate this strategy, we first define the useful parameters and their key properties that we will later use.

The *backbone* of S_1 (S_2) is the path going through all the ports of the copies of Δ_m within S_1 (S_2) and the *skeleton* of S_1 is the set of nodes within the backbone of S_1 together with their siblings.

The *nesting level* of a node x of S_1 or S_2 is the minimal number j such that x belongs to a context Δ_j or $U_{i,j}$. We set the nesting level of the nodes that are in any copy of a forest $s_1, \dots, s_\ell$ to 0. The notion of nesting level is illustrated in Figure 6. Note that this number is equal in siblings and may only increase when going up in the tree. When this number is low, we are near the leaves of the forest and we need to make sure that the current positions of the game point to isomorphic subtrees. Recall that because of the construction of the context $U_{i,j}$, a node of nesting level j always has, for all $i' \leq \ell$, a descendant that is at the root of a copy of $T_{i',j-1}$ and, for all $j' < j$ a descendant that is at the root of a copy of $\Delta_{j'}$.

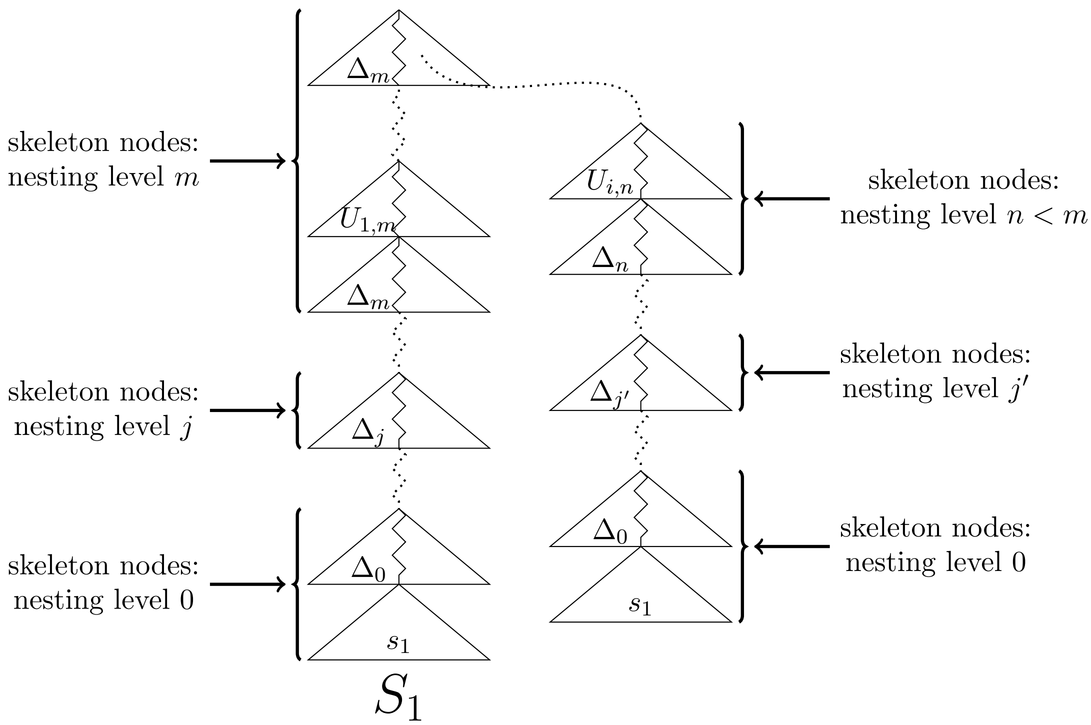

Figure 6: Illustration of the notion of nesting level in the proof of Lemma 7.2.

The *upward number* of a node $x \in S_1$ (or $x \in S_2$) is the number of occurrences of Δ_m in the path from x to the root of S_1 (see Figure 7). When this number is low, we are near the roots and we need to make sure the current positions are identical. Fortunately the two forests S_1 and S_2 are identical up to a certain depth. This number is equal in siblings. When moving up in the tree this number may only decrease and, by construction, it can only decrease when traversing a copy of Δ_m and therefore the resulting node must be on the backbone of S_1 (S_2).

Given a node $x \in S_1$ (or $x \in S_2$), the *horizontal number* of x is the maximal number $n \leq k$ such that for all strict ancestors y of x, there exists a node z on the backbone of Δ_m such that $y \sim_n^X z$. Note that this number is equal in siblings and can only increase when going up in the tree. Recall also that if y is an ex-port-node by pseudo-saturation $y \sim_k^X z$

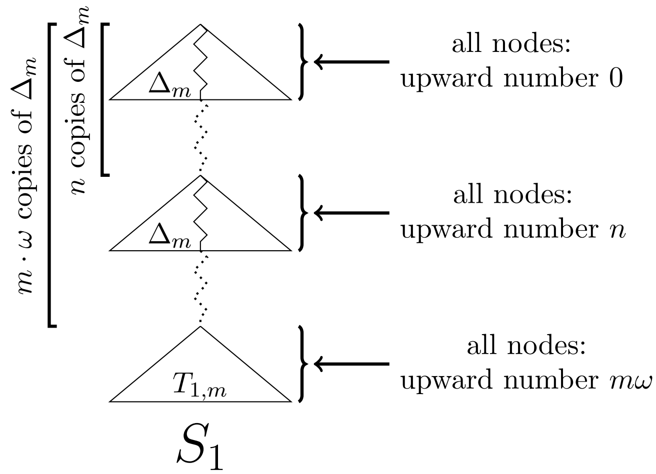

Figure 7: Illustration of the notion of upward number in the proof of Lemma 7.2.

for some z on the backbone of Δ_m. Hence, if all the strict ancestors of x are ex-port-nodes, then its horizontal number is k. In particular all nodes x in the skeleton of S_1 (or S_2) have horizontal number k.

We now state a property $\mathcal{P}(n)$ that depends on an integer n and two nodes $x \in S_1$ and $y \in S_2$. We then show that when $\mathcal{P}(n+1)$ holds in a game starting at x, y, then Duplicator can play one move while enforcing $\mathcal{P}(n)$. As it is easy to see that $\mathcal{P}(k)$ holds for the roots of S_1 and S_2, this will conclude the proof of Lemma 7.2. The inductive property $\mathcal{P}(n)$ is a disjunction of three cases:

(1) There exist ancestors $\hat{x}, \hat{y}$ of x, y such that $\hat{x}$ and $\hat{y}$ have nesting level $\geq 2n+1$, upward number $\geq n$ and horizontal number $\geq n$. Furthermore, either
   (a) $\hat{x} \sim_n^X \hat{y}$ and Duplicator has a winning strategy in the n-move game played on the *subtrees at* $\hat{x}$ and $\hat{y}$, starting at positions x and y, or,
   (b) Duplicator has a winning strategy in the n-move game played on the *subforests of* $\hat{x}$ and $\hat{y}$, starting at positions x and y.
(2) The nodes x and y are at the same relative position within the copy of the context $(\Delta_m)^{m\omega}$ in their respective forest.
(3) The upward numbers of x and y are $\geq n$, their nesting levels are $\geq 2n+1$ and their horizontal number are $\geq n$. Moreover, we have $x \sim_n^X y$.

Observe that there is a factor 2 involved in the conditions on the nesting levels of the nodes. We need this factor in order to be able to use Lemma 7.3.

Assume we are in a situation where $\mathcal{P}(n+1)$ holds. We show how Duplicator can play to enforce $\mathcal{P}(n)$. The strategy depends on why $\mathcal{P}(n+1)$ holds. In all cases we assume that Spoiler moves the pebble from x in S_1. The case when Spoiler moves the pebble from y in S_2 is symmetrical. Recall that $n < k$, and $m = 2^k > 2n$.

7.3.1. **Case 1.** $\mathcal{P}(n+1)$ holds because of Item (1).

In this case we have two nodes $\hat{x}$, and $\hat{y}$ satisfying the properties of Item (1).

• *Spoiler moves from x to a node that is still in the subtree at $\hat{x}$.* In that case, Duplicator simply responds in the subtree at $\hat{y}$ using the strategy provided by Item (1) of $\mathcal{P}(n+1)$ and $\mathcal{P}(n)$ holds because Item (1) remains true.

• *Spoiler moves to a sibling x' of $\hat{x}$.* This can only occur if $x = \hat{x}$ and $y = \hat{y}$. If *a*) holds, by hypothesis we have $\hat{x} \sim^X_{n+1} \hat{y}$, therefore, Duplicator can answer with a sibling y' of $\hat{y}$ such that $x' \sim^X_n y'$. Since x' (y') is a sibling of $\hat{x}$ ($\hat{y}$), it has the same upward number, nesting level and horizontal number as $\hat{x}$ ($\hat{y}$). Hence by hypothesis, all those numbers satisfy Item (3) of $\mathcal{P}(n)$ and we are done. If *b*) holds Duplicator simply responds in the subforest of $\hat{y}$ using the strategy provided by *b*) and $\mathcal{P}(n)$ holds because Item (1.b) remains true.

• *Spoiler moves to an ancestor x' of $\hat{x}$.*

Assume first that the upward number of x' is $< n$. Recall that by hypothesis the upward number of $\hat{x}$ is $\geq n+1$. Hence x' is on the backbone of S_1. As, by hypothesis, $\hat{y}$ has also an upward number $\geq (n+1)$, the copy y' of x' in the other forest is also an ancestor of $\hat{y}$. Duplicator then selects y' as her answer, satisfying Item (2) of $\mathcal{P}(n)$.

Assume now that the upward number of x' is $\geq n$. Since the horizontal number of $\hat{x}$ is $\geq n+1$, there exists a node z on the skeleton of Δ_m such that $x' \sim^X_{n+1} z$. By hypothesis the upward number of $\hat{y}$ is $\geq (n+1)$. Hence we can find above $\hat{y}$ an occurrence of Δ_m of upward number $\geq n$. Duplicator answers by the copy of z in this occurrence of Δ_m. By construction, x', y' have upward numbers $\geq n$. Moreover x' (y') is an ancestor of $\hat{x}$ ($\hat{y}$) and therefore has a bigger nesting level. As by hypothesis the latter was $\geq 2(n+1)+1$, x' and y' have nesting level $\geq 2n+1$. For the same reason the horizontal number of x' is larger than the one of $\hat{x}$ and is therefore $\geq n$. It follows that Item (3) of $\mathcal{P}(n)$ is satisfied.

7.3.2. **Case 2.** $\mathcal{P}(n+1)$ holds because of Item (2).

In this case x and y are at the same relative position within the copy of the context $(\Delta_m)^{m\omega}$ in their respective forest.

• *Spoiler moves to a node x' that remains within the context $(\Delta_m)^{m\omega}$.* Then Duplicator copies this move in the other forest and $\mathcal{P}(n)$ is satisfied because of Item (2).

• *Spoiler moves to a node x' that is within the subforest $T_{1,m}$ of S_1.* In particular this means that x, y are on the backbones of S_1, S_2. By construction the subforest $T_{2,m}$ of S_2 contains at least one copy of the forest $T_{1,m-1}$ that can be chosen such that all nodes occurring on the path to this copy are ex-port-nodes. It follows from Lemma 7.3 that there exists a node y' in $T_{1,m-1}$ such that Duplicator has a winning strategy in the n-move game played on $T_{1,m}$ and $T_{1,m-1}$, starting at positions x' and y'. This is Duplicator's answer.

Set $\hat{x}$ and $\hat{y}$ as the roots of the copies $T_{1,m}$ and $T_{1,m-1}$ in S_1 and S_2. Observe that by construction, $\hat{x}$ and $\hat{y}$ have nesting level $\geq m-1 \geq 2n+1$, upward number $m\omega \geq n$ and horizontal number $k \geq n$. Moreover, by choice of y', Duplicator has a winning strategy in the n-move game played on the subforest of $\hat{x}$ and $\hat{y}$, starting at positions x' and y'. We conclude that $\mathcal{P}(n)$ holds because of Item (1.b).

7.3.3. **Case 3.** $\mathcal{P}(n+1)$ holds because of Item (3).

In this case the upward numbers of x and y are $\geq n+1$, their nesting levels are $\geq 2(n+1)+1$ and their horizontal number are $\geq n+1$. Moreover we have $x \sim^X_{n+1} y$.

• *Spoiler moves horizontally.* Then Duplicator moves according to the winning strategy provided by $x \sim^X_{n+1} y$ and Item (3) of $\mathcal{P}(n)$ holds.

• *Spoiler moves to an ancestor x' of x.*

If the upward number of x' is $< n$, as the upward number of $x \geq n+1$, x' must be on the backbone of S_1. Duplicator answers by the copy y' of x' in the other forest, satisfying Item (2) of $\mathcal{P}(n)$. Note that the upward number of y is $\geq n+1$. Therefore y', having an upward number $< n$ is indeed an ancestor of y.

If the upward number of x' is $\geq n$. By hypothesis, the horizontal number of x is $\geq n+1$, therefore, there exists a node z in the skeleton of Δ_m such that $x' \sim_{n+1}^{X} z$. By hypothesis the upward number of y is $\geq n+1$. Hence we can find above y an occurrence of Δ_m of upward number n. Duplicator answers by the copy y' of z in this occurrence of Δ_m. By construction we have $x' \sim_n^X y'$. By hypothesis, for x', and by construction, for y', both have upward number $\geq n$. As this is a move up, the nesting level increases and therefore remains $\geq 2n+1$. Hence Item (3) of $\mathcal{P}(n)$ is satisfied.

• *Spoiler moves down to some node x'.* Note that this means that x is a port-node and therefore either an ex-X-node or an ex-port-node. Moreover, since $x \sim_{n+1}^X y$, this is also true of y. Set T_{i_x,j_x} and T_{i_y,j_y} as the subforests below x and y. Observe that by hypothesis $j_x, j_y \geq 2(n+1)$. We distinguish two cases:

Assume first that x and y are both ex-X-nodes and $i_x = i_y$. Set $\hat{x}$ as x and $\hat{y}$ as y. Using Lemma 7.3, we get that Duplicator wins the $(n+1)$-moves game played on the subtrees at $\hat{x}$ and $\hat{y}$. This gives Duplicator's answer y' to x' and Item (1.a) of $\mathcal{P}(n)$ holds.

Otherwise, we use pseudo-saturation to prove that there exists a node z on the backbone of Δ_m such that $z \sim_n^X x \sim_n^X y$ and provide an answer satisfying Item (1.b) of $\mathcal{P}(n)$ for Duplicator.

When either x or y is an ex-port-node node, the existence of z is immediate from pseudo-saturation and transitivity of $\sim_n^X$. In the only remaining case, x and y are both ex-X-nodes and $i_x \neq i_y$. Therefore, Spoiler is allowed to use the safety move in the pseudo-X-relaxed game $x \sim_{n+1}^X y$, and we get a ex-port-node z' in the shallow multicontext of y such that $z' \sim_n^X x$. By pseudo-saturation we then obtain z on the backbone of Δ_m such that $z \sim_k^X z'$. By transitivity, we get that $z \sim_n^X x \sim_n^X y$.

