Wreath Products of Distributive Forest Algebras

Michael Hahn
Stanford University
mhahn2@stanford.edu

Andreas Krebs
University of Tübingen
mail@krebs-net.de

Howard Straubing
Boston College
howard.straubing@bc.edu

Abstract

It is an open problem whether definability in Propositional Dynamic Logic (PDL) on forests is decidable. Based on an algebraic characterization by Bojańczyk, *et. al.,* (2012) in terms of forest algebras, Straubing (2013) described an approach to PDL based on a k-fold iterated distributive law. A proof that all languages satisfying such a k-fold iterated distributive law are in PDL would settle decidability of PDL. We solve this problem in the case $k = 2$: All languages recognized by forest algebras satisfying a 2-fold iterated distributive law are in PDL. Furthermore, we show that this class is decidable. This provides a novel nontrivial decidable subclass of PDL, and demonstrates the viability of the proposed approach to deciding PDL in general.

1 Introduction

1.1 Motivation

A much-studied problem in the theory of automata is that of determining whether a given regular language L can be defined by a formula of some logic–in other words, to give an effective characterization of the precise expressive power of the logic. For automata over words, there is by now a large collection of such results, giving effective tests for definability in many temporal and predicate logics.

For tree automata, the situation is quite different: the problems of effectively deciding expressibility in *CTL*, *CTL**, first-order logic with ancestor, and propositional dynamic logic (*PDL*) remain open to this day.

In [6] Bojańczyk, *et. al.,* proposed to attack this problem by adapting the algebraic tools that have proved so successful in the case of word languages. They proved (working in the setting of languages of finite unordered forests) that the languages definable in each of the four logics cited above can be characterized as those recognized by iterated wreath products of forest algebras, where the factors in the wreath product all belong to a particular decidable variety of algebras. For example, languages in *PDL*, which are the focus of the present paper, are exactly those recognized by wreath products of forest algebras, each of which has an idempotent and commutative horizontal monoid, and which satisfies a distributive law. (See below for precise definitions).

Straubing, in [17] detailed a possible approach to *PDL* by noting that forest algebras that divide an iterated wreath product of k distributive algebras satisfy a kind of order k generalized distributive law (analogous to solvable groups, which satisfy an order k commutative law, for some $k > 0$). Determining whether a given forest algebra satisfies such a generalized distributive law for some k is a decidable problem. So if one could prove that every such generalized distributive forest algebra divides a wreath product of distributive algebras, the question of definability in *PDL* would be settled.

Here, we solve this problem in the case $k = 2$. More precisely, we show that every 2-distributive finite forest algebra divides a wreath product of four distributive algebras, and that further, 2-distributivity is itself a decidable property. Thus we have identified a decidable nontrivial subclass of *PDL*, and demonstrated the viability of the proposed approach for deciding *PDL* in general.

PDL contains the logics *CTL* and *CTL**, the latter being the bisimulation-invariant part of first-order logic on trees. The graded equivalent of *PDL*, also known as Chain Logic, fully contains first-order logic with ancestor. *PDL* and Chain Logic are the largest among the tree logics that have been considered in [6] and related work on finding effective characterizations. Indeed, *PDL* could be seen as the 'largest' nontrivial bisimulation-invariant logic on finite trees: It seems that no nontrivial logic class has been found between *PDL* and the bisimulation-invariant Boolean Formula Value problem, which is not definable in any of these logics [14, 17]. Strikingly, for all of these logics, decidability is still open despite several attempts, and a family of decidability results obtained for smaller fragments of these logics (e.g., [2, 3, 5, 12]). Furthermore, all of these logics were characterized in [6] in terms of iterated wreath products of certain forest algebras that satisfy a distributive law. While the representations of *CTL* and *CTL** place restrictions on these algebras, *PDL* is characterized by products of arbitrary distributive forest algebras.

1.2 Outline of the paper

In Section 2 we recall the basic definitions of forests and forest algebras. In Section 3, we review the algebraic characterization of Propositional Dynamic Logic (PDL) in terms of wreath products of distributive forest algebras. In Section 4, we discuss 2-distributive forest algebras, the main object of study in this paper. In Section 5, we review a generalization of the classical Derived Category Theorem to the setting of forest algebras, recently introduced by [18]. In Section 6, we prove the main result: Languages recognized by 2-distributive forest algebras are in PDL. We will discuss the two contributions on which this result rests, namely a separation theorem and a Local-Global theorem. We discuss the role of our results in Section 8.

2 Background on Finite Forests and Forest Algebras

Definition 1 (Forest Algebras)**.** A tuple (H, V) is called a *forest algebra* if the following conditions hold:

1. H is a monoid, whose operation is written +, with neutral element 0_H
2. V is a monoid, whose operation is written $\cdot$, with neutral element 1_V
3. There is an operation $V \times H \to H$, also written $\cdot$
4. This operation is an action: $v \cdot (v' \cdot h) = (vv') \cdot h$
5. The action is faithful: If $v \cdot h = v' \cdot h$ for all $h \in H$, then $v = v'$
6. There is a map $I_\cdot : H \to V$ such that $I_h h' = h + h'$.

LICS'18, July 09–12, 2018, Oxford
2018.

Note that, due to faithfulness, this map is uniquely determined. We will write $h + v$ for $I_h \cdot v$ (that is, the product of I_h and v as elements of V).

7. For each $h \in H$, there is $v \in V$ such that $h = v \cdot 0_H$.

If (H, V), (H', V') are forest algebras, then a pair $\phi = (\phi_H, \phi_V)$ of maps $\phi_H : H \to H'$, $\phi_V : V \to V'$ is called a *morphism* if it respects these structures. More formally, we require that (1) ϕ_H, ϕ_V are monoid morphisms, (2) $\phi_V(v)\phi_H(h) = \phi_H(vh)$, (3) $I_{\phi_H(h)} = \phi_V(I_h)$ for all $h \in H$, $v \in V$.

We will often omit the $\cdot$ operator for the multiplication on V and the action of V on H. But we will never omit the + operator for the addition on H.

We will use $\mathfrak{F}, \mathfrak{G}$ as variables for forest algebras. For a forest algebra $\mathfrak{F} = (H, V)$, the elements of H are called *forest types*, while the elements of V are called *context types*. Given a forest algebra $\mathfrak{F} = (H, V)$, we will sometimes write $H_{\mathfrak{F}}$, $V_{\mathfrak{F}}$ for H and V, respectively.

Trees and Forests Let Σ be a finite set, referred to as *alphabet*. By *trees over* Σ, we refer to finite (well-founded) trees, all of whose nodes are labeled with symbols in Σ. We do not allow empty trees.

Contexts, Free Forest Algebra A *context* is a forest where (exactly) one leaf is labeled with a variable instead of a symbol from Σ.

Contexts form a monoid V_Σ: We define $v \cdot v'$ to be the context obtained by replacing the variable in v with the context v'. The result is again a context. This operation is associative. The identity element is the context consisting of only a variable. Forests form a monoid H_Σ, with union as the monoid operation +, and the empty forest as the identity element. The monoid of contexts acts on the monoid of forests, with insertion of forests into the hole of a context as the operation. Taken together, the monoid of forests and the monoid of contexts form a forest algebra, the *free forest algebra* $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$.

Definition 2 (Recognition)**.** A forest algebra (H, V) *recognizes* a forest language $\mathcal{L} \subseteq H_\Sigma$ (that is, a set of forests) if and only if there is a forest algebra morphism $\phi : \Sigma^\Delta \to (H, V)$ such that $\mathcal{L} = \phi^{-1}(\phi(\mathcal{L}))$.

Many notions from Universal Algebra carry over to forest algebras. If a tuple (H', V') consists of subsets $H' \subseteq H$ and $V' \subseteq V$, then it is a *subalgebra* of a forest algebra (H, V) if it is closed under the forest algebra operations of (H, V). A pair of equivalence relations on H and V is called a *congruence* of (H, V) if it respects the forest algebra operations. The *quotient* (H', V') of (H, V) by a congruence is formed by taking H' and V' to be the sets of equivalence classes of the respective equivalence relations given by the congruence. Since congruences respect forest algebra operations, the action $\cdot$ and the operation $I.$ are well-defined on the quotient.

Definition 3 (Division)**.** Let (H, V), (H', V') be forest algebras. Then $(H, V) \prec (H', V')$ if (H, V) is a quotient of a subalgebra of (H', V').

If $\mathfrak{F} \prec \mathfrak{F}'$, then any language recognized by $\mathfrak{F}$ is also recognized by $\mathfrak{F}'$. This is shown in analogy to the parallel result for word languages and monoids [8].

Horizontal Idempotency and Commutativity A forest algebra (H, V) is *horizontally commutative and idempotent* if $h + h = h$ and $h_1 + h_2 = h_2 + h_1$ hold for all $h, h_1, h_2 \in H$. From now on, we will assume that all forest algebras are horizontally commutative and idempotent. This is no loss of generality, since all PDL languages are recognized by horizontally commutative and idempotent forest algebras [6].

Furthermore, we will consider *trees* without regard to order and multiplicity of children. More formally, we can define trees inductively as follows: The set of trees over Σ is the smallest set such that (1) $\alpha[\emptyset]$ is a tree whenever $\alpha \in \Sigma$, and (2) whenever C is a finite set of trees, and $\alpha \in \Sigma$, then $\alpha[C]$ is also a tree. Note that this means that the free forest algebra Σ^Δ is also horizontally commutative and idempotent.

3 PDL and Wreath Products of Forest Algebras

Having introduced the general algebraic background for studying wreath products of forest algebras, we now discuss *distributive* forest algebras and the forest logic that we are focusing on, Propositional Dynamic Logic. We refer to [6] for the definition of Propositional Dynamic Logic as a temporal logic on trees and forests. For our purposes, the algebraic characterization from [6] will be sufficient. We will first introduce the forest algebra wreath product.

3.1 Wreath Products of Forest Algebras

In [6], Bojańczyk *et al.* introduced the following wreath product operation on forest algebras:

Definition 4 ([6])**.** Let (H_1, V_1), (H_2, V_2) be forest algebras. Then define the *wreath product* as

$$(H_1, V_1) \wr (H_2, V_2) := (H_1 \times H_2, V_1^{H_2} \times V_2)$$

with the following operations: For $(f, v) \in V_1^{H_2} \times V_2$ and $(h_1, h_2) \in H_1 \times H_2$, let

$$(f, v)(h_1, h_2) := (f(h_2)h_1, vh_2)$$

For $(f, v), (f', v') \in V_1^{H_2} \times V_2$, let

$$(f, v)(f', v') := (f'', vv')$$

with $f''(h) := (f(v'h)) \cdot (f'(h))$. For the operation on $H_1 \times H_2$, we use the structure of the direct product.

[6] showed that $(H_1, V_1) \wr (H_2, V_2)$ is a forest algebra. Also, the wreath product is associative up to isomorphism [6]: $((H_1, V_1) \wr (H_2, V_2)) \wr (H_3, V_3) \equiv (H_1, V_1) \wr ((H_2, V_2) \wr (H_3, V_3))$. Therefore, it makes sense to talk about iterated wreath products of classes of forest algebras.

The wreath product has been applied to finite forest algebras in previous work, but nothing in the definition precludes application to infinite forest algebras ($V_1^{H_2}$ will then be uncountable). We will make reference to wreath products of infinite forest algebras for illustrative purposes, but our main result will not depend on this.

3.2 Distributive Forest Algebras and PDL

A forest algebra (H, V) is called *distributive* [6] if

$$v(h_1 + h_2) = vh_1 + vh_2 \tag{1}$$

for all $v \in V$, $h_1, h_2 \in H$. Equivalently, (H, V) is distributive if, for each morphism $\phi : \Sigma^\Delta \to (H, V)$, for all contexts c, and for all forests f_1, f_2, the following equality holds:

$$\phi(c(f_1 + f_2)) = \phi(cf_1 + cf_2) \tag{2}$$

Recall that, additionally, we require idempotency ($h + h = h$ for $h \in H$) and commutativity ($h_1 + h_2 = h_2 + h_1$ for $h_1, h_2 \in H$) for all forest algebras in this paper.

There is a close connection between distributive forest algebras and the sets of paths of forests. If $\phi : \Sigma^\Delta \to \mathfrak{F}$ is a morphism to a distributive algebra $\mathfrak{F}$, then, for any forest f, the value $\phi(f)$ only depends on the set of (not necessarily maximal) paths in the forest f. If L is a regular language of words, then the language of forests that have (not necessarily maximal) paths in L is recognized by a finite distributive forest algebra. More generally, the class of forest languages recognized by such algebras is exactly the Boolean algebra generated by languages of this form (Proposition 6).

