Deciding Piecewise Testable Separability for Regular Tree Languages

Jean Goubault-Larrecq, Sylvain Schmitz

Deciding Piecewise Testable Separability for Regular Tree Languages

Jean Goubault-Larrecq and Sylvain Schmitz

LSV, ENS Cachan & CNRS & Inria, Université Paris-Saclay, France
{goubault,schmitz}@lsv.fr

Abstract

The piecewise testable separability problem asks, given two input languages, whether there exists a piecewise testable language that contains the first input language and is disjoint from the second. We prove a general characterisation of piecewise testable separability on languages in a well-quasi-order, in terms of ideals of the ordering. This subsumes the known characterisations in the case of finite words. In the case of finite ranked trees ordered by homeomorphic embedding, we show using effective representations for tree ideals that it entails the decidability of piecewise testable separability when the input languages are regular. A final byproduct is a new proof of the decidability of whether an input regular language of ranked trees is piecewise testable, which was first shown in the ranked case for a coarser ordering by Place [CSL 2008], and in the unranked case for the homeomorphic embedding relation by Bojańczyk, Segoufin, and Straubing [Log. Meth. in Comput. Sci., 8(3:26), 2012].

1998 ACM Subject Classification F.4.3 Formal Languages, F.4.1 Mathematical Logic

Keywords and phrases Well-quasi-order, ideal, tree languages, first-order logic

1 Introduction

The *separability* problem for a class $\mathcal{C}$ of input languages and a class $\mathcal{S}$ of separators, asks given two input languages L and L' from $\mathcal{C}$ whether there exists a language S from $\mathcal{S}$ such that $L \subseteq S$ and $S \cap L' = \emptyset$. This classical problem has been studied in formal language theory since the 70's [35], but has sparked renewed interest due in particular to its connection with the *definability* problem: given L from $\mathcal{C}$, does L belong to $\mathcal{S}$? When $\mathcal{C}$ is effectively closed under complement, this last question reduces to the separability of L and of its complement. Separability thus generalises definability, and has been instrumental in the recent advances on the definability problem for the alternation hierarchy over finite words [31].

We focus in this paper on the class of *piecewise testable* languages as class $\mathcal{S}$ of separators. Over finite words, this coincides with the languages defined by $\mathcal{B}\Sigma_1(<)$, the Boolean combinations of existential first-order sentences with order, and is one of the oldest and best-known classes of languages with decidable definability: Simon [34] showed that a language is piecewise-testable if and only if its syntactic monoid is $\mathcal{J}$*-trivial*. Lately, this has been extended in two different directions:

- Over finite ordered trees, Place [30] and Bojańczyk, Segoufin, and Straubing [7] generalise Simon's algebraic approach, in the ranked and unranked cases respectively, and characterise the syntactic algebræ of piecewise testable tree languages defined by several signatures. This yields the decidability of the piecewise testable definability problem for regular tree languages.
- Over finite words, Almeida and Zeitoun [2, 3] first proved the decidability of the PTL separability problem for regular input languages, based on a topological characterisation of separability. This abstract characterisation has been turned into a much more efficient characterisation in terms of patterns found simultaneously in L and L' [32, 12], which

culminated in the work of Czerwiński, Martens, van Rooijen, Zeitoun, and Zetzsche [13] with a proof of the decidability of piecewise separability for many classes $\mathcal{C}$ of input languages, which even includes higher-order languages [21, 9]. The crux of this proof is a reduction to the computation of representations for downward-closures [38] ordered by scattered subword embedding.

Separability for Tree Languages. Considering these two lines of research side by side begs the question whether piecewise testable separability might be decidable over richer structures than words, and in particular over trees. In this paper, we answer positively by proving:

▶ **Theorem 1.** *Piecewise testable separability is decidable for regular languages of ranked trees ordered by homeomorphic embedding.*

Since the class of regular tree languages is effectively closed under complement, Theorem 1 entails as a sub-problem the decidability of the corresponding definability problem:

▶ **Corollary 2.** *Piecewise testable definability is decidable for regular languages of ranked trees ordered by homeomorphic embedding.*

Thus Theorem 1 also yields as a byproduct a new proof of decidability for the piecewise testable definability problems studied by Bojańczyk et al. [7] in the unranked case, and by Place [30] in the ranked case but with a coarser ordering.

Order-Theoretic Framework. We employ vastly different techniques from Place's and Bojańczyk et al.'s, and Theorem 1 should be thought of as a proof of concept for our main contribution, which is a very general order-theoretic framework for piecewise testable separability:

- We characterise in Section 3 separability by piecewise testable languages over any combinatorial *well-quasi-order*; this encompasses for instance words ordered by scattered subword embedding and trees ordered by homeomorphic embedding or minor ordering. Our characterisation proceeds similarly to the topological characterisation of Almeida [2] by comparing the closures of L and L', but is stated in purely order-theoretic terms and compares the *ideals*—i.e. the irreducible downwards-closed sets—adherent to L and L'. The advantage of this approach is that it scales to more complex structures than words—we have a toolkit of effective ideal representations at our disposal [16, 18]—and leads to a very simple and general proof, that delegates syntactic manipulations to the ideal representation. Our techniques encompass the word case: e.g. the patterns derived by Czerwiński et al. [13] turn out to be representations for word ideals [24, 1], and we eschew the technical work on patterns that absorbed a large part of Czerwiński et al.'s proof.
- We then show in Section 4 how to turn our characterisation into a generic decision procedure, by reduction to the *adherence membership* problem for the class $\mathcal{C}$. Provided we have effective representations for ideals, this yields directly the decidability for regular tree languages. We also derive the reduction of Czerwiński et al. to the computation of downward-closures in full generality.
- In order to instantiate this framework for tree languages, we provide in Section 5 our last ingredient: an effective representation for ideals of ranked trees ordered by homeomorphic embedding. This is a contribution of independent interest, which fixes an issue with the representation proposed in [16].

We start in Section 2 by defining piecewise testable languages in the general setting of combinatorial quasi-orders, and by recalling their logical characterisation. We assume the

reader is already familiar with tree languages; see [10] for a general reference. Due to space constraints, some material will be found in the appendices.

2 Piecewise Testable Languages

We work in the first part of this paper with *combinatorial* classes X of elements, where X is a countable set equipped with a *size* function $|.|: X \to \mathbb{N}$ such that $X_{\leq n} \stackrel{\text{def}}{=} \{x \in X \mid |x| \leq n\}$ is finite for every n. As soon as we equip X with a quasi-ordering $\leq$, we can define a notion of piecewise-testable languages (see Section 2.1). The usual setting for piecewise testable languages is however that of classes of finite structures along with the embedding relation, in which case those sets also enjoy a logical characterisation (see Section 2.2).

2.1 Pieces

Two elements x and y are *n-piecewise equivalent*, noted $x \equiv_n y$, if for every $z \in X_{\leq n}$, $z \leq x$ if and only if $z \leq y$. A subset S of X is called *n-piecewise testable* (n-PTL) over $(X, \leq)$ if S is a union of n-piecewise equivalence classes. A subset S of X is *piecewise testable* (PTL) if it is an n-PTL for some $n \geq 0$. This is well-known to be equivalent to the following formulation:

▶ **Fact 3.** *S is an n-PTL over $(X, \leq)$ if and only if S is a finite Boolean combination of* principal filters $\uparrow x \stackrel{\text{def}}{=} \{x' \in X \mid x \leq x'\}$ *where $x \in X_{\leq n}$.*

2.2 Logical Characterisation

Although the choice of the particular quasi-order $(X, \leq)$ is irrelevant to Fact 3 and the treatment in Section 3, one normally chooses X to be a class $\mathcal{K}$ of finite structures over some signature σ and $\leq$ to be the *induced substructure* ordering, which we denote by '$\sqsubseteq$'. When $(\mathcal{K}, \sqsubseteq)$ is a *well-quasi-order*, this leads to a well-known logical characterisation in terms of the Boolean closure of existential first-order formulæ—one of the chief motivations for studying piecewise testable languages. Other orderings may lead to different logical characterisations, or have no logical characterisation.

Finite Structures. We consider finite relational structures $\mathfrak{M} = \langle M, (R_i)_{i \in I} \rangle$ over a finite signature $\sigma = (R_i)_{i \in I}$ (we conflate the names of relations in σ with their interpretations in $\mathfrak{M}$). The finite set M is the (potentially empty) domain of $\mathfrak{M}$ and each relation $R_i \subseteq M^{r_i}$ has a prescribed arity $r_i > 0$. The size $|\mathfrak{M}|$ of $\mathfrak{M}$ is the cardinality of its domain. A class $\mathcal{K}$ of structures is a set of such finite structures sharing the same signature σ and closed under isomorphism.

▶ **Example 4** (Finite Words)**.** Let Σ be a finite alphabet; we write Σ^* for the set of finite words over Σ. Such a word w can be seen as a relational structure over $\sigma_s \stackrel{\text{def}}{=} ((P_a)_{a \in \Sigma}, <)$ with its set of positions $\{1, \ldots, |w|\}$ as domain (where $|w|$ denotes the length of w, and thus coincides with its size), and $P_a(x)$ for $x \in \{1, \ldots, |w|\}$ holds if and only if the xth symbol of w is $a \in \Sigma$, while $x < y$ has the obvious interpretation.

▶ **Example 5** (Ranked Trees)**.** Fix a finite ranked alphabet $\mathcal{F}$, where each symbol f is mapped to a single rank $r(f) \geq 0$, and write $\mathcal{F}_r$ for the subset of $\mathcal{F}$ with rank r. A (finite, ranked, ordered) tree is defined inductively as a term $f(t_1, \ldots, t_r)$ where $f \in \mathcal{F}_r$ and $t_1, \ldots, t_r$ are trees [see e.g. 10]. The notions of node, child, parent, descendant, and ancestor relations between nodes of a tree are defined as usual. We denote by $T(\mathcal{F})$ the set of trees over $\mathcal{F}$.

A tree can be seen as a structure with its set of nodes as domain—note that this domain is never empty by definition—; several signatures are then pertinent [see 7, and Appendix B]. We shall focus on the signature $\sigma_T \stackrel{\text{def}}{=} ((P_f)_{f\in\mathcal{F}}, <, <_{\text{dfs}}, \sqcap)$ where, for all nodes x, y,

- $P_f(x)$ holds whenever x is labelled by f,
- $x < y$ whenever x is an ancestor of y,
- $x <_{\text{dfs}} y$ whenever x is visited before y in a depth-first, left-first traversal of the tree—this is known as the *document order*—, and
- $z = x \sqcap y$ whenever z is the *least* common ancestor of x and y, i.e. z is the unique descendant of all the nodes which are ancestors of both x and y.

Embeddings. Recall that $\mathfrak{A} \sqsubseteq \mathfrak{B}$ holds between two structures $\mathfrak{A}$ and $\mathfrak{B}$ if and only if there exists an *embedding* from $\mathfrak{A}$ to $\mathfrak{B}$, i.e. an injective mapping e from A to B that preserves the relations: for all $i \in I$ and $a_1, \dots, a_{r_i} \in A$, $(a_1, \dots, a_{r_i}) \in R_i$ in $\mathfrak{A}$ if and only if $(e(a_1), \dots, e(a_{r_i})) \in R_i$ in $\mathfrak{B}$. As $\sqsubseteq$ is transitive and reflexive, $(\mathcal{K}, \sqsubseteq)$ is a *quasi-order* (a qo).

Well-Quasi-Orders. Given a qo $(X, \leq)$ and a subset $S \subseteq X$, the *downward-closure* of S is $\downarrow S \stackrel{\text{def}}{=} \{x \in X \mid \exists s \in S \,.\, x \leq s\}$. A subset $D \subseteq X$ is *downwards-closed* if $D = \downarrow D$; when D is a singleton $\{x\}$ we write more succinctly $\downarrow x$. The notions of *upward-closure* and *upwards-closed* subsets are defined similarly.

A qo $(X, \leq)$ is a *well-quasi-order* (wqo) if in every infinite sequence $x_0, x_1, \dots$ of elements of X, one can find an infinite sequence of indices $i_0 < i_1 < \cdots$ such that $x_{i_0} \leq x_{i_1} \leq \cdots$ [23, 25]. Equivalently, $(X, \leq)$ is well-founded (there are no infinite descending sequences $x_0 > x_1 > \cdots$) and satisfies the *finite antichain condition* (FAC), i.e. all its antichains are finite. Still equivalently, $(X, \leq)$ has the *descending chain condition*: there are no infinite descending sequences $D_0 \supsetneq D_1 \supsetneq \cdots$ of downwards-closed subsets $D_i \subseteq X$. For instance, any finite set ordered by equality is a wqo by the Pigeonhole Principle. As another example, if $(X, \leq_X)$ and $(Y, \leq_Y)$ are two wqos, then by Dickson's Lemma, their Cartesian product $X \times Y$ is well-quasi-ordered by the *product ordering* defined by $(x, y) \leq (x', y')$ if and only if $x \leq_X x'$ and $y \leq_Y y'$.