We can now describe Duplicator's answer. By hypothesis, x' belongs to T_{i_x,j_x} and by definition T_{i_y,j_y} contains at least one copy of the forest T_{i_x,j_y-1} that can be chosen such that all nodes occurring on the path to this copy are ex-port-nodes. Since $j_x, j_y \geq 2(n+1)$, it follows from Lemma 7.3 that there exists a node y' in T_{i_x,j_y-1} such that Duplicator has a winning strategy in the n-move game played on T_{i_x,j_x} and T_{i_x,j_y-1}, starting at positions x' and y'. This is Duplicator's answer.

Set $\hat{x}$ and $\hat{y}$ as the roots of the copies T_{i_x,j_x} and T_{i_x,j_x-1} in S_1 and S_2. Observe that by definition, $\hat{x}$ and $\hat{y}$ have nesting level $j_x, j_y - 1 \geq 2n+1$ and upward numbers $\geq n+1 > n$ (the same as x, y). Moreover, all ancestors of $\hat{x}, \hat{y}$ are either ancestors of x, y, ex-port-nodes or x, y themselves. Since we proved that there exists z on the backbone of Δ_m such that $z \sim_n^X x \sim_n^X y$, it follows that $\hat{x}, \hat{y}$ have horizontal number $\geq n$. Finally, by choice of y', Duplicator has a winning strategy in the n-move game played on the subforests of $\hat{x}$ and $\hat{y}$, starting at positions x' and y'. We conclude that $\mathcal{P}(n)$ holds because of Item (1.b).

This concludes the proof of Lemma 7.2 and therefore the proof of Proposition 7.1.

8. Sufficiency of the properties

For this section we fix a regular forest language L recognized by a morphism $\alpha : \mathbb{A}^{\triangle} \to (H, V)$ into a finite forest algebra (H, V). Assume that H and V satisfy Identities (6.2) and (6.3) and that the leaf completion of α is closed under saturation. We prove that any language recognized by α, including L, is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$, concluding the proof of Theorem 6.2.

Recall that given a forest s (a context p) we refer to its image by α as the *forest type* of s (the *context type* of p). In view of Lemma 4.1, we assume without loss of generality that α itself is leaf surjective and closed under saturation. By definition, this implies in particular that for each $h \in H$ there exists a tree consisting of a single node whose forest type is h.

As mentioned earlier, we will often manipulate shallow multicontexts modulo $\equiv_k$ for some fixed integer k. We start by defining a suitable k. Given a shallow multicontext q and a forest s we denote by $q[\bar{s}]$ the forest constructed from q by placing s at each port of q.

Lemma 8.1. *There exists a number k' such that for all $k \geq k'$, for all shallow multicontexts $p \equiv_k p'$ and for all forests s, $p[\bar{s}]$ and $p'[\bar{s}]$ have the same forest type.*

Proof. This is a consequence of Theorem 4.2 and the fact that H satisfies Identity (6.2). Consider strings over H as alphabet and the natural morphism $\beta : H^+ \to H$. Since H satisfies Identity (6.2), it follows from Theorem 4.2 that for every $h \in H$, $\beta^{-1}(h)$ is definable using a formula of φ_h of $\mathrm{FO}^2(<)$. We choose k' as the maximal rank of all these formulas.

Let $k \geq k'$ and take $p \equiv_k p'$ and s some forest. Let $t_1, \dots, t_n$ be the sequence of trees occurring in $p[\bar{s}]$ and $t'_1, \dots, t'_{n'}$ be the sequence of trees occurring in $p'[\bar{s}]$. For all i let $h_i = \alpha(t_i)$ and $h'_i = \alpha(t'_i)$. As $p \equiv_k p'$ the strings $h_1 \dots h_n$ and $h'_1 \dots h'_{n'}$ satisfy the same formulas of $\mathrm{FO}^2(<)$ of rank k' over the alphabet H. Let $h = \beta(h_1 \dots h_n)$, by our choice of k' it follows that $h'_1 \dots h'_{n'} \models \varphi_h$. Hence $\beta(h_1 \dots h_n) = h = \beta(h'_1 \dots h'_{n'})$. Therefore $\alpha(p[\bar{s}]) = \alpha(p'[\bar{s}])$. □

As α is closed under saturation, there is an integer k'' such that α is closed under k''-saturation. We set k as the maximum of k' as given by Lemma 8.1 and k''. By Lemma 6.1, α remains closed under k-saturation. Recall that a set P of shallow multicontexts is k-definable if it is a union of equivalence classes of $\equiv_k$.

Recall that V^1 is the monoid obtained from V by adding a neutral element 1_V. For each $h \in H$, $v \in V^1$ and each set P of shallow multicontexts let

$$L^P_{v,h} = \{t \mid v\alpha(t) = h \text{ and } t \text{ is } P\text{-valid}\}$$

Our goal in this section is to show that:

Proposition 8.2. *For all $h \in H$, all $v \in V^1$ and all sets P of shallow multicontexts, there exists a language definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ that agrees with $L^P_{v,h}$ on P-valid forests.*

Theorem 6.2 is a direct consequence of Proposition 8.2. Let L' be the union of all definable languages resulting from applying Proposition 8.2 to all $L^P_{v,h}$ where $h \in \alpha(L)$, $v = 1_V$ and P is the set of all shallow multicontexts. By definition L' is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ and agrees with L on all P-valid forests. Hence $L = L' \cup \{a \in A \mid a \in L\}$ which is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$.

The remainder of this section is devoted to the proof of Proposition 8.2. Assume that v, h and P are fixed as in the statement of the proposition, we prove that there exists a definable language that agrees on $L^P_{v,h}$ on P-valid forests. We begin by considering the special case when P is not branching (i.e. contains only shallow multicontexts of arity 0 or 1). In that case we conclude directly by applying Theorem 4.2.

8.1. **Special Case: P is not branching.** In this case we treat our forests as strings and use the known results on strings. Since all shallow multicontexts in P have arity 0 or 1, any P-valid forest t is of the form:

$$c_1 \cdots c_k s$$

where k is possibly 0 and 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 $u \in V^1$ and $g \in H$, consider the languages:

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

Notice that $L^P_{v,h}$ is the union of those languages where $vug = h$. We show that for any u and g, there exists a language definable in $\text{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ that agrees with $M_{u,g}$ on P-valid forests. This will conclude this case.

By definition shallow multicontexts of arity 1 are contexts. Let $\{v_1, ..., v_n\}$ be the context types that are images of shallow multicontexts of arity 1 in P.

Let P' be the set of shallow multicontexts from P of arity 1. Let $p, p' \in P'$, by Lemma 8.1 if $p \equiv_k p'$ for all forests s, $p[s]$ and $p'[s]$ have the same forest type. Hence, p and p' have the same context type. This means that for all v_i the set of shallow multicontexts of context type v_i is k-definable. Therefore, by Claim 5.2, there is a formula $\theta_{v_i}(x)$ of $\text{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ testing whether the shallow multicontext of x has v_i as forest type.

Let $\Gamma = \{d_1, ..., d_n\}$ be an alphabet and define a morphism $\beta : \Gamma^* \to V$ by $\beta(d_i) = v_i$. Since V satisfies Identity (6.3), for each $u \in V$ there is a $\text{FO}^2(<)$ formula φ_u such that the strings of Γ^* satisfying φ_u are the strings of type u under β. From φ_u we construct a formula Ψ_u of $\text{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ defining all P'-valid contexts having u as context type. This is done by replacing in φ_u all atomic formulas $P_{d_i}(x)$ with $\theta_{v_i}(x)$. We can also easily define in $\text{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ the set of shallow multicontexts of arity 0 such that $\alpha(s) = g$. After combining this last formula with Ψ_u we get the desired language definable in $\text{FO}^2(<_\mathbf{v}, <_\mathbf{h})$ and agreeing with $M_{u,g}$ on P-valid forests.

In the remainder of the proof we assume that P is branching, i.e. it contains one shallow multicontext of arity at least 2. Recall, that by Claim 5.1, it follows that there exists a unique maximal P-reachable class H_P. The rest of the proof is by induction on three parameters that we now define.

8.2. **Induction Parameters.** The first and most important of our induction parameters is the size of the set of P-valid forest types. We denote this set by X. Observe that by definition $H_P \subseteq X$.

Our second parameter is an index defined on sets P of shallow multicontexts. During the proof we will construct from P new sets P' by replacing some of their port-nodes with X-nodes. Our definition ensures that the index of P' will be smaller than the index of P, hence guarantees termination of the induction. It is based on following preorder on shallow multicontexts called simulation modulo X.

Given two shallow multicontexts p and p', we say that p *simulates* p' *modulo* X if p' is obtained from p by replacing some of its port-nodes $b(\square)$ by an X-node $b(a)$ with the same inner label. Observe that simulation modulo X is a partial order.

For each shallow multicontext p its X-*number* is the number of non $\cong^X_{k+2}$-equivalent shallow multicontexts q (not necessarily in P) that can be simulated modulo X by p. For each set P of shallow multicontexts the n-*index of* P is the number of non $\cong^X_{k+2}$-equivalent shallow multicontexts $p \in P$ of X-number n. Our second induction parameter, called the *index of* P, is the sequence of its n-indexes ordered by decreasing n.

The third parameter is based on v. Consider the preorder on context types defined by the quotient of the P-reachability relation by the P-equivalence relation. The *P-depth* of a context type v is the maximal length of a path in this preorder from the empty context to v.

We prove Proposition 8.2 by induction on the following three parameters, given below in their order of importance:

(i) $|X|$
(ii) the index of P
(iii) the P-depth of v

We distinguish three cases: a base case and two cases in which we will use antichain composition and induction. We say that a context type u *P-preserves* v if v is P-reachable from vu. A context c *P-preserves* v if its context-type P-preserves v.

8.3. **Base Case: P is reduced and v is P-preserved by a (P,k)-saturated context Δ.** We use saturation to prove that v is constant over the set $X = H_P$ of P-valid forest types, i.e. for any P-valid $h_1, h_2 \in H$, $vh_1 = vh_2$. Since all forests in $L^P_{v,h}$ are P-valid, it follows that $L^P_{v,h}$ is either empty or the language of all P-valid forests. The desired definable language is therefore either the empty language of the language of all forests.

Since Δ P-preserves v, there exists a P-valid context c such that $v\alpha(\Delta c) = v$. It follows that $v = v\alpha(\Delta c)^\omega$. Moreover, observe that since Δ is P-saturated and c is P-valid, Δc is P-saturated as well. It then follows from saturation that

$$vh_1 = v\alpha(\Delta c)^\omega h_1 = v\alpha(\Delta c)^\omega h_2 = vh_2$$

This terminates the proof of the base case. We now consider two cases in which we conclude by induction.

8.4. **Case 1: P is not reduced, Bottom-Up Induction.** By definition, since P is not reduced there exists a P-valid forest type $g \in X \setminus H_P$. We choose g to be minimal with respect to P-reachability, i.e., any P-valid forest type g' is either P-equivalent to g or g is not P-reachable from g'. Let G be the set of P-valid forest types that are P-equivalent g. Observe that by minimality of g, in any P-valid forest s whose type is in G, all subforests of s that are not single leaves have a forest type in G (recall that a subforest consists of *all* the children of some node). Moreover, by choice of g, $G \cap H_P = \emptyset$ and all $g' \in X$ such that G is P-reachable from g' are in G. We obtain the desired definable language for $L^P_{v,h}$ via the Antichain Composition Lemma using languages that we prove to be definable in $FO^2(<_{\mathbf{v}},<_{\mathbf{h}})$ by induction on $|X|$. Correctness of the construction relies on both Equation (6.2) and Equation (6.3).

Outline. Our agenda is as follows. We construct from P a k-definable set P' and prove that a P-valid forest has a type in G iff it contains no shallow multicontext of P'. Since k-definable sets of shallow multicontexts can be expressed in $FO^2(<_{\mathbf{v}},<_{\mathbf{h}})$, we use P' to define an antichain formula φ which selects all positions whose subforest contains a shallow multicontext in P' (i.e. has a forest type outside G) but have no descendant with that property (i.e. all descendants of the position have a subforest of type in G). This formula splits P-valid forests into two parts: a lower part and upper part. In the lower part all subforests have type in $G \subsetneq X$ and in the upper part the set of valid forest types is included in $X \setminus G$. In both cases we get definable languages by induction on $|X|$ we glue them back together using the Antichain Composition Lemma. This situation is illustrated in Figure 8.

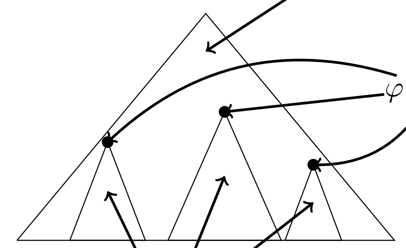

Figure 8: Illustration of the Antichain Composition Lemma for Case 1. The marked are the lowest nodes whose subforest contains a shallow multicontext in P'

Definition of P'. Let s be some arbitrarily chosen P-valid forest such that $\alpha(s) \in G$. We set

$$P' = \{p \;\mid\; \alpha(p[\bar{s}]) \notin G\}$$

We prove in the next lemma that P' is well-defined, i.e. that its definition does not depend on the choice of s.

Lemma 8.3. *Let p be a shallow multicontext of arity n and T and T' be two sequences of n P-valid forests of forest type in G. We have:*

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

Proof. We use Identity (6.3) to prove this lemma. Let $T = (t_1, ..., t_n)$ 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 port and t'_j by t_j for $j > i$. Notice that by hypothesis on p, T and T', c_i is P-valid for all $i \leq n$. For all $i \leq n$, we write $u_i = \alpha(c_i)$, $h_i = \alpha(t_i)$ and $h'_i = \alpha(t'_i)$. We first show that:

$$\forall i \leq n,\; u_i h_i \in G \;\Leftrightarrow\; u_i h'_i \in G \tag{8.1}$$

Assuming that $u_i h_i \in G$, we show that $u_i h'_i \in G$. By symmetry this will prove (8.1). As G is closed under mutual P-reachability, it is enough to show that $u_i h'_i$ is mutually P-reachable from h'_i. By definition $u_i h'_i$ is P-reachable from h'_i, therefore it remains to show that h'_i is P-reachable from $u_i h'_i$. From $u_i h_i \in G$ we get that h'_i is P-reachable from $u_i h_i$ and therefore there is a P-valid context u such that $h'_i = u u_i h_i$. By hypothesis h_i is P-reachable from h'_i and therefore there exists a P-valid context u' such that $h_i = u' h'_i$. A little bit of algebra and Identity (6.3) yields:

$$\begin{aligned} h'_i &= uu_iu'h'_i \\ &= (uu_iu')^{\omega+1}h'_i \\ &= uu_i(u'uu_i)^{\omega}u'h'_i \\ &= uu_i(u'uu_i)^{\omega}uu_i(u'uu_i)^{\omega}u'h'_i && \text{using Identity (6.3)} \\ &= (uu_iu')^{\omega}uu_i(uu_iu')^{\omega+1}h'_i \\ &= (uu_iu')^{\omega}u\ u_ih'_i \end{aligned}$$

as $(uu_iu')^{\omega}u$ is P-valid, h'_i is P-reachable from $u_ih'_i$ and (8.1) is proved.