Definition 5. If f is a forest, we write $\pi(f)$ for the set of its (not necessarily maximal) paths, starting at the root. Thus, $\pi(f)$ is a finite subset of Σ^* closed under taking prefixes ($wv \in X \Rightarrow w \in X$).

Proposition 6. [Theorem 5.3 from [6]] A language $\mathcal{L} \subseteq H_\Sigma$ is recognized by a finite distributive algebra if and only if it is a finite Boolean combination of languages of the form

$$\mathcal{L}_I := \{f \in H_\Sigma : \pi(f) \cap I \neq \emptyset\}$$

with $I \subseteq \Sigma^*$ regular word languages.

We can now state the algebraic characterization of Propositional Dynamic Logic (PDL) by [6]:

Theorem 7 ([6]). A regular language $\mathcal{L} \subset H_\Sigma$ is definable in PDL if and only if there are finite distributive forest algebras $\mathfrak{F}_1, ..., \mathfrak{F}_k$ such that $\mathfrak{F}_1 \wr ... \wr \mathfrak{F}_k$ recognizes $\mathcal{L}$.

By this result, the problem of deciding definability of a language $\mathcal{L}$ in PDL can be reduced to the problem of determining whether it is recognized by an iterated wreath product of finite distributive algebras. This, in turn, is equivalent to the question whether the syntactic forest algebra of $\mathcal{L}$ divides such a product.

4 2-Distributive Forest Algebras

We now define 2-distributive forest algebras, which will be the subject of our main result. In the Discussion (Section 8), we will discuss how this notion and results in this section generalize to $k > 2$, and how this notion relates to the general approach to settling decidability of PDL. A forest algebra (H, V) is *2-distributive* if, for any alphabet Σ and for all morphisms $\phi : \Sigma^\Delta \to (H, V)$, for all contexts $v \in V_\Sigma$, and for all forests $f_1, f_2 \in H_\Sigma$ with $\pi(f_1) = \pi(f_2)$, the following equality holds:

$$\phi(v(f_1 + f_2)) = \phi(vf_1 + vf_2)$$

That is, we take the same condition as for distributivity, but require this only in the case when $\pi(f_1) = \pi(f_2)$. In this sense, we are describing a 2-fold iterated distributive law. As before, we further require horizontal idempotency ($h + h = h$ for $h \in H$) and commutativity ($h_1 + h_2 = h_2 + h_1$ for $h_1, h_2 \in H$).

Remark 8. By definition, the property of 2-distributivity is expressed by a collection of identities between explicit operations, and thus 2-distributive forest algebras form a (Birkhoff) variety. We will not explicitly make use of varieties here. However, it deserves mentioning that most classes of forest languages that have been characterized using identities involve *profinite* identities involving implicit operations [3], while defining 2-distributivity does not involve such implicit operations.

We will now show that 2-distributive algebras are closely connected to wreath products of distributive forest algebras. The following proposition is not hard to prove:

Proposition 9. If $\mathfrak{F}_1, \mathfrak{F}_2$ are distributive forest algebras, then $\mathfrak{F}_1 \wr \mathfrak{F}_2$ is 2-distributive.

Proof. Let $\mathfrak{F}_1 = (H_1, V_1)$, $\mathfrak{F}_2 = (H_2, V_2)$ be distributive forest algebras (finite or infinite). Take any alphabet Σ and a morphism $\phi : \Sigma^\Delta \to \mathfrak{F}_1 \wr \mathfrak{F}_2$. Let $v \in V_\Sigma$, $f_1, f_2 \in H_\Sigma$ with $\pi(f_1) = \pi(f_2)$. Note that context types and forest types in $\mathfrak{F}_1 \wr \mathfrak{F}_2$ are tuples, whose right elements are elements of $\mathfrak{F}_2$. As in the case of monoid wreath products, it is easy to see that the projection of elements of $\mathfrak{F}_1 \wr \mathfrak{F}_2$ on the second component is a morphism from $\mathfrak{F}_1 \wr \mathfrak{F}_2$ to $\mathfrak{F}_2$. That is, $\pi^{(2)} \circ \phi : \Sigma^\Delta \to \mathfrak{F}_2$. Since $\mathfrak{F}_2$ is distributive and $\pi(f_1) = \pi(f_2)$, we know $\pi^{(2)}\phi(f_1) = \pi^{(2)}\phi(f_2)$. Let's call this element h_0. Let us compute $\phi(v(f_1 + f_2))$. By definition of $\mathfrak{F}_1 \wr \mathfrak{F}_2$, there is $f \in V_1^{H_2}$ and $u \in V_2$, such that $\phi(v) = (f, u)$. Similarly, $\phi(f_i) = (h_i, \pi^{(2)}\phi(f_i))$ for some $h_i \in H_1$, for $i = 1, 2$. Thus,

$$\begin{aligned} \phi(v(f_1 + f_2)) &= (f, u)((h_1^1, h_1^2) + (h_2^1, h_2^2)) \\ &= (f, u)(h_1 + h_2, \pi^{(2)}\phi(f_1 + f_2)) \\ &= (f(h_0)(h_1 + h_2), uh_0) \end{aligned} \tag{3}$$

On the other hand,

$$\begin{aligned} \phi(vf_1 + vf_2) &= (f, u)(h_1, h_0) + (f, u)(h_1, h_0) \\ &= (f(h_0)h_1, uh_0) + (f(h_0)h_2, uh_0) \\ &= (f(h_0)h_1 + f(h_0)h_2, uh_0) \end{aligned} \tag{4}$$

Given that $\mathfrak{F}_1$ is distributive, the last lines of (3) and (4) are the same. □

This statement has a converse, which can be shown using the Local-Global Theorem 19:

Theorem 10. A forest algebra $\mathfrak{F}$ (finite or infinite) is 2-distributive if and only if it divides a wreath product of two (possibly infinite) distributive forest algebras.

Proof Sketch. The 'if' direction is the previous proposition. For the 'only if' direction, the proof closely follows that of Theorem 16. Let $\sim$ be the congruence on Σ^Δ induced by $v[h + h'] = vh + vh'$. The quotient of Σ^Δ by $\sim$ is an infinite distributive forest algebra, which we denote Σ^Δ_D. We can extend π to a morphism $\Sigma^\Delta \to \Sigma^\Delta_D$. Let $\phi : \Sigma^\Delta \to \mathfrak{F}$ be any morphism. Consider $D_{\phi,\pi}$ (Definition 13), an infinite forest category. From the definition of 2-distributivity, one can show that $D_{\phi,\pi}$ is locally-distributive (Definition 18). Similar to Theorem 19, one can then show that $D_{\phi,\pi}$ divides an infinite distributive forest algebra Σ'^Δ_D where Σ' is an extended alphabet. The Derived Category Theorem 15 implies that $\mathfrak{F}$ divides $\Sigma'^\Delta_D \wr \Sigma^\Delta_D$. □

Allowing infinite algebras is crucial here: Even if $\mathfrak{F}$ is finite, we cannot readily conclude that it divides a product of two *finite* distributive forest algebras. Nonetheless, this characterization is interesting: It shows that 2-distributive algebras represent the second level in the hierarchy of iterated wreath products of infinite distributive algebras. This hierarchy can be viewed as an infinitary counterpart to PDL, as it allows products of infinite algebras.

Using this characterization, we can give a simple example:

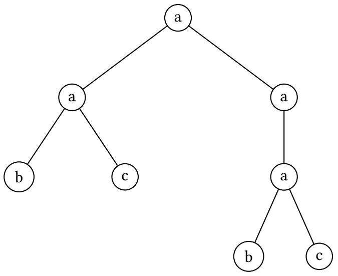


Figure 1. Illustration for Example 11

Example 11. Any distributive algebra is evidently also 2-distributive. For a less trivial example, consider the language of forests satisfying the following conditions: (1) Each maximal path has the form a^*b or a^c, (2) each b-node has a c-sibling, (3) each c-node has a b-sibling (see Figure 1). This language is not recognized by a distributive algebra. It is not hard to show that it is recognized by a wreath product of two finite forest algebras, and thus, using Proposition 9, by a 2-distributive forest algebra.

In Example 11, the recognizing algebra is not just 2-distributive but also divides the wreath product of two finite distributive algebras (the language is therefore in PDL). However, this is not the case in general: Finite 2-distributive algebras need not divide a wreath product of two *finite* distributive forest algebras. We will show this in the example below. The main result of this paper will imply that a wreath product of *four* finite distributive forest algebras will still be enough in this case. This proves that all languages recognized by finite 2-distributive forest algebras are definable in PDL.

Example 12. Consider the following languages (see Figure 2 for illustration):

$\mathcal{L}_1$ is the set of nonempty forests where (1) each maximal path has the form a^*b or a^*c, (2) each b-node has a c-sibling, (3) each c-node has a b-sibling (see Figure 2a).

$\mathcal{L}_2$ is the set of nonempty forests where (1) each maximal path has the form a^*b or a^+c, (2) each b-node has an a-sibling which has a c-child, (3) each a-node with a c-child has a b-sibling (see Figure 2b).

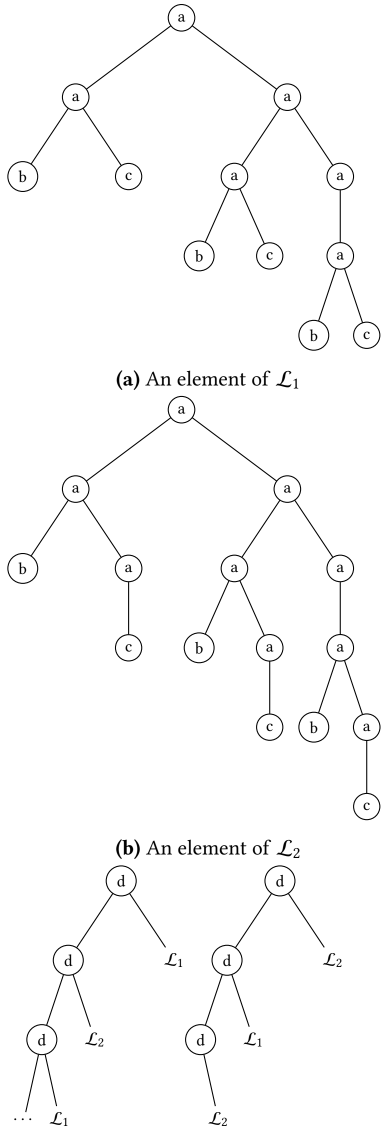
$\mathcal{L}_3^a$ is the set of (possibly empty) forests where each tree has the form $c_d(f_1 + f_2)$, where c_d denotes the context consisting of only a node labeled d and a variable below it, with $f_1 \in \mathcal{L}_3^b$, $f_2 \in \mathcal{L}_1$.

$\mathcal{L}_3^b$ is the set of (possibly empty) forests where each tree has the form $c_d(f_1 + f_2)$, with $f_1 \in \mathcal{L}_3^a$ and $f_2 \in \mathcal{L}_2$.

Set $\mathcal{L} := \mathcal{L}_3^a + \mathcal{L}_3^b$ (see Figure 2c).

It can be shown that $\mathcal{L}$ is recognized by a wreath product of two infinite distributive algebras and thus is 2-distributive by Proposition 10. However, $\mathcal{L}$ is not recognized by any wreath product of two *finite* distributive algebras.

The proof is based on facts about *separation* by morphisms to distributive algebras: If $\mathcal{L}$ is a language of forests, $\pi(\mathcal{L}) \subseteq Pow(\Sigma^*)$ is the image of $\mathcal{L}$ under π, a set of finite pathsets. First, it is not hard to see that $\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2)$ is empty, and thus the language $\pi^{-1}(\pi(\mathcal{L}_1))$ separates these: $\mathcal{L}_1 \subset \pi^{-1}(\pi(\mathcal{L}_1)) \subset (H_\Sigma - \mathcal{L}_2)$. The syntactic forest algebra of $\pi^{-1}(\pi(\mathcal{L}_1))$ is distributive, but infinite (it crucially needs to count at arbitrary depths). From this fact, one can derive using Theorem 10 that $\mathcal{L}$ is indeed 2-distributive.



(a) An element of $\mathcal{L}_1$

(b) An element of $\mathcal{L}_2$

(c) A schematic depiction of $\mathcal{L}$: The left tree belongs to $\mathcal{L}_3^a$, the right tree one to $\mathcal{L}_3^b$.