▶ **Example 6** (Scattered Subword Embedding)**.** The signature σ_s of Example 4 for finite words in Σ^* gives rise to an embedding relation $\sqsubseteq_s$ known as the (scattered) *subword embedding*: $u \sqsubseteq_s v$ if and only if $u = a_1 \cdots a_m$ and $v = v_0 a_1 v_1 \cdots v_{m-1} a_m v_m$ for some $a_1, \dots, a_m \in \Sigma$ and $v_0, \dots, v_m \in \Sigma^*$.

By Higman's Lemma [23], the subword embedding relation $\sqsubseteq_s$ well-quasi-orders Σ^*.

▶ **Example 7** (Homeomorphic Tree Embedding)**.** Using the signature σ_T of Example 5 for trees in $T(\mathcal{F})$, the ordering $t \sqsubseteq_T t'$ is better known as the *homeomorphic tree embedding* relation, and holds for $t = f(t_1, \dots, t_r)$ and $t' = g(t'_1, \dots, t'_s)$ if and only if

- there exists $1 \leq i \leq s$ such that $t \sqsubseteq_T t'_i$, or
- $f = g$ (and thus $r = s$) and for every $1 \leq i \leq r$, $t_i \sqsubseteq_T t'_i$;

see Appendix B for a proof of this folklore result. Note that $(\Sigma^*, \sqsubseteq_s)$ is isomorphic to $(T(\mathcal{F}_\Sigma), \sqsubseteq_T)$ where $\mathcal{F}_\Sigma \stackrel{\text{def}}{=} \Sigma \uplus \{\$\}$ assigns rank 1 to letters in Σ and rank 0 to \$, so our results on ranked trees properly generalise the case of finite words.

As first shown by Higman [23], the homeomorphic tree embedding relation $\sqsubseteq_T$ similarly well-quasi-orders $T(\mathcal{F})$, and Kruskal's Tree Theorem [25] shows that this generalises to unranked trees.

Examples of WQOs on Relational Structures. Finite trees are also well-quasi-ordered by the coarser *tree minor* relation [19] (this is the main ordering considered by Bojańczyk et al. [7]; see Appendix B for details). More examples of well-quasi-ordered classes of structures under embedding are provided by finite undirected graphs of bounded tree-depth over a signature containing only the edge relation [14]—i.e. the embedding relation is the induced subgraph ordering—; see also [4] for some classes of labelled graphs.

Existential First-Order Logic. Consider first-order logic over the signature σ, and a class of structures $\mathcal{K}$ over σ. Given a sentence φ, the set of structures $\mathfrak{M} \in \mathcal{K}$ such that $\mathfrak{M} \models \varphi$ is called the (first-order) *language* of φ. It is well-known that *existential sentences* in $\Sigma_1(\sigma)$ define *upwards-closed* first-order languages S with respect to embeddings, i.e. such that $S = \uparrow S \stackrel{\text{def}}{=} \{\mathfrak{M}' \in \mathcal{K} \mid \exists \mathfrak{M} \in S . \mathfrak{M} \sqsubseteq \mathfrak{M}'\}$; however the converse does not necessarily hold [36].

We are interested here in the Boolean closure $\mathcal{B}\Sigma_1(\sigma)$ of $\Sigma_1(\sigma)$:

▶ **Fact 8.** *If a set S is an n-PTL over $(\mathcal{K}, \sqsubseteq)$, then it is definable by a sentence in $\mathcal{B}\Sigma_1(\sigma)$ with at most n variables. Conversely, assuming additionally that $(\mathcal{K}, \sqsubseteq)$ is a wqo, if S is definable by a sentence in $\mathcal{B}\Sigma_1(\sigma)$, then it is a PTL over $(\mathcal{K}, \sqsubseteq)$.*

3 Ideal Characterisation of PTL Separability

A set $S \subseteq X$ *separates* $L \subseteq X$ from $L' \subseteq X$ if $L \subseteq S$ and $S \cap L' = \emptyset$. This can be turned into a decision problem when restricting L and L' to a class $\mathcal{C} \subseteq 2^X$ of finitely representable sets:

▶ **Problem :** PTL separability for $\mathcal{C}$ over $(X, \leq)$
input representations of L and L' from $\mathcal{C}$
question are L and L' *PTL separable*, i.e. does there exist S a PTL over $(X, \leq)$ that separates L from L'?

Throughout this section, we assume that $(X, \leq)$ is a well-quasi-order. Under this assumption, we establish in Section 3.2 a characterisation of PTL separability in terms of *ideals* of $(X, \leq)$ (whose definition we recall in Section 3.1). This yields a generic framework in which the decidability of PTL separability can be tackled, which we present in sections 4 and 5 and instantiate for the case of $(T(\mathcal{F}), \sqsubseteq_T)$ the set of ranked trees together with homeomorphic embeddings against the class $\mathcal{C} = \mathrm{Reg}(T(\mathcal{F}))$ of regular tree languages.

3.1 Ideals

An *ideal* of a qo $(X, \leq)$ is a downwards-closed and (up-)*directed* subset $I \subseteq X$, where this last condition ensures that I is non-empty and that, given any $x \in I$ and $y \in I$, there exists $z \in I$ with $x \leq z$ and $y \leq z$. A related notion is the following. A downwards-closed subset D of $(X, \leq)$ is *irreducible* if and only if it is non-empty, and for any two downwards-closed subsets D_1, D_2 such that $D \subseteq D_1 \cup D_2$, D is contained in D_1 or in D_2 already; equivalently, D is non-empty and cannot be written as the union of two proper, downwards-closed subsets.

▶ **Fact 9** (c.f. Lemma 1 in [8]). *The following are equivalent for a downwards-closed subset D of a qo: (a) D is an ideal; (b) D is directed; (c) D is irreducible.*

We write $\mathrm{Idl}(X)$ for the set of ideals of X. Ideals are especially useful when $(X, \leq)$ is a wqo: for one thing, when X is countable, $\mathrm{Idl}(X)$ is then also countable [8, Theorem 1], and furthermore any downwards-closed subset has a decomposition as a finite union of ideals:

▶ **Fact 10** (c.f. Lemma 2 in [8]). *A qo is FAC if and only if every downwards-closed subset is a finite union of ideals.*

3.2 Characterising Separability

Over any qo $(X, \leq)$, the case where L and L' are not PTL separable has a well-known characterisation in terms of sequences of indistinguishable witnesses, which is a straightforward consequence of the definitions:

▶ **Lemma 11.** *L and L' are* not *PTL separable over a qo $(X, \leq)$ if and only if there exist two sequences of elements $(x_n)_{n\in\mathbb{N}}$ in L and $(x'_n)_{n\in\mathbb{N}}$ in L' such that for every n, $x_n \equiv_n x'_n$.*

Proof Idea. It suffices to observe that L and L' are not n-PTL separable if and only if there exist $x_n \in L$ and $x'_n \in L'$ such that $x_n \equiv_n x'_n$. See Appendix A for a detailed proof. ◀

When $(X, \leq)$ is a wqo, we have another characterisation in terms of directed subsets of L and L' defining the same ideal. The fact that $(X, \leq)$ is wqo is needed in order for Lemma 12 to hold: otherwise, Appendix B.4 presents a counter-example for the *subtree ordering* over $T(\mathcal{F})$.

▶ **Lemma 12** (Key Lemma)**.** *L and L' are not PTL separable over a wqo $(X, \leq)$ if and only if there exist two directed sets $\Delta \subseteq L$ and $\Delta' \subseteq L'$ such that $\downarrow\Delta = \downarrow\Delta'$.*

Proof of 'if'. Let $\Delta \subseteq L$ and $\Delta' \subseteq L'$ be two directed sets with $\downarrow\Delta = \downarrow\Delta'$. Let us show that for every $n \in \mathbb{N}$, there exist $x_n \in L$ and $x'_n \in L'$ such that $x_n \equiv_n x'_n$, from which Lemma 11 yields the result.

Write I for the ideal $\downarrow\Delta = \downarrow\Delta'$. Observe that, for every $n \in \mathbb{N}$, $I \cap X_{\leq n}$ is finite since X is a combinatorial class. Furthermore, for every $z \in I \cap X_{\leq n}$, by definition of I there exist $x_z \in \Delta$ with $z \leq x_z$ and $x'_z \in \Delta'$ with $z \leq x'_z$. Since Δ and Δ' are directed, we can find $x_n \in \Delta$ and $x'_n \in \Delta'$ greater or equal to all those finitely many x_z and x'_z when z ranges over $I \cap X_{\leq n}$ (if $I \cap X_{\leq n} = \emptyset$ then any $x_n \in \Delta$ and $x'_n \in \Delta'$ fit). Then $x_n \equiv_n x'_n$, since for any $z \in X_{\leq n}$, either $z \in I$ and then both $z \leq x_z \leq x_n$ and $z \leq x'_z \leq x'_n$, or $z \notin I$ and then both $z \not\leq x_n$ and $z \not\leq x'_n$ since I is downwards-closed. ◀

Proof of 'only if'. Assume L and L' are not PTL separable, hence by Lemma 11 there exist two infinite sequences $(x_n)_n$ in L and $(x'_n)_n$ in L' with $x_n \equiv_n x'_n$ for every n.

Let us consider the infinite sequence of *pairs* $(x_n, x'_n)_{n\in\mathbb{N}}$. By Dickson's Lemma, $X \times X$ is a wqo for the product ordering, hence there exists an infinite sequence of indices $i_0 < i_1 < \cdots$ such that $x_{i_j} \leq x_{i_{j+1}}$ and $x'_{i_j} \leq x'_{i_{j+1}}$ for every $j \in \mathbb{N}$. Define $\Delta \stackrel{\text{def}}{=} \{x_{i_j} \mid j \in \mathbb{N}\}$ and $\Delta' \stackrel{\text{def}}{=} \{x'_{i_j} \mid j \in \mathbb{N}\}$. These two sets are directed; they are actually infinite chains $x_{i_0} \leq x_{i_1} \leq \cdots$ and $x'_{i_0} \leq x'_{i_1} \leq \cdots$.

It remains to show that $\downarrow\Delta = \downarrow\Delta'$. By symmetry, it suffices to show $\Delta \subseteq \downarrow\Delta'$. Consider some $x_{i_j} \in \Delta$; then there exists some index $i_k > \max(i_j, |x_{i_j}|)$. Hence $x_{i_j} \leq x_{i_k}$, and since $x_{i_k} \equiv_{i_k} x'_{i_k}$, $x_{i_j} \leq x'_{i_k}$ and thus $x_{i_j} \in \downarrow\Delta'$. ◀

Related Work over Finite Words. Let S be a subset of X. We define the *adherence* of S as the set of ideals defined by the directed subsets of S:

$$\mathrm{Adh}(S) \stackrel{\text{def}}{=} \{\downarrow\Delta \in \mathrm{Idl}(X) \mid \Delta \subseteq S \text{ is directed } \}\,. \tag{1}$$

Lemma 12 can be restated as saying that L and L' are PTL separable if and only if their adherences are disjoint, i.e. $\mathrm{Adh}(L) \cap \mathrm{Adh}(L') = \emptyset$. This makes Lemma 12 quite reminiscent of several results on separability for word languages over Σ^*. Almeida [2] showed that two regular languages over Σ are PTL separable over $(\Sigma^*, \sqsubseteq_s)$ if and only if their topological closures inside a specific profinite semigroup do not intersect. This lead to a first decision

procedure [3] by explicitly constructing representations for these topological closures and testing their intersection for emptiness—the counterpart in terms of Lemma 12 would be to compute the adherences of L and L'.

Major improvements were brought by Place et al. [32] and Czerwiński et al. [12] and culminate in Theorem 2.1 of [13], by reducing this non-empty intersection check to identifying a common pattern 'densely' matched by both L and L'; this leads to a decidable PTL separability for many classes $\mathcal{C} \subseteq 2^{\Sigma^*}$, for instance context-free languages and languages of labelled vector addition systems [13] and higher-order IO languages [21, 9]. It turns out that these patterns are essentially finite representations for ideals of $(\Sigma^*, \sqsubseteq_s)$, so Lemma 12 subsumes Theorem 2.1 of Czerwiński et al. [13]—and has a considerably simpler proof.

4 Deciding PTL Separability

While Lemma 12 provides a general characterisation for PTL separability, turning it into a decision procedure requires finite representations and effectiveness assumptions on its various ingredients. We define a set of such assumptions in Section 4.1, which is sufficient to derive a generic algorithm. We describe the latter in two steps: we first show in Section 4.2 a reduction to the *adherence membership problem*. The final step in Section 4.3 is to show that this last problem is decidable for the set of regular tree languages over $(T(\mathcal{F}), \sqsubseteq_T)$.