For concluding the proof of the lemma, notice that by construction $\alpha(p[T]) = u_1h_1$, $\alpha(p[T']) = u_nh'_n$ and $u_ih'_i = u_{i+1}h_{i+1}$. As from (8.1) we get $u_ih_i \in G$ iff $u_ih'_i \in G$, this implies by induction on i that for all $i \leq n$, $u_1h_1 \in G$ iff $u_ih_i \in G$ iff $u_ih'_i \in G$. The case $i = n$ proves the lemma. □

We now prove that P' can be used to test whether a P-valid forest has a type in G.

Lemma 8.4. *A P-valid forest has type in G iff it is $(P \setminus P')$-valid.*

Proof. This is a consequence of Lemma 8.3. Let t be a P-valid forest such that $\alpha(t) \notin G$. We prove that t contains a shallow multicontext of P'. Consider a minimal subforest t' of t whose type is not in G. Then we have $t' = p[T]$ where p is a shallow multicontext and T a sequence of forests of forest type in G (possibly empty if p is of arity 0). Let T' be the sequence $\bar{s}$ for some s with $\alpha(s) \in G$. By Lemma 8.3 $\alpha(p[T']) \notin G$ and therefore $p \in P'$.

Conversely, if $\alpha(t) \in G$, by minimality of G, all subforests of t are in G. It is then immediate by definition of P' and Lemma 8.3 that t cannot contain a shallow multicontext of P'. □

Setting up the Composition. Let φ be the antichain formula which holds at port-nodes (p, x) such that $p \in P'$ and x has no descendant with that property. It follows from the next lemma that φ is expressible in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$.

Lemma 8.5. *P' is k-definable.*

Proof. This is a consequence of Lemma 8.1 (which is itself a consequence of (6.2)). Set $p \in P'$ and $p' \equiv_k p$. We prove that $p' \in P'$. By definition of P', $\alpha(p[\bar{s}]) \notin G$. As $p \equiv_k p'$, by choice of k and Lemma 8.1 we get $\alpha(p'[\bar{s}]) = \alpha(p[\bar{s}])$. Hence $\alpha(p'[\bar{s}]) \notin G$ and $p' \in P'$. □

We now define the languages that we will use to apply the Antichain Composition Lemma.

Lemma 8.6. *For any $g \in G$, there exists a language definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ that agrees with $L^P_{1_V,g}$ on P-valid forests.*

Proof. Notice that the set of all $(P \setminus P')$-valid forest types is $G \subsetneq X$. Hence by induction on the first parameter in Proposition 8.2 there exists a language L_1 definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ that agrees with $L^{(P\setminus P')}_{1_V,g}$ on $(P \setminus P')$-valid forests. By Lemma 8.4, a P-valid forest has type $g \in G$ iff it is $(P \setminus P')$-valid. Hence L_1 agrees with $L^P_{1_V,g}$ on P-valid forests. □

Assume $G = \{g_1, \cdots, g_n\}$. For all $i \leq n$, let L_i be a language definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ that agrees with $L^P_{1_V, g_i}$ on P-valid forests given by Lemma 8.6.

Let Q be the set of shallow multicontexts q that can be obtained from some $p \in P$ by replacing some port-nodes (possibly none) with G-nodes of the same inner label and such that either:

- q has arity greater than 1 (i.e. one port-node of p was left unchanged)
- or q has arity 0 and $p \in P'$ (hence $\alpha(q) \notin G$).

We have:

Lemma 8.7. *There is a language K definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ that agrees with $L^Q_{v,h}$ on Q-valid forests.*

Proof. Let Y be the set of Q-valid forest types. We prove that $Y \subseteq X$ and $Y \cap G = \emptyset$. It will follow that $|Y| < |X|$. Hence K is obtained by applying Proposition 8.2 by induction on the first parameter.

Let $h \in Y$, by definition, there exists a Q-valid forest s such that $\alpha(s) = h$. All shallow multicontexts $q \in Q$ occurring in s are constructed from $p \in P$ by replacing some port-nodes of p with G-nodes. As G contains only P-valid forest types, for any $g \in G$ there exists a P-valid forest whose type is g. By replacing the newly introduced G-nodes in s by the correspond P-valid forest with the same type we get a P-valid forest s' whose type remains h. Hence $h \in X$. Moreover, for any shallow multicontext of arity 0 occurring in s, the corresponding shallow multicontext occurring in s' must belong to P'. It follows that s' contains at least one shallow multicontext in P' and by Lemma 8.4 that $h \notin G$. □

Applying Antichain Composition. We now apply the Antichain Composition Lemma to the languages K, and $L_1 \cdots L_n$ defined above. The situation is depicted in Figure 8.

Recall that $G = \{g_1, \cdots, g_n\}$. For any $i \leq n$, let $a_i \in A$ be such that $\alpha(a_i) = g_i$. Set $L = \{t \mid t[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n] \in K\}$. Since $K, L_1, \ldots, L_n$ are definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$, it follows from Lemma 3.3 that L is definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. We terminate the proof by proving that L agrees with $L^P_{v,h}$ on P-valid forests.

Lemma 8.8. *Let t be a P-valid forest, then $\alpha(t) = \alpha(t[(L_1, \varphi) \to a_1, \cdots, (L_n, \varphi) \to a_n])$.*

Proof. This is immediate by definition of Q, K and $L_1 \cdots L_n$. □

8.5. **Case 2: P is reduced but there exists no (P, k)-saturated context Δ that P-preserves v, Top-Down Induction.** In this case we use again the Antichain Composition Lemma using languages that we prove to be definable by induction on the index of P and the P-depth of v. Recall that since P is reduced, $X = H_P$. Correctness relies on Identity (6.3).

Outline. We proceed as follows. First we use our hypothesis to define a port-node $(p, x) \in P$ with the following properties. For any P-valid forest t and port-node (p', x') of t such that $(p, x) \cong^X_k (p', x')$, the context c obtained from t by replacing the subforest below x' by a port does not P-preserve v. Since by Claim 5.5 all such nodes (p', x') can be defined in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$, this gives an antichain formula φ which selects such nodes having no ancestor with that property. Such a formula splits a forest in two parts: an upper part and a lower part. For the upper part, we will prove that the set of occurring shallow multicontexts has smaller index and use induction on that parameter. Moreover, observe that by choice of

(p, x), each subforest in the lower part is below a context that has larger P-depth than v, we will use induction on this parameter. This situation is depicted in Figure 9.

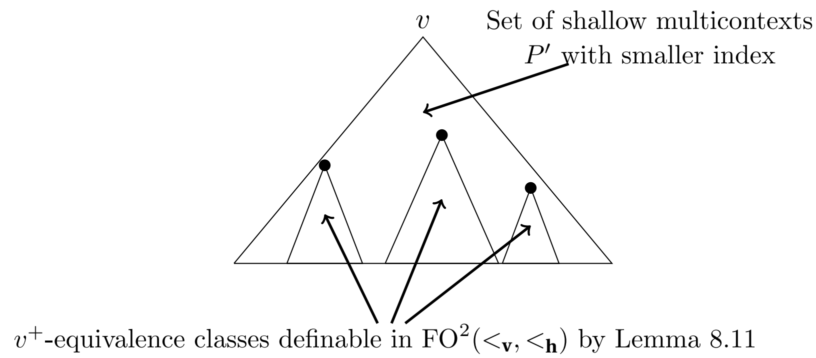

Figure 9: Illustration of the Antichain Composition Lemma for Case 2. The marked nodes are the topmost nodes equivalent to (p, x).

Definition of (p, x)**.** Let (p, x) be a port-node in P. We say that (p, x) is *P-bad for v* iff there exists no P-valid context c satisfying the two following properties:

(1) c P-preserves v.
(2) the port-node (p', x') above the port of c verifies $(p, x) \cong_k^X (p', x')$.

Lemma 8.9. *There exists a port-node $(p, x) \in P$ that is P-bad for v.*

Proof. This is where Identity (6.3) is used. We proceed by contradiction and assume that no port-node $(p, x) \in P$ is P-bad for v.

By definition, for all port-nodes $(p, x) \in P$ we get a P-valid context $c_{p,x}$ that P-preserves v and such that the port-node (p', x') above the port of $c_{p,x}$ verifies $(p, x) \cong_k^X (p', x')$. Note that since $\cong_k^X$ is of finite index, we may assume that there are finitely many different contexts $c_{p,x}$ for all $(p, x) \in P$. Let Δ be the context obtained by concatenating all these finitely many contexts $c_{p,x}$ for all (p, x). By definition Δ is (P, k)-saturated. We use Identity (6.3) to prove that Δ P-preserves v which contradicts the hypothesis of this case. This is an immediate consequence of the next claim.

Claim 8.10. *Let $u, u' \in V$ such that both u and u' P-preserve v. Then uu' P-preserves v.*

We finish the proof of Lemma 8.9 by proving Claim 8.10. By hypothesis, we have $w, w' \in V$ that are P-valid and such that $vuw = v$ and $vu'w' = v$. Set $e = (wu'w'u)^\omega$, a little algebra yields $vue = vu$. Applying Identity (6.3), we get that

$$vu = vue = vueu'w'ue = vuu'w'ue$$

Hence $v = vuu'w'uew$ and since $w'uew$ is P-valid, this terminates the proof. □

Setting-up the Composition. For the remainder of the proof we set $(p, x) \in P$ as a port-node which is P-bad for v as given by Lemma 8.9. We define our antichain formula φ as the formula holding exactly at all port-nodes (p', x') such that $(p, x) \cong_k^X (p, x')$ and having no ancestor with that property. By definition, φ is antichain and by Claim 5.5, φ is expressible in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$. We now define the languages $L_1, \ldots, L_n$ and K necessary for applying the Antichain Composition Lemma.

Given two elements g and g' of H, we say that g is v^+-equivalent to g' if for all context types u which do not P-preserve v (hence the P-depth of vu is strictly higher that the P-depth of v) we have $vug = vug'$. Set $\{\gamma_1, \dots, \gamma_n\}$ as the set of all v^+-equivalence classes. For all i, we define $M_i = \{s \mid \alpha(s) \in \gamma_i \text{ and } s\ P\text{-valid}\}$.

Lemma 8.11. *For all i, there is a language L_i definable in* $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ *that agrees with M_i on P-valid forests.*

Proof. Fix a v^+-equivalence class γ_i and let $g \in \gamma_i$. For any u such that vu is not P-reachable from v, by induction in Proposition 8.2, the third parameter has increased and the other two are unchanged, there is a language K_u definable in $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ that agrees with $L^P_{vu,vug}$ on P-valid forests. The lemma then follows by taking for L_i the intersection of all languages K_u for u such that vu is not P-reachable from v. □

Let $P' = \{p' \mid p \cong^X_{k+2} p'\}$. Observe that by Claim 5.3, any shallow multicontext p' in P' contain at least one position x' such that $(p, x) \cong^X_k (p', x')$. For $p' \in P'$, let $x_1, \cdots, x_\ell$ be all the port-nodes of p' such that $(p, x) \cong^X_k (p', x_i)$. Let $b(\Box)$ be the label of all the x_i in p'. Let $\Delta_{p'}$ be the set of all the shallow multicontexts that are constructed from p' by replacing at all the positions x_i, $b(\Box)$ by a label $b(a)$ (possibly different for each position), for a such that $\alpha(a) \in X$. Let $\hat{P}$ be the union of all $\Delta_{p'}$ for $p' \in P'$. Finally, let $Q = (P \setminus P') \cup \hat{P}$.

Lemma 8.12. *There is a language K definable in* $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ *that agrees with $L^Q_{v,h}$ on Q-definable forests.*

Proof. Let Y be the set of all Q-valid forest types. We first observe that $Y \subseteq X$. The argument is similar to the one in the proof of Lemma 8.7 as the newly introduced shallow multicontexts can be represented by P-definable forests. If $Y \subsetneq X$, then the lemma follows by induction on the first parameter in Proposition 8.2. Otherwise $X = Y$ and we prove that Q has smaller index than P. The result is then immediate by induction on the second parameter in Proposition 8.2. Note that this is where we use the fact that our notion of equivalence between positions is weaker than $\equiv_k$. With a stronger notion, it would not be possible to prove that the index has decreased.

Set $n \in \mathbb{N}$ as the largest integer such that there exists a shallow multicontext in P' with X-number n. We prove that all $\hat{p} \in \hat{P}$ have a X-number that is strictly smaller than n. It will then be immediate from the definitions that Q has smaller index than P.

Let $\hat{p} \in \hat{P}$. By definition, there exists $p' \in P'$ and $x_1, \cdots, x_\ell \in p'$ such that for all i, $(p, x) \cong^X_k (p', x_i)$ and replacing the labels $b(\Box)$ at all positions x_i in p' by $b(a)$ for $\alpha(a) \in X$ yields $\hat{p}$. In particular this means that p' simulates $\hat{p}$ modulo X and that the X-number of $\hat{p}$ is smaller or equal to that of p' and hence smaller or equal to n. We prove that $\hat{p}$ does not simulate any $p'' \cong^X_{k+2} p'$ modulo X. It will follow that the inequality is strict which terminates the proof.

We proceed by contradiction, assume that there exists $p'' \cong^X_{k+2} p'$ such that $\hat{p}$ simulates p'' modulo X. By definition, $p'' \cong^X_{k+2} p' \cong^X_{k+2} p$, hence, by Claim 5.3, p'' contains a port-node x'' such that $(p'', x'') \cong^X_k (p, x)$. By definition of simulation, x'' corresponds to a port-node $\hat{x}$ in $\hat{p}$ and a port-node x' in p'. Moreover, since $\hat{x}$ is a port-node, this means that $x' \notin \{x_1, \cdots, x_\ell\}$, i.e. (p', x') is not $\cong^X_k$-equivalent to (p, x). This contradicts the following claim.

Claim 8.13. *$(p'', x'') \cong^X_k (p', x') \cong^X_k (p, x)$.*

It remains to prove Claim 8.13. We prove that $(p'',x'') \cong_k^X (p',x')$. By definition, p' and p'' use the same set of labels in $\mathbb{A}_s$ and x',x'' have the same label $b(\Box)$. We give a winning strategy for Duplicator in the X-relaxed game between (p'',x'') and (p',x'). By definition p'' is obtained from p' by replacing some port nodes with X-nodes with the same inner-node label. Therefore as long as Spoiler does not use a safety move, Duplicator can answer by playing the isomorphism. Assume now that Spoiler does a safety move. Then the pebbles are on positions $z' \in p'$ and $z'' \in p''$ with labels $b(\Box)$ and $b(a)$ as Duplicator's strategy disallow any other possibility such as $b(a), b(a')$ where $a \neq a'$. Is Spoiler selects z'', then Duplicator continues to play the isomorphism by leaving the other pebble on z'. If Spoiler selects z', observe that since $p' \cong_{k+2}^X p''$, we can use Claim 5.3 and get a node $y'' \in p''$ such that $(p',x') \cong_k^X (p'',y'')$, this is Duplicator's answer. Duplicator can then continue to play by using the strategy given by $(p',x') \cong_k^X (p'',y'')$.

□

Applying Antichain Composition. Let K and $L_1 \cdots L_n$ be languages definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ as given by Lemma 8.12 and Lemma 8.11. For all i let $a_i \in A$ be such that $\alpha(a_i) \in \gamma_i$. Set $L = \{t \mid t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_n] \in K\}$. It follows from Lemma 3.3 that L is definable in $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$. We terminate the proof by proving that L agrees with $L^P_{v,h}$ on P-valid forests.