Figure 2. Illustration for Example 12. $\mathcal{L}$ is 2-distributive and regular, but not recognized by the wreath product of two finite distributive forest algebras.

However, even though $\mathcal{L}_1$ and $\mathcal{L}_2$ are both regular, no language recognized by a *finite* distributive algebra can separate them. From this, one can deduce using the Derived Category Theorem 15 that $\mathcal{L}$ is not recognized by the wreath product of any two finite distributive forest algebras. However, it is not hard to show that $\mathcal{L}_1$ and $\mathcal{L}_2$ are both recognized by a wreath product of two finite distributive algebras. Fom this, one can derive that $\mathcal{L}$ is recognized by a wreath product of *three* finite distributive forest algebras.

5 The Derived Forest Category

The proof of our main result will construct wreath product decompositions by separately studying the left and right factors of a wreath product. Given a 2-distributive forest algebra (H_2, V_2), we will use the Separation Lemma 17 to construct an intended right-hand factor (H_2, V_2) which is already known to be a wreath product of finite distributive forest algebras. The remaining problem is then to find a left-hand factor (H, V) such that

$$(H_1, V_1) \prec (H, V) \wr (H_2, V_2) \tag{5}$$

holds. If we can show that this factor can be chosen to be distributive, the problem is solved. In order to do this, we seek a general strategy to obtain a 'minimal' left-hand factor (H, V). In the case of groups, the solution to this problem is provided by the kernel group, ker ϕ, when $\phi : G \to H$: Then G is embedded in $\ker \phi \wr H$.

For monoids, the analog to $\ker \phi$ is not a monoid any more, but a category. This classical construction is known as the *Derived Category* ([21], [15]) and originated from the study of regular word languages via wreath products of finite monoids [7], [19], [16]. Recently, [18] showed that this construction generalizes to the setting of forest algebras.

The sense in which this construction is 'optimal' is made precise in Tilson's Derived Category Theorem [21]. In the case of forest algebras, we will essentially see that the decomposition in (5) holds if and only if (H, V) is divided by a certain forest category determined from (H_1, V_1) and (H_2, V_2) and morphisms from Σ^Δ into these, where the notion of 'division' by a category will be made precise below.

In this section, we will review the definition of the Derived Forest Category and the Derived Category Theorem from [18]. The Derived Forest Category is a category with some structure added to it, and the relation between forest categories and categories is akin to the relation between forest algebras and monoids. It is possible to define a general notion of Forest Categories [18], but for our purposes, it is sufficient to consider the Derived Forest Category:

Definition 13. [Derived Forest Category, [18]] Let Σ be a finite alphabet. Consider a pair of surjective forest algebra morphisms

$$(H_1, V_1) \xleftarrow{\alpha} \Sigma^\Delta \xrightarrow{\beta} (H_2, V_2)$$

The derived category $D_{\alpha,\beta}$ is defined as follows:

1. The set of *objects* of the category is $\mathrm{Obj}(D_{\alpha,\beta}) := H_2$
2. As in ordinary categories, *arrows* connect objects. To define the arrows, we fix $h, h' \in H_2$ and introduce an equivalence relation on the set of triples (h, p, h') with $p \in V_\Sigma$ for which $\beta(p) \cdot h = h'$. We set $(h, p, h') \sim (h, q, h')$ if for all $s \in H_\Sigma$ with $\beta(s) = h$, we have $\alpha(ps) = \alpha(qs)$.
   We then set $\mathrm{Arr}(h, h')$ to be the set of equivalence classes of $\sim$. Its elements are called *arrows*.
   We depict an arrow as $h' \xleftarrow{p} h$, with the understanding that the same arrow can have many distrinct representations in this form.
   For consistency with notation here, the direction of arrows in this graphical notation is inverted relative to [18].
3. To obtain a category, we now want to define the multiplication of arrows. We set
   $$\left(h_3 \xleftarrow{p} h_2\right) \cdot \left(h_2 \xleftarrow{q} h_1\right) = h_3 \xleftarrow{pq} h_1$$
   or shortened
   $$h_3 \xleftarrow{p} h_2 \xleftarrow{q} h_1 = h_3 \xleftarrow{pq} h_1$$
   It can be shown that this is a well-defined arrow, independently of the representation chosen for the arrows [18]. Since the multiplication on V_Σ is associative, this multiplication is associative. The arrow $h \xleftarrow{1_{V_\Sigma}} h$ is the identity at $h \in H_2$.
   Up to this point, we have defined a category. We now add some additional structure:
4. For $h \in H_2$, we set
   $$\mathrm{HArr}(h) := \{(\alpha(s), h) : s \in H_\Sigma, \beta(s) = h\}$$
   The elements of this set are called *half-arrows*. They can be thought of as being arrows that end in an object, but do not start in any object.
   We depict the half-arrow (h_1, h_2) as $h_2 \xleftarrow{h_1}$.
5. We set $\mathrm{HArr}(D_{\alpha,\beta})$ to be the set of all half-arrows. Note that $\mathrm{Obj}(D_{\alpha,\beta})$ and $\mathrm{HArr}(D_{\alpha,\beta})$ are monoids, and that the projection of a half-arrow onto the second element (that is, the object at the end of the arrow in our graphical notation) is a homomorphism from $\mathrm{HArr}(D_{\alpha,\beta})$ to $\mathrm{Obj}(D_{\alpha,\beta})$.
6. Viewed in analogy to forest algebras, arrows correspond to context types, while half-arrows correspond to forest types. We therefore want arrows to act on half-arrows. We define the action of an arrow on a half-arrow by
   $$\left(h_2' \xleftarrow{p} h_2\right) \cdot \left(h_2 \xleftarrow{h_1}\right) = h_2' \xleftarrow{\alpha(p)h_1}$$
   or shortened
   $$h_2' \xleftarrow{p} h_2 \xleftarrow{h_1} = h_2' \xleftarrow{\alpha(p)h_1}$$
7. In analogy to forest algebras, we want to be able to add arrows and half-arrows. We set
   $$\left(h' \xleftarrow{p} h\right) + \left(h_2 \xleftarrow{h_1}\right) = \left(h' + h_2\right) \xleftarrow{p+s} h$$
   where $s \in H_\Sigma$ such that $\alpha(s) = h_1$, $\beta(s) = h_2$.

For proofs of well-definedness, we refer the reader to [18].

To formulate the Derived Category Theorem connecting Derived Categories with wreath products, it is necessary to generalize the notion of division to the setting of forest categories dividing forest algebras:

Definition 14. [Division, [18]] If C is a derived forest category and (H, V) a forest algebra, then we write $C \prec (H, V)$, and say C *divides* (H, V), if for each $\left(x \xleftarrow{c}\right) \in \mathrm{HArr}(C)$ there exists a nonempty set $K_{x \xleftarrow{c}} \subseteq H$, and for each $\left(y \xleftarrow{d} x\right) \in \mathrm{Arr}(C)$ there exists a nonempty set $K_{y \xleftarrow{d} x} \subseteq V$ satisfying the following properties:

1. (Preservation of Operations) For all $x \xleftarrow{c}$, $y \xleftarrow{d} \in \mathrm{HArr}(C)$, $y \xleftarrow{e} x, z \xleftarrow{f} y \in \mathrm{Arr}(C)$,
   a. $K_{z \xleftarrow{f} y} \cdot K_{y \xleftarrow{e} x} \subseteq K_{z \xleftarrow{f} y \xleftarrow{e} x}$
   b. $K_{y \xleftarrow{e} x} \cdot K_{x \xleftarrow{c}} \subseteq K_{y \xleftarrow{e} x \xleftarrow{c}}$
   c. $K_{x \xleftarrow{c}} + K_{y \xleftarrow{d}} \subseteq K_{x \xleftarrow{c} + y \xleftarrow{d}}$
   d. $K_{x \xleftarrow{c}} + K_{z \xleftarrow{f} y} \subseteq K_{x \xleftarrow{c} + yz \xleftarrow{f}}$
   e. $K_{z \xleftarrow{f} y} + K_{x \xleftarrow{c}} \subseteq K_{z \xleftarrow{f} y + x \xleftarrow{c}}$
2. (Injectivity)

a. If $y \overset{c}{\leftarrow} x$ and $y \overset{c'}{\longleftarrow} x$ are distinct arrows, then $K_{y \overset{c}{\leftarrow} x} \cap K_{y \overset{c'}{\longleftarrow} x} = \emptyset$

b. If $y \overset{c}{\leftarrow}$ and $y \overset{c'}{\longleftarrow}$ are distinct half-arrows, then $K_{y \overset{c}{\longleftarrow}} \cap K_{y \overset{c'}{\longleftarrow}} = \emptyset$.

In the special case where a derived forest category has exactly one object, it can be viewed as a forest algebra. In this case, the notion of division reduces to ordinary division of forest algebras.

We are now ready to state the Derived Category Theorem connecting categories with wreath products. We state only the direction required for our main result:

Theorem 15. [Derived Category Theorem, [18]] Let Σ be an alphabet, and let α, β be morphisms from Σ^Δ onto forest algebras (H_1, V_1), (H_2, V_2), respectively. Let (H, V) be a forest algebra. Assume $D_{\alpha,\beta} \prec (H, V)$. Then

$$(H_1, V_1) \prec (H, V) \wr (H_2, V_2)$$

6 Main Result

Our aim is to prove that any language recognized by a finite 2-distributive forest algebras is definable in PDL:

Theorem 16. Let $\mathfrak{F}$ be a finite 2-distributive forest algebra. Then every language recognized by $\mathfrak{F}$ is definable in PDL.

In the Discussion (Section 8), we will discuss the relevance of this result for the question of deciding definability in PDL.

Our proof proceeds by solving two sub-problems related to the left and right factors in wreath product decompositions: To obtain the right factor of a wreath product decomposition, we study the problem of *separating* forest languages by the map π. To then obtain the left factor, we start at the Derived Category, and show that it has a certain local property – in our case, local distributivity. To conclude a decomposition result, we then prove that this local property entails a global property. These steps are remarkably similar to results from the theory of logic on words and finite monoids which also reduce the problem of decidability to separation [13] and Local-Global theorems [9].

6.1 Separation Lemma

We will first state the Separation Lemma. Recall that $\pi(f)$ is the set of paths in the forest f. If $\mathcal{L}$ is a language of forests, $\pi(\mathcal{L}) \subseteq Pow(\Sigma^*)$ is the image of $\mathcal{L}$ under π, a set of finite pathsets.

Lemma 17 (Separation Lemma)**.** Let $\mathcal{L}_1, \mathcal{L}_2 \subseteq H_\Sigma$ be regular forest languages such that

$$\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2) = \emptyset$$

Then there are finite distributive algebras $\mathfrak{F}_1, \mathfrak{F}_2, \mathfrak{F}_3$ and a language $X \subseteq \Sigma^\Delta$ recognized by $\mathfrak{F}_1 \wr \mathfrak{F}_2 \wr \mathfrak{F}_3$ such that

$$\mathcal{L}_1 \subseteq X \subseteq (H_\Sigma - \mathcal{L}_2)$$

That is, X *separates* $\mathcal{L}_1$ from $\mathcal{L}_2$.

Proof. The proof considers a forest algebra (H, V) recognizing both $\mathcal{L}_1$ and $\mathcal{L}_2$ via a morphism ϕ, and constructs PDL languages 'approximating' each $\phi^{-1}(h)$ for $h \in H$. □

Let's consider how this is useful for proving Theorem 16 by sketching the proof idea for the theorem – we will make this more precise in Section 6.3. If $\mathfrak{F}$ is a forest algebra with morphism $\phi_{\mathfrak{F}} : \Sigma^\Delta \to \mathfrak{F}$, we can apply this lemma to all pairs of languages $\phi_{\mathfrak{F}}^{-1}(h)$ for $h \in H_{\mathfrak{F}}$ and obtain separators $X_{h,h'}$ for each pair whenever $\pi\phi^{-1}(h) \cap \pi\phi^{-1}(h') = \emptyset$. Combining the resulting separators, we can build finite distributive algebras $\mathfrak{F}_1, \mathfrak{F}_2, \mathfrak{F}_3$ and a morphism $\phi_X : \Sigma^\Delta \to \mathfrak{F}_1 \wr \mathfrak{F}_2 \wr \mathfrak{F}_3$ which recognizes each $X_{h,h'}$. We will examine the derived category $D_{\phi_{\mathfrak{F}}, \phi_X}$. If we can show that this derived category divides a finite distributive algebra $\mathfrak{F}_0$, we can apply the Derived Category Theorem to conclude $\mathfrak{F} \prec \mathfrak{F}_0 \wr \mathfrak{F}_1 \wr \mathfrak{F}_2 \wr \mathfrak{F}_3$, which then will prove Theorem 16.