4.1 Effectiveness Assumptions

In order to put Lemma 12 into practice, we need to consider in more details how we are going to represent PTLs over $(X, \leq)$, ideals in $\mathrm{Idl}(X)$, and languages in $\mathcal{C}$.

Effective PTL Representations. Fact 3 provides a natural representation for PTLs as finite Boolean combinations of principal filters, i.e. more concretely as terms of the free Boolean algebra with elements of X as atoms.

Effective Ideal Representations. Recall that over a countable wqo $(X, \leq)$, $\mathrm{Idl}(X)$ is also countable [8]. In Section 4.2, we only need to have explicit *ideal representations* as a means of enumerating ideals, while Corollary 15 further needs a means of computing representations as regular tree languages. Section 5 fulfils both requirements by providing ideal representations for $(T(\mathcal{F}), \sqsubseteq_T)$ as regular tree expressions.

Effective Classes of Languages. Our last effectiveness assumptions regard the class of languages $\mathcal{C}$. We call $\mathcal{C}$ *PTL-effective* over a qo $(X, \leq)$ if

- $\mathcal{C}$ has *decidable emptiness*: given a representation for $L \in \mathcal{C}$, there is an algorithm that decides whether $L = \emptyset$, and if
- $\mathcal{C}$ is *effectively closed under intersection with PTLs*: given a representation for $L \in \mathcal{C}$ and one for S a PTL over $(X, \leq)$, there is an algorithm that constructs a representation for $L \cap S$ in $\mathcal{C}$.

For instance, both over $(\Sigma^*, \sqsubseteq_s)$ and over $(T(\mathcal{F}), \sqsubseteq_T)$, a principal filter $\uparrow x$ is a regular language, and since regular languages are closed under Boolean operations by Fact 3 any PTL is regular. Thus, PTL-effective classes of word and tree languages abound, starting with regular languages themselves, but also context-free, etc.

4.2 Reducing to Adherence Membership

We describe our decision procedure in two steps. The first one reduces the PTL separability problem for a PTL-effective class $\mathcal{C}$ to the following problem:

▶ **Problem :** Adherence Membership for $\mathcal{C}$ over $(X, \leq)$
input a representation for $L \in \mathcal{C}$ and one for $I \in \mathrm{Idl}(X)$
question is $I \in \mathrm{Adh}(L)$?

▶ **Proposition 13.** *Let $(X, \leq)$ be a wqo with ideal representations and $\mathcal{C} \subseteq 2^X$ be PTL-effective over $(X, \leq)$. Then there is a Turing reduction from the PTL separability problem to the adherence membership problem.*

Proof. Given an oracle for the adherence membership problem for $\mathcal{C}$ over $(X, \leq)$, our algorithm consists of two semi-decision procedures that take representations for $L \in \mathcal{C}$ and $L' \in \mathcal{C}$ as input. The first attempts to show that L and L' are PTL separable, and enumerates representations of PTLs S until $L \cap (X \setminus S) = \emptyset$ and $L' \cap S = \emptyset$. This is possible because $\mathcal{C}$ is PTL-effective and complementing a PTL representation is trivial. The second relies on Lemma 12 and attempts to show that L and L' are not PTL separable by enumerating representations of ideals I until $I \in \mathrm{Adh}(L)$ and $I \in \mathrm{Adh}(L')$, using the oracle for adherence membership for the tests. ◀

4.3 Deciding Adherence Membership

Regular Languages. In the case of regular tree languages over $T(\mathcal{F})$, the adherence membership problem is decidable thanks to the following lemma:

▶ **Lemma 14.** *Let $(X, \leq)$ be a qo and $L \subseteq X$. Then $I \in \mathrm{Adh}(L)$ if and only if $I \subseteq {\downarrow}(I \cap L)$.*

Proof. The 'only if' part is immediate since $I = {\downarrow}\Delta$ for $\Delta \subseteq L$ implies $\Delta \subseteq I \cap L$ and thus $I \subseteq {\downarrow}(I \cap L)$. For the 'if' part we show that $I \cap L$ is directed. Since I is non-empty and included in ${\downarrow}(I \cap L)$, $I \cap L$ is also non-empty. Furthermore, if x, y are in $I \cap L$, then since I is directed there exists $z \in I$ such that $x \leq z$ and $y \leq z$, and since $I \subseteq {\downarrow}(I \cap L)$ there exists $z' \in I \cap L$ such that $z \leq z'$. ◀

Now, if L is a regular tree language, I is also effectively regular using the ideal representations from Section 5, and so are $I \cap L$ and ${\downarrow}(I \cap L)$, hence $I \subseteq {\downarrow}(I \cap L)$ is decidable since inclusion of regular tree languages is decidable, proving Theorem 1:

▶ **Corollary 15.** *Adherence membership is decidable for regular tree languages over $(T(\mathcal{F}), \sqsubseteq_T)$.*

Generic Approach. Let us finally generalise the previous idea. Our issue is that, for instance when $\mathcal{C}$ is the class of context-free languages over Σ^*, while I is a regular language and ${\downarrow}(I \cap L)$ is a computable context-free language, the inclusion test between a regular language and a context-free one is in general undecidable. Thankfully, ${\downarrow}(I \cap L)$ is regular for arbitrary languages L by Fact 10, since it is a finite union of ideals and ideals are regular. However, the issue is then to compute a representation of ${\downarrow}(I \cap L)$ *as a regular language*, or more generally to solve the following problem:

▶ **Problem :** Ideal Decomposition for $\mathcal{C}$ over $(X, \leq)$
input a representation for $L \in \mathcal{C}$
output ${\downarrow}L$ as a finite union of representations of ideals in $\mathrm{Idl}(X)$.

The following result requires effective operations on ideal representations. We refer the reader to [16, 18] for a systematic study of algorithms on finite ideal representations for wqos; here we use the notion of *effective* ideal representations as defined by Goubault-Larrecq et al. [18] to prove (see Appendix C.2):

▶ **Proposition 16.** *Let $(X, \leq)$ be a wqo with effective ideal representations and $\mathcal{C} \subseteq 2^X$ be PTL-effective over $(X, \leq)$. Then the adherence membership problem and the ideal decomposition problem are Turing-equivalent.*

Two remarks are in order. The first is that the seemingly innocuous hypothesis that $\mathcal{C}$ is PTL-effective is actually crucial: Theorem V.2 and Theorem VIII.1 in [27] provide an instance of undecidable adherence membership but computable ideal decompositions over a wqo with effective ideal representations; in these results, $\mathcal{C}$ is the set of run structures between two configurations of a vector addition system, and has decidable emptiness but lacks effective closure under intersection with PTLs. The second is that it might seem overkill to ask for ideal decompositions when regular representations for $\downarrow L$ would work just as well in practice; however the proofs in Section 5 also entail that ideal decompositions are computable for regular tree languages over $(T(\mathcal{F}), \sqsubseteq_T)$, hence the two are equivalent.

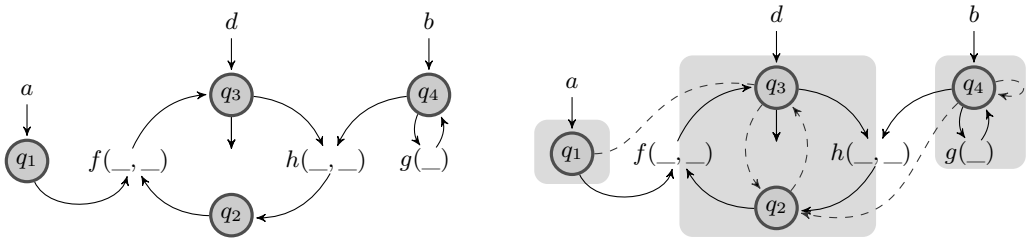
Related Work over Finite Words. Propositions 13 and 16 together show that PTL separability reduces to the computation of ideal decompositions for downward-closures over any wqo with effective ideal representations. This includes as a special case $(\Sigma^*, \sqsubseteq_s)$ using the representations in [24, 1], and since ideal decompositions of downward-closures are computable for many classes of PTL-effective word languages [38]—including context-free languages [37, 11], reachability languages of vector addition systems [20], matrix languages [38], and higher-order languages [21, 9]—, PTL separability is decidable for them.

In fact, propositions 13 and 16 subsume most of Theorem 2.5 of Czerwiński et al. [13], which is stated for *full trios* $\mathcal{C}$ over Σ^*, i.e. classes of languages closed under rational transductions—which are thus effectively closed under intersection with PTLs. There is a small price to pay for our level of generality, which is that their Theorem 2.5 also shows a converse reduction over $(\Sigma^*, \sqsubseteq_s)$ from the ideal decomposition problem back to the PTL separability problem. In our general setting, the best we know is that $L \stackrel{\text{def}}{=} I$ and $L' \stackrel{\text{def}}{=} X \setminus \downarrow(I \cap S)$ are PTL-separable if and only if $I \in \mathrm{Adh}(S)$ by Lemma 14, but this either uses the closure of $\mathcal{C}$ under complementation and downward-closure, or already uses computable ideal decompositions. The former however holds for regular languages:

▶ **Proposition 17.** *There is a many-one reduction from the adherence membership problem to the PTL separability problem for regular tree languages over $(T(\mathcal{F}), \sqsubseteq_T)$.*

5 Ideals for Ranked Trees with Homeomorphic Embedding

In this section, we provide finite representations for ideals of ranked trees ordered by homeomorphic embedding. These representations are expressed as tree regular expressions [10, Section 2.2]. We show in Section 5.1 that any downwards-closed subset of $T(\mathcal{F})$ can be represented as a *simple tree regular expression* (STRE), which we construct from its tree automaton. In Section 5.2, we then characterise ideals in this syntax as *tree products*, obtained as the summands of the normal form according to a rewrite system. Thanks to this particular proof strategy, the entire construction also solves the ideal decomposition problem: given a regular tree language, first build the STRE for its downward-closure, then normalise this STRE into a union of tree products representing its ideals.



(a) Initial tree automaton. **(b)** ε-strongly connected components.

Figure 1 Converting tree automata to STREs.

Note that a concrete syntax for ranked tree ideals was proposed in [16]. However, no proof was given, and indeed the proposed tree regular expressions failed to be downwards-closed.

5.1 Simple Tree Regular Expressions

Let $L \subseteq T(\mathcal{F})$ be a downwards-closed language. Its complement is upwards-closed and has finitely many minimal elements, i.e. $T(\mathcal{F}) \setminus L = {\uparrow}\{t_1, \ldots, t_n\} = \bigcup_{1 \leq i \leq n} {\uparrow} t_i$, hence this is a regular tree language and L is also regular. Thus L is recognised by a finite (bottom-up) tree automaton $\mathcal{A}$, or equivalently by Kleene's Theorem for tree languages,by a tree regular expression [10, Section 2.2].

We shall see that $\mathcal{A}$ is equivalent to an expression of a specific shape, called a *simple tree regular expression* (STRE). We describe next a general procedure that converts any (ε-free) finite tree automaton $\mathcal{A}$ to an STRE S whose language is the downward-closure ${\downarrow} L(\mathcal{A})$ of the language recognised by $\mathcal{A}$. This procedure is best explained on an example: see Figure 1a, where there is one 0-ary transition a (from no state) to state q_1, one binary transition f from the pair of states q_1, q_2 to q_3, and so on. In textual form, we write these transitions as rewrite rules [10]: $a \to q_1$, $f(q_1, q_2) \to q_3$, $h(q_3, q_3) \to q_2$, $c \to q_3$, $g(q_4) \to q_4$, $b \to q_4$. A tree t is *recognised* at a state s if and only if $t \to^* s$, using the rewrite rules of $\mathcal{A}$. There is a set of final states, marked with an outgoing arrow with dangling end: in our example, just q_3. The language $L(\mathcal{A})$ of $\mathcal{A}$ is the set of trees recognised at some final state.

We first extend our automaton with ε-transitions. An ε-transition from s to s' will be drawn as a dashed arrow (see Figure 1b), and is just a rewrite rule of the new form $s \to s'$. This implies that every tree recognised at s is also recognised at s'. For each transition, say $f(s_1, s_2, \cdots, s_n) \to s$, of $\mathcal{A}$, we add n ε-transitions $s_1 \to s$, $s_2 \to s$, $\ldots$, $s_n \to s$. (To make things clear, we are assuming that $\mathcal{A}$ does not originally contain any ε-transition.)Call the resulting automaton ${\downarrow}\mathcal{A}$. It is an easy exercise to show that $L({\downarrow}\mathcal{A}) = {\downarrow} L(\mathcal{A})$.

There is a graph underlying ${\downarrow}\mathcal{A}$, whose vertices are the states of ${\downarrow}\mathcal{A}$, and whose edges are the ε-transitions. Consider its strongly connected components, shown against a grey background in Figure 1b. Any two states in the same strongly connected component C recognise exactly the same trees, so it makes sense to talk of the language $L_C({\downarrow}\mathcal{A})$ of those trees recognised at any state of C. Let $C \to C'$ if and only if $s \to s'$ for some $s \in C$, $s' \in C'$. The strict ordering $\prec \stackrel{\text{def}}{=} \to^+$ is well-founded, and we shall build an expression S_C whose language is $L_C({\downarrow}\mathcal{A})$ by induction along $\prec$.