Lemma 8.14. *For any P-valid forest t, $v\alpha(t) = v\alpha(t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k])$.*

Proof. This is because (p,x) is P-bad for v. The proof goes by induction on the number of occurrences in t of port-node (q,y) such that $(q,y) \cong_k^X (p,x)$. If there is no occurrence, this is immediate as the substitution does nothing.

Consider a node y of a shallow multicontext q such that $(q,y) \cong_k^X (p,x)$ and no node above y satisfies that property. Let s be the subforest below y in t and let i be such that $\alpha(s) \in \gamma_i$. Let c be the context formed from t by replacing s by a port and let u_c be its type. Since (p,x) is P-bad for v, u_c does not P-preserve v. Hence, $v\alpha(t) = vu_c\alpha(s) = vu_c\alpha(a_i)$ by definition of v^+-equivalence.

We write $t' = ca_i$, we already know that $v\alpha(t') = v\alpha(t)$. Observe that by construction $t'[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k]$ is $t[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k]$. By induction we have that $v\alpha(t') = v\alpha(t'[(L_1,\varphi) \to a_1, \cdots, (L_k,\varphi) \to a_k])$ which terminates the proof. □

9. Decidability

In this section we prove that the characterization of $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ given in Theorem 6.2 is decidable.

Theorem 9.1. *Let L be a regular language of forests. It is decidable whether L is definable in* $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$.

In view of Theorem 6.2 the decision procedure works as follows. From L we first compute its syntactic morphism $\alpha : \mathbb{A}^\Delta \to (H,V)$. Then we check that (6.2) holds in H, that (6.3) holds in V and that α is closed under saturation. This is straightforward for (6.2) and (6.3) as H and V contain only finitely many elements. However it is not obvious from the definitions that closure under saturation can be decided. The main result of this section is an algorithm which, given as input a morphism α, decides whether α is closed under saturation.

Recall the definition of saturation. It requires the existence of a number k such that for all branching and reduced sets P of shallow multicontexts and all (P, k)-saturated contexts a property holds. The main problem is that all these quantifications range over infinite sets. In the first part of this section we introduce an "abstract" version of these sets with finitely many objects together with an associated "abstract" notion of saturation and show that closure under saturation corresponds to closure under the abstract notion of saturation.

Then, in the remaining part of the section we present an algorithm that computes the needed abstract sets.

9.1. **Abstraction.** Let $\mathbb{A} = (A, B)$ be a finite alphabet and $\alpha : \mathbb{A}^\Delta \to (H, V)$ be a morphism into a finite forest algebra (H, V). Recall that we see shallow multicontexts as strings over $\mathbb{A}_s$, i.e. as elements of $\mathbb{A}_s^+$. In order to stay consistent with our notation on shallow multicontexts, we will denote by $+$ the concatenation within $\mathbb{A}_s^+$. Recall that if $Q \subseteq \mathbb{A}_s^+$ is a set of shallow multicontexts, then we write $(p, x) \in Q$ instead of x is a node of some shallow multicontext $p \in Q$.

We start with some terminology. Let p be a shallow multicontext of arity n and let $G \subseteq H$. We denote by $p(G)$ the set of forest types $h \in H$ such that there exists a sequence T of n forests which all have a type in G and such that $\alpha(p[T]) = h$. For a port-node x of p, we denote by $p(G, x)$ the set of context types $v \in V$ such that there exists a sequence T of $n-1$ forests which have all a type in G such that $\alpha(p[T, x]) = v$. If x is not a port-node of p then we set $p(G, x) = \emptyset$ for all G. The following fact is immediate.

Fact 9.2. *Let (p, x) and (q, y) be nodes and $r = p + q$. Then for any $G \subseteq H$, $r(G) = p(G) + q(G)$, $r(G, x) = p(G, x) + q(G)$ and $r(G, y) = p(G) + q(G, y)$.*

Abstracting shallow multicontexts: Profiles. We now define an abstract version of positions in shallow multicontexts that we call *profiles*.

Consider a pair (p, x) where p is a shallow multicontext and x a position in p. The profile of (p, x), denoted $\beta(p, x)$ is the quadruple $\mathbf{v} = (i, \mathbb{B}_s, f_H, f_V)$ where

(1) $i \in \{0, 1, 2\}$ is the *arity* of p counted up to threshold 2,
(2) $\mathbb{B}_s \subseteq \mathbb{A}_s$ is the *alphabet* p, i.e. the set of labels used in p,
(3) $f_H : 2^H \to 2^H$ is the *forest mapping* of p, defined as the mapping $G \mapsto p(G)$,
(4) $f_V : 2^H \to 2^V$ is the *context mapping* of (p, x), defined as the mapping $G \mapsto p(G, x)$.

Observe that if p has arity 0 (i.e. p is a forest) then f_H is the mapping $G \mapsto \{\alpha(p)\}$. Moreover, whenever x is not a port-node, f_V is the mapping $G \mapsto \emptyset$. We let $\mathbf{P}$ be the set of profiles of all shallow multicontexts. Observe that $\mathbf{P}$ is finite:

$$\mathbf{P} \subseteq \{0, 1, 2\} \times 2^{\mathbb{A}_s} \times (2^H)^{2^H} \times (2^V)^{2^H}$$

In the rest of this section we shall denote by $\mathbf{u}, \mathbf{v}, \dots$ profiles (elements of $\mathbf{P}$), by $\mathbf{U}, \mathbf{V}, \dots$ sets of profiles (subsets of $\mathbf{P}$) and by $\mathcal{U}, \mathcal{V}, \dots$ sets of sets of profiles (subsets of $2^{\mathbf{P}}$).

Let us first present two semigroup operations for $\mathbf{P}$. Both operations are adapted from the concatenation operation between shallow multicontexts. If (p, x) and (p', x') are pairs where p, p' are shallow multicontexts and x, x' are positions of p, p', then one can use concatenation to construct two new pairs: $(p + p', x)$ in which we keep the position x of p and $(p + p', x')$ in which we keep the position x' of p'.

Abstracted on profiles, this yields the two following operations. Let $\mathbf{v}, \mathbf{v}' \in \mathbf{P}$ be two profiles and set $(i, \mathbb{B}_s, f_H, f_V) = \mathbf{v}$ and $(i', \mathbb{B}'_s, f'_H, f'_V) = \mathbf{v}'$. We define, two new profiles $\mathbf{v} +_\ell \mathbf{v}' \in \mathbf{P}$ and $\mathbf{v} +_r \mathbf{v}' \in \mathbf{P}$ as follows,

$\mathbf{v} +_\ell \mathbf{v}' = (j, \mathbb{C}_s, g_H, g_V)$ with
- $j = min(i + i', 2)$
- $\mathbb{C}_s = \mathbb{B}_s \cup \mathbb{B}'_s$.
- $g_H : G \mapsto f_H(G) + f'_H(G)$.
- $g_V : G \mapsto f_V(G) + f'_H(G)$.

$\mathbf{v} +_r \mathbf{v}' = (j, \mathbb{C}_s, g_H, g'_V)$ with
- $j = min(i + i', 2)$
- $\mathbb{C}_s = \mathbb{B}_s \cup \mathbb{B}'_s$.
- $g_H : G \mapsto f_H(G) + f'_H(G)$.
- $g'_V : G \mapsto f_H(G) + f'_V(G)$.

On the shallow multicontext level, the definition exactly means that for any (p, x), $(p', x') \in \mathbb{A}_s^+$ such that $\mathbf{v} = \beta(p, x)$ and $\mathbf{v}' = \beta(p', x')$, we have $\mathbf{v} +_\ell \mathbf{v}' = \beta(p + p', x)$ and $\mathbf{v} +_r \mathbf{v}' = \beta(p + p', x')$. One can verify that $+_r$ and $+_\ell$ are both semigroup operations. Moreover, the following fact is immediate from the definitions and states that one can use the operations $+_\ell$ and $+_r$ to compute the whole set $\mathbf{P}$ from the profiles of one-letter shallow multicontexts.

Fact 9.1. $\mathbf{P}$ is the smallest subset of $\{0, 1, 2\} \times 2^{\mathbb{A}_s} \times (2^H)^{2^H} \times (2^V)^{2^H}$ such that:
- $\mathbf{P}$ contains the profiles of one-letter shallow multicontexts: for all $c \in \mathbb{A}_s$, $\beta(c, x) \in \mathbf{P}$ (where x is the unique position in c)
- $\mathbf{P}$ is closed under $+_\ell$.
- $\mathbf{P}$ is closed under $+_r$.

Abstracting sets of shallow multicontexts: Configurations. Recall the definition of saturated contexts: let P be a set of shallow multicontexts, a context is (P, k)-saturated iff it is P-valid and for all $(p, x) \in P$, there exists a $\cong_k^X$-equivalent position on the backbone of the context. This means that we need to define an abstraction of sets of shallow multicontexts P that contains two informations:
- the set of P-valid types.
- the set of images under α of (P, k)-saturated contexts.

For this we introduce the notion of *configurations*. Notice that this abstraction needs to be parametrized by the equivalence $\cong_k^X$. In order to do this, we will abstract this equivalence on profiles which are our abstraction of the objects compared by $\cong_k^X$.

There is an issue however. Intuitively, we want two profiles $\mathbf{u}$ and $\mathbf{v}$ to be "equivalent" if one can find (p, x) and (q, y) such that $(p, x) \cong_k^X (q, y)$, $\beta(p, x) = \mathbf{v}$ and $\beta(q, y) = \mathbf{u}$. Unfortunately, this is not the right definition as the relation we obtain is not transitive in general and hence not an equivalence anymore. This is a problem since the definition of a saturated context requires to pick one position (p, x) among a *set* of equivalent ones. We solve this problem by abstracting sets of equivalent positions directly by sets of profiles.

Moreover, if $P \subseteq \mathbb{A}_s^+$, the configuration that abstracts P needs to have exhaustive information about *all* sets of equivalent positions that can be found in P. Therefore we define a configuration as a sets of sets of profiles, i.e. an element of the set:

$$\mathfrak{C} = 2^{2^{\mathbf{P}}}$$

Of course, we are only interested in elements of $\mathfrak{C}$ that correspond to actual sets of shallow multicontexts. Let $k \in \mathbb{N}$ and $X \subseteq H$, we let $\mathfrak{I}_k[\alpha, X]$ be the set of (X, k)-*relevant* configurations:

$\mathfrak{I}_k[\alpha, X] = \{\mathcal{V} \in \mathfrak{C} \mid \exists Q \subseteq \mathbb{A}_s^+$ such that

1. $\forall q, q' \in Q, \exists x, x' \in q, q'$ s.t. $(q, x) \cong_k^X (q', x')$
2. $\forall (q, x) \in Q$ there exist $(q_1, x_1) \cong_k^X \cdots \cong_k^X (q_n, x_n) \cong_k^X (q, x) \in Q$ such that $\{\beta(q_1, x_1), \ldots, \beta(q_n, x_n)\} \in \mathcal{V}$
3. $\forall \mathbf{V} \in \mathcal{V}$ there exist $(q_1, x_1) \cong_k^X \cdots \cong_k^X (q_n, x_n) \in Q$ such that $\mathbf{V} = \{\beta(q_1, x_1), \ldots, \beta(q_n, x_n)\}$

$\}$

Note that condition (1) restricts the definition to sets of shallow multicontexts that are $\cong_k^X$-equivalent. This will later be necessary when computing the sets of relevant configurations. However, when considering saturation, we will actually work with unions of relevant configurations.

This definition takes care of the quantification over the infinite set of sets of shallow multicontexts in the definition of saturation. One quantification still needs to be dealt with: quantification over $k \in \mathbb{N}$. We achieve this by defining the set of X-*relevant* configurations as the intersection of the previous sets for all k:

$$\mathfrak{I}[\alpha, X] = \bigcap_k \mathfrak{I}_k[\alpha, X]$$

We will present an algorithm for computing $\mathfrak{I}[\alpha, X]$ in the second part of this section. The following fact is immediate from the definitions:

Fact 9.3. *For any $k, k' \in \mathbb{N}$ such that $k \leq k'$ and $X \subseteq H$, $\mathfrak{I}[\alpha, X] \subseteq \mathfrak{I}_{k'}[\alpha, X] \subseteq \mathfrak{I}_k[\alpha, X]$. In particular, there exists $\ell \in \mathbb{N}$ such that for all $k \geq \ell$ and all $X \subseteq H$, $\mathfrak{I}[\alpha, X] = \mathfrak{I}_k[\alpha, X]$.*

Note that while proving the existence of ℓ in Fact 9.3 is simple, computing an actual bound on ℓ is more difficult and will be a consequence of our algorithm computing $\mathfrak{I}[\alpha, X]$.

Finally we equip $\mathfrak{C}$ with a semigroup operation by generalizing the operations $+_\ell$ and $+_r$ defined on $\mathbf{P}$. Observe that $+_\ell$ and $+_r$ can be generalized on sets of profiles: we define the sum of two sets as the set of all possible sums of elements of the two sets. If $\mathcal{U}, \mathcal{V} \in \mathfrak{C}$, we can now define $\mathcal{U} + \mathcal{V}$ as the set

$$\{\mathbf{U} +_\ell \bigcup_{\mathbf{V} \in \mathcal{V}} \mathbf{V} \mid \mathbf{U} \in \mathcal{U}\} \cup \{\bigcup_{\mathbf{U} \in \mathcal{U}} \mathbf{U} +_r \mathbf{V} \mid \mathbf{V} \in \mathcal{V}\}$$

The following fact can be verified from the definitions.

Fact 9.4. $(\mathfrak{C}, +)$ *is a semigroup.*

Validity and Reachability for Configurations. Let $\mathcal{V}$ be a configuration and let $\mathbf{V} \in 2^{\mathbf{P}}$ be the union of all sets in $\mathcal{V}$. The set of $\mathcal{V}$-valid forest types is the smallest $X \subseteq H$ such that for every $(i, \mathbb{B}_s, f_H, f_V) \in \mathbf{V}$, $f_H(H) \subseteq X$ when $i = 0$ and $f_H(X) \subseteq X$ otherwise. $\mathcal{V}$-valid context types are defined as the smallest $Y \subseteq V$ such that $Y \cdot Y \subseteq Y$ and for all $(i, \mathbb{B}_s, f_H, f_V) \in \mathbf{V}$, $f_V(X) \subseteq Y$ (with X the set of $\mathcal{V}$-valid forest types).

Finally, given $\mathcal{V}$-valid forest types h and h', we say that h is $\mathcal{V}$-reachable from h' iff there exists $v \in V$ that is $\mathcal{V}$-valid and such that $h = vh'$. The following fact can be verified from the definitions.