6.2 Locally-Distributive Categories

Recall the equation $v[h_1 + h_2] = vh_1 + vh_2$ defining distributive forest algebras. To apply this to derived forest categories, we would want to interpret v as an arrow and h_1, h_2 as half-arrows. In general, this equation does not make sense, since the action of an arrow on a half-arrow is only defined in certain cases. The equation becomes sensible when we restrict it to those arrows and half-arrows for which the action is defined:

Definition 18 (Locally Distributive)**.** We say that a derived forest category C is *locally distributive* if the following equation holds for any $h \in \mathrm{Obj}(C)$ and any two half-arrows $h_1, h_2 \in \mathrm{HArr}(h)$, and for any arrow $v \in \mathrm{Arr}(h, h')$ $(h' \in \mathrm{Obj}(C))$:

$$v[h_1 + h_2] = vh_1 + vh_2$$

Rewriting this in arrow-based notation, we want the following for any half-arrows $h \overset{h_1}{\longleftarrow}$ and $h \overset{h_2}{\longleftarrow}$, and for any arrow $h' \overset{v}{\leftarrow} h$:

$$\left(h' \overset{v}{\leftarrow} h\right) \cdot \left(h \overset{h_1}{\longleftarrow} + h \overset{h_2}{\longleftarrow}\right) = \left(h' \overset{v}{\leftarrow} h \overset{h_1}{\longleftarrow}\right) + \left(h' \overset{v}{\leftarrow} h \overset{h_2}{\longleftarrow}\right)$$

Any derived forest category that divides a distributive forest algebra must be locally distributive. We now show that the converse is also true: Any locally-distributive category divides some distributive forest algebra. In analogy to results from the theory of ordinary finite categories, we refer to this as a Local-Global Theorem – showing that being locally distributive entails a global property of the category:

Theorem 19 (Local-Global)**.** Let C be a locally-distributive finite derived forest category. Then it divides a finite distributive forest algebra.

Proof. The proof proceeds by considering terms built from arrows and half-arrows and using local distributivity to transform them into a normal form that only depends on the paths in these terms (viewing them as forests). We then apply Proposition 6 to construct a finite distributive forest algebra and a division. □

The idea of introducing a 'local' version of distributivity that is appropriate for forest categories, and then relating it to distributive forest algebras in a 'Local-Global' Theorem is related to a long tradition in the theory of semigroups and monoids, where local pseudovarieties of categories have been an important object of study (e.g., [21], [10], [1]), and where such Local-Global theorems have been applied to prove decidability of logic classes [9].

In order to prove Theorem 16, our goal will be to prove that the derived category $D_{\phi_{\mathfrak{F}}, \phi_X}$ mentioned above is locally distributive, then being able to apply Theorem 19. The details are given in Section 6.3.

6.3 Concluding the proof of Theorem 16

Proof of the Theorem. Let $\mathcal{L}$ be a language recognized by a 2-distributive finite algebra $\mathfrak{F}$ via morphism $\phi : \Sigma^{\Delta} \to \mathfrak{F}$. For each pair $h_1, h_2 \in H_{\mathfrak{F}}$ such that $\pi(\phi^{-1}(h_1)) \cap \pi(\phi^{-1}(h_2)) = \emptyset$, we can apply Lemma 17 to the languages $\mathcal{L}_1 := \phi^{-1}(h_1)$ and $\mathcal{L}_2 := \phi^{-1}(h_2)$. From the lemma we get a language X_{h_1,h_2} separating the preimages of h_1 and h_2. That is, we have

$$\phi^{-1}(h_1) \subseteq X_{h_1,h_2} \subseteq (H_{\Sigma} - \phi^{-1}(h_2))$$

We also get an algebra $\mathfrak{G}_{h_1,h_2} = \mathfrak{F}_1^{h_1,h_2} \wr \mathfrak{F}_2^{h_1,h_2} \wr \mathfrak{F}_3^{h_1,h_2}$ which recognizes X via morphism $\psi_{h_1,h_2} : \Sigma^{\Delta} \to \mathfrak{G}_{h_1,h_2}$. By the lemma, the three algebras $\mathfrak{F}_1^{h_1,h_2}, \mathfrak{F}_2^{h_1,h_2}, \mathfrak{F}_3^{h_1,h_2}$ are finite and distributive.

Let

$$\mathfrak{G} := \left(\bigtimes_{h_1,h_2} \mathfrak{F}_1^{h_1,h_2} \right) \wr \left(\bigtimes_{h_1,h_2} \mathfrak{F}_2^{h_1,h_2} \right) \wr \left(\bigtimes_{h_1,h_2} \mathfrak{F}_3^{h_1,h_2} \right)$$

where we take products over all pairs h_1, h_2 for which $\pi(\phi^{-1}(h_1)) \cap \pi(\phi^{-1}(h_2)) = \emptyset$ holds. Then $\mathfrak{G}$ is a wreath product of three finite distributive algebras. Furthermore, it is divided by each $\mathfrak{G}_{h_1,h_2}$, and thus recognizes each of the separators X_{h_1,h_2} via some morphism $\psi : \Sigma^{\Delta} \to \mathfrak{G}$.

We now consider the derived forest category $D_{\phi,\psi}$. We want to show that it is locally distributive, then being able to apply the Derived Category Theorem. Let h, h' be objects, let $f_1, f_2 \in \mathrm{HArr}(h)$, and let $v \in \mathrm{Arr}(h, h')$. By the definition of the Derived Category, we can write f_1 as $h \xleftarrow{h_1}$ and f_2 as $h \xleftarrow{h_2}$. Also, we can write v as $h \xleftarrow{p} h'$, with $p \in V_{\mathfrak{F}}$. We want to prove the equality from Definition 18.

Note that $h_1, h_2 \in H_{\mathfrak{F}}$. In view of the construction of the half-arrows in the derived category, there are forests $t_1, t_2 \in H_{\Sigma}$ such that $\phi(t_i) = h_i$ and $\psi(t_i) = h$ for $i = 1, 2$. For a contradiction, let us assume $\pi(\phi^{-1}(h_1)) \cap \pi(\phi^{-1}(h_2)) = \emptyset$. Given the way ψ was constructed, the equality $\psi(t_i) = h$ entails $\psi_{h_1,h_2}(t_1) = \psi_{h_1,h_2}(t_2)$. This is a contradiction to the way in which we have chosen ψ_{h_1,h_2}. The assumption about the empty intersection must have been incorrect, and we have

$$\pi(\phi^{-1}(h_1)) \cap \pi(\phi^{-1}(h_2)) \neq \emptyset$$

So there are forests b_1, b_2 such that $\phi(b_i) = h_i$ and $\pi(b_1) = \pi(b_2)$.

Recall $v = h \xleftarrow{p} h'$, with $p \in V_{\mathfrak{F}}$. Let $\alpha' \in \phi^{-1}(p)$. Since $\mathfrak{F}$ is 2-distributive, we have $\phi(\alpha'[b_1 + b_2]) = \phi(\alpha' b_1 + \alpha' b_2)$. Applying ϕ, this means

$$p(h_1 + h_2) = p(h_1) + p(h_2)$$

In the derived category, this translates to

$$v(f_1 + f_2)) = vf_1 + vf_2$$

or, in arrow-based notation,

$$h' \xleftarrow{p} \left(h \xleftarrow{h_1} + h \xleftarrow{h_2} \right) = \left(h' \xleftarrow{p} h \xleftarrow{h_1} \right) + \left(h' \xleftarrow{p} h \xleftarrow{h_2} \right)$$

Thus, $D_{\phi,\psi}$ is locally distributive. It is also finite (the two algebras involved in its construction are finite), so it divides a finite distributive algebra $\mathfrak{F}'$. By the Derived Category Theorem, $\mathcal{L}$ is recognized by $\mathfrak{F}' \wr \mathfrak{G}$, which is the wreath product of four finite distributive algebras. □

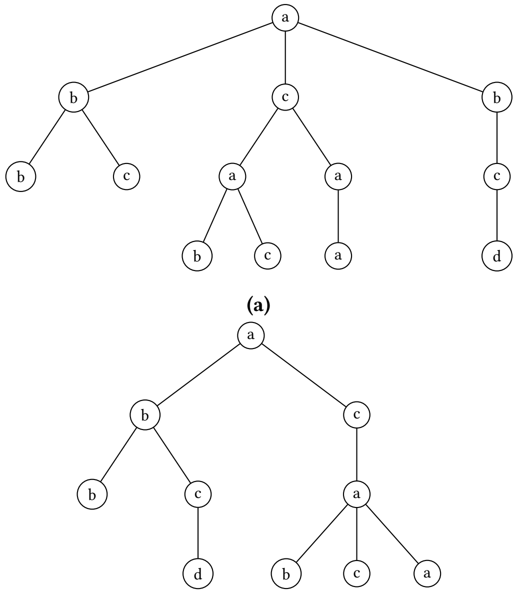


Figure 3. Illustration for Definition 20. Applying Ψ to the tree in (a) results in the tree in (b). The trees have the same (not necessarily maximal) paths. In (b), no two siblings have the same label.

7 Decidability of 2-Distributivity

We have shown that languages recognized by finite 2-distributive algebras form a subclass of PDL. We now show that 2-distributivity is decidable.

To decide whether $\mathfrak{F}$ is 2-distributive, we need to check for morphisms $\phi : \Sigma^{\Delta} \to \mathfrak{F}$ whether $\phi(v(f_1 + f_2)) = \phi(vf_1 + vf_2)$ holds whenever $\pi(f_1) = \pi(f_2)$, for all forests $f_1, f_2 \in H_{\Sigma}$ and all contexts $v \in V_{\Sigma}$. To do this algorithmically, we want to find those pairs $h, h' \in H$ such that $\pi(\phi^{-1}(h))$ and $\pi(\phi^{-1}(h'))$ have nonempty intersection. For these h, h', we then need to check whether $v(h + h') = vh + vh'$ for all $v \in V$. If we can find these pairs h, h' algorithmically, decidability is shown (We will see that looking at one specific morphism ϕ is enough.).

Thus, the problem boils down to deciding, given two regular forest languages $\mathcal{L}_1, \mathcal{L}_2$, whether $\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2)$ is empty. We will reduce this to the problem of deciding whether two regular forest languages – computed from $\mathcal{L}_1, \mathcal{L}_2$ – have nonempty intersection. We will use the following tool:

Definition 20 (Distributive Normal Form)**.** Define a map $\Psi : H_{\Sigma} \to H_{\Sigma}$ as follows.

Consider a forest $f := \beta_1[f_1] + \ldots + \beta_n[f_n]$ $(n \geq 0)$. Here, $\beta_1, \ldots, \beta_n$ are symbols from Σ, some or all of which can be identical, and $f_1, \ldots, f_n$ are forests. For each $\beta \in \{\beta_1, .., \beta_n\}$, define

$$F_{\beta} := \{f_i : \beta_i = \beta\} \subset \{f_1, \ldots, f_n\}$$

Then, set

$$\Psi(f) := \sum_{\beta \in \{\beta_1, \ldots, \beta_n\}} \beta \left(\Psi \left[\sum_{f' \in F_{\beta}} f' \right] \right)$$

An example is provided in Figure 3.

Proposition 21. Let $f, f' \in H_\Sigma$.

1. No two distinct sibling nodes in $\Psi(f)$ are labeled with the same symbol.
2. $\pi(f) = \pi(\Psi(f))$.
3. $\pi(f) = \pi(f')$ if and only if $\Psi(f) = \Psi(f')$.

Proof. (1) By induction over the height of forests. (2) is immediate from the definition of Ψ. For (3), the 'if' direction follows from (2). For the 'only if' direction, observe that, for any given pathset, there is only a single forest (up to order of siblings) having this pathset and satisfying the condition that no sibling nodes have the same symbol. □

Due to the second property, we will use $\Psi(f)$ as a suitable representative forest for the path set $\pi(f)$. In certain ways, $\Psi(f)$ will be better-behaved than a general forest f. The following proposition shows that the image of languages under Ψ is also well-behaved:

Proposition 22. Let $\mathcal{L}$ be a regular forest language. Then $\Psi(\mathcal{L})$ is a regular forest language and can be effectively constructed from (an automaton for) $\mathcal{L}$.