Trivial Components. If C is a trivial strongly connected component (one state s, no self-edge), then enumerate its incoming non-ε transitions $f_i(s_{i1}, s_{i2}, \cdots, s_{in_i}) \to s$, $1 \leq i \leq m$. Let S_{ij} be an expression whose language is the set of trees recognised at s_{ij}, which is given

by induction hypothesis. Then $S_C \stackrel{\text{def}}{=} P_1 + \cdots + P_m$ with $P_i \stackrel{\text{def}}{=} f_i^?(S_{i1}, S_{i2}, \cdots, S_{in_i})$ is our desired expression. For instance, the set of trees recognised at the leftmost state q_1 of Figure 1b is the language of $S_1 \stackrel{\text{def}}{=} a^?$.

Here, a tree t is in the language of an expression $P = f^?(S_1, \ldots, S_n)$ (written '$t \in P$') for $f \in \mathcal{F}_n$ and expressions $S_1, \ldots, S_n$ if and only if either t is of the form $f(t_1, \cdots, t_n)$ with $t_i \in S_i$ for every i, $1 \leq i \leq n$, *or* if $t \in \bigcup_{i=1}^n S_i$; as the notation suggests, for $S = P_1 + \cdots + P_m$, $t \in S$ if and only if $t \in P_j$ for some j, $1 \leq j \leq m$.

Iterators. If C is a non-trivial strongly connected component, then enumerate the non-ε transitions $f_i(s_{i1}, s_{i2}, \cdots, s_{in_i}) \to s_i$, $1 \leq i \leq m$, whose end state s_i is in C. For each pair i, j, if s_{ij} is in C, then $S_{\Box ij} \stackrel{\text{def}}{=} \Box$ is a placeholder; otherwise, let $S_{\Box ij}$ be an expression whose language is the set of trees recognised at s_{ij}, which we obtain by induction hypothesis. Then $S_C \stackrel{\text{def}}{=} (A_1 + \cdots + A_m)^*.0$ for $A_i \stackrel{\text{def}}{=} f_i(S_{\Box i1}, S_{\Box i2}, \cdots, S_{\Box in_i})$ is a suitable expression. For example, the rightmost strongly connected component $\{q_4\}$ of Figure 1b yields the expression $S_4 \stackrel{\text{def}}{=} (b + g(\Box))^*.0$. One might have expected an expression of the more intuitive form $(g(\Box))^*.b^?$, however, as we are going to see, they define exactly the same language.

Here, 0 denotes the empty sum with empty language, and intuitively the language of an *iterator* $C^*.S$ for $C = A_1 + \cdots + A_m$ should consist of all trees obtained by repeatedly applying *contexts* from $A_1, \ldots, A_m$—i.e. trees over the extended alphabet $\mathcal{F} \cup \{\Box\}$ where $\Box$ has arity 0—, until one reaches a tree in S. More precisely, the language of an *atom* $A = f(S_{\Box 1}, \cdots, S_{\Box n})$ consists of those contexts in $T(\mathcal{F} \cup \{\Box\})$ of the form $f(c_1, \cdots, c_n)$ where $c_i \in S_{\Box i}$ for every i. In turn, $c \in S_\Box$ if and only if either $S_\Box = \Box$ and c is the trivial context $\Box$, or $S_\Box$ is an expression S, c is a tree t in $T(\mathcal{F})$, and $t \in S$. The language of $C = A_1 + \cdots + A_m$ is the union of the languages of A_j, $1 \leq j \leq m$. For example, $(f(\Box))^*.a^?$ will recognise all trees of the form $f^n(a)$, $n \in \mathbb{N}$. There are however two catcheswith the semantics of iterators:

1. The first one has to do with atoms A where the placeholder $\Box$ occurs more than once: as usual with tree regular expressions, when replacing $\Box$ by a tree from S, several occurrences of $\Box$ can be replaced by *different* trees from S. Hence $(f(\Box, \Box))^*.(a^? + b^?)$ consists of all binary-branching trees with inner nodes labelled f and leaves labelled a or b, including $f(f(a, a), a)$ and $f(f(b, b), b)$ but also $f(f(a, b), a))$ or $f(f(b, b), a)$ among others. (We assume f binary, and a and b of rank 0.)For a context c, and a set of trees S, we shall write $c[S]$ for the set of trees obtained from c by replacing each occurrence of $\Box$ by a (possibly different) tree in S.
2. The second catch has to do with downward-closure. It is tempting to define the trees of $C^*.S$ as those in $c_1[\cdots[c_k[S]]\cdots]$, for some $k \in \mathbb{N}$ and some $c_1, \cdots, c_k \in C$. However, there are cases where that language would fail to be downwards-closed, e.g., $(f(a^?, \Box))^*.b^?$ would contain $f(a, b)$ but not a, according to that semantics.

   We repair that as follows. For $A = f(S_{\Box 1}, \cdots, S_{\Box n})$, define $\operatorname{supp} A$ as the set of trees $t \in T(\mathcal{F})$ such that some context $f(\cdots, t, \cdots)$ (i.e., with one of its arguments equal to t) is in the language of A. Alternatively: if those $S_{\Box i}$, $1 \leq i \leq n$, that are different from $\Box$ define non-empty languages, then $\operatorname{supp} A$ is the union of those languages; if some $S_{\Box i} \neq \Box$ has an empty language, then $\operatorname{supp} A = \emptyset$. Hence, for example, $\operatorname{supp} f(\Box, \Box) = \emptyset$, $\operatorname{supp} f(a^?, \Box) = a^? = \{a\}$, and $\operatorname{supp} f'(a^?, \Box, 0) = \emptyset$. For $C = A_1 + \cdots + A_m$, let $\operatorname{supp} C = \bigcup_{j=1}^m \operatorname{supp} A_j$.

We are now ready to define the language of $C^*.S$, as the language of trees in $c_1[\cdots[c_k[S \cup \operatorname{supp} C]]\cdots]$, for some $k \in \mathbb{N}$ and some $c_1, \cdots, c_k \in C$.(In writing $S \cup \operatorname{supp} C$, we confuse the expression S with its language.)

$$
\begin{array}{rcll}
P + P' & \to_1 & P' & \text{if } P \subseteq P' \\
0 + P & \to_1 & P & \\
0^*.S & \to_1 & S & \\
f^?(\vec{S}_1, 0, \vec{S}_2) & \to_1 & 0 & \\
f(\vec{S}_{\Box 1}, 0, \vec{S}_2) & \to_1 & 0 & \\
C^*.0 & \to_1 & 0 & \\
C^*.(S + S') & \to_1 & C^*.S + C^*.S' & \text{if } C \text{ is } \Box\text{-linear}
\end{array}
\qquad
\begin{array}{rcll}
A + A' & \to_1 & A' & \text{if } A \subseteq A' \\
0 + A & \to_1 & A & \\
(C + f(S_1, \cdots, S_n))^*.S & \to_1 & C^*.(S + f^?(S_1, \cdots, S_n)) & \\
f^?(\vec{S}_1, S + S', \vec{S}_2) & \to_1 & f^?(\vec{S}_1, S, \vec{S}_2) + f^?(\vec{S}_1, S', \vec{S}_2) & \\
f(\vec{S}_{\Box 1}, S_\Box + S'_\Box, \vec{S}_{\Box 2}) & \to_1 & f(\vec{S}_{\Box 1}, S_\Box, \vec{S}_{\Box 2}) + f(\vec{S}_{\Box 1}, S'_\Box, \vec{S}_{\Box 2}) & \\
\multicolumn{4}{r}{\text{if } C = \textstyle\sum_{i=1}^m f_i(\Box, \cdots, \Box) \text{ and no } f_i \text{ is 0-ary}}
\end{array}
$$

Figure 2 The rewrite relation $\to_1$.

Finally, $\downarrow L(\mathcal{A})$ is the union of the languages of the strongly connected components containing a final state of $\mathcal{A}$; in our example the strongly connected component in the middle yields the final expression $(d + f(S_1, \Box) + h(\Box, S_4))^*.0$.

General Syntax. Summing up, we define *simple tree regular expressions* (STREs) by the following abstract syntax:

$$
\begin{array}{lll}
S ::= P_1 + \cdots + P_m & P ::= f^?(S, \cdots, S) \mid C^*.S & \\
C ::= A + \cdots + A & A ::= f(S_\Box, \cdots, S_\Box) & S_\Box ::= S \mid \Box
\end{array}
$$

where $f \in \mathcal{F}_r$ in $f^?(S_1, \cdots, S_r)$ and in $f(S_{\Box 1}, \cdots, S_{\Box r})$, the sum operation $+$ is associative and commutative (we shall sometimes write $\sum_{i=1}^m P_i$ for $P_1 + \cdots + P_m$) with 0 denoting the empty sum, and $\Box \notin \mathcal{F}$ is a placeholder. Note that $\Box$ is not meant to denote a family of placeholders, rather a single one. The STREs of the form P are called *tree pre-products*. Among them, the *tree products* will be our notations for ideals, see Section 5.2.

▶ **Proposition 18.** *Every STRE defines a downwards-closed language of $(T(\mathcal{F}), \sqsubseteq_T)$. Every downwards-closed language of $(T(\mathcal{F}), \sqsubseteq_T)$ is the language of some STRE.*

Proof. The first part can be shown by structural induction on STREs; see Appendix C.3 for details. The second part was sketched above. ◀

5.2 Tree Products

We now characterise the STREs that define ideals of $(T(\mathcal{F}), \sqsubseteq_T)$. We define a rewrite relation $\to_1$ on STREs that moves all $+$ signs to the outside: for a $\to_1$-normal STRE $S = P_1 + \cdots + P_m$, each P_i will be irreducible, hence S will be an ideal if and only if $m = 1$ by Fact 9.

The rewrite relation $\to_1$ is defined in Figure 2. Recall that $+$ is understood modulo associativity and commutativity. Letters matter, too: S, S', S_1, …, S_n are STREs, while P, P' are those special STREs of the form $f^?(S_1, \cdots, S_n)$ or $C^*.S$, etc. In particular, the third rule of the second column applies provided the pattern $f(S_1, \cdots, S_n)$ does not contain $\Box$ at all. In the first rules, note that inclusion of STREs is decidable. For the two bottom rules, we need some auxiliary definitions. Say that a pattern $A = f(S_{\Box 1}, \cdots, S_{\Box n})$ is $\Box$-*linear* if and only if at most one $S_{\Box i}$ is the placeholder $\Box$. Writing C as $A_1 + \cdots + A_m$, we say that C is $\Box$-linear if and only if every non-empty A_i is $\Box$-linear. The $\Box$-linearity restriction imposed on the last two rules is needed for the following to hold.

▶ **Fact 19.** *If $S \to_1^* S'$ then S and S' define the same language.*

▶ **Lemma 20.** *Every STRE S has a normal form with respect to $\to_1$.*

Proof. Using Bachmair and Plaisted's associative path ordering $>_{apo}$ [6] on a precedence where $+$ is minimal, $f > f^?$ for each symbol f, and the $(\_)^*.\_$ operator has lexicographic status, we see that $\to_1$ is even a terminating relation. (Bachmair and Plaisted's ordering has been improved upon many times, but is sufficient in the case of just one associative commutative symbol $+$.) ◀

▶ **Definition 21.** A *tree product* is any $\to_1$-normal tree pre-product P.

▶ **Lemma 22.** *Every ideal of $T(\mathcal{F})$ is the language of some tree product.*

Proof. By Proposition 18, an ideal I is the language of some STRE S, which has a $\to_1$-normal form by Lemma 20; write it $P_1 + \cdots + P_m$. Since I is non-empty, $m \geq 1$, and since I is irreducible (Fact 9), it is included in, hence equal to, the language of some P_i. ◀

Conversely, the language of any tree product is directed (see Appendix C.4 for a proof), thus:

▶ **Theorem 23.** *The ideals of $T(\mathcal{F})$ are exactly the languages of tree products.*

6 Concluding Remarks

We have presented a general order-theoretic characterisation of PTL separability, and shown that it could be applied to decide PTL separability of languages beyond finite words, namely of ranked regular tree languages ordered by homeomorphic embedding. Our work further adds to the growing body of algorithmic applications of downwards-closed sets and ideals of well-quasi-orders in logic and verification, e.g. in forward analysis [16, 17], backward analysis [26], inference of inductive invariants [29], and reachability in vector addition systems [27].

We are confident our techniques apply to unranked trees, by defining suitable ideal representations. In the same vein, it would be interesting to develop ideal representations for the tree minor ordering, and to try to decide PTL separability for context-free tree languages.