Fact 9.5. *Let Q be a set of shallow multicontexts such that $\mathbf{V} = \{\beta(q,x) \mid (q,x) \in Q\}$. Then $h \in H$ (resp. $v \in V$) is $\mathcal{V}$-valid iff it is Q-valid. Moreover, for all $\mathcal{V}$-valid $h, h' \in H$, h is $\mathcal{V}$-reachable from h' iff h is Q-reachable from h'.*

We say that $\mathcal{V}$ is *branching* iff $\mathbf{V}$ contains a profile of arity 2. One can verify that this implies the existence of a maximal $\mathcal{V}$-reachability class denoted $H_{\mathcal{V}}$. Finally, we say that a branching $\mathcal{V}$ is *reduced* when all $\mathcal{V}$-valid forest types are mutually reachable, i.e. $H_{\mathcal{V}}$ is the whole set of $\mathcal{V}$-valid forest types.

Profile Saturation. We are now ready to rephrase saturation as a property of the sets $\mathfrak{I}[\alpha, X]$. Set $X \subseteq H$, we say that a configuration $\mathcal{V} \in \mathfrak{C}$ is *X-compatible* iff it is branching, it is reduced, $H_{\mathcal{V}} = X$ and $\mathcal{V} = \bigcup_i \mathcal{V}_i$ with $\mathcal{V}_i \in \mathfrak{I}[\alpha, X]$ for all i.

Let $\mathcal{V}$ be an X-compatible configuration. We say that $v \in V$ is $\mathcal{V}$-saturated iff there exist $v_1 \cdots v_n = v$ such that:

- for all j, v_j is $\mathcal{V}$-valid.
- for all $\mathbf{V} \in \mathcal{V}$ there exists $(i, \mathbb{B}_s, f_H, f_V) \in \mathbf{V}$ such that either $i = 0$ (i.e. $\mathbf{V}$ abstracts a set of non-port nodes) or $v_j \in f_V(H_{\mathcal{V}})$ for some j.

Let ℓ be as defined in Fact 9.3. The following fact is a simple consequence of the definitions.

Fact 9.6. *Set $X \subseteq H$ and v be an idempotent of V. For every $k \geq \ell$ the following properties are equivalent:*

(1) There exists a branching and reduced $Q \subseteq \mathbb{A}_s^+$ such that $X = H_Q$ and v is the image of some (Q,k)-saturated context.
(2) There exists a X-compatible $\mathcal{V} \in \mathfrak{C}$ such that v is $\mathcal{V}$-saturated.

Proof sketch. From top to bottom. Let Q and v be as in (1). Let Δ be the (Q,k)-saturated context such that $\alpha(\Delta) = v$. We construct $\mathcal{V}$ and $v_1 \cdots v_n$ witnessing (2) as follows. To each $(q,x) \in Q$, we associate the set $\mathbf{U} = \{\beta(q',x') \mid (q',x') \cong_k^X (q,x)\}$. We then set $\mathcal{V}$ as the set of all such sets $\mathbf{U}$. It follows from the definition and Fact 9.5 that $H_Q = X = H_{\mathcal{V}}$. Moreover, $\mathcal{V}$ is by definition a union of elements of $\mathfrak{I}_k[\alpha, X]$ (and hence of $\mathfrak{I}_k[\alpha, X]$ by definition of k) and is therefore X-compatible. It is then immediate to check that the (Q,k)-saturation of Δ implies the existence of $v_1 \cdots v_n$ with the desired properties. Note that we did not use the hypothesis that v is idempotent, it is only required for the other direction.

From bottom to top. Let $\mathcal{V}$ and $v_1 \cdots v_n$ be as required for (2). By definition, we have $\mathcal{V} = \bigcup_i \mathcal{V}_i$ where each $\mathcal{V}_i$ is $H_{\mathcal{V}}$-relevant and therefore $(H_{\mathcal{V}}, k)$-relevant. This means that for all i, there exists a set of shallow multicontext Q_i for $\mathcal{V}_i$ as in the definition of $\mathfrak{I}_k[\alpha, H_{\mathbf{Q}}]$. We set $Q = \bigcup_i Q_i$. It is immediate from the definition of Q and Fact 9.5 that $H_Q = X = H_{\mathcal{V}}$ and that $v_1, \ldots, v_n$ are Q-valid. We construct the desired (Q,k)-saturated context Δ as follows. For any node $(p,x) \in Q$, we construct a Q-valid context of type v having a node (p',x') on its backbone satisfying $(p',x') \cong_k^{H_Q} (p,x)$. It will then suffice to define Δ as the concatenation of all these contexts. Since v is idempotent Δ will have type v as well.

Let (p,x) with p in Q and x a port-node of p, by definition, there exist some i, some $\mathbf{V} \in \mathcal{V}_i$ and some $(i, \mathbb{B}_s, f_H, f_V) \in \mathbf{V}$ such that $(i, \mathbb{B}_s, f_H, f_V) = \beta(q,y)$ with $(p,x) \cong_k^{H_Q} (q,y)$. As x is a port-node, so is y and we have $f_V(H_Q) \neq \emptyset$ and therefore by $\mathcal{V}$-saturation of v, we get $(i', \mathbb{B}'_s, f'_H, f'_V) \in \mathbf{V}$ such that $f'_V(H_Q)$ contains v_j for some j. By definition, we get $(p',x') \cong_k^{H_Q} (q,y) \cong_k^{H_Q} (p,x)$ such that $\beta(p',x') = (i', \mathbb{B}'_s, f'_H, f'_V)$. Hence we can create a Q-valid context of type v_j, with a unique position (p',x') on its backbone. Since $v_1, \ldots, v_n$ are all Q-valid, this context can then be completed into a Q-valid context of type v which terminates the proof. □

We say that α is *closed under profile saturation* iff for all $X \subseteq H$, for all X-compatible $\mathcal{V} \in \mathfrak{C}$, for all $v \in V$ that are $\mathcal{V}$-saturated and all $h_1, h_2 \in H_{\mathcal{V}}$:

$$v^\omega h_1 = v^\omega h_2$$

Observe that all quantifications in the definition range over finite sets. Therefore, if one can compute the X-compatible configurations for all X, one can decide closure under profile saturation by testing all possible combinations. In the next proposition, we prove that this is equivalent to testing closure under saturation.

Proposition 9.7. *Let $\alpha : \mathbb{A}^\Delta \to (H, V)$ be a morphism into a finite forest algebra. Then the following three properties are equivalent:*

(1) α is closed under saturation.
(2) α is closed under ℓ-saturation
(3) α is closed under profile saturation.

Proof. We prove that 1) $\Rightarrow$ 3) $\Rightarrow$ 2) $\Rightarrow$ 1). That 2) $\Rightarrow$ 1) is immediate by definition of saturation.

We now prove 1) $\Rightarrow$ 3). Assume that α is closed under saturation. By Lemma 6.1, α is closed by k-saturation for some $k \geq \ell$. We need to prove that α is closed under profile saturation. Let $X \subseteq H$, $\mathcal{V} \in \mathfrak{C}$ that is X-compatible $\mathcal{V} \in \mathfrak{C}$, $v \in V$ that is $\mathcal{V}$-saturated and $h_1, h_2 \in H_{\mathcal{V}}$. Using Fact 9.6, we get $Q \subseteq \mathbb{A}_s^+$ such that $H_{\mathcal{V}} = H_Q$ and v^ω is the image of some (Q, k)-saturated context. It is now immediate from k-saturation that $v^\omega h_1 = v^\omega h_2$.

It remains to prove that 3) $\Rightarrow$ 2). Assume that α is closed under profile saturation. We need to prove that α is closed under ℓ-saturation. Let $Q \subseteq \mathbb{A}_s^+$, Δ that is (Q, ℓ)-saturated and $h_1, h_2 \in H_Q$. Using Fact 9.6,we get $\mathcal{V} \in \mathfrak{C}$ such that $H_{\mathcal{V}} = H_Q$ and $\alpha(\Delta^\omega)$ is $\mathcal{V}$-saturated. It is now immediate from profile saturation that $\alpha(\Delta)^\omega h_1 = \alpha(\Delta)^\omega h_2$. □

In view of Proposition 9.7, it is enough to show that closure under profile saturation is decidable in order to prove Theorem 9.1. Because all the quantifications inside the definition of profile saturation range over finite sets, it is enough to show that those finite sets, namely $\mathfrak{I}[\alpha, X]$ for all $X \subseteq H$, can be computed.

This is immediate in the case of ranked trees. Indeed for trees of rank l, the set of legal shallow multicontexts is a subset of $\mathbb{A}_s^l$. Therefore $\mathfrak{I}[\alpha, X] = \mathfrak{I}_{l+1}[\alpha, X]$ can now be computed by considering all the finitely many possible sets $Q \subseteq \mathbb{A}_s^l$. Hence Theorem 9.1 is proved for regular languages of ranked trees.

In the general case it is not obvious how to compute $\mathfrak{I}[\alpha, X]$ and this is the goal of the remaining part of this section.

9.2. **Computing the Sets of X-indistinguishable Configurations.** We present an algorithm which, given as input $\alpha : \mathbb{A}^\Delta \to (H, V)$ and $X \subseteq H$, computes the set $\mathfrak{I}[\alpha, X]$. This is a fixpoint algorithm that starts from trivial configurations corresponding to sets of shallow multicontexts that are singletons composed of a single letter shallow multicontext and saturate the set with two operations.

Our first operation is the semigroup operation on $\mathfrak{C}$ (recall Fact 9.4) which corresponds to concatenating shallow multicontexts. Our second and most important operation is derived from a well-known property of FO$^2(<)$ on strings. Let C be a finite string alphabet and let $u, u' \in C^+$ such that u, u' both contain all labels in C. Then for all $k \in \mathbb{N}$ and any $u'' \in C^+$:

$$(u)^k u''(u')^k \equiv_k (u)^k(u')^k$$

In our case however, the situation will be slightly more complicated as we work with the weaker equivalence $\cong_k^X$ in which tests on labels are relaxed.

Remark 9.8. *By definition for any $\mathcal{V} \in \mathfrak{I}[\alpha, X]$ all profiles contained in sets of $\mathcal{V}$ have the same alphabet. Therefore, we will assume implicitly that this is true of all sets of profiles we consider from now and whenever we refer to "the alphabet of $\mathcal{V}$" we mean this common alphabet.*

Fixpoint Algorithm. Recall from Fact 9.4 that $\mathfrak{C}$ is equipped with a semigroup operation. We start with a few definitions about alphabets that we will need in order to present the algorithm. To each alphabet $\mathbb{B}_s \subseteq \mathbb{A}_s$, we associate a configuration $\llbracket \mathbb{B}_s \rrbracket$ as follows,

$$\llbracket \mathbb{B}_s \rrbracket = \{\{\beta(p,x) \mid p \text{ has alphabet } \mathbb{B}_s \text{ and } x \text{ has label } c\} \mid c \in \mathbb{B}_s\} \in \mathfrak{C}$$

Observe that for any $\mathbb{B}_s$, it is simple to compute $\llbracket \mathbb{B}_s \rrbracket$ from α. Indeed, for any $c \in \mathbb{B}_s$, if y denotes the unique position in the shallow multicontext c, one can verify that,

$$\left\{\beta(p,x) \mid \begin{array}{c} p \text{ has alphabet } \mathbb{B}_s \\ \text{and } x \text{ has label } c \end{array}\right\} = \mathbf{P} +_r \{\beta(c,y)\} +_\ell \mathbf{P} \bigcap \{\mathbf{v} \in \mathbf{P} \mid \mathbf{v} \text{ has alphabet } \mathbb{B}_s\}$$

An important remark is that while $\llbracket \mathbb{B}_s \rrbracket$ is a configuration, in general, it is not an X-relevant configuration (for any X). The main idea behind the fixpoint algorithm is that $\llbracket \mathbb{B}_s \rrbracket$ can become X-relevant if one adds "appropriate" X-relevant configurations to its left and to its right. The definition of "appropriate" is based on the notion of *X-approximation* of an alphabet that we define now.

In the X-relaxed game, there are three types of nodes, port-nodes, X-nodes and $\overline{X}$-nodes. Let $\mathbb{B}_s \subseteq \mathbb{A}_s$, and let $c, c' \in \mathbb{B}_s$ we say that c, c' are *$\mathbb{B}_s[X]$-equivalent* iff $c = c'$ or there exists $b(\square) \in \mathbb{B}_s$ such that c, c' are port-nodes or X-nodes labels of inner label b. Finally, an *X-approximation* of $\mathbb{B}_s$ is an alphabet $\mathbb{C}_s \subseteq \mathbb{B}_s$ such for any $c \in \mathbb{B}_s$, there exists $c' \in \mathbb{C}_s$ that is *$\mathbb{B}_s[X]$-equivalent* to c.

We can now present the algorithm. Set $\mathfrak{T}[\alpha] \subseteq \mathfrak{I}[\alpha, X]$ for all X as the set of configurations associated to sets of shallow multicontexts of the form $\{c\}$ where c is a single letter in $\mathbb{A}_s$. More precisely, $\mathfrak{T}[\alpha]$ is the set of configurations:

$$\{\{\{\beta(c,x)\}\} \mid c \in \mathbb{A}_s \text{ and } x \text{ the unique position in } c\}$$

We set $Sat[X, \alpha]$ as the smallest set $S \subseteq \mathfrak{C}$ containing $\mathfrak{T}[\alpha]$ and such that:

(1) For all $\mathcal{V}, \mathcal{V}' \in S$, $\mathcal{V} + \mathcal{V}' \in S$.
(2) For all $\mathbb{B}_s \subseteq \mathbb{A}_s$, if $\mathcal{V}, \mathcal{V}' \in S$ have (possibly different) alphabets that are both X-approximations of $\mathbb{B}_s$, then $\omega\mathcal{V} + \llbracket \mathbb{B}_s \rrbracket + \omega\mathcal{V}' \in S$.

where $\omega = \omega(\mathfrak{C})$. Clearly $Sat[X, \alpha]$ can be computed from α. It is connected to $\mathfrak{I}[\alpha, X]$ via the proposition below. For $\mathcal{U}_1, \mathcal{U}_2 \in \mathfrak{C}$, we write $\mathcal{U}_1 \sqsubseteq \mathcal{U}_2$ iff

(1) For every $\mathbf{V}_1 \in \mathcal{U}_1$ there exists $\mathbf{V}_2 \in \mathcal{U}_2$ such that $\mathbf{V}_1 \subseteq \mathbf{V}_2$.
(2) For every $\mathbf{V}_2 \in \mathcal{U}_2$ there exists $\mathbf{V}_1 \in \mathcal{U}_1$ such that $\mathbf{V}_1 \subseteq \mathbf{V}_2$.

One can verify that $\sqsubseteq$ is a preorder. If $\mathfrak{V} \subseteq \mathfrak{C}$, the *downset* of $\mathfrak{V}$ is set $\downarrow\mathfrak{V} = \{\mathcal{V} \mid \exists \mathcal{U} \in \mathfrak{V}$ such that $\mathcal{V} \sqsubseteq \mathcal{U}\}$.