It is important to note that the image $\Psi(\mathcal{L})$ is not recognized by a horizontally idempotent forest algebra, as multiplicity of children does matter. The recognizing finite forest algebra will be horizontally commutative but not idempotent. This proposition and proof is the one place in this paper where we deviate from our convention that all forest algebras are horizontally commutative and idempotent.

Proof. Let $\mathfrak{F} = (H, V)$ be a finite forest algebra recognizing $\mathcal{L}$ via morphism ϕ.

Let $H' := Pow(Pow(H))^\Sigma \cup \{\bot\}$ with the operation: $f + f' = \bot$ if there is $\alpha \in \Sigma$ such that $f(\alpha), f'(\alpha) \neq \emptyset$, or if one of f, f' is already equal to $\bot$. Otherwise, $(f + f')(\alpha) = f(\alpha) \cup f'(\alpha)$. It should be noted that this operation is not idempotent due to the first condition, which makes $f + f = \bot$ unless $f(\alpha) = \emptyset$ for all α. With this operation, H' is a commutative finite monoid with identity f_0 given by $f_0(\alpha) = \emptyset$ for $\alpha \in \Sigma$.

Let V' be the finite monoid of all maps $H' \to H'$, which naturally acts on H'. It is easy to show that (H', V') is a finite forest algebra (though not horizontally idempotent).

Then define a morphism $\psi : \Sigma^\Delta \to (H', V')$ by first constructing the images of the contexts consisting of only a single letter: $\psi(\alpha) := g_\alpha$ where by $\psi(\alpha)$ we denote the image of the context consisting of only α and a variable below it ($\alpha[X]$). Once we have chosen $g_\alpha \in V'$ for each α, it is not hard to see that we obtain a unique forest algebra morphism $\psi : \Sigma^\Delta \to (H', V')$ extending this map. Recall that g_α will need to be a map $H' \to H'$. We first set $g_\alpha(\bot) = \bot$ for all α. For $f \in H' - \{\bot\}$, so $f : \Sigma \to Pow(Pow(H))$, we furthermore set

$$\psi(\alpha)(f)(\beta) = \emptyset \text{ when } \alpha \neq \beta$$

Finally, considering the case $\alpha = \beta$, then for any $Q \subset H$, we set $Q \in \psi(\alpha)(f)(\alpha)$ if and only if there are sets $Q_1, ..., Q_l \subset H$ such that for each $\gamma \in \Sigma$ such that $f(\gamma) \neq \emptyset$, there is $P_\gamma \in f(\gamma)$ such that

$$Q_1 \cup ... \cup Q_l = \bigcup_\gamma P_\gamma$$

and

$$Q = \left\{ \phi(\alpha) \cdot \left[\sum_{h \in Q_i} h \right] : i = 1, ..., l \right\}$$

We now claim that, for $h \in H$, the language $\Psi(\phi^{-1}(h)) - \{\emptyset\}$ (that is, removing the empty forest if it is in the language) is equal to

$$\psi^{-1}(\{f : \exists \alpha : f(\alpha) \neq \emptyset \wedge \forall \alpha \in \Sigma : f(\alpha) = \emptyset \vee \{h\} \in f(\alpha)\})$$

where ψ is the forest algebra morphism $\psi : \Sigma^\Delta \to (H', V')$ that we just constructed. This is shown by induction over forests.

Considering that the empty forest is the only element of the preimage of the identity element of H', this implies that $\Psi(\mathcal{L})$ is recognized by (H', V') via ψ. □

We can now show decidability of 2-distributivity:

Theorem 23. It is decidable whether a finite forest algebra is 2-distributive.

Proof. Given a finite forest algebra $\mathfrak{F} = (H, V)$, choose $\Sigma := V$, and let $\phi : \Sigma^\Delta \to (H, V)$ be the (unique) morphism extending the identity map $\phi : \Sigma \to V$, that is, mapping the context $v[X]$ to $v \in V$.

Given two regular forest languages $\mathcal{L}_1, \mathcal{L}_2$, it is decidable whether $\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2)$ is empty. To prove this, we use the mapping Ψ introduced in Definition 20. We can use Proposition 22 to effectively check whether the regular forest languages $\Psi(\mathcal{L}_1)$ and $\Psi(\mathcal{L}_2)$ have nonempty intersection. From Proposition 21, we know that this happens if and only if $\pi(\phi^{-1}(h))$ and $\pi(\phi^{-1}(h'))$ also have nonempty intersection.

Using this resut, we can then, for each pair $h, h' \in H$, effectively check whether $\pi(\phi^{-1}(h))$ and $\pi(\phi^{-1}(h'))$ have nonempty intersection. If this is the case, we can check for each context type $v \in V$ whether $v[h + h'] = vh + vh'$. This equality holds for each v and for each selected pair h, h' if and only if $\phi(c(f + f')) = \phi(cf + cf')$ for all $c \in V_\Sigma$ and each $f, f' \in H_\Sigma$ such that $\pi(f) = \pi(f')$. This is a necessary condition for $\mathfrak{F}$ to be 2-distributive.

To prove that this is also sufficient, consider another alphabet Σ' and a morphism $\psi : \Sigma'^\Delta \to (H, V)$. We can build a morphism $\eta : \Sigma'^\Delta \to \Sigma^\Delta$, generated by the map $\Sigma' \to \Sigma$ defined by $\eta(\alpha) := \psi(\alpha) \in \Sigma$ for $\alpha \in \Sigma'$. Then $\psi = \phi \circ \eta$. Let $f_1, f_2 \in H_{\Sigma'}$ with $\pi(f_1) = \pi(f_2)$, and let $c \in V_{\Sigma'}$. Then $\pi(\eta(f_1)) = \pi(\eta(f_2))$, and, by assumption, $\psi(c[f_1 + f_2]) = \phi(\eta(c[f_1 + f_2])) = \phi(\eta(c)[\eta(f_1) + \eta(f_2)]) = \phi(\eta(c)\eta(f_1) + \eta(c)\eta(f_2)) = \phi(\eta(cf_1 + cf_2)) = \psi(cf_1 + cf_2)$. □

8 Discussion

We have shown that 2-distributive finite forest algebras recognize a subclass of PDL, and that 2-distributivity is a decidable property.

As we outlined in the Introduction, generalizing this approach to $k > 2$ would settle decidability of PDL. Our notion of 2-distributivity can be generalized in the following way, slightly different than the one given in [17]:

Definition 24. For each $k \geq 1$, define a congruence $\sim_k$ on $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$ as follows:

1. $\sim_1$ is the smallest congruence such that $f \sim_1 f'$ whenever $f, f' \in H_\Sigma$ and $\pi(f) = \pi(f')$.
2. For any $k \geq 1$, $\sim_{k+1}$ is the smallest congruence such that
$$v[f + f'] \sim_{k+1} vf + vf'$$
for any $v \in V_\Sigma$ and any $f, f' \in H_\Sigma$ such that $f \sim_k f'$.

For each k, the congruence $\sim_k$ encodes a k-fold iteration of the distributive law. A forest algebra $\mathfrak{F}$ is *k-distributive* if, for all morphisms $\phi : \Sigma^{\Delta} \to \mathfrak{F}$, $\phi(f) = \phi(f')$ whenever $f \sim_k f'$. For $k = 2$, this coincides with our definition above.

In analogy to Proposition 9 and a result from [17], it can be shown that the wreath product of k distributive forest algebras is k-distributive. Determining, given a finite forest algebra $\mathfrak{F}$, whether it is k-distributive for some k is a decidable problem. If one could show that any finite k-distributive forest algebra only recognizes languages in PDL, definability in PDL would therefore be shown decidable [17]. We have solved this problem in the case $k = 2$.

Indeed, generalizing our Local-Global Theorem to $k \geq 2$ is feasible, and the proof method of our main result might be adapted to construct an inductive proof. It would be sufficient to, given a general k-distributive algebra, construct a wreath product of finite distributive algebras and show that an appropriate derived forest algebra is $k - 1$-distributive. To carry this out, a suitable strengthening of our Separation Lemma to a property stronger than separation would be required.

PDL is a member of a larger family of forest languages for which decidability of expressibility is still unknown, in spite of longstanding interest and several attempts [14, 20]. For a range of tree and forest logics, decidable characterizations have been obtained (see [3] for a survey up to 2008; more recent results include [2, 4, 5, 11, 12] among others). However, for many more prominent logics, including First-Order Logic with ancestor, CTL, CTL^*, PDL, and Chain Logic, this problem remains open. As described in the introduction, PDL extends both CTL and CTL^*. All of these logics were shown by [6] to correspond to iterated wreath products of specific types of forest algebras satisfying a distributive law. Among these, PDL stands out because it is characterized through products of arbitrary distributive forest algebras, and can – at least at the level $k = 2$ – be captured in terms of a k-fold iterated distributive law. Therefore, our results might also shed light on this larger family of open problems.

In the field of regular word languages and logic on words, the study of finite monoids has been tremendously successful. Our proof strategy highlights how the classical theory developed for studying logic on words via wreath products of monoids carries over faithfully to the setting of forest algebras: Our proof proceeds by solving two sub-problems related to the left and right factors in wreath product decompositions: a separation result and a Local-Global theorem, which are then combined via the Derived Category Theorem [21]. These steps are remarkably similar to results from the theory of logic on words and finite monoids which also reduce the problem of decidability to separation [13] and Local-Global theorems [9].

References

[1] Jorge Almeida. 1996. A syntactical proof of locality of DA. *International Journal of Algebra and Computation* 06, 02 (April 1996), 165–177.

[2] Michael Benedikt and Luc Segoufin. 2009. Regular tree languages definable in FO and in FO_*mod*. *ACM Trans. Comput. Log.* 11, 1 (Oct. 2009), 1–32.

[3] Mikolaj Bojańczyk. 2008. Effective characterizations of tree logics. In *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, Maurizio Lenzerini and Domenico Lembo (Eds.). ACM, 53–66.

[4] Mikolaj Bojanczyk and Luc Segoufin. 2010. Tree Languages Defined in First-Order Logic with One Quantifier Alternation. *Logical Methods in Computer Science* 6, 4 (2010).

[5] Mikolaj Bojańczyk, Luc Segoufin, and Howard Straubing. 2012. Piecewise testable tree languages. *Logical Methods in Computer Science* 8, 3 (2012), 442–451.

[6] Mikolaj Bojańczyk, Howard Straubing, and Igor Walukiewicz. 2012. Wreath Products of Forest Algebras, with Applications to Tree Logics. *Logical Methods in Computer Science* 8, 3 (2012).

[7] J. A. Brzozowski and Imre Simon. 1973. Characterizations of locally testable events. *Discrete Mathematics* 4, 3 (March 1973), 243–271.

[8] S. Eilenberg and B. Tilson. 1976. *Automata, languages, and machines.* Academic Pr, New York.

[9] Andreas Krebs and Howard Straubing. 2012. An effective characterization of the alternation hierarchy in two-variable logic. In *FSTTCS*. 86–98.

[10] Jean-Éric Pin, Howard Straubing, and Denis Thérien. 1988. Locally trivial categories and unambiguous concatenation. *Journal of Pure and Applied Algebra* 52, 3 (1988), 297–311.

[11] Thomas Place and Luc Segoufin. 2009. A Decidable Characterization of Locally Testable Tree Languages. In *Automata, Languages and Programming, 36th Internatilonal Collogquium, ICALP 2009, Rhodes, greece, July 5-12, 2009, Proceedings, Part II (Lecture Notes in Computer Science)*, Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E Nikoletseas, and Wolfgang Thomas (Eds.), Vol. 5556. Springer, 285–296.

[12] Thomas Place and Luc Segoufin. 2010. Deciding Definability in $FO_2(<_h, <_v)$ on Trees. In *2010 25th Annual IEEE Symposium on Logic in Computer Science*. IEEE, 253–262.

[13] Thomas Place and Marc Zeitoun. 2014. Going Higher in the First-Order Quantifier Alternation Hierarchy on Words. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II (Lecture Notes in Computer Science)*, Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias (Eds.), Vol. 8573. Springer, 342–353.

[14] Andreas Potthoff. 1995. First-Order Logic on Finite Trees. In *TAPSOFT'95: Theory and Practice of Software Development, 6th International Joint Conference CAAP/FASE, Aarhus, Denmark, May 22-26, 1995, Proceedings (Lecture Notes in Computer Science)*, Peter D Mosses, Mogens Nielsen, and Michael I Schwartzbach (Eds.), Vol. 915. Springer, 125–139.