An open-ended question is how to finely relate our order-theoretic characterisation with the algebraic and topological characterisations typically employed for the definability and separability problems. For instance, can one derive the characteristic equations of Bojańczyk et al. [7] for piecewise-testable tree languages from Lemma 12?

Acknowledgements. The authors thank Wojciech Czerwiński, Luc Segoufin, and Georg Zetzsche for their insightful comments, and an anonymous reviewer for correcting Fact 8.

References

1 P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Form. Methods in Syst. Des.*, 25 (1):39–65, 2004.

2 J. Almeida. Some algorithmic problems for pseudovarieties. *Publ. Math. Debrecen*, 54 (suppl.):531–552, 1999.

3 J. Almeida and M. Zeitoun. The pseudovariety J is hyperdecidable. *Theor. Inform. Appl.*, 31:457–482, 1997.

4 A. Atminas and V. V. Lozin. Labelled induced subgraphs and well-quasi-ordering. *Order*, 32(3):313–328, 2014.

5 A. Atserias, A. Dawar, and M. Grohe. Preservation under extensions on well-behaved finite structures. *SIAM J. Comput.*, 38(4):1364–1381, 2008.

6 L. Bachmair and D. A. Plaisted. Termination orderings for associative-commutative rewriting systems. *J. Logic Comput.*, 1(4):329–349, 1985.

7 M. Bojańczyk, L. Segoufin, and H. Straubing. Piecewise testable tree languages. *Logic. Meth. in Comput. Sci.*, 8(3:26), 2012.

8 R. Bonnet. On the cardinality of the set of initial intervals of a partially ordered set. In *Infinite and finite sets: to Paul Erdős on his 60th birthday, Vol. 1*, Coll. Math. Soc. János Bolyai, pages 189–198. North-Holland, 1975.

9 L. Clemente, P. Parys, S. Salvati, and I. Walukiewicz. The diagonal problem for higher-order recursion schemes is decidable. In *LICS 2016*. ACM, 2016. To appear.

10 H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. *Tree Automata Techniques and Applications*. 2007. URL `http://tata.gforge.inria.fr/`.

11 B. Courcelle. On constructing obstruction sets of words. *Bull. EATCS*, 44:178–186, 1991.

12 W. Czerwiński, W. Martens, and T. Masopust. Efficient separability of regular languages by subsequences and suffixes. In *ICALP 2013*, volume 7966 of *Lect. Notes in Comput. Sci.*, pages 150–161. Springer, 2013.

13 W. Czerwiński, W. Martens, L. van Rooijen, M. Zeitoun, and G. Zetzsche. A characterization for decidable separability by piecewise testable languages. Preprint, 2015. URL `http://arxiv.org/abs/1410.1042v2`. An extended abstract appeared as:
W. Czerwiński, W. Martens, L. van Rooijen, and M. Zeitoun. A note on decidable separability by piecewise testable languages. In *FCT 2015*, volume 9210 of *LNCS*, pages 173–185. Springer, 2015.

14 G. Ding. Subgraphs and well-quasi-ordering. *J. Graph Theory*, 16(5):489–502, 1992.

15 D. Duris. Extension preservation theorems on classes of acyclic finite structures. *SIAM J. Comput.*, 39(8):3670–3681, 2010.

16 A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In *STACS 2009*, volume 3 of *Leibniz Int. Proc. Inf.*, pages 433–444. LZI, 2009.

17 A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: Complete WSTS. *Logic. Meth. in Comput. Sci.*, 8(3:28), 2012.

18 J. Goubault-Larrecq, P. Karandikar, K. Narayan Kumar, and Ph. Schnoebelen. The ideal approach to computing closed subsets in well-quasi-orderings. In preparation, 2016. See also an earlier version in:
J. Goubault-Larrecq. On a generalization of a result by Valk and Jantzen. Research Report LSV-09-09, LSV, ENS Cachan, 2009. URL `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2009-09.pdf`.

19 A. Gupta. A constructive proof that trees are well-quasi-ordered under minors. In *LFCS 1992*, volume 620 of *Lect. Notes in Comput. Sci.*, pages 174–185. Springer, 1992.

20 P. Habermehl, R. Meyer, and H. Wimmel. The downward-closure of Petri net languages. In *ICALP 2010*, volume 6199 of *Lect. Notes in Comput. Sci.*, pages 466–477. Springer, 2010.

21 M. Hague, J. Kochems, and C.-H. L. Ong. Unboundedness and downward closures of higher-order pushdown automata. In *POPL 2016*, pages 151–163. ACM, 2016.

22 F. Harwath, L. Heimberg, and N. Schweikardt. Preservation and decomposition theorems for bounded degree structures. *Logic. Meth. in Comput. Sci.*, 11(4:17), 2015.

23 G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc.*, 3(2): 326–336, 1952.

24 P. Jullien. *Contribution à l'étude des types d'ordres dispersés*. Thèse de doctorat, Université de Marseille, 1969.

25 J. B. Kruskal. Well-quasi-ordering, the Tree Theorem, and Vazsonyi's Conjecture. *Trans. Amer. Math. Soc.*, 95(2):210–225, 1960.

26 R. Lazić and S. Schmitz. The ideal view on Rackoff's coverability technique. In *RP 2015*, volume 9328 of *Lect. Notes in Comput. Sci.*, pages 1–13. Springer, 2015.

27 J. Leroux and S. Schmitz. Demystifying reachability in vector addition systems. In *LICS 2015*, pages 56–67. IEEE Press, 2015.

28 C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Math. Proc. Cambridge Philos. Soc.*, 59(4):833–835, 1963.

29 O. Padon, N. Immerman, S. Shoham, A. Karbyshev, and M. Sagiv. Decidability of inferring inductive invariants. In *POPL 2016*, pages 217–231. ACM, 2016.

30 T. Place. Characterization of logics over ranked tree languages. In *CSL 2008*, volume 5213 of *Lect. Notes in Comput. Sci.*, pages 401–415. Springer, 2008.

31 T. Place and M. Zeitoun. Automata column: The tale of the quantifier alternation hierarchy of first-order logic over words. *SIGLOG News*, 2(3):4–17, 2015.

32 T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *MFCS 2013*, volume 8087 of *Lect. Notes in Comput. Sci.*, pages 729–740. Springer, 2013.

33 A. Sankaran, B. Adsul, and S. Chakraborty. A generalization of the Łoś-Tarski preservation theorem over classes of finite structures. In *MFCS 2014*, volume 8634 of *Lect. Notes in Comput. Sci.*, pages 474–485. Springer, 2014.

34 I. Simon. Piecewise testable events. In *Automata Theory and Formal Languages*, volume 33 of *Lect. Notes in Comput. Sci.*, pages 214–222. Springer, 1975.

35 T. G. Szymanski and J. H. Williams. Noncanonical extensions of bottom-up parsing techniques. *SIAM J. Comput.*, 5(2):231–250, 1976.

36 W. W. Tait. A counterexample to a conjecture of Scott and Suppes. *J. Symb. Log.*, 24(1): 15–16, 1959.

37 J. van Leeuwen. Effective constructions in well-partially-ordered free monoids. *Disc. Math.*, 21(3):237–252, 1978.

38 G. Zetzsche. An approach to computing downward closures. In *ICALP 2015*, volume 9135 of *Lect. Notes in Comput. Sci.*, pages 440–451. Springer, 2015.

A Basic Facts

We gather in this appendix the proofs of several basic facts about piecewise testable languages and relational structures. These all seem to be folklore results, but it cannot hurt to include formal proofs, especially since we are working in a fairly general setting.

A.1 Piecewise Testable Languages

Let us define $\downarrow_n x \stackrel{\text{def}}{=} \{x' \in X_{\leq n} \mid x' \leq x\}$ as the set of *n-pieces* of x, which is lifted to sets $S \subseteq X$ of elements by $\downarrow_n S \stackrel{\text{def}}{=} \bigcup_{x \in S} \downarrow_n x$.

Then, by definition of $\equiv_n$, $x \equiv_n x'$ if and only if $\downarrow_n x = \downarrow_n x'$, and therefore every element x in an n-piecewise equivalence class $C \in X/\equiv_n$ has $\downarrow_n x = \downarrow_n C$. Since X is a combinatorial class, $X_{\leq n}$ is finite and this entails in particular that there are *finitely many* different n-piecewise equivalence classes in X for every fixed n:

▶ **Fact 24.** *For every fixed n, $\equiv_n$ has finite index.*

Characterisation by Principal Filters. We are now equipped to prove Fact 3:

▶ **Fact 3.** *S is an n-PTL over $(X, \leq)$ if and only if S is a finite Boolean combination of* principal filters $\uparrow x \stackrel{\text{def}}{=} \{x' \in X \mid x \leq x'\}$ *where $x \in X_{\leq n}$.*

Proof. By the above observations, if C is an equivalence class in $X/\equiv_n$, then $\downarrow_n C \subseteq X_{\leq n}$ is a finite set, and we get an equivalent formulation for C as:

$$C = \bigcap_{x \in \downarrow_n C} \uparrow x \quad \cap \quad \bigcap_{x' \in X_{\leq n} \setminus \downarrow_n C} (X \setminus \uparrow x') \,. \tag{2}$$

Then, if S is a union of n-piecewise equivalence classes, it is necessarily a finite union by Fact 24, and can be expressed as a finite union of Boolean combinations of principal filters as in (2).

Conversely, observe that if $z \in X_{\leq n}$, $x \in \uparrow z$, and $x \equiv_n y$, then by definition $y \in \uparrow z$. Thus a principal filter $\uparrow z$ with $z \in X_{\leq n}$ is necessarily an n-PTL. We conclude by noting that being a union of equivalence classes is preserved under Boolean operations, and thus n-PTLs are closed under Boolean combinations. ◀

Separability. PTL separability has a straightforward characterisation in terms of indistinguishable witnesses:

▶ **Lemma 11.** *L and L' are* not *PTL separable over a qo $(X, \leq)$ if and only if there exist two sequences of elements $(x_n)_{n \in \mathbb{N}}$ in L and $(x'_n)_{n \in \mathbb{N}}$ in L' such that for every n, $x_n \equiv_n x'_n$.*

Proof. It suffices to show that L and L' are not n-PTL separable if and only if there exist $x_n \in L$ and $x'_n \in L'$ such that $x_n \equiv_n x'_n$. Since L and L' are PTL separable if and only if there exists n such that they are n-PTL separable, the statement will follow.

Assume S is an n-PTL such that $L \subseteq S$ and $L' \cap S = \emptyset$, but that there exist $x \in L$ and $x' \in L'$ with $x \equiv_n x'$. Then $x \in S$ and therefore $x' \in S$, contradicting $S \cap L' = \emptyset$.

Conversely, if for all $x \in L$ and $x' \in L'$, $x \not\equiv_n x'$, then we can define $S \stackrel{\text{def}}{=} \bigcup_{x \in L} [x]_n$ as the union of the n-piecewise equivalence classes of the elements of L. This is an n-PTL, and furthermore $S \cap L' = \emptyset$ as desired. ◀

A.2 Finite Model Theory

Assume $\leq$ is chosen as the induced substructure ordering $\sqsubseteq$ over a class $\mathcal{K}$ of finite structures.

WQOs and Extension Preservation. In the case of a class of finite structures $\mathcal{K}$, the embedding $\sqsubseteq$ ordering is always well-founded, hence $(\mathcal{K}, \sqsubseteq)$ is a wqo if and only if it is FAC.

One might wonder how contrived our focus to wqos is. We argue that the classes of structures $\mathcal{K}$ such that $(\mathcal{K}, \sqsubseteq)$ is wqo are *well-behaved*, in a formal sense defined by Atserias, Dawar, and Grohe [5]. A first-order formula φ is *preserved under extensions in* $\mathcal{K}$ (with an implicit signature σ) if and only if its language is upwards-closed for $\sqsubseteq$, i.e. for every $\mathfrak{M}, \mathfrak{M}' \in \mathcal{K}$, if $\mathfrak{M} \sqsubseteq \mathfrak{M}'$ and $\mathfrak{M} \models \varphi$, then $\mathfrak{M}' \models \varphi$. Two formulae are *equivalent* (in $\mathcal{K}$) if and only if they have the same models in $\mathcal{K}$. Let us say that $\mathcal{K}$ has the *extension preservation property* if a first-order formula φ is preserved under extensions in $\mathcal{K}$ if and only if φ is equivalent in $\mathcal{K}$ to an existential first-order formula. The Łoś-Tarski Theorem states that for any signature σ, the class of all—finite and infinite—structures over σ has this property; it is known to fail on the class of all finite structures [36], but to hold on several other classes [5, 15, 22].