Fact 9.9. *Within $\mathfrak{C}$, $+$ is compatible with $\sqsubseteq$ (i.e. $\mathcal{V}_1 \sqsubseteq \mathcal{V}_2$ and $\mathcal{U}_1 \sqsubseteq \mathcal{U}_2$ implies $\mathcal{V}_1 + \mathcal{U}_1 \sqsubseteq \mathcal{V}_2 + \mathcal{U}_2$).*

Proposition 9.10. *Let $\ell = 2|\mathbb{A}_s|^2(|\mathfrak{C}|+1)$ and $X \subseteq H$, then for any $k \geq \ell$:*

$$\mathfrak{I}[\alpha, X] = \mathfrak{I}_k[\alpha, X] = \downarrow Sat[X, \alpha]$$

It follows from Proposition 9.10 that $\mathfrak{I}[\alpha, X]$ can be computed for any $X \subseteq H$. By combining this with Proposition 9.7, we obtain the desired corollary:

Corollary 9.11. *Let $\alpha : \mathbb{A}^\Delta \to (H, V)$ be a morphism into a finite forest algebra. It is decidable whether α is closed under saturation.*

Observe that Proposition 9.10 also contains a bound for ℓ in Fact 9.3. This bound is of particular interest: as explained in Proposition 9.7, ℓ is also a bound for saturation, if α is closed under saturation, then it is closed under ℓ-saturation.

It now remains to prove Proposition 9.10. We prove that for any $k \geq \ell$, $\mathfrak{I}[\alpha, X] \subseteq \mathfrak{I}_k[\alpha, X] \subseteq \downarrow Sat[X, \alpha] \subseteq \mathfrak{I}[\alpha, X]$. Observe that $\mathfrak{I}[\alpha, X] \subseteq \mathfrak{I}_k[\alpha, X]$ is immediate by Fact 9.3. We give the two remaining inclusions their own subsections.

9.3. **Proof of Correctness.** We prove that $Sat[X, \alpha] \subseteq \mathfrak{I}[\alpha, X]$. One can then verify that $\downarrow \mathfrak{I}[\alpha, X] = \mathfrak{I}[\alpha, X]$ and therefore that $\downarrow Sat[X, \alpha] \subseteq \mathfrak{I}[\alpha, X]$. Recall that $\mathfrak{I}[\alpha, X]$ is defined as $\bigcap_{k \in \mathbb{N}} \mathfrak{I}_k[\alpha, X]$. This means that it suffices to prove that for all $k \in \mathbb{N}$, $Sat[X, \alpha] \subseteq \mathfrak{I}_k[\alpha, X]$. We fix such a $k \in \mathbb{N}$ for remainder of the proof.

By definition, $\mathfrak{T}[\alpha] \subseteq \mathfrak{I}_k[\alpha, X]$ for every $k \in \mathbb{N}$. We prove that $\mathfrak{I}_k[\alpha, X]$ is closed under the two operations in the definition of Sat. We begin with Operation (1).

Operation (1)**.** Let $\mathcal{V}, \mathcal{V}' \in \mathfrak{I}_k[\alpha, X]$ and let Q, Q' be the sets of shallow multicontexts witnessing the membership of $\mathcal{V}$ and $\mathcal{V}'$ in $\mathfrak{I}_k[\alpha, X]$. Set $R = \{q + q' \mid q \in Q \text{ and } q' \in Q'\}$, we prove that R witnesses the membership of $\mathcal{V} + \mathcal{V}'$ in $\mathfrak{I}_k[\alpha, X]$. This is a consequence of Fact 9.2 and the following lemma:

Lemma 9.12. *Let $(p_1, x_1), (p_2, x_2) \in Q$ such that $(p_1, x_1) \cong_k^X (p_2, x_2)$ and $(p_1', x_1'), (p_2', x_2') \in Q'$ such that $(p_1', x_1') \cong_k^X (p_2', x_2')$. Then if $r_1 = p_1 + p_1'$ and $r_2 = p_2 + p_2'$,*

$$(r_1, x_1) \cong_k^X (r_2, x_2) \quad \textit{and} \quad (r_1, x_1') \cong_k^X (r_2, x_2')$$

Proof. This is a composition lemma whose proof is immediate using Ehrenfeucht-Fraïssé games. □

We have three conditions to check. That (1) holds is immediate from Lemma 9.12. For (2), if $(r, x) \in R$, we have $r = p + p'$ with $p, p' \in Q, Q'$. By symmetry, assume that $x \in p$. By definition of Q, we get $(p_1, x_1), \dots, (p_n, x_n) \in Q$ such that $(p, x) \cong_k^X (p_1, x_1) \cong_k^X \cdots \cong_k^X (p_n, x_n)$ and $\mathbf{V} = \{\beta(p_i, x_i) \mid i \leq n\} \in \mathcal{V}$. Set $\mathbf{V}'$ as the union of all sets in $\mathcal{V}'$ and set R' as the set of pairs (q, y) such that $q = p_j + p''$ with $j \leq n$ and $p'' \in Q'$. By Lemma 9.12 for all $(q, y) \in R'$, $(r, x) \cong_k^X (q, y)$. Moreover, one can verify using Fact 9.2 that

$$\beta(R') = \mathbf{V} +_\ell \mathbf{V}' \in \mathcal{V}$$

It remains to verify (3). Set $\mathbf{U} \in \mathcal{V} + \mathcal{V}'$. By symmetry, assume that $\mathbf{U} = \mathbf{V} +_\ell \mathbf{V}'$ with $\mathbf{V} \in \mathcal{V}$ and $\mathbf{V}'$ the union of all sets of $\mathcal{V}'$. By definition of Q, Q' there exists $(p_1, x_1) \cong_k^X \cdots \cong_k^X (p_n, x_n) \in Q$ such that $\mathbf{V} = \{\beta(p_i, x_i) \mid i \leq n\}$. Using the same set R' of pairs (q, y) as above we get that all pairs in R' are $\cong_k^X$-equivalent and

$$\beta(R') = \mathbf{V} +_\ell \mathbf{V}' = \mathbf{U}$$

Operation (2)**.** Let $\mathbb{B}_s \subseteq \mathbb{A}_s$ and $\mathcal{V}, \mathcal{V}' \in \mathfrak{I}_k[\alpha, X]$ having alphabets $\mathbb{C}_s, \mathbb{C}'_s$ that are X-approximations of $\mathbb{B}_s$ and let $\mathbf{V}$ and $\mathbf{V}'$ be the unions of all sets in $\mathcal{V}$ and $\mathcal{V}'$ respectively. Let Q and Q' be the sets of shallow multicontexts witnessing the membership of $\mathcal{V}$ and $\mathcal{V}'$ into $\mathfrak{I}_k[\alpha, X]$. Furthermore, set R as the set of all shallow multicontexts of alphabet $\mathbb{B}_s$. We prove that $P = k\omega Q + R + k\omega Q'$ witnesses the fact that $k\omega\mathcal{V} + \llbracket \mathbb{B}_s \rrbracket + k\omega\mathcal{V}' = \omega\mathcal{V} + \llbracket \mathbb{B}_s \rrbracket + \omega\mathcal{V}'$ belongs to $\mathfrak{I}_k[\alpha, X]$. Most of the proof is based on the following property of the equivalence $\cong_k^X$.

Lemma 9.13. *Let $\hat{q_1}, \hat{q_2} \in k\omega Q$, $\hat{q_1}'\hat{q_2}' \in k\omega Q'$ and $r_1, r_2 \in R$. Then for every nodes x_1, x_2 of r_1, r_2 with the same label $c \in \mathbb{B}_s$*

$$(\hat{q_1} + r_1 + \hat{q_1}', x_1) \cong_k^X (\hat{q_2} + r_2 + \hat{q_2}', x_2)$$

Proof. We give a winning strategy for Duplicator in the X-relaxed game. We simplify the argument by assuming that $\hat{q_1} = \hat{q_2} = k\omega q$ for some $q \in Q$ and that $\hat{q_1}' = \hat{q_2}' = k\omega q'$ for some $q' \in Q'$. Since all shallow multicontexts in Q (resp. Q') are $\cong_k^X$-equivalent (see Item (1) in the definition of $\mathfrak{I}_k[\alpha, X]$), one can then obtain a strategy for the general case by adapting this special case.

Recall that q (resp. q') has alphabet $\mathbb{C}_s$ (resp. $\mathbb{C}'_s$) and that $\mathbb{C}_s$ and $\mathbb{C}'_s$ are X-approximations of $\mathbb{B}_s$. Therefore, using a standard game argument, one can verify that Duplicator can win k moves of the X-relaxed game between $(k\omega q + r_1 + k\omega q', x_1)$ and $(k\omega q + r_2 + k\omega q', x_2)$ as long as no safety move is played. In case a safety move is played the X-approximation hypothesis guarantees that r_1, r_2 of alphabet $\mathbb{B}_s$ contains the appropriate letter with a port-node label. □

We now prove that the set P satisfies the definition of $\mathfrak{I}_k[\alpha, X]$ for $k\omega\mathcal{V} + \llbracket \mathbb{B}_s \rrbracket + k\omega\mathcal{V}'$. We have three conditions to verify. That (1) holds is immediate from Lemma 9.13.

For (2), consider $(p, x) \in P$. By definition, $p = \hat{q} + r + \hat{q}'$ with $r \in R$, $\hat{q} \in k\omega Q$ and $\hat{q}' \in k\omega Q'$. We treat the case when $x \in r$ (the other cases are treated with a similar argument). Let c be the label of x and let $(r_1, x_1) \dots (r_n, x_n)$ be all nodes such that $r_i \in R$ and x_i is a node of r_i of label c. By definition of $\llbracket \mathbb{B}_s \rrbracket$ and R, we have

$$\mathbf{U} = \{\beta(r_1, x_1), \dots, \beta(r_n, x_n)\} \in \llbracket \mathbb{B}_s \rrbracket$$

Let $(p_1, y_1) \dots (p_m, y_m)$ be all nodes such that $p_i \in P$ and y_i is a node of label c in the "R-part" of p_i. From Lemma 9.13 we have:

$$(p, x) \cong_k^X (p_1, y_1) \cong_k^X \dots \cong_k^X (p_m, y_m)$$

Observe that viewed as nodes of the "R-part" of p_i, the nodes y_i are exactly the nodes x_j. Using Fact 9.2 one can then verify that

$$\{\beta(p_i, y_i) \mid i \leq m\} = \underbrace{\mathbf{V} +_r \cdots +_r \mathbf{V}}_{k\omega \textit{ times}} +_r \mathbf{U} +_\ell \underbrace{\mathbf{V}' +_\ell \cdots +_\ell \mathbf{V}'}_{k\omega \textit{ times}} \in k\omega\mathcal{V} + \llbracket \mathbb{B}_s \rrbracket + k\omega\mathcal{V}'$$

It remains to prove (3). Let $\mathbf{U} \in \omega\mathcal{V} + \llbracket \mathbb{B}_s \rrbracket + \omega\mathcal{V}'$. Again, we concentrate on the case when

$$\mathbf{U} = \underbrace{\mathbf{V} +_r \cdots +_r \mathbf{V}}_{k\omega \textit{ times}} +_r \mathbf{W} +_\ell \underbrace{\mathbf{V}' +_\ell \cdots +_\ell \mathbf{V}'}_{k\omega \textit{ times}}$$

with $\mathbf{W} \in \llbracket \mathbf{R} \rrbracket$ (other cases are treated in a similar way). By definition of $\llbracket \mathbb{B}_s \rrbracket$, we have

$$\mathbf{W} = \{\beta(r_1, x_1), \dots, \beta(r_n, x_n)\} \in \llbracket \mathbb{B}_s \rrbracket$$

with $(r_1, x_1) \dots (r_n, x_n)$ as all nodes such that $r_i \in R$ and x_i is a node of r_i of label c for some fixed c. Let $(p_1, y_1), \dots, (p_m, y_m) \in P$ be all the shallow multicontexts of P such that y_i is a node of p_i with label c in the "R-part" of p_i. By Lemma 9.13, we have

$$(p_1, y_1) \cong_k^X \dots \cong_k^X (p_m, y_m)$$

Observe that viewed as nodes of the "R-part" of p_i, the nodes y_i correspond exactly to the nodes x_j. Using Fact 9.2 one can then verify that

$$\{\beta(p_i, y_i) \mid i \leq m\} = \mathbf{U}$$

9.4. **Proof of Completeness.** Let ℓ be defined as in Proposition 9.10. We prove that for any $k \geq \ell$, $\mathfrak{I}_k[\alpha, X] \subseteq Sat[X, \alpha]$. We will need the following definition.

Let $k \in \mathbb{N}$, $X \subseteq H$. To every shallow multicontext $q \in \mathbb{A}_s^+$, we associate a configuration $\mathcal{G}_k[X](q) \in \mathfrak{I}[\alpha, X]$. For any p, x set $\mathbf{V}_{p,x} = \{\beta(p', x') \mid (p, x) \cong_k^X (p', x')\}$. We set

$$\mathcal{G}_k[X](q) = \{\mathbf{V}_{q,y} \mid y \in q\}$$

The following two facts are immediate consequences of the definitions:

Fact 9.14. *For all $k \leq k' \in \mathbb{N}$, $X \subseteq H$ and $q \in \mathbb{A}_s^+$ we have $\mathcal{G}_{k'}[X](q) \sqsubseteq \mathcal{G}_k[X](q)$.*

Fact 9.15. *For all $k \in \mathbb{N}$ and $X \subseteq H$ we have $\mathfrak{I}_k[\alpha, X] = \downarrow\{\mathcal{G}_k[X](q) \mid q \in \mathbb{A}_s^+\}$.*

We can now finish the proof of Proposition 9.10. The proof is by induction on the size of the alphabet as stated in the proposition below.

Proposition 9.16. *Let $\mathbb{B}_s \subseteq \mathbb{A}_s$, $k \geq 2|\mathbb{B}_s|^2(|\mathfrak{C}|+1)$ and p a shallow multicontext such that p contains only labels in $\mathbb{B}_s$. Then $\mathcal{G}_k[X](p) \in \downarrow Sat[X, \alpha]$.*

Using Proposition 9.16 with $\mathbb{B}_s = \mathbb{A}_s$, we obtain that for any $k \geq \ell$ and any $p \in \mathbb{A}_s^+$, we have $\mathcal{G}_k[X](p) \in \downarrow Sat[X, \alpha]$. It then follows from Fact 9.15 that $\mathfrak{I}_k[\alpha, X] \subseteq \downarrow Sat[X, \alpha]$ which terminates the proof of Proposition 9.10. It now remains to prove Proposition 9.16. The remainder of the section is devoted to this proof.

For the sake of simplifying the presentation, we assume that p can be an empty shallow multicontext denoted 'ε' and that $Sat[X, \alpha]$ contains an artificial neutral element '0' such that $\mathcal{G}_k[X](\varepsilon) = 0$ for any k. As ε will be the only shallow multicontext having that property this does not harm the generality of the proof.

As explained above, the proof is by induction on the size of $\mathbb{B}_s$. The base case happens when $\mathbb{B}_s = \emptyset$. In that case, $p = \varepsilon$ and $\mathcal{G}_k[X](\varepsilon) \in Sat[X, \alpha]$ by definition. Assume now that $\mathbb{B}_s \neq \emptyset$, we set $k \geq 2|\mathbb{B}_s|^2(|\mathfrak{C}|+1)$ and p as a shallow multicontext containing only labels in $\mathbb{B}_s$. We need to prove that $\mathcal{G}_k[X](p) \in \downarrow Sat[X, \alpha]$.