[15] John Rhodes and Benjamin Steinberg. 2009. *The q-theory of finite semigroups.* Springer Science & Business Media.

[16] Howard Straubing. 1985. Finite semigroup varieties of the form V * D. *Journal of Pure and Applied Algebra* 36 (1985), 53–94.

[17] Howard Straubing. 2013. New applications of the wreath product of forest algebras. *RAIRO - Theor. Inf. and Applic.* 47, 3 (2013), 261–291.

[18] Howard Straubing. 2018. Forest Categories. *arXiv:1801.04337 [cs]* (Jan. 2018). arXiv: 1801.04337.

[19] Denis Thérien and Alex Weiss. 1985. Graph congruences and wreath products. *Journal of Pure and Applied Algebra* 36 (Jan. 1985), 205–215.

[20] W. Thomas. 1984. Logical aspects in the study of tree languages. In *Ninth Colloquium on Trees in Algebra and in Programming (Proc. CAAP 84)*, Courcelle and B. (Eds.). Cambridge University Press, 31–51.

[21] Bret Tilson. 1987. Categories as algebra: An essential ingredient in the theory of monoids. *Journal of Pure and Applied Algebra* 48, 1-2 (Sept. 1987), 83–198.

*

A General Definitions and Lemmas

Definition 25. [Pathsets: π, H_Σ^D, Π] Recall the map $\pi : H_\Sigma \to Pow(\Sigma^*)$ mapping forests to their path sets. The image of π is called H_Σ^D. That is, H_Σ^D is the set of nonempty finite subsets of Σ^* that are closed under prefixes ($wv \in X \Rightarrow w \in X$).

We extend π to an operation on forest languages:

$$\pi(\mathcal{L}) := \{\pi(f) : f \in \mathcal{L}\} \subset Pow(\Sigma^*)$$

for $\mathcal{L} \subseteq H_\Sigma$.

When $\mathcal{L} \subseteq H_\Sigma$, then $\Pi(\mathcal{L}) \subseteq \Sigma^*$ is the set of all paths occurring in some forest in $\mathcal{L}$:

$$\Pi(\mathcal{L}) := \bigcup_{f \in \mathcal{L}} \pi(f)$$

Since Π is defined in terms of π, Π is also well-defined on subsets of H_Σ^D: If $X \subseteq H_\Sigma^D$, then $\Pi(X) := \bigcup_{\phi \in X} \phi$.

Proposition 26. Let $\mathcal{L} \subset H_\Sigma$ be a regular forest language. Then $\Pi(\mathcal{L}) \subset \Sigma^*$ is a regular word language, and can be effectively constructed from (an automaton for) $\mathcal{L}$.

Proof. Let $\mathfrak{F} = (H, V)$ be a finite forest algebra recognizing $\mathcal{L}$ via morphism ϕ. We want to construct a finite automaton recognizing the word language $\Pi(\mathcal{L})$. Let $Q := Pow(V)$ be the state set of this automaton, and consider the transition function $Q \times \Sigma \to Q$ given by

$$(\{v_1, ..., v_k\}, \alpha) \mapsto \{v : \exists j = 1, ..., k, \exists h \in H : v = v_j \cdot \phi(\alpha) \cdot I_h\}$$

where by $\phi(\alpha)$ we refer to the image of the context consisting of only α and a variable below it ($\alpha[X]$), and $I_h \in V$ is the insertion operation defined in Definition 1. We take the starting state to be $\{1_V\} \in Q$, and the accepting states to be all nonempty subsets of $\{v : \phi^{-1}(v \cdot 0_H) \subset \mathcal{L}\}$. By induction over the length of words, it is shown that this automaton exactly accepts the language $\Pi(\mathcal{L})$. □

For reference, we formulate the full version of the Derived Category Theorem. It is not required for our main result, but will be used in Appendix F for the proof of Example 12.

Theorem 27. [Full Version of the Derived Category Theorem, [18]] Let Σ be an alphabet, and let α, β be morphisms from Σ^Δ onto forest algebras (H_1, V_1), (H_2, V_2), respectively. Let (H, V) be a forest algebra.

(a) Assume $D_{\alpha,\beta} \prec (H, V)$. Then

$$(H_1, V_1) \prec (H, V) \wr (H_2, V_2)$$

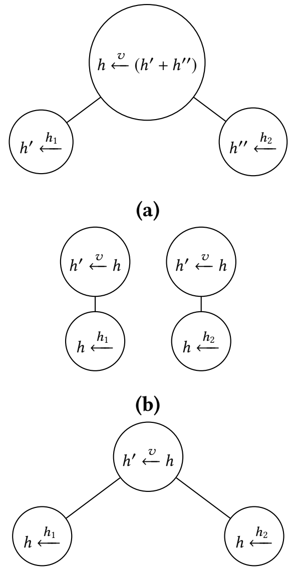
(b) Suppose α factors as

$$\alpha = \gamma\delta : \Sigma^\Delta \to (H_1, V_1)$$

where

$$\delta : \Sigma^\Delta \to (H, V) \wr (H_2, V_2)$$
$$\gamma : \text{Im } \delta \to (H_1, V_1)$$

and that $\beta = \pi^{(2)}\delta$, where $\pi^{(2)}$ is the projection homomorphism from the wreath product onto its right-hand factor. Then $D_{\alpha,\beta} \prec (H, V)$.



(a)

(b)

(c)

Figure 4. Examples of forest diagrams. In any locally distributive derived forest category, the diagrams in (b) and (c) result in the same value under the map Val.

B Proving the Local-Global Theorem

We will need the notion of *forest diagrams*:

Definition 28. Given a derived forest category C, a *forest diagram* is a forest whose leaves are labeled with half-arrows of C and whose internal nodes are labeled with arrows of C, subject to the following consistency condition: Let n be a (non-leaf) node labeled with an arrow $h_2 \xleftarrow{v} h_1$, and let $h_3, ..., h_k \in \text{Obj}(C)$ be the endpoints of the arrows or half-arrows that label the root nodes of the children of n. Then $h_3 + ... + h_k = h_1$.

Examples are shown in Figure 4.

To any forest diagram one can assign a half-arrow of C by recursively adding up the half-arrows assigned to siblings, and multiplying out the action of arrows on the half-arrows assigned to their children. Let Val be the map assigning these values to forest-diagrams.

One could think of forest diagrams as forming a 'free forest category' – in analogy to Σ^Δ, and Val being a 'forest category morphism' to C, but we will not develop this perspective here.

Assume that C is a finite and locally-distributive derived forest category. Let d be a forest diagram over C. Let us assume that d contains two sibling nodes labeled with the same arrow $h' \xleftarrow{v} h$ (see Figure 4b). Let d_1, d_2 be the forest diagrams below these two nodes. The value of the sum of these two sibling nodes is

$$\left(h' \xleftarrow{v} h\right) \cdot \text{Val}(d_1) + \left(h' \xleftarrow{v} h\right) \cdot \text{Val}(d_2)$$

Since C is locally distributive, we know that this is equal to

$$\left(h' \xleftarrow{v} h\right) \cdot \text{Val}(d_1 + d_2)$$

Therefore, if we replace the two sibling nodes by a single node and put $d_1 + d_2$ below it (see Figure 4c), we obtain a new forest diagram which evaluates to the same half-arrow in C as d did.

If we apply this procedure iteratively, we will, after finitely many steps, arrive at a forest diagram $\widehat{d}$ where no two sibling nodes are labeled by the same half-arrow, and for which $\mathrm{Val}(\widehat{d}) = \mathrm{Val}(d)$. Indeed, taking Ψ from Definition 20, the result $\widehat{d}$ is equal to $\Psi(d)$.

Let Σ_C be the alphabet consisting of all arrows and half-arrows of C. Then each forest-diagram is a forest in H_{Σ_C}.

With this view, $\widehat{d}$ and d have the same pathsets: $\pi(\widehat{d}) = \pi(d)$ (Proposition 21.2). Furthermore, it is not hard to see that for any other forest diagram d' with $\pi(d') = \pi(d)$, we would have arrived at the same modified diagram $\widehat{d}$ (Proposition 21.3). This implies that $\mathrm{Val}(d)$ is determined by $\pi(d)$.

Having established this, we can prove the theorem:

Proof of the Local-Global Theorem 19. Assume that C is a finite and locally-distributive derived forest category.

Let $h \overset{c}{\leftarrow}$, $h' \overset{c'}{\longleftarrow}$ be distinct half-arrows in C. By the previous argument,

$$\pi(\mathrm{Val}^{-1}(h \overset{c}{\leftarrow})) \cap \pi(\mathrm{Val}^{-1}(h' \overset{c'}{\longleftarrow})) = \emptyset$$

Certainly, the sum of the two half-arrows has to be distinct from at least one of them: At least, $h \overset{c}{\leftarrow} +h' \overset{c'}{\longleftarrow} \neq h \overset{c}{\leftarrow}$ or $h \overset{c}{\leftarrow} +h' \overset{c'}{\longleftarrow} \neq h' \overset{c'}{\longleftarrow}$. We'll assume that the first one holds. Then, we know $\pi(\mathrm{Val}^{-1}(h \overset{c}{\leftarrow} +h' \overset{c'}{\longleftarrow})) \cap \pi(\mathrm{Val}^{-1}(h \overset{c}{\leftarrow})) = \emptyset$

Recall the operation Π from Definition 25: For any forest language $\mathcal{L}$, we define $\Pi(\mathcal{L}) \subseteq (\Sigma_C)^*$ as the set of all paths occurring in some forest in $\mathcal{L}$:

$$\Pi(\mathcal{L}) := \bigcup_{f \in \mathcal{L}} \pi(f)$$

Now consider the sets $\Pi_1 := \Pi(\mathrm{Val}^{-1}(h \overset{c}{\leftarrow}))$ and $\Pi_2 := \Pi(\mathrm{Val}^{-1}(h' \overset{c'}{\longleftarrow}))$. Assume that there is $d_1 \in \mathrm{Val}^{-1}(h \overset{c}{\leftarrow})$ such that $\pi(d_1) \subseteq \Pi_2$. Then we can find $d_3 \in \mathrm{Val}^{-1}(h' \overset{c'}{\longleftarrow})$ such that $\pi(d_1) \subseteq \pi(d_3)$. Observe $\pi(d_1) = \pi(d_3 + d_1)$ and $d_3 + d_1 \in \mathrm{Val}^{-1}(h \overset{c}{\leftarrow} +h' \overset{c'}{\longleftarrow})$. This means $\pi(d_1) \in \pi(\mathrm{Val}^{-1}(h \overset{c}{\leftarrow} +h' \overset{c'}{\longleftarrow})) \cap \pi(\mathrm{Val}^{-1}(h \overset{c}{\leftarrow}))$, contradiction.

This means that there is no $d_1 \in \mathrm{Val}^{-1}(h \overset{c}{\leftarrow})$ such that $\pi(d_1) \subseteq \Pi_2$. Said differently, for any $d_1 \in \mathrm{Val}^{-1}(h \overset{c}{\leftarrow})$, we have $\pi(d_1) \cap (\Pi_1 - \Pi_2) \neq \emptyset$.

This means that, in order to separate forest diagrams evaluating to $h \overset{c}{\leftarrow}$ from those evaluating to $h' \overset{c'}{\longleftarrow}$, it is sufficient to check whether the pathset of the diagram in question contains an element of $\Pi_1 - \Pi_2$.

We want to show that $\Pi_1, \Pi_2 \subset \Sigma_C^*$ are regular word languages over the finite alphabet Σ_C. Being a finite derived forest category, C is equal to $D_{\phi,\psi}$ for some forest algebra morphisms ϕ, ψ into finite forest algebras. Thus, $\mathrm{Val}^{-1}(h \overset{c}{\leftarrow})$ and $\mathrm{Val}^{-1}(h' \overset{c'}{\longleftarrow})$ are recognized by combining a morphism into the (finite) image of ϕ that disregards the start- and end-points of (half-)arrows and otherwise behaves like ϕ, with a regular language that checks whether the start- and end-points of the node labels are locally consistent. By Lemma 26, it follows that Π_1 and Π_2 are regular word languages over Σ_C.

Thus, by Proposition 6, there is a finite distributive forest algebra $\mathfrak{F}$ and a morphism $\eta : (\Sigma_C)^\Delta \to \mathfrak{F}$ which recognizes the set of forests whose path set intersects $\Pi_1 - \Pi_2$. While we cannot hope that this morphism recognizes the set of such forest diagrams, it certainly separates the class of diagrams whose path sets intersect $\Pi_1 - \Pi_2$ from those for which this doesn't hold.