The interest here is that the extension preservation property holds when $(\mathcal{K}, \sqsubseteq)$ is a wqo; this does not seem to be so widely known, but is proven e.g. by Sankaran et al. [33, Proposition 4.1]:

▶ **Fact 25.** *If $(\mathcal{K}, \sqsubseteq)$ is a wqo then $\mathcal{K}$ has the extension preservation property.*

Proof. Consider a first-order sentence φ such that its language $L(\varphi) \subseteq \mathcal{K}$ is upwards-closed, and consider the set $\min_{\sqsubseteq} L(\varphi)$ of minimal models of φ: then $L(\varphi) = \bigcup_{\mathfrak{M} \in \min_{\sqsubseteq} L(\varphi)} {\uparrow}\mathfrak{M}$.

Note that, for all finite structures $\mathfrak{M}$ and $\mathfrak{M}'$, if $\mathfrak{M} \sqsubseteq \mathfrak{M}'$ and $\mathfrak{M}' \sqsubseteq \mathfrak{M}$, then $\mathfrak{M}$ and $\mathfrak{M}'$ are isomorphic. Hence $\min_{\sqsubseteq} L(\varphi)$ is an antichain up to isomorphism. Since $(\mathcal{K}, \sqsubseteq)$ is a wqo, there are therefore finitely many minimal models in $\min_{\sqsubseteq} L(\varphi)$ up to isomorphism, hence $L(\varphi)$ is a finite union of principal filters. We conclude using the fact that each one of these finitely many principal filters can be defined by an existential sentence (as shown e.g. in the proof of Fact 8). ◀

The converse of Fact 25 does not hold in general [33, Proposition 4.3]; intuitively, the extension preservation property only cares about first-order definable sets of structures, whereas wqos require the whole of $\mathcal{K}$ to be well-behaved.

Characterisation by Existential First-Order Sentences. When $(\mathcal{K}, \sqsubseteq)$ is a wqo, PTLs are characterised by $\mathcal{B}\Sigma_1$ sentences:

▶ **Fact 8.** *If a set S is an n-PTL over $(\mathcal{K}, \sqsubseteq)$, then it is definable by a sentence in $\mathcal{B}\Sigma_1(\sigma)$ with at most n variables. Conversely, assuming additionally that $(\mathcal{K}, \sqsubseteq)$ is a wqo, if S is definable by a sentence in $\mathcal{B}\Sigma_1(\sigma)$, then it is a PTL over $(\mathcal{K}, \sqsubseteq)$.*

Proof. The proof relies on Fact 3 for both statements.

Any principal filter ${\uparrow}\mathfrak{M}$ can be defined by an existential sentence φ with $|\mathfrak{M}|$ variables. This uses one existentially quantified variable per element of $\mathfrak{M}$ and a conjunction of atoms and negated atoms describing which relations hold between all these elements.

Conversely, consider an existential sentence φ from Σ_1 and assume $(\mathcal{K}, \sqsubseteq)$ is a wqo. The set of models of φ is upwards-closed, and by the finite antichain condition it has a finite set of $\sqsubseteq$-minimal models up to equivalence, hence can be written as a finite union of principal filters. ◀

The requirement for $(\mathcal{K}, \sqsubseteq)$ to be a wqo in the second part of Fact 8 is necessary. For instance, over the signature $\sigma_h = ((P_f)_{f\in\mathcal{F}}, <_h, \sqcap)$ investigated in Appendix B.4, $(T(\mathcal{F}), \sqsubseteq_h)$ is in general not a wqo; for instance $A \stackrel{\text{def}}{=} \{f(t_1, t_2) \mid t_1, t_2 \in T(\{g, a\})\}$ is an infinite antichain over $\mathcal{F} \stackrel{\text{def}}{=} \{f, g, a\}$ where f, g have arity two and a arity zero. Observe that the set of models of $\exists x.P_f(x)$ is $\uparrow A$, and therefore cannot be expressed as a finite Boolean combination of principal filters.

B Relational Signatures on Trees

While we focus in the main text on the signature $\sigma_T = ((P_f)_{f\in\mathcal{F}}, <, <_{\text{dfs}}, \sqcap)$, which has the advantage of matching the well-studied homeomorphic tree embedding relation, other choices are possible. Place [30] and Bojańczyk et al. [7] consider in fact several other signatures in their articles on PTL definability, and we provide here a quick comparison of the embedding relations they define.

B.1 Homeomorphic Tree Embeddings

A first remark is that the ancestor relation can be defined by a quantifier-free formula using the least common ancestor relation as $x < y \equiv x = x \sqcap y \wedge x \neq y$, hence any embedding that preserves $\sqcap$ also preserves $<$ and we could define σ_T equivalently as $((P_f)_{f\in\mathcal{F}}, <_{\text{dfs}}, \sqcap)$.

The following lemma is a folklore result, which we prove for the sake of completeness:

▶ **Lemma 26.** *Let $t = f(t_1, \dots, t_r)$ and $t' = g(t'_1, \dots, t'_s)$ be two trees in $T(\mathcal{F})$. Then $t \sqsubseteq_T t'$ if and only if $t \sqsubseteq_T t'_i$ for some $1 \leq i \leq s$, or $f = g$ and $t_i \sqsubseteq_T t'_i$ for every $1 \leq i \leq s$.*

Proof of 'only if'. It suffices to check that $t \sqsubseteq_T t'$ in both cases. If t embeds into t'_i, then the embedding can be expanded into an embedding from t to t'. If for every $1 \leq i \leq r$, t_i embeds into t'_i by some e_i and $f = g$, then an embedding of t into t' is obtained by mapping the root of t to that of t' and using the embeddings e_i for the other nodes. In both cases, the relations of σ_T are preserved. ◀

Proof of 'if'. Assume t embeds into t' through some mapping e. If e maps the root of t to some node of t'_i for some $1 \leq i \leq s$, then it also has to map all the other nodes of t inside t'_i in order to preserve the ancestor relation, i.e. e can be seen as embedding t into t'_i.

Otherwise, e maps the root of t to the root of t', and therefore they have the same label $f = g$ in order to preserve P_f and P_g (and thus $r = s$). Consider now the images of the roots x_i of each t_i by e; these must be different from the root of t' since e is injective. Assume for the sake of contradiction that e maps two different roots x_i and x_j for $i \neq j$ into nodes inside a single t'_k for some $1 \leq k \leq r$: then the least common ancestor of $e(x_i)$ and $e(x_j)$ would be a node inside t'_k, and would be different from the image of their least common ancestor, which is the root of t'. Hence e must induce a permutation π of $\{1, \dots, r\}$ and maps x_i to a node in $t'_{\pi(i)}$. Furthermore, any inner node of t_i must be mapped into an inner node of $t'_{\pi(i)}$, as otherwise the ancestor relation $<$ would not be preserved. Hence the restriction of e to t_i is an embedding from t_i into $t'_{\pi(i)}$. Finally, if π were not the identity, then there would exist $1 \leq i < j \leq r$ such that $\pi(j) < \pi(i)$ and thus $x_i <_{\text{dfs}} x_j$ but $e(x_{\pi(j)}) <_{\text{dfs}} e(x_{\pi(i)})$ and the document order relation $<_{\text{dfs}}$ would not be preserved. Thus e maps each t_i into t'_i. ◀

B.2 Ordered Tree Minors

The main signature considered by Bojańczyk et al. [7] is $\sigma_m \stackrel{\text{def}}{=} ((P_f)_{f\in\mathcal{F}}, <, <_{\text{dfs}})$. The corresponding embedding relation is again a known one: this is a labelled version of the

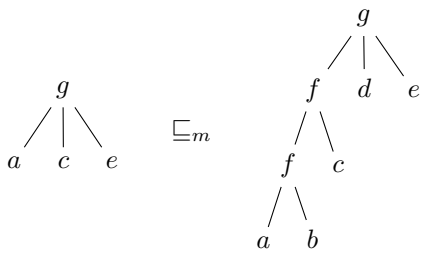


Figure 3 Two trees related by the tree minor ordering.

(ordered) *tree minor* ordering $\sqsubseteq_m$, studied for instance by Gupta [19], and can be defined through edge contractions [7]. Tree minors well-quasi-order the set of finite trees ($\sqsubseteq_T$ refines $\sqsubseteq_m$, and also in the case of unranked trees), and our techniques could apply, provided one identified effective ideal representations for $\sqsubseteq_m$ and proved the decidability of the adherence membership problem.

Figure 3 illustrates this relation on two trees, and shows that tree minors do not preserve least common ancestors (the two trees are not related by $\sqsubseteq_T$).

Note that due to our restriction to ranked trees, multiples edge contractions might be necessary before we find again a properly labelled minor—in the example of Figure 3, we had to contract the edges leading to the two 'f' nodes and to 'b' and 'd' (since the latter are constants, contractions delete the corresponding subtrees), but none of the intermediate trees we obtained was respecting the symbols' arities. Tree minors are thus more natural in the setting of unranked trees, which incidentally is the one adopted by Bojańczyk et al. [7].

B.3 Unordered Tree Minors

The signature investigated by Place [30] is $\sigma_{m'} \stackrel{\text{def}}{=} ((P_f)_{f\in\mathcal{F}}, <)$, i.e. it removes the $<_{\text{dfs}}$ relation from σ_m. The corresponding embedding relation $\sqsubseteq_{m'}$ coincides with *unordered tree minors*; this is a coarser well-quasi-ordering on finite trees than the tree minor relation.

B.4 Subtree Relation

Let $<_h$ denote the *horizontal ordering*: $x <_h y$ if and only if x is a sibling of y and $x <_{\text{dfs}} y$. Consider the signature $\sigma_h \stackrel{\text{def}}{=} ((P_f)_{f\in\mathcal{F}}, <_h, \sqcap)$. This signature is considered by Bojańczyk et al. [7, Section 6.4], and they briefly mention that their proof for σ_T carries over to that case. The corresponding embedding relation $\sqsubseteq_h$ over $T(\mathcal{F})$ is again a rather natural one in the case of ranked trees:

▶ **Lemma 27.** *Let $\mathcal{F}$ be a ranked alphabet without unary symbols, and t and t' be two trees in $T(\mathcal{F})$. Then $t \sqsubseteq_h t'$ if and only if t is (isomorphic to) a subtree of t'.*

Proof. If t is a subtree of t', then there is an obvious embedding of t into t' that preserves all the relations. Let us prove conversely that $t \sqsubseteq_h t'$ implies that t is a subtree of t' by induction on the height of t'. Let $t = f(t_1, \ldots, t_r)$ and $t' = g(t'_1, \ldots, t'_s)$.

Let us show by induction on the height of t that, if the embedding e that witnesses $t \sqsubseteq_h t'$ maps the root of t to that of t', then $t = t'$. First, $f = g$ in order to preserve the predicates P_f and P_g and thus $r = s$. Second, if $r = s = 0$, then we are done. Otherwise, $r = s \geq 2$ since there are no unary symbols, and letting $x_1, \ldots, x_r$ and $x'_1, \ldots, x'_r$ denote respectively the roots of $t_1, \ldots, t_r$ and $t'_1, \ldots, t'_r$, e must map every x_i to a node inside a different t'_j in order to preserve the least common ancestor relation $\sqcap$. Third, e must also preserve the

horizontal order $<_h$, and therefore the images of the various x_i must be siblings: thus e maps x_i to x'_i. Fourth, e must map all the nodes in t_i to nodes inside t'_i in order to preserve the ancestor relation. Hence $t_i \sqsubseteq_h t'_i$ for every $1 \leq i \leq r$ and the restriction of e to each t_i maps the root of t_i to that of t'_i. By induction hypothesis, $t_i = t'_i$.

Now, if $t \sqsubseteq t'$ through some mapping e, let t'' be the subtree of t' rooted by the image of the root of t by e. As usual, the images of the nodes of t must all be in t'' in order to preserve the ancestor relation, hence $t \sqsubseteq_h t''$. By the previous claim, $t'' = t$. Hence t is a subtree of t'. ◀

Observe that $\sqsubseteq_h$ strictly refines $\sqsubseteq_T$; e.g. $f(a,b) \sqsubseteq_T f(g(a,a),b)$ but $f(a,b) \not\sqsubseteq_h f(g(a,a),b)$. In fact, it is *too* fine: $\sqsubseteq_h$ does not well-quasi-order $T(\mathcal{F})$:

▶ **Corollary 28.** *If $\mathcal{F}$ contains at least two symbols of arity at least two, then $(T(\mathcal{F}), \sqsubseteq_h)$ is not wqo.*

Proof. Let us assume $\{f, g, a\} \subseteq \mathcal{F}$ where f, g are binary and a nullary (f and g with other arities greater than one can be handled similarly). Consider the trees $t_n \stackrel{\text{def}}{=} f(a, (g(a, \Box))^n \cdot a)$: those are 'comb-shaped' trees with n occurrences of g under a single occurrence of f; e.g. $t_1 = f(a, g(a, a))$ and $t_2 = f(a, g(a, g(a, a)))$. By Lemma 27 above, they form an infinite antichain $\{t_n \mid n \geq 0\}$. ◀

Hence our techniques do not apply directly in the case of $(T(\mathcal{F}), \sqsubseteq_h)$. In fact, the infinite antichain in the previous proof can be exploited to show that Lemma 12 *requires* $(X, \leq)$ to be wqo in order to apply:

▶ **Proposition 29.** *There exist L and L' not PTL separable over $(T(\mathcal{F}), \sqsubseteq_h)$ where all the directed $\Delta \subseteq L$ and $\Delta' \subseteq L'$ are such that $\downarrow\Delta \neq \downarrow\Delta'$.*

Proof. We split the infinite antichain $\{t_n \mid n \geq 0\}$ used in the proof of Corollary 28 into its odd and even parts: $L \stackrel{\text{def}}{=} \{t_{2i} \mid i \geq 0\}$ and $L' \stackrel{\text{def}}{=} \{t_{2j+1} \mid j \geq 0\}$.