First observe that when p does not contain *all* labels in $\mathbb{B}_s$, the result is immediate by induction. Therefore, assume that p contains all labels in $\mathbb{B}_s$. We proceed as follows. First, we define a new notion called a $(\mathbb{B}_s[X], n)$-pattern. Intuitively, a shallow multicontext q contains a $(\mathbb{B}_s[X], n)$-pattern iff all labels in $\mathbb{B}_s$ (modulo $\mathbb{B}_s[X]$-equivalence) are repeated at least n times in q. Then, we prove that if p contains a $(\mathbb{B}_s[X], n)$-pattern for a large enough n, then $\mathcal{G}_k[X](p)$ can be decomposed in such a way that it can be proved to be in $Sat[X, \alpha]$ by using induction on the factors, and Operations (1) and (2) to compose them. Otherwise, we prove that $\mathcal{G}_k[X](p)$ can be decomposed as a sum of bounded length whose elements can be proved to be in $Sat[X, \alpha]$ by induction. We then conclude using Operation (1). We begin with the definition of $(\mathbb{B}_s[X], n)$-patterns.

$(\mathbb{B}_s[X], n)$-patterns. Consider the $\mathbb{B}_s[X]$-equivalence of labels in $\mathbb{B}_s$ and let m be the number of equivalence classes. We fix an arbitrary order on these classes that we denote by $C_0, \dots, C_{m-1} \subseteq \mathbb{B}_s$. Recall that $\mathbb{C}_s$ is a an X-approximation of $\mathbb{B}_s$ iff $\mathbb{C}_s$ contains at least one element of each class. Let $n \in \mathbb{N}$. We say that a shallow multicontext q contains a $(\mathbb{B}_s[X], n)$-pattern iff q can be decomposed as

$$q = q_0 + c_0 + q_1 + c_1 + \cdots + q_n + c_n + q_{n+1}$$

such that for all $i \leq n$, $c_i \in C_j$ (with $j = i \bmod m$) and q_i is a (possibly empty) shallow multicontext. In particular, the decomposition above is called the *leftmost decomposition* iff for all $i \leq n$ no label in C_j (with $j = i \bmod m$) occurs in q_i. Symmetrically, in the *rightmost decomposition*, for all $i \geq 0$, no label in C_i (with $j = i \bmod m$) occurs in q_{i+1}. Observe that by definition the leftmost and rightmost decompositions are unique. In the proof, we use the following decomposition lemma.

Lemma 9.17 (Decomposition Lemma)**.** *Let $n \in \mathbb{N}$. Let q be a shallow multicontext that contains a $(\mathbb{B}_s[X], n)$-pattern and let $q = q_0 + c_0 + \cdots + c_n + q_{n+1}$ be the associated leftmost or rightmost decomposition. Then*

$$\mathcal{G}_k[X](q) \sqsubseteq \mathcal{G}_{k-n}[X](q_0) + \mathcal{G}_{k-n}[X](c_0) + \cdots + \mathcal{G}_{k-n}[X](c_n) + \mathcal{G}_{k-n}[X](q_{n+1})$$

Proof. This is a simple Ehrenfeucht-Fraïssé game argument. Because of the missing boundary labels within the q_j, using at most n moves, Spoiler can make sure that the game stays within the appropriate segment q_j and can use the remaining $k - n$ moves for describing that segment. □

This finishes the definition of patterns. Set $n = m(|\mathfrak{C}| + 1)$. We now consider two cases depending on whether our shallow multicontext p contains a $(\mathbb{B}_s[X], 2n)$-pattern.

Case 1: p does not contain a $(\mathbb{B}_s[X], 2n)$-pattern. In that case we conclude using induction and Operation (1). Let n' be the largest number such that p contains a $(\mathbb{B}_s[X], n')$-pattern. By hypothesis $n' < 2n$. Let $p = p_0 + c_0 + \cdots + c_{n'} + p_{n'+1}$ be the associated leftmost decomposition. Observe that by definition, for $i \leq n'$, p_i uses a strictly smaller alphabet than $\mathbb{B}_s$. Moreover, since p does not contain a $(\mathbb{B}_s[X], n' + 1)$-pattern this is also the case for $p_{n'+1}$. Set $\tilde{k} = k - n'$, by choice of k, we have $\tilde{k} \geq 2(|\mathbb{B}_s| - 1)^2(|\mathfrak{C}| + 1)$. Therefore, we can use our induction hypothesis and for all i we get,

$$\mathcal{G}_{\tilde{k}}[X](p_i) \in {\downarrow} Sat[X, \alpha]$$

Moreover, for all i, $\mathcal{G}_{\tilde{k}}[X](c_i) \in \mathfrak{T}[\alpha] \subseteq Sat[X, \alpha]$. Finally, using Lemma 9.17 we obtain

$$\mathcal{G}_k[X](p) \sqsubseteq \mathcal{G}_{\tilde{k}}[X](p_0) + \mathcal{G}_{\tilde{k}}[X](c_0) + \cdots + \mathcal{G}_{\tilde{k}}[X](c_{n'}) + \mathcal{G}_{\tilde{k}}[X](p_{n'+1})$$

From Operation (1) the right-hand sum is in ${\downarrow} Sat[X, \alpha]$. We then conclude that $\mathcal{G}_k[X](p) \in {\downarrow} Sat[X, \alpha]$ which terminates this case.

Case 2: p contains a $(\mathbb{B}_s[X], 2n)$-pattern. In that case we conclude using induction, Operation (1) and Operation (2). By hypothesis, we know that p contains a $(\mathbb{B}_s[X], n)$-pattern, let $p = p_0 + c_0 + \cdots + c_n + p_{n+1}$ be the associated leftmost decomposition. Since p contains a $(\mathbb{B}_s[X], 2n)$-pattern, p_{n+1} must contain a $(\mathbb{B}_s[X], n)$-pattern. We set $p_{n+1} = p' + c'_0 + \cdots + c'_n + p'_{n+1}$ as the associated rightmost decomposition. In the end we get

$$p = p_0 + c_0 + \cdots + c_n + p' + c'_0 + p'_1 + \cdots + c'_n + p'_{n+1}$$

Set $\tilde{k} = k - 2n$ and observe that by choice of k, $\tilde{k} \geq 2(|\mathbb{B}_s| - 1)^2(|\mathfrak{C}| + 1)$. Therefore, as in the previous case, we get by induction that for all i, $\mathcal{G}_{\tilde{k}}[X](p_i) \in {\downarrow} Sat[X, \alpha]$, $\mathcal{G}_{\tilde{k}}[X](p'_i) \in$

$\downarrow Sat[X,\alpha]$, $\mathcal{G}_{\tilde{k}}[X](c_i) \in \downarrow Sat[X,\alpha]$ and $\mathcal{G}_{\tilde{k}}[X](c'_i) \in \downarrow Sat[X,\alpha]$. Using the same inductive argument for p' may not be possible as p' might contain all labels in $\mathbb{B}_s$.

If p' does not contain all labels in $\mathbb{B}_s$, then, by induction, $\mathcal{G}_{\tilde{k}}[X](p') \in \downarrow Sat[X,\alpha]$ and we can then use Lemma 9.17 as in Case 1 to conclude that $\mathcal{G}_k[X](p) \in \downarrow Sat[X,\alpha]$. Assume now that p' contains all labels in $\mathbb{B}_s$. Recall that m is the number of $\mathbb{B}_s[X]$-equivalence classes. For all $j \leq |\mathfrak{C}|$, set

$$\mathcal{V}_j = \sum_{i=jm}^{m-1+jm} (\mathcal{G}_{\tilde{k}}[X](p_i) + \mathcal{G}_{\tilde{k}}[X](c_i)) \qquad \mathcal{V}'_j = \sum_{i=jm}^{m-1+jm} (\mathcal{G}_{\tilde{k}}[X](c'_i) + \mathcal{G}_{\tilde{k}}[X](p'_{i+1}))$$

Observe that for all j, by definition $\mathcal{V}_j, \mathcal{V}'_j$ have an alphabet which is an X-approximation of $\mathbb{B}_s$ and by Operation (1), $\mathcal{V}_j, \mathcal{V}'_j \in \downarrow Sat[X,\alpha]$. Moreover, it follows from a pigeon-hole principle argument that the sequences $\mathcal{V}_0 + \cdots + \mathcal{V}_{|\mathfrak{C}|}$ and $\mathcal{V}'_0 + \cdots + \mathcal{V}'_{|\mathfrak{C}|}$ must contain "loops", i.e. there exists $j_1 < j_2$ and $j'_1 < j'_2$ such that

$$\begin{array}{rcl} \mathcal{V}_0 + \cdots + \mathcal{V}_{j_1} & = & \mathcal{V}_0 + \cdots + \mathcal{V}_{j_2} \\ \mathcal{V}'_{j_2} + \cdots + \mathcal{V}'_{|\mathfrak{C}|} & = & \mathcal{V}'_{j_1} + \cdots + \mathcal{V}'_{|\mathfrak{C}|} \end{array}$$

Set $\mathcal{U}_1 = \mathcal{V}_0 + \cdots + \mathcal{V}_{j_1}$, $\mathcal{U}_2 = \mathcal{V}_{j_1+1} + \cdots + \mathcal{V}_{j_2}$, $\mathcal{U}'_1 = \mathcal{V}'_{j'_1} + \cdots \mathcal{V}'_{j'_2-1}$ and $\mathcal{U}'_2 = \mathcal{V}'_{j'_2} + \cdots + \mathcal{V}'_{|\mathfrak{C}|}$. Observe that by Operation (1), we have $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}'_1, \mathcal{U}'_2 \in \downarrow Sat[X,\alpha]$ and that by construction the alphabets of $\mathcal{U}_2, \mathcal{U}'_1$ are X-approximations of $\mathbb{B}_s$. Moreover, a little algebra yields $\mathcal{U}_1 = \mathcal{U}_1 + \mathcal{U}_2 = \mathcal{U}_1 + \omega\mathcal{U}_2$ and $\mathcal{U}'_2 = \mathcal{U}'_1 + \mathcal{U}'_2 = \omega\mathcal{U}'_1 + \mathcal{U}'_2$.

Set $p'' = p_{j_2m} + \cdots + c_n + p' + c'_0 + \cdots + p'_{j'_1m-1}$. Observe that by hypothesis on p', p'' contains all labels in $\mathbb{B}_s$. It follows from Fact 9.14 and Lemma 9.17 that

$$\mathcal{G}_k[X](p) \sqsubseteq \mathcal{U}_1 + \mathcal{U}_2 + \mathcal{G}_{\tilde{k}}[X](p'') + \mathcal{U}'_1 + \mathcal{U}'_2 = \mathcal{U}_1 + \omega\mathcal{U}_2 + \mathcal{G}_{\tilde{k}}[X](p'') + \omega\mathcal{U}'_1 + \mathcal{U}'_2$$

Moreover, since p'' has alphabet $\mathbb{B}_s$, it is immediate that $\mathcal{G}_{\tilde{k}}[X](p'') \sqsubseteq \llbracket \mathbb{B}_s \rrbracket$. Therefore,

$$\mathcal{G}_k[X](p) \sqsubseteq \mathcal{U}_1 + \omega\mathcal{U}_2 + \llbracket \mathbb{B}_s \rrbracket + \omega\mathcal{U}'_1 + \mathcal{U}'_2$$

It is now immediate from Operation (2) $\omega\mathcal{U}_2 + \llbracket \mathbb{B}_s \rrbracket + \omega\mathcal{U}'_1 \in \downarrow Sat[X,\alpha]$. By combining this with Operation (1), we obtain

$$\mathcal{G}_k[X](p) \sqsubseteq \mathcal{U}_1 + \omega\mathcal{U}_2 + \llbracket \mathbb{B}_s \rrbracket + \omega\mathcal{U}'_1 + \mathcal{U}'_2 \in \downarrow Sat[X,\alpha]$$

We conclude that $\mathcal{G}_k[X](p) \in \downarrow Sat[X,\alpha]$ which terminates the proof.

10. Other logics

It turns out that the proof of Theorem 6.2 depends on the horizontal modalities of the logic only via the notion of definability within shallow multicontexts. It can therefore be adapted to many other horizontal modalities assuming those can at least express the fact that two nodes are siblings (ie. can talk about the shallow multicontext of a given node). By tuning this notion one can obtain several new characterizations. We illustrate this feature in this section with the horizontal predicates $\mathbf{X_h}$, $\mathbf{X_h}^{-1}$, $\mathbf{S}$ and $\mathbf{S}^{\neq}$, adopting the point of view of temporal logic.

The semantic of these predicates is defined as follows. The formula $\mathbf{S}^{\neq}\varphi$ holds at a node x if φ holds at some sibling of x distinct from x. It is a shorthand for $\mathbf{F_h}\varphi \vee \mathbf{F_h}^{-1}\varphi$. The formula $\mathbf{S}\varphi$ holds at x if φ holds at some sibling of x including x. It is a shorthand for $\varphi \vee \mathbf{S}^{\neq}\varphi$. The predicates $\mathbf{X_h}$ and $\mathbf{X_h}^{-1}$ are the usual next sibling and previous sibling modalities.

The vertical navigational modalities remain the same and the corresponding logics are denoted by $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S})$, $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S}^{\neq})$, $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{X_h},\mathbf{F_h},\mathbf{X_h}^{-1},\mathbf{F_h}^{-1})$ and a characterization can be obtained for each of them using the same scheme as for $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{F_h},\mathbf{F_h}^{-1})$.

As before $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S}^{\neq})$ and $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{X_h},\mathbf{F_h},\mathbf{X_h}^{-1},\mathbf{F_h}^{-1})$ are equivalent to two-variable fragments of first-order logic. $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S}^{\neq})$ has the same expressive power as $\mathrm{FO}^2(s,<_{\mathbf{v}})$ while $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{X_h},\mathbf{F_h},\mathbf{X_h}^{-1},\mathbf{F_h}^{-1})$ corresponds to $\mathrm{FO}^2(Succ_{\mathbf{h}},<_{\mathbf{h}},<_{\mathbf{v}})$. Here $s(x,y)$ is a binary predicate that holds when x and y are siblings and $Succ_{\mathbf{h}}$ is a binary predicate that holds when y is the next sibling of x. These facts can be proved along the same way as the equivalence between $\mathrm{FO}^2(<_{\mathbf{v}},<_{\mathbf{h}})$ and $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{F_h},\mathbf{F_h}^{-1})$, see Theorem 3.1.

Note that this is no longer the case for $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S})$ as languages defined in this formalism are closed under bisimulation while in the two variable fragment of first-order logic it is possible to have quantifications over incomparable nodes by using the equality and negation which rules out closure under bisimulation.

The proof techniques presented in the previous sections require at least the power of testing whether two nodes are sibling in order to extract a shallow multicontext within a forest. Hence it cannot be applied to $\mathrm{FO}^2(<_{\mathbf{v}})$ and finding a decidable characterization for this logic remains an open question. Similarly, we rely on the fact that the child relation cannot be expressed and finding a decidable characterization in the presence of this predicate remains also an open question.