For each pair of half-arrows $h \overset{c}{\leftarrow}$, $h' \overset{c'}{\longleftarrow}$ in C, we obtain a finite distributive forest algebra $\mathfrak{F}_{h \overset{c}{\leftarrow}, h' \overset{c'}{\longleftarrow}}$ and a morphism $\eta_{h \overset{c}{\leftarrow}, h' \overset{c'}{\longleftarrow}}$ in this manner. We then let $\widehat{\mathfrak{F}}$ be the direct product of these finitely many algebras, and $\widehat{\eta}$ the direct product of these morphisms. Certainly, $\widehat{\mathfrak{F}}$ is finite and distributive. To construct a division $C \prec \widehat{\mathfrak{F}}$ (recall Definition 14), we assign to each half-arrow $h \overset{c}{\leftarrow}$ in C the set of half-arrows in $\widehat{\mathfrak{F}}$ that are in the image of $\mathrm{Val}^{-1}(h \overset{c}{\leftarrow})$ under $\widehat{\eta}$, and similarly for arrows. One can verify that these assignments preserve operations as required in Definition 14. More importantly, the previous arguments guarantee that they satisfy the injectivity condition required in Definition 14:

For a contradiction, let $h \overset{c}{\leftarrow}$, $h' \overset{c'}{\longleftarrow}$ be distinct half-arrows in C that are mapped to overlapping sets in $H_{\widehat{\mathfrak{F}}}$: $g \in K_{h \overset{c}{\leftarrow}} \cap K_{h' \overset{c'}{\longleftarrow}}$ This means that there are forest diagrams f_1, f_2 such that $\mathrm{Val}(f_1) = K_{h \overset{c}{\leftarrow}}$, $\mathrm{Val}(f_2) = K_{h' \overset{c'}{\longleftarrow}}$, but $\widehat{\eta}(f_1) = \widehat{\eta}(f_2) = g$. In particular, $\eta_{h \overset{c}{\leftarrow}, h' \overset{c'}{\longleftarrow}}(f_1) = \eta_{h \overset{c}{\leftarrow}, h' \overset{c'}{\longleftarrow}}(f_2)$. By the choice of $\eta_{h \overset{c}{\leftarrow}, h' \overset{c'}{\longleftarrow}}$, this is impossible. The argument for arrows is similar.

In conclusion, we have shown that the assignment is injective and C divides the finite distributive forest algebra $\widehat{\mathfrak{F}}$.

□

C Proving the Separation Lemma

Recall from the beginning that trees and forests are horizontally commutative and idempotent. Informally, order and multiplicity of children in a tree and of trees in a forest don't matter. To fix notation, we denote a tree with root symbol α and children set $C = \{t_1, ..., t_n\}$ as $\alpha[C]$.

We are given regular forest languages $\mathcal{L}_1, \mathcal{L}_2$. We want to show that, if the images of $\mathcal{L}_1, \mathcal{L}_2$ under π are disjoint, then some language recognized by a wreath product of three finite distributive algebras separates $\mathcal{L}_1, \mathcal{L}_2$.

We will prove a stronger statement to 'load the induction hypothesis'. This stronger statement will be the Main Lemma 30. In order to state and prove it, a few more notions are needed.

C.1 Preliminaries

We will need a notion of *rules* that is analogous to transitions in bottom-up tree automata:

Definition 29 (Rules). Let $\mathfrak{F} = (H, V)$ be a forest algebra, and $\phi : \Sigma^\Delta \to (H, V)$ a morphism.

A *rule* is a tuple

$$(h_0, \alpha, \{h_1, ..., h_k\}) \in H \times \Sigma \times Pow(H)$$

such that

$$h_0 = \phi(\alpha)[h_1 + ... + h_k]$$

The set of rules for $\mathfrak{F} = (H, V)$ with morphism ϕ is called $Rules(\mathfrak{F}, \phi)$.

For $\rho \in Rules(\mathfrak{F})$, let $\mathcal{L}(\rho)$ be the set of nonempty forests where all top-level rule applications are of rule $\rho = (h_0, \alpha, \{h_1, ..., h_k\})$. Formally, a nonempty forest f belongs to $\mathcal{L}(\rho)$ iff, for any tree $t \in f$, (1) the root symbol of t is α, and (2) if C is the set of trees that are children of the root node of t, then $\{\phi(c) : c \in C\} = \{h_1, ..., h_k\}$.

For convenience, we will use the same notation for elements of H: $\mathcal{L}(h) := \phi^{-1}(h)$ for $h \in H$.

C.2 Main Lemma and Proof of Separation Lemma

Lemma 30 (Main Lemma). Let $\mathfrak{F} = (H, V)$ be a finite forest algebra, and $\phi : \Sigma^\Delta \to (H, V)$ a morphism. There is a map $\mathcal{P}$ assigning to each $\rho \in Rules(\mathfrak{F})$ a language $\mathcal{P}(\rho) \subseteq H_\Sigma$ such that the following four statements hold:

(A) For any $\rho \in Rules(\mathfrak{F}, \phi)$, we have

$$\mathcal{L}(\rho) \subseteq \mathcal{P}(\rho)$$

(B) For all $\rho, \rho' \in Rules(\mathfrak{F}, \phi)$, $\mathcal{P}(\rho) \cap \mathcal{P}(\rho')$ is empty if and only if $\pi\mathcal{L}(\rho) \cap \pi\mathcal{L}(\rho')$ is empty.

(C) For any $\rho_1, ...\rho_k \in Rules(\mathfrak{F}, \phi)$, the language

$$\{f = f_1 + ... + f_k : f_i \in \mathcal{P}(\rho_i)\}$$

is recognized by a wreath product of three finite distributive forest algebras.

(D) For any $\rho \in Rules(\mathfrak{F}, \phi)$ and any forest $f = \{t_1, ..., t_l\}$, where each t_i is a tree, we have $f \in \mathcal{P}(\rho)$ if and only if all the singleton forests $\{t_i\}$ $(i = 1, ..., l)$ are in $\mathcal{P}(\rho)$.

Each $\mathcal{P}(\rho)$ can be seen as a PDL approximator of the language $\mathcal{L}(\rho)$. While (A) guarantees that these approximators are sufficiently 'large', (B) guarantees that they suffice to construct a PDL separator whenever $\mathcal{L}(\rho)$ and $\mathcal{L}(\rho')$ are separable by π at all. We'll prove the Main Lemma below in Appendix D. Using the Main Lemma, we prove the Separation Lemma:

Proof of the Separation Lemma 17. Since $\mathcal{L}_1, \mathcal{L}_2$ are regular, there is a finite forest algebra $\mathfrak{F} = (H, V)$ which recognizes both $\mathcal{L}_1$ and $\mathcal{L}_2$ via a morphism $\phi : \Sigma^\Delta \to \mathfrak{F}$.

Any forest f in H_Σ can be uniquely written as a sum of forests from languages $\mathcal{L}(\rho)$, with $\rho \in Rules(\mathfrak{F}, \phi)$. If, for each forest f, we use R_f to denote the collection of these ρ's, we can assign to each $h \in H$ a set

$$Q_h := \{R_f : f \in \phi^{-1}(h)\}$$

We can assume that ϕ is onto, so this set is always nonempty. Each element of this set is a subset of the finite set $Rules(\mathfrak{F}, \phi)$. Thus, Q_h itself is finite. Given the choice of Q_h, we can write

$$\phi^{-1}(h) := \bigcup_{q \in Q_h} \sum_{\rho \in q} \mathcal{L}(\rho)$$

Using the map $\mathcal{P}$ from the Main Lemma, we then define

$$\mathcal{P}_h := \bigcup_{q \in Q_h} \sum_{\rho \in q} \mathcal{P}(\rho)$$

By condition (C) of the Main Lemma, each $\sum_{\rho \in q} \mathcal{P}(\rho)$ is recognized by a wreath product of three finite distributive algebras. Since Q_h is finite, $\mathcal{P}_h$ is also recognized by a wreath product of three finite distributive algebras.

Furthermore, let $\mathfrak{G}$ be a wreath product of three finite distributive algebras that recognizes *all* languages $\mathcal{P}_h$ via a morphism $\phi_\mathfrak{G}$. Again, this is possible since H is finite.

Let $h_1, h_2 \in H$. Assume $\pi(\phi^{-1}(h_1)) \cap \pi(\phi^{-1}(h_2)) = \emptyset$. Then for all $q \in Q_{h_1}, q' \in Q_{h_2}$, we have

$$\pi\left(\sum_{\rho \in q} \mathcal{L}(\rho)\right) \cap \pi\left(\sum_{\rho \in q'} \mathcal{L}(\rho)\right) = \emptyset$$

Hence, there is $\rho_0 \in q$ such that $\pi(\mathcal{L}(\rho_0)) \cap \pi(\mathcal{L}(\rho')) = \emptyset$ for all $\rho' \in q'$, or the same with q, q' reversed.

> *Proof of this.* Assume for each $\rho \in q$, there is $\rho' \in q'$ such that $\pi(\mathcal{L}(\rho)) \cap \pi(\mathcal{L}(\rho')) \neq \emptyset$, and the same with q, q' reversed. First, for $\rho \in q$, let $\phi_\rho \in \pi(\mathcal{L}(\rho)) \cap \pi(\mathcal{L}(\rho'))$. Then, for $\rho \in q'$, let $\phi'_\rho \in \pi(\mathcal{L}(\rho)) \cap \pi(\mathcal{L}(\rho'))$. Then define $\widehat{\phi} := \sum_{\rho \in q} \phi_s + \sum_{\rho \in q'} \phi'_\rho$. By definition, $\widehat{\phi} \in \pi\left(\sum_{\rho \in q} \mathcal{L}(\rho)\right) \cap \pi\left(\sum_{\rho \in q'} \mathcal{L}(\rho)\right)$. □

From condition (B) of the Main Lemma, we can deduce $\mathcal{P}(\rho_0) \cap \mathcal{P}(\rho') = \emptyset$ for all $\rho' \in q'$ (or q, depending on where ρ_0 came from).

From this, we want to deduce

$$\left(\sum_{\rho \in q} \mathcal{P}_\rho\right) \cap \left(\sum_{\rho \in q'} \mathcal{P}_\rho\right) = \emptyset$$

This follows from condition (D) of the Main Lemma: If this intersection were nonempty and contained a forest f, this forest would contain a tree t such that $\{t\} \in \mathcal{P}(\rho_0)$ and $\{t\} \in \mathcal{P}(\rho')$ for some $\rho' \in q'$ – a contradiction.

Since q, q' were chosen arbitrarily,

$$\left(\bigcup_{q \in Q_{h_1}} \sum_{\rho \in q} \mathcal{P}_\rho\right) \cap \left(\bigcup_{q \in Q_{h_2}} \sum_{\rho \in q} \mathcal{P}_\rho\right) = \emptyset$$

which means that $\mathcal{P}_{h_1} \cap \mathcal{P}_{h_2} = \emptyset$.

Now consider

$$\mathcal{P}_1 := \bigcup_{h \in \phi(\mathcal{L}_1)} \mathcal{P}_h$$

$$\mathcal{P}_2 := \bigcup_{h \in \phi(\mathcal{L}_2)} \mathcal{P}_h$$

These languages are recognized by $\mathfrak{G}$ with morphism $\phi_{\mathfrak{G}}$. Now given $\pi\mathcal{L}_1 \cap \pi\mathcal{L}_2 = \emptyset$, we have $\pi\phi^{-1}(h) \cap \pi\phi^{-1}(h')$ for all $h \in \phi(\mathcal{L}_1)$, $h' \in \phi(\mathcal{L}_2)$. By our previous reasoning, we get $\mathcal{P}_h \cap \mathcal{P}_{h'} = \emptyset$ for any such h, h'. Putting this together, we get $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$, and thus $\mathcal{P}_1 \subseteq (H_\Sigma - \mathcal{P}_2)$.

Since $\mathfrak{F}$ recognizes $\mathcal{L}_1, \mathcal{L}_2$ via the morphism ϕ, we can write

$$\mathcal{L}_1 = \bigcup_{h \in \phi(\mathcal{L}_1)} \phi^{-1}(h)$$

$$\mathcal{L}_2 = \bigcup_{h \in \phi(\mathcal{L}_2)} \phi^{-1}(h)$$

By condition (A) of the Main Lemma, $\phi^{-1}(h) \subseteq \mathcal{P}_h$ holds for any $h \in H$. Given the definition of $\mathcal{P}_1, \mathcal{P}_2$, we can concude $\mathcal{L}_1 \subseteq \mathcal{P}_1$ and $\mathcal{L}_2 \subseteq \mathcal{P}_2$.