Any directed $\Delta \subseteq L$ is then a singleton $\{t_{2i}\}$ for some $i \geq 0$, and similarly $\Delta' \subseteq L'$ is directed if and only if Δ' is a singleton $\{t_{2j+1}\}$ for some $j \geq 0$. Hence, for all directed $\Delta \subseteq L$ and $\Delta' \subseteq L'$, $\downarrow\Delta = \downarrow t_{2i}$ and $\downarrow\Delta' = \downarrow t_{2j+1}$ are two different, and even incomparable principal ideals.

If $(T(\mathcal{F}), \sqsubseteq_h)$ were a wqo, then Lemma 12 would thus yield that L and L' are PTL separable. However, $(T(\mathcal{F}), \sqsubseteq_h)$ is not a wqo, and on the contrary we observe that for every $n \geq 0$,

$$\downarrow_n t_{2n} = \downarrow_n t_{2n+1} = \{(g(a, \Box)^m \cdot a \mid 2m + 1 \leq n\} \,, \tag{3}$$

and thus $t_{2n} \equiv_n t_{2n+1}$. Therefore, by Lemma 11, L and L' are not PTL separable. ◀

Note that the previous proof could also be carried out over $(\Sigma^*, \leq_{\text{suf}})$ the set of finite words ordered by *suffix* (over $\Sigma \stackrel{\text{def}}{=} \{f, g\}$): $u \leq_{\text{suf}} v$ if and only if there exists $w \in \Sigma^*$ such that $v = wu$. Czerwiński et al. [12] show that PTL separability for regular languages over $(\Sigma^*, \leq_{\text{suf}})$ is decidable, and it would be interesting to check whether PTL separability for $(T(\mathcal{F}), \sqsubseteq_h)$ is decidable.

B.5 'Strong' Tree Minors

The case of the signature $\sigma_M \stackrel{\text{def}}{=} ((P_f)_{f \in \mathcal{F}}, <, <_h)$ is also considered by Bojańczyk et al. [7], and they provide a decidable characterisation of the corresponding piecewise testable languages [7, Theorem 6.4].

The embedding relation $\sqsubseteq_M$ induced by this signature refines the minor ordering (e.g. in Figure 3, none of the nodes 'a', 'c', and 'e' are siblings in the tree on the right, thus the two trees are not in relation by $\sqsubseteq_M$). This ordering is incomparable with the homeomorphic tree ordering: for instance, $f(a,b) \sqsubseteq_T f(g(a),g(b))$ but $f(a,b) \not\sqsubseteq_M f(g(a),g(b))$, and conversely $f(a,b) \not\sqsubseteq_T f(g(a,b),c)$ but $f(a,b) \sqsubseteq_M f(g(a,b),c)$.

Unlike the previous cases, this relation seems a bit exotic, and as far as we know it has not been studied in the wqo literature: write $t \sqsubseteq'_M t'$ if there is an embedding for σ_M that maps the root of t to that of t'. Then one can prove, following the same lines as in Lemma 27:

▶ **Lemma 30.** *Let $t = f(t_1,\dots,t_r)$ and t' be two trees in $T(\mathcal{F})$.*

1. *$t \sqsubseteq'_M t'$ if and only if*
   - **a.** *t' has root label f and*
   - **b.** *there exists a subtree $g(t'_1,\dots,t'_s)$ of t' for some $s \geq r$ and an strictly monotone mapping φ from $\{1,\dots,r\}$ to $\{1,\dots,s\}$ such that $t_i \sqsubseteq'_M t'_{\varphi(i)}$ for every $1 \leq i \leq r$.*
2. *$t \sqsubseteq_M t'$ if and only if there exists a subtree t'' of t' such that $t \sqsubseteq'_M t''$.*

Proof. Point **2** is a direct consequence of point **1**. Regarding the proof of point **1**, we can check that conditions **a** and **b** indeed entail $t \sqsubseteq'_M t'$ by suitably assembling an embedding that preserves the relations in σ_M. For the converse direction, assume that $t \sqsubseteq'_M t'$ through a mapping e that sends its root to the root of t'. Then the root of t' must be labelled by f in order to preserve P_f, thus it satisfies condition **a**. Furthermore, let $x_1,\dots,x_r$ denote the roots of $t_1,\dots,t_r$. We know that the images $e(x_1) <_h \cdots <_h e(x_r)$ must be siblings, hence there exists a subtree $g(t'_1,\dots,t'_s)$ of t' such that each $e(x_i)$ roots some t'_j for $1 \leq j \leq s$, and we let in that case $\varphi(i) \stackrel{\text{def}}{=} j$. Then φ is indeed a strictly increasing mapping and $r \leq s$ as desired. Finally, all the nodes in t_i must be mapped to nodes in t'_i in order to preserve the ancestor relation $<$, hence the restriction of e to t_i embeds it into t'_i and sends its root to the root of t'_i, i.e. $t_i \sqsubseteq'_M t'_i$, and condition **b** is satisfied. ◀

The previous infinite antichain $\{f(g^n(a)) \mid n \geq 0\}$ for $\sqsubseteq_h$ becomes an infinite chain for $\sqsubseteq_M$, and in fact $(T(\mathcal{F}),\sqsubseteq_M)$ is a wqo (essentially the same proof arguments could be used for unranked trees):

▶ **Theorem 31.** *$(T(\mathcal{F}),\sqsubseteq_M)$ is a wqo.*

Proof. We proceed using a 'minimal infinite bad sequence' argument as first employed by Nash-Williams [28] in his proof of Kruskal's Tree Theorem, and prove the stronger statement that $(T(\mathcal{F}),\sqsubseteq'_M)$ is a wqo. Note that the subtree ordering is well-founded, thus any non-empty set of trees has at least one minimal element.

Assume for the sake of contradiction that $(T(\mathcal{F}),\sqsubseteq'_M)$ is not a wqo. Then there exists a *minimal* infinite bad sequence $s = t_0, t_1, \dots$ for $\sqsubseteq'_M$, i.e. where t_0 was chosen among the minimal trees (for the subtree ordering) that could start an infinite bad sequence, and inductively for every i, t_{i+1} was chosen among the minimal trees (for the subtree ordering) that could yield an infinite bad sequence starting with $t_0,\dots,t_i$.

Write each t_i in the sequence s as $f_i(t'_{i,1},\dots,t'_{i,r_i})$ for some $f_i \in \mathcal{F}$ and $t'_{i_j} \in T(\mathcal{F})$. Then we claim that the set $S \stackrel{\text{def}}{=} \{t'_{i,j} \mid i \in \mathbb{N}, 1 \leq j \leq r_i\}$ together with $\sqsubseteq'_M$ forms a wqo $(S,\sqsubseteq'_M)$.

Indeed, assuming the contrary there would exist an infinite bad sequence $s' = t_{i_0,j_0}, t_{i_1,j_1}, \dots$ of elements from S. But then the infinite sequence $t_0,\dots,t_{i_0-1},s'$ is bad. However, t_{i_0,j_0} is strictly smaller than t_{i_0}, contradicting the minimality of s.

Since $(S,\sqsubseteq'_M)$ is a wqo, by Dickson's Lemma the set $S' \stackrel{\text{def}}{=} \{(f_i, t_{i,1}\cdots t_{i,r_i})\} \subseteq \bigcup_{r \leq |\mathcal{F}|} \mathcal{F}_r \times S^r$ is well-quasi-ordered by $(f, t_1\cdots t_r) \leq_{S'} (g, t'_1\cdots t'_s)$ defined by $f = g$ and $t_j \sqsubseteq'_M t'_j$ for every $1 \leq j \leq r$.

With the infinite bad sequence s, we associate the infinite sequence $(f_0, t'_{0,1} \cdots t'_{0,r_0}), (f_1, t'_{1,1}, \ldots, t'_{1,r_1}), \ldots$ of elements in S'. Since $(S', \leq_{S'})$ is wqo, there exists $i < k$ such that $(f_i, t'_{i,1} \cdots t'_{i,r_i}) \leq_{S'} (f_k, t'_{k,1} \cdots t'_{k,r_k})$. Then $t_i = f_i(t'_{i,1}, \ldots, t'_{i,r_i})$ and $t_k = f_k(t'_{k,1}, \ldots, t'_{k,r_k})$ are such that $f_i = f_k$ and $t'_{i,j} \sqsubseteq'_M t'_{k,j}$ for every $1 \leq j \leq r_i$. By Lemma 30 this proves $t_i \sqsubseteq'_M t_k$, contradicting the fact that s was assumed bad. ◀

Theorem 31 shows that our techniques might also apply to the case of σ_M, provided one worked out an effective ideal representation for $(T(\mathcal{F}), \sqsubseteq_M)$ and proved the decidability of the adherence membership problem.

C Ideals and Tree Products

C.1 Effective Ideal Representations

Goubault-Larrecq et al. [18] study ideal representations for various wqos $(X, \leq)$, and focus in particular on their algorithmic properties. In the following, we shall require the following procedures on representations for elements of X and for ideals of $\mathrm{Idl}(X)$, and say that $(X, \leq)$ has *effective* ideal representations when these procedures exist:

ideals of X (XI) compute the ideal decomposition of X;

ideal containment (IC) decide whether $I \subseteq J$ given two ideals $I, J \in \mathrm{Idl}(X)$;

ideal intersection (II) compute the ideal decomposition of $I \cap J$ given two ideals $I, J \in \mathrm{Idl}(X)$;

filter complementation (CU') compute the ideal decomposition of $X \setminus {\uparrow} x$ given an element $x \in X$.

These procedures can be employed to write procedures for the following problems:

ideal membership (IM) decide whether $x \in I$ given an ideal $I \in \mathrm{Idl}(X)$. Compute $X \setminus {\uparrow} x = J_1 \cup \cdots \cup J_n$ using **CU'**; by irreducibility of I it then suffices to check that $I \not\subseteq J_i$ for every $1 \leq i \leq n$ using **IC**.

ideal complementation (CD) compute a finite set B of elements x such that $\bigcup_{x \in B} {\uparrow} x = X \setminus I$ given an ideal $I \in \mathrm{Idl}(X)$. This can be done using oracles for **II** and **CU'** thanks to the *Generalised Valk-Jantzen Lemma* [18].

C.2 Adherence Membership

▶ **Proposition 16.** *Let $(X, \leq)$ be a wqo with effective ideal representations and $\mathcal{C} \subseteq 2^X$ be PTL-effective over $(X, \leq)$. Then the adherence membership problem and the ideal decomposition problem are Turing-equivalent.*

Proof. The first reduction from the adherence membership problem to the ideal decomposition problem is an application of Lemma 14. Using **CD**, I has a representation as a PTL, hence $I \cap L$ is in $\mathcal{C}$ and can be constructed. An ideal decomposition $I_1 \cup \cdots \cup I_k$ of ${\downarrow}(I \cap L)$ is then computed using the oracle, and since I is irreducible the test $I \subseteq {\downarrow}(I \cap L)$ is handled by k calls to **IC**.

The converse reduction from the ideal decomposition problem to the adherence membership problem uses an abstraction refinement algorithm already described in [27]. Given L, the algorithm computes a descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ of downwards-closed subsets $D_i \supseteq {\downarrow} L$, and halts at step k if $D_k = {\downarrow} L$; termination is guaranteed by the descending chain property of the wqo $(X, \leq)$. Each D_i is represented by its unique ideal decomposition, starting with $D_0 \stackrel{\text{def}}{=} X$ using **XI**. At every step $k = 0, 1, \ldots$, if there is an ideal I in the

decomposition of D_k such that $I \not\subseteq \downarrow L$, then we extract some $x \in I \setminus \downarrow L$ by enumerating the elements of I using **IM** until $\uparrow x \cap L = \emptyset$, which is decidable since $\uparrow x$ is a PTL, and set $D_{k+1} \stackrel{\text{def}}{=} D_k \setminus \uparrow x$ using **II** and **CU'**. The trick is that, in that situation, [27, Lemma IV.7] shows that $I \not\subseteq \downarrow L$ if and only if $I \notin \mathrm{Adh}(L)$, which can be decided thanks to our oracle. ◀

▶ **Proposition 17.** *There is a many-one reduction from the adherence membership problem to the PTL separability problem for regular tree languages over* $(T(\mathcal{F}), \sqsubseteq_T)$.