As we don't have a vertical successor modality, the characterizations we obtain for $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S})$, $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S}^{\neq})$ and $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{X_h},\mathbf{F_h},\mathbf{X_h}^{-1},\mathbf{F_h}^{-1})$ still require Identity (6.3) on the vertical monoid V of the forest algebra. Identity (6.2) is now replaced by the appropriate identity corresponding to the new horizontal expressive power. Finally the notion of saturation is adapted by replacing $\cong_k^X$ with a notion reflecting the horizontal expressive power of the logic. It is defined as in Section 5 by only modifying the allowed moves in the game in order to reflect the horizontal expressive power of the associated logic (the constraints on the labels remaining untouched within each game). In a similar way, $\equiv_k$ is replaced in the proof by the suitable game. Besides these changes at the level of definitions, the characterization is stated and proved as for Theorem 6.2.

10.1. $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S})$**.** In this case the games on shallow multicontexts are defined with no navigational constraints on Duplicators moves: Duplicator can respond by choosing an arbitrary node, the restriction being only on its label.

Note that the games no longer depend on k, as only the presence or absence of a given symbol of $\mathbb{A}_s$ inside the shallow multicontext matters. We write $\mathbf{S}$-$\equiv$ and $\mathbf{S}$-$\cong^X$ the equivalence relations resulting from this game and its X-relaxed variant.

The following analog of Claim 5.5 is an immediate consequences of the definitions.

Claim 10.1. *Let $X \subseteq H$ and (p,x) be a node. There is a $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S})$ formula $\psi_{p,x}$ having one free variable and such that for any forest s, $\psi_{p,x}$ holds exactly at all nodes (p',x') such that (p,x) $\mathbf{S}$-$\cong^X$ (p',x').*

Recall the definition of saturation given in Section 6.2. The notion of $\mathbf{S}$-saturation is obtained identically after replacing $\cong_k^X$ with $\mathbf{S}$-$\cong^X$. With these new definitions we get:

Theorem 10.2. *A regular forest language L is definable in $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{S})$ iff its syntactic morphism $\alpha : \mathbb{A}^{\Delta} \to (H,V)$ satisfies:*

a) H satisfies the identities

$$2h = h \text{ and } f + g = g + f \tag{10.1}$$

b) V satisfies Identity (6.3)

$$(uv)^\omega v(uv)^\omega = (uv)^\omega$$

c) the leaf completion of α *is closed under* $\mathbf{S}$*-saturation.*

Note that (10.1) simply states that the logic is closed under bisimulation, hence reflecting exactly the horizontal expressive power of $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S})$.

Concerning the proof of Theorem 10.2, aside from the initial choice of the integer k which is no longer necessary here, it is identical to the one we gave for Theorem 6.2 after replacing Lemma 8.1 by the following result:

Lemma 10.3. *Let L be a language whose syntactic forest algebra satisfies the identities stated in Theorem 10.2. For all shallow multicontexts* p $\mathbf{S}$-$\equiv$ p' *and for all forests* s, $p[\bar{s}]$ *and* $p'[\bar{s}]$ *have the same forest type.*

Proof. Since p $\mathbf{S}$-$\equiv$ p' the forests $p[\bar{s}]$ and $p'[\bar{s}]$ contain the same symbols but possibly with a different number of occurrences. It follows from (10.1) that $\alpha(p[\bar{s}]) = \alpha(p'[\bar{s}])$. ☐

10.2. **$\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S}^{\neq})$ and $\mathbf{FO}^2(s, <_\mathbf{v})$.** As before the key point is the allowed moves in the Ehrenfeucht-Fraïssé games. In this case we only require that Duplicator moves in a different position as soon as Spoiler does.

As in the previous section, the games no longer depend on k as it only matters whether or not a label occurs and whether or not it occurs twice. We write $\mathbf{S}^{\neq}$-$\equiv$ and $\mathbf{S}^{\neq}$-$\cong^X$ the equivalence relations resulting from this game and its X-relaxed variant.

The following analog of Claim 5.5 is an immediate consequences of the definitions.

Claim 10.4. *Let* $X \subseteq H$ *and* (p, x) *be a node. There is a* $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S}^{\neq})$ *formula* $\psi_{p,x}$ *having one free variable and such that for any forest* s, $\psi_{p,x}$ *holds exactly at all nodes* (p', x') *such that* (p, x) $\mathbf{S}^{\neq}$-$\cong^X$ (p', x').

As in the previous case, replacing $\cong_k^X$ with $\mathbf{S}^{\neq}$-$\cong^X$ in the definition of saturation yields a new notion of saturation that we call $\mathbf{S}^{\neq}$-saturation. We can show:

Theorem 10.5. *A regular forest language L is definable in* $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S}^{\neq})$ *iff its syntactic morphism* $\alpha : \mathbb{A}^\Delta \to (H, V)$ *satisfies:*

a) H satisfies the identities

$$3h = 2h \text{ and } f + g = g + f \tag{10.2}$$

b) V satisfies Identity (6.3)

$$(uv)^\omega v(uv)^\omega = (uv)^\omega$$

c) the leaf completion of α *is closed under* $\mathbf{S}^{\neq}$*-saturation.*

Notice that (10.2) reflects exactly the horizontal expressive power of $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S}^{\neq})$: no horizontal order and counting up to threshold 2.

Concerning the proof of Theorem 10.5, aside from the initial choice of the integer k which is no longer necessary here, it is identical to the one we gave for Theorem 6.2 after replacing Lemma 8.1 by the following result:

Lemma 10.6. *Let L be a language whose syntactic forest algebra satisfies the identities stated in Theorem 10.5. For all shallow multicontexts* p $\mathbf{S}^{\neq}$-$\equiv$ p' *and all forests* s, $p[\bar{s}]$ *and* $p'[\bar{s}]$ *have the same forest type.*

Proof. Since p $\mathbf{S}^{\neq}$-$\equiv$ p' the forests $p[\bar{s}]$ and $p'[\bar{s}]$ contain the same symbols with the same number of occurrences up to threshold 2. It follows from (10.2) that $\alpha(p[\bar{s}]) = \alpha(p'[\bar{s}])$. $\square$

10.3. $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{X_h},\mathbf{F_h},\mathbf{X_h}^{-1},\mathbf{F_h}^{-1})$ **and** $\mathbf{FO}^2(Succ_{\mathbf{h}},<_{\mathbf{h}},<_{\mathbf{v}})$**.** In this case Duplicator not only must respect the direction in which Spoiler has moved his pebble, but she also must place her pebble on the successor (predecessor) of the current position if this was also the situation for Spoiler.

The games now depends on k and we write $\mathbf{Suc}\text{-}\equiv_k$ and $\mathbf{Suc}\text{-}\cong_k^X$ the equivalence relations resulting from this game and its X-relaxed variant.

The following analog of Claim 5.5 is an immediate consequences of the definitions.

Claim 10.7. *Let $X \subseteq H$ and (p,x) be a node. There is a $\mathbf{EF}+\mathbf{F}^{-1}(\mathbf{X_h},\mathbf{F_h},\mathbf{X_h}^{-1},\mathbf{F_h}^{-1})$ formula $\psi_{p,x}$ having one free variable and such that for any forest s, $\psi_{p,x}$ holds exactly at all nodes (p',x') such that (p,x) $\mathbf{Suc}\text{-}\cong_k^X$ (p',x').*

As in the previous cases, we obtain from $\mathbf{Suc}\text{-}\cong_k^X$ a new notion of saturation that we call $\mathbf{X_h}$-saturation. We can show:

Theorem 10.8. *A regular forest language L is definable in $\mathrm{FO}^2(Succ_{\mathbf{h}},<_{\mathbf{h}},<_{\mathbf{v}})$ iff its syntactic morphism $\alpha : \mathbb{A}^{\Delta} \to (H,V)$ satisfies:*

a) H satisfies for all $h,g \in H$, for all $e \in H$ such that $2e = e$:

$$\omega(e+h+e+g+e)+g+\omega(e+h+e+g+e) = \omega(e+h+e+g+e) \qquad (10.3)$$

b) V satisfies Identity (6.3)

$$(uv)^{\omega}v(uv)^{\omega} = (uv)^{\omega}$$

c) the leaf completion of α is closed under $\mathbf{X_h}$-saturation.

Equation (10.3) is extracted from the following result which is essentially proved in [TW98] based on a result of [Alm96] (see Footnote on page 9).

Theorem 10.9 ([TW98],[Alm96])**.** *A regular string language L is definable in $\mathrm{FO}^2(Succ,<)$ iff its syntactic semigroup S satisfies for all $u,v \in S$, for all $e \in S$ such that $e^2 = e$:*

$$(eueve)^{\omega}v(eueve)^{\omega} = (eueve)^{\omega}$$

Again, the proof of Theorem 10.8 follows the lines of the proof of Theorem 6.2 after replacing Lemma 8.1 by the following simple result:

Lemma 10.10. *Let L be a language whose syntactic forest algebra satisfies the identities stated in Theorem 10.8. There exists a number k' such that for all $k \geq k'$, all shallow multicontexts p $\mathbf{Suc}\text{-}\equiv_k$ p' and all forests s, $p[\bar{s}]$ and $p'[\bar{s}]$ have the same forest type.*

Proof. This is a consequence of the fact that H satisfies Identity (10.3). The proof is identical to the one we provided for Lemma 8.1 replacing Theorem 4.2 by Theorem 10.9 and $\equiv_k$ with $\mathbf{Suc}\text{-}\equiv_k$. $\square$

10.4. **Decidability.** Deciding whether a regular forest language is definable in $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S})$ and $\mathbf{EF} + \mathbf{F}^{-1}(\mathbf{S}^{\neq})$ is simple from Theorem 10.2 and Theorem 10.5. As in Section 9 we prove that the corresponding notions of saturation are equivalent to their abstract variant. The latter are decidable because they don't depend on k and, up to equivalence, only finitely many $Q \subseteq \mathbb{A}_s^+$ needs to be considered.

However, for $\mathrm{FO}^2(\mathrm{Succ}, <)$, it is not clear how to generalize the construction of the indistinguishable sets. We leave this and the status of deciding definability in $\mathrm{FO}^2(Succ_{\mathbf{h}}, <_{\mathbf{h}}, <_{\mathbf{v}})$ as an open problem.

11. Discussion

We have obtained a characterization for $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$, using identities on the syntactic forest algebra and the new notion of saturation. Our proof technique applies to many other logical formalisms assuming these only differ from $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ by their horizontal expressive power and that they can at least express the fact that two nodes are siblings.

We have shown all these characterizations to be decidable except for $\mathrm{FO}^2(Succ_{\mathbf{h}}, <_{\mathbf{h}}, <_{\mathbf{v}})$. We leave this case as an open problem. As explained in Section 10, it would be enough to generalize our algorithm for computing profiles (i.e. Proposition 9.10) to the appropriate notion of profile for $\mathrm{FO}^2(Succ_{\mathbf{h}}, <_{\mathbf{h}}, <_{\mathbf{v}})$.

Since $\mathrm{FO}^2(<_{\mathbf{v}})$ is unable to express the sibling relation, it cannot be covered by our techniques and we leave open the problem of finding a decidable characterization for this logic.

It would also be interesting to incorporate the vertical successor in our proofs to obtain a decidable characterization for $\mathrm{FO}^2(Succ_{\mathbf{h}}, <_{\mathbf{h}}, Succ_{\mathbf{v}}, <_{\mathbf{v}})$. This would yield a decidable characterization of the navigational core of XPath. We believe this requires new ideas.

It terms of complexity, a rough analysis of the proof of Theorem 9.1 yields a 4-EXPTIME upper bound on the complexity of the problem. It is likely that this can be improved. Recall that the complexity of the same problem for the corresponding logics over words, which amounts to checking (6.3), is polynomial is the size of the syntactic monoid.

It would also be interesting to obtain an equivalent characterization of $\mathrm{FO}^2(<_{\mathbf{v}}, <_{\mathbf{h}})$ which remains decidable while avoiding the cumbersome notion of saturation. For instance it is not clear whether the notion of confusion introduced in [BSW12] can be used as a replacement. We leave this as an open problem.

Acknowledgment. We thanks the reviewers for their comments on earlier versions of this article. Their comments led to significant improvements of the paper.

References

[Alm96] Jorge Almeida. A syntactical proof of locality of **DA**. *International Journal of Algebra and Computation*, 6:165–177, 1996.

[Boj09] Mikołaj Bojańczyk. Two-way unary temporal logic over trees. *Logical Methods in Computer Science*, 5(3), 2009.

[BS09] Michael Benedikt and Luc Segoufin. Regular Languages Definable in FO and FOmod. *ACM Transactions of Computational Logic*, 11(1), 2009.

[BS10] Mikołaj Bojańczyk and Luc Segoufin. Tree languages defined in first-order logic with one quantifier alternation. *Logical Methods in Computer Science*, 6(4), 2010.

[BSS12] Mikołaj Bojańczyk, Luc Segoufin, and Howard Straubing. Piecewise testable tree languages. *Logical Methods in Computer Science*, 8(3), 2012.

[BSW12] Mikołaj Bojańczyk, Howard Straubing, and Igor Walukiewicz. Wreath products of forest algebras, with applications to tree logics. *Logical Methods in Computer Science*, 8(3), 2012.

[BW06] Mikołaj Bojańczyk and Igor Walukiewicz. Characterizing EF and EX tree logics. *Theoretical Computer Science*, 358, 2006.

[BW07] Mikołaj Bojańczyk and Igor Walukiewicz. Forest algebras. In *Automata and Logic: History and Perspectives*, pages 107–132. Amsterdam University Press, 2007.

[CDG+] Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications. Available on http://tata.gforge.inria.fr. release 12th October 2007.

[EVW02] Kousha Etessami, Moshe Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. *Inf. Comput.*, 179(2):279–295, 2002.

[ÉW05] Zoltán Ésik and Pascal Weil. Algebraic characterization of regular tree languages. *Theoretical Computer Science*, 340:291–321, 2005.

[Kuf06] Manfred Kufleitner. *Logical Fragments for Mazurkiewicz Traces: Expressive Power and Algebraic Characterizations*. Dissertation, Institut für Formale Methoden der Informatik, Universität Stuttgart, 2006.

[Mar05] Maarten Marx. First order paths in ordered trees. In *Proceedings of the 10th International Conference in Database Theory (ICDT'05)*, pages 114–128, 2005.

[Pla08] Thomas Place. Characterization of logics over ranked tree languages. In *Proceedings of the 17th Annual EACSL Conference on Computer Science Logic (CSL'08)*, pages 401–415, 2008.

[Pot94] Andreas Potthoff. Modulo-counting quantifiers over finite trees. *Theoretical Computer Science (TCS)*, 126(1):97–112, 1994.

[PS10] Thomas Place and Luc Segoufin. Deciding definability in $FO_2(<)$ (or XPath) on trees. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS'10)*, pages 1–1, 2010.

[PS11] Thomas Place and Luc Segoufin. A decidable characterization of locally testable tree languages. *Logical Methods in Computer Science*, 7(4), 2011.

[Str94] Howard Straubing. *Finite Automata, Formal Logic and Circuit Complexity*. Birkhauser Verlag GmbH, 1994.

[TW98] Denis Thérien and Thomas Wilke. Over words, two variables are as powerful as one quantifier alternation. In *Proceedings of the 30th ACM Symposium on the Theory of Computing (STOC'98)*, pages 234–240, 1998.

[Wil96] Thomas Wilke. An algebraic characterization of frontier testable tree languages. *Theoretical Computer Science*, 154(1):85–106, 1996.