Proof. Let us first show that, if $(X, \leq)$ is a wqo with effective ideal representations and $\mathcal{C}$ is a PTL-effective class over $(X, \leq)$, then given as input two representations for $I \in \mathrm{Idl}(X)$ and $S \in \mathcal{C}$, $L \stackrel{\text{def}}{=} I$ and $L' \stackrel{\text{def}}{=} X \setminus \downarrow(I \cap S)$ are PTL-separable if and only if $I \in \mathrm{Adh}(S)$. If $I \in \mathrm{Adh}(S)$, then by Lemma 14 $I \subseteq \downarrow(I \cap S)$, hence L and L' are disjoint and I is a separator. The latter is a PTL since it is the complement of an upwards-closed set and $(X, \leq)$ is a wqo. Conversely, if $I \notin \mathrm{Adh}(S)$, then still by Lemma 14 there exists $x \in I \setminus \downarrow(I \cap S)$, hence L and L' are not disjoint and no separator of any sort can exist.

We conclude in the case where $\mathcal{C}$ is the class of regular tree languages over $(T(\mathcal{F}), \sqsubseteq_T)$, by observing that I, $I \cap S$, $\downarrow(I \cap S)$, and $T(\mathcal{F}) \setminus \downarrow(I \cap S)$ are all computable regular languages. ◀

C.3 Simple Tree Regular Expressions

▶ **Lemma 32.** *Every STRE defines a downwards-closed language of* $(T(\mathcal{F}), \sqsubseteq_T)$.

Proof. We use induction on the size of the STRE, and the main point consists in checking that if $t \in C^*.S$ then any smaller tree s (w.r.t. $\sqsubseteq_T$) is also in $C^*.S$. By induction hypothesis, S is downwards-closed. We use a secondary induction on the k used in the definition of the language of $C^*.S$ as $c_1[\cdots[c_k[S \cup \operatorname{supp} C]]\cdots]$. If $k = 0$, then $t \in S$, hence $s \in S$ as well. Otherwise, $k \geq 1$, one of the summands in c_1 is of the form $f(S_{\Box 1}, \cdots, S_{\Box n})$, and $t = f(t_1, \cdots, t_n)$. If s is smaller than some t_i, then either $S_{\Box i} = \Box$ and $t_i \in c_2[\cdots[c_k[S \cup \operatorname{supp} C]]\cdots]$, which allows us to conclude by the inner induction hypothesis; or $S_{\Box i}$ is of the form P, and is downwards-closed by the outer induction hypothesis. If instead $s = f(s_1, \cdots, s_n)$ where s_i is smaller than t_i for each i, we conclude similarly that $s_i \in C^*.S$ for each i, hence $s \in C^*.S$ as well, by applying c_1 again. ◀

C.4 Tree Products

Let us introduce some additional notation. We say that a pattern $A = f(S_{\Box 1}, \ldots, S_{\Box n})$ is $\Box$-*generated* if and only if at least one $S_{\Box_i}$ is; it is *empty* if and only if some $S_{\Box i}$ is different from $\Box$ and has an empty language. Writing C as $A_1 + \cdots + A_m$, we say that C is $\Box$-generated if and only if every non-empty A_i is $\Box$-generated, and is empty if and only if every A_i is empty.

By inspection of the rules defining $\to_1$, we see:

▶ **Lemma 33.** *The tree products are exactly the STREs of the form:*

- $f^?(P_1, \cdots, P_n)$ *where* $P_1, \ldots, P_n$ *are tree products;*
- *or* $C^*.(P_1 + \cdots + P_n)$ *where* $n \in \mathbb{N}$, $P_1, \ldots, P_n$ *are tree products,* $C = \sum_{i=1}^{m} f_i(P_{\Box i1}, \cdots, P_{\Box in_i})$, *each pattern* $P_{\Box ij}$ *is either a tree product or the placeholder* $\Box$, C *is* $\Box$*-generated, and one of the following conditions is satisfied: (a)* C *is not* $\Box$*-linear and* $n \geq 1$, *or (b)* C *is not* $\Box$*-linear,* $n = 0$, *and* $P_{\Box ij} \neq \Box$ *for some* i, j, *or (c)* C *is* $\Box$*-linear and* $n = 1$.

▶ **Lemma 34.** *If* $S_1, \ldots, S_n$ *are directed STREs, then so is* $f^?(S_1, \cdots, S_n)$.

Proof. Non-emptiness is clear, since $S_1, \ldots, S_n$ are non-empty. Let t, t' any two trees in $f^?(S_1, \cdots, S_n)$. If $t = f(t_1, \cdots, t_n)$ and $t' = f(t'_1, \cdots, t'_n)$ with $t_i, t'_i \in S_i$ for every i, then we can find $t''_i \geq t_i, t'_i$ in S_i, and then $f(t''_1, \cdots, t''_n) \geq t, t'$ is in $f^?(S_1, \cdots, S_n)$.

If t is in some S_j and $t' = f(t'_1, \cdots, t'_n)$ with $t'_i \in S_i$ for every i, then build the tree $s = f(s_1, \cdots, s_{j-1}, t, s_{j+1}, \cdots, s_n)$, where s_i is an arbitrary tree from the non-empty set S_i, $i \neq j$. Clearly, s is in $f^?(S_1, \cdots, S_n)$, so, by reduction to the previous case, there is a tree $t'' \in f^?(S_1, \cdots, S_n)$ such that $s, t' \sqsubseteq_T t''$. Since $t \sqsubseteq_T s$, we obtain that $t, t' \sqsubseteq_T t''$.

Similarly if t' is in some S_k, we build a new tree $s' = f(s'_1, \cdots, s'_{k-1}, t', s'_{k+1}, \cdots, s'_n)$ and conclude by a similar argument that there is a tree $t'' \in f^?(S_1, \cdots, S_n)$ such that $t' \sqsubseteq_T s' \sqsubseteq_T t''$ and $t \sqsubseteq_T t''$. ◀

The case of STREs of the form $C^*.S$ is more complex. The three cases (a), (b) and (c) match those of Lemma 33, second item.

▶ **Lemma 35.** *Let $C = A_1 + \cdots + A_m$ be a sum of patterns $A_i = f_i(S_{\Box i1}, \cdots, S_{\Box in_i})$, where each $S_{\Box ij}$ that is different from $\Box$ has a non-empty language. If any of the following conditions is satisfied, then $C^*.S$ is directed:*

- (a) *C is not $\Box$-linear, and S has a non-empty language, or*
- (b) *C is not $\Box$-linear, and some $S_{\Box ij}$ is different from $\Box$, or*
- (c) *C is $\Box$-linear and $\Box$-generated, and S is irreducible.*

Proof. The fact that $C^*.S$ is non-empty is easy exercise: in case (a), $C^*.S$ contains S; in case (b), it contains $S_{\Box ij}$; in case (c), it contains S and every irreducible subset is necessarily non-empty. Let t, t' be any two trees in $C^*.S$.

In case (a), some non-empty A_i is of the form $f(S_{\Box 1}, \cdots, S_{\Box n})$, where $\Box$ occurs at least twice, say at positions j and j', $j \neq j'$. For every k, $1 \leq k \leq n$, define a tree t_k as follows: if $S_{\Box k} = \Box$ and $k \neq j'$ (including the case $k = j$), let $t_k = t$, if $k = j'$ then let $t_k = t'$, and if $S_{\Box k} \neq \Box$ then pick any tree for t_k from the language of $S_{\Box k}$, which is non-empty by assumption. We check that $f(t_1, \cdots, t_n)$ is in $C^*.S$, and $t = t_j$, $t' = t_{j'}$ both embed into $f(t_1, \cdots, t_n)$.

Case (b) reduces to (a), since if $S_{\Box ij} \neq \Box$, then $C^*.S$ defines the same language as $C^*.(S + S_{\Box ij})$.

In case (c), every non-empty A_i is of the form $f(\cdots, \Box, \cdots)$ with a unique occurrence of $\Box$. In that case, the language of $C^*.S$ can be described more simply: it consists of those trees of the form $c_1[c_2[\cdots c_k[s]\cdots]]$, where $k \in \mathbb{N}$, each c_k is a context in the language of C, with just one occurrence of $\Box$ each, and s is a tree in $S \cup \operatorname{supp} C$. For short, say that a context c is *in* C^* if and only if it is of the form $c_1[c_2[\cdots c_k[\Box]\cdots]]$, where $k \in \mathbb{N}$ and each c_k is in C. Such contexts have exactly one occurrence of $\Box$. Hence the language of $C^*.S$ consists of those trees $c[s]$ where c is in C^* and $s \in S \cup \operatorname{supp} C$. Given any two such trees $t = c[s]$ and $t' = c'[s']$, we find a tree $t'' \in C^*.S$ in which both t and t' embed, as follows.

If both s and s' are in S, then by directedness there is an $s'' \in S$ such that $s, s' \sqsubseteq_T s''$: we define t'' as $c[c'[s'']]$.

If $s \in S$ and $s' \in \operatorname{supp} C$, then there is an $A_i = f_i(S_{\Box i1}, \cdots, S_{\Box in_i})$ and a position j', $1 \leq j' \leq n_i$, such that $S_{\Box ij'} \neq \Box$ and the language of $S_{\Box ij'}$ contains s'. The unique position j at which $S_{\Box ij} = \Box$ is different from j'. Let $u = f_i(u_1, \cdots, u_n)$ where: $u_j = s$, $u_{j'} = s'$, for every position $k \neq j, j'$ u_k is an arbitrary tree from $S_{\Box ik}$. By construction, $u \in C^*.S$, hence so is $t'' = c[c'[u]]$. Additionally, since s and s' both embed in u, t and t' both embed in t''.

The same argument applies when $s \in \operatorname{supp} C$ and $s' \in S$. Finally, we consider the case where s, s' are both in $\operatorname{supp} C$. Then $s \in \operatorname{supp} A_i$ for some i, say $A_i = f_i(S_{\Box i1}, \cdots, S_{\Box in_i})$, $S_{\Box ij'} \neq \Box$ and s in in the language of $S_{\Box ij'}$. Since S is irreducible, it is non-empty, hence we

can pick a tree s_0 from it. Let $u_{j'} = s$, $u_j = s_0$ where j is the unique position where $S_{\Box ij} = \Box$, and pick u_k from the non-empty language $S_{\Box ik}$ for every $k \neq j, j'$; define $u = f_i(u_1, \cdots, u_{n_i})$, a tree from $C^*.S$ in which s embeds. Similarly, since $s' \in \operatorname{supp} C$, s' is in the support of some $A_{i'}$, say $A_{i'} = f_{i'}(S_{\Box i'1}, \cdots, S_{\Box i'n_{i'}})$, $S_{\Box i'j'''} \neq \Box$ and s' is in the language of $S_{\Box i'j'''}$. Let $v_{j'''} = s'$ (instead of s in our previous step), $v_{j''} = u$ (instead of s_0) where j'' is the unique position where $S_{\Box i'j''} = \Box$, and pick v_k from the non-empty language $S_{\Box i'k}$ for every $k \neq j'', j'''$. The tree $v = f_{i'}(v_1, \cdots, v_{n_{i'}})$ is again in $C^*.S$, and now both s and s' embed into it. Finally, we define t'' as $c[c'[v]]$. ◀

▶ **Theorem 23.** *The ideals of $T(\mathcal{F})$ are exactly the languages of tree products.*

Proof. One direction is Lemma 22. In the other direction, we show that the language of every tree product P is directed by structural induction on P, using Lemma 34 or Lemma 35, depending on its shape, as given by Lemma 33. In doing the proof, one needs to observe that any $\to_1$-normal STRE $S = P_1 + \cdots + P_m$ (where the language of each P_i is an ideal by induction hypothesis) has an empty language if and only if $m = 0$; this is because ideals are never empty. ◀

▶ **Proposition 36.** *$(T(\mathcal{F}), \sqsubseteq_T)$ has effective ideal representations.*

Proof. By Theorem 23, any ideal of $(T(\mathcal{F}), \sqsubseteq_T)$ can be represented as a tree product, which is a regular tree language. Let us check that we have the required procedures for effectiveness:

(XI) $T(\mathcal{F}) = \left(\sum_{r>0} \sum_{f \in \mathcal{F}_r} f(\Box, \ldots, \Box)\right)^*.\mathcal{F}_0$;

(IC) containment of regular tree languages is decidable;

(II) the intersection of two regular tree languages is a regular tree language, and using the construction of Proposition 18 and the rewrite system of Figure 2, we can obtain its decomposition into a sum of tree products;

(CU') $\uparrow t$ is a regular tree language, thus an automaton for $T(\mathcal{F}) \setminus \uparrow t$ is constructible, and using the construction of Proposition 18 and the rewrite system of Figure 2, we can obtain its decomposition into a sum of tree products. ◀