Putting these results together, we find that

$$\mathcal{L}_1 \subseteq \mathcal{P}_1 \subseteq (H_\Sigma - \mathcal{P}_2) \subseteq (H_\Sigma - \mathcal{L}_2)$$

We can thus take $X := \mathcal{P}_1$, completing the proof of the Separation Lemma 17. □

D Proving the Main Lemma

The goal in this section is to prove the Main Lemma 30.

For the proof, we will utilize the following bit of notation. Recall the set H_Σ^D from Definition 25.

Definition 31. Let $\mathfrak{F} = (H, V)$. Let $R, S \subseteq \mathit{Rules}(\mathfrak{F}, \phi)$ or $R, S \subseteq H$. Then $\langle R, S \rangle$ is a subset of H_Σ^D defined as follows:

$d \in \langle R, S \rangle$ if and only if there are forests f_1, f_2 such that $\pi(f_1) = \pi(f_2) = d$, and there are $r_1, \ldots, r_n \in R$, $s_1, \ldots, s_k \in S$ ($n, k > 0$) and

$$f_1 \in \mathcal{L}(r_1) + \ldots + \mathcal{L}(r_n)$$

$$f_2 \in \mathcal{L}(s_1) + \ldots + \mathcal{L}(s_k)$$

Recall the operation Π mapping forest languages to subsets of Σ^*. We will also apply it to subsets of H_Σ^D: For $X \subseteq H_\Sigma^D$, set $\Pi(X) := \bigcup_{x \in X} x$.

Definition 32. The *height* of a forest is the length of the longest paths. Formally, we define $\text{height}(\alpha[C]) := 1 + \max_{t \in C} \text{height}(t)$, with $\max \emptyset := 0$. If f is a forest, we set $\text{height}(f) := \max_{t \in f} \text{height}(t)$, with $\max \emptyset := 0$.

We will frequently use:

$$Z_n(\mathcal{L}) := \{f \in \mathcal{L} : \text{height} f \leq n\}$$

Since the height of a forest only depends on its pathset and the elements of H_Σ^D are finite sets, we can write $Z_n X$ even when $X \subseteq H_\Sigma^D$.

The goal is to prove the Main Lemma 30. We prove the following stronger version. Recall the operation Π from Definition 25, and $Z_n\mathcal{L}$ from Definition 32.

Lemma 33 (Stronger Version of Main Lemma)**.** Let $\mathfrak{F} = (H, V)$ be a finite forest algebra, and $\phi : \Sigma^\Delta \to (H, V)$ a morphism. There is a map $\mathcal{P}$ assigning to each $R \subseteq \mathit{Rules}(\mathfrak{F})$ a language $\mathcal{P}(R) \subseteq H_\Sigma$ such that the following four statements hold:

(A) For any $R \subseteq \mathit{Rules}(\mathfrak{F}, \phi)$, and for any $\rho_1, \ldots, \rho_k \in R$ ($r \geq 1$), we have

$$\mathcal{L}(\rho_1) + \ldots + \mathcal{L}(\rho_k) \subseteq \mathcal{P}(R)$$

That is, $\mathcal{P}(R)$ extends the languages obtained by adding forests that evaluate to rules in R.

(B) The following relationship between $\mathcal{P}(\cdot)$ and the construct $\langle \cdot, \cdot \rangle$ holds for all $R, S \subseteq \mathit{Rules}(\mathfrak{F}, \phi)$, and $n \in \mathbb{N}$:

$$\Pi\left(Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S)\right) = \Pi Z_n \langle R, S \rangle$$

In words: The following two operations yield the same sets of paths:

1. Intersect the languages $\mathcal{P}(R), \mathcal{P}(S)$, restricted to forests of height $\leq n$. Then compute the set of paths occurring in the resulting forest language.
2. Compute the set $\langle R, S \rangle \subseteq H_\Sigma^D$, and restrict to pathsets of height $\leq n$. Then compute the set of paths, that is, the union over the resulting subset of H_Σ^D.

(C) For any $\rho_1, \ldots \rho_k \in \mathit{Rules}(\mathfrak{F}, \phi)$, the language

$$\{f = f_1 + \ldots + f_k : f_i \in \mathcal{P}(\{\rho_i\})\}$$

is recognized by a wreath product of three finite distributive forest algebras.

(D) For any $R \subseteq \mathit{Rules}(\mathfrak{F}, \phi)$ and any forest $f = \{t_1, \ldots, t_l\}$ (each t_i being a tree), we have $f \in \mathcal{P}(R)$ if and only if all the singleton forests $\{t_i\}$ ($i = 1, \ldots, l$) are in $\mathcal{P}(R)$.

To obtain Lemma 30, we take $\mathcal{P}(\rho)$ from that version to be the map $\mathcal{P}$ given here applied to the singleton $\{\rho\}$. Conditions (A), (C), and (D) immediately follow. For condition (B), note that $\mathcal{P}(\{\rho\}) \cap \mathcal{P}(\{\rho'\})$ is empty if and only if $Z_n\mathcal{P}(\{\rho\}) \cap Z_n\mathcal{P}(\{\rho'\})$ is empty for all n. Now, since even empty forests have a nonempty pathset (consisting of the empty path), $Z_n\mathcal{P}(\{\rho\}) \cap Z_n\mathcal{P}(\{\rho'\})$ is empty if and only if $\Pi(Z_n\mathcal{P}(\{\rho\}) \cap Z_n\mathcal{P}(\{\rho'\}))$ is. Now, by (B) from Lemma 33, this is empty if and only if $\Pi Z_n \langle \{\rho\}, \{\rho'\}\rangle$ is empty. By definition of $\langle\cdot,\cdot\rangle$, the term $\langle \{\rho\}, \{\rho'\}\rangle$ is actually equal to $\pi\mathcal{L}(\rho) \cap \pi\mathcal{L}(\rho')$. Reversing the previous arguments from this paragraph, $\Pi Z_n(\pi\mathcal{L}(\rho) \cap \pi\mathcal{L}(\rho'))$ is empty for all n if and only if $\pi\mathcal{L}(\rho) \cap \pi\mathcal{L}(\rho')$ is empty. We have proven Condition (B) from Lemma 30. This shows that Lemma 30 follows once Lemma 33 is proven.

We first construct the map $\mathcal{P}$ and show (A), (C), and (D). Proving (B) will be a bigger task. We split this up into three sections: (1) the right-to-left inclusion, (2) the left-to-right inclusion in the case of small heights $n \leq N$, (3) the left-to-right inclusion on the case of large heights $n > N$.

D.1 Distributive Approximators

We first define a family of languages recognized by finite distributive forest algebras, acting as the right-most factors of the wreath products we want to build. For each rule ρ, the language $\Delta(\rho)$ will be a distributive approximation of $\mathcal{L}(\rho)$, and should at least contain the trees belonging to $\mathcal{L}(\rho)$. Intuitively, $\Delta(\rho)$ should be the smallest language extending $\mathcal{L}(\rho)$ that is recognized by a finite distributive forest algebra. Such a language would have to be a superset of $\pi^{-1}\pi\mathcal{L}(\rho)$. However, such a smallest language does not in general exist. For any fixed height limit N, we can find a finite distributive forest algebra which agrees with $\pi^{-1}\pi(\mathcal{L}(\rho))$ for forests of height up to N, but in general, no finite forest algebra can recognize $\pi^{-1}\pi(\mathcal{L}(\rho))$. Our strategy will be to specify a height limit N up to which $\Delta(\rho)$ will (in a certain sense) agree with $\pi^{-1}\pi(\mathcal{L}(\rho))$. This height limit is specified by the following lemma:

Lemma 34 (Constant). There is $N \in \mathbb{N}$ such that, for any choice of $s \in H_{\mathfrak{F}}$, $S_1, S_2 \subseteq \mathit{Rules}(\mathfrak{F})$, and $n \geq N$, at least **one** of the following two statements holds:

1. There is $f \in \mathcal{L}(s)$ such that
$$\pi(f) \subseteq \Pi Z_n \langle S_1, S_2\rangle$$
2. For each $f \in \mathcal{L}(s)$, we have
$$\pi(f) \cap (\Pi(\mathcal{L}(s)) - \Pi(\langle S_1, S_2\rangle)) \neq \emptyset$$

Proof. We show that violation of (2) entails (1).

Fix s, S_1, S_2. Assume $f \in \mathcal{L}(s)$ and $\pi(f) \cap (\Pi(\mathcal{L}(s)) - \Pi(\langle S_1, S_2\rangle)) = \emptyset$. Since $\pi(f) \subseteq \Pi(\mathcal{L}(s))$, we also get $\pi(f) \subseteq \Pi(\langle S_1, S_2\rangle)$. Due to horizontal imdepotency/commutativity, there is a forest $g \in \langle S_1, S_2\rangle$ such that $\pi(f) \subseteq \pi(g)$. Let $N_{s,S_1,S_2} :=$ height(g). Thus, (†) $\pi(f) \subseteq \Pi Z_{N_{s,S_1,S_2}} \langle S_1, S_2\rangle$.

Since $H_{\mathfrak{F}}$ is finite, we can take an integer $N := \max_{s,S_1,S_2} N_{s,S_1,S_2}$. Furthermore, $Z_{N_{s,S_1,S_2}} \langle S_1, S_2\rangle \subseteq Z_n\mathcal{L}_{S_1,S_2}$ for any s, S_1, S_2. In view of (†), if (2) is violated, then (1) holds. □

To define Δ, we collect some properties expressible by distributive algebras that represent 'minimal requirements' that any tree in $\mathcal{L}(\rho)$ would certainly fulfil.

Definition 35. *(Distributive Approximators)* For $\rho \in \mathit{Rules}(A)$, define a set of trees $\Delta(\rho)$ as follows:

Let t be a tree, with root symbol α. Then $t \in \Delta(\rho)$ iff these conditions hold:

- The root symbol α is the symbol of the rule ρ. That is, ρ has the form $\cdots \leftarrow \alpha[\dots]$.
- If t has height $\leq N$, then there is a tree t' of height $\leq N$ such that $\{t'\} \in \mathcal{L}(\rho)$ and $\pi t' \subseteq \pi t$
- Take any $s \in H_{\mathfrak{F}}$, $S_1, S_2 \subseteq \mathit{Rules}(\mathfrak{F})$, and assume that
$$\pi(g) \cap (\alpha \circ (\Pi(\mathcal{L}(s)) - \Pi(\langle S_1, S_2\rangle))) \neq \emptyset$$
for any $g \in \mathcal{L}(\rho)$, where $\circ$ denotes concatenation: $\alpha \circ A := \{\alpha w : w \in A\}$. Then, we have
$$\pi(t) \cap (\alpha \circ (\Pi(\mathcal{L}(s)) - \Pi(\langle S_1, S_2\rangle))) \neq \emptyset$$

At this point, let's point out the formal similarity between the third condition in this definition to the second condition in Lemma 34. We can think of the third condition as a coarse distributive approximation to $\mathcal{L}(\rho)$ that is as fine-grained as allowed by sets of the form $\Pi(\langle S_1, S_2\rangle)$. We will later use the definition of $\Pi(\langle S_1, S_2\rangle)$ and Lemma 34 to see that this requirement, while quite weak, is useful at heights $> N$.

D.2 Defining Approximators $\mathcal{P}$

We now define the map $\mathcal{P}$. The idea is that languages $\mathcal{P}(R)$ approximate each $\pi\mathcal{L}(\rho)$ ($\rho \in \mathit{Rules}(\mathfrak{F}_i)$) exactly up to depth N given in Lemma 34, and up to the granularity provided by sets of the form $\Pi\langle R, S\rangle$ at greater depths.

Definition 36 (Trails, Traces). *(Trails):* Let f a forest. A *trail* in f is a sequence of nodes starting at the root of one member tree, continuing towards the leaf until it ends. Trails do not have to be maximal.

(Traces): Let Traces$_{\mathfrak{F}} \in \mathit{Rules}(\mathfrak{F}, \phi)^*$ as follows: $\tau \in$ Traces$_{\mathfrak{F}}$ iff, for all $1 \leq i < i+1 \leq |\tau|$, there is $s_i \in H_{\mathfrak{F}}$ such that

$$\tau_i \text{ has the form } \cdots \leftarrow \dots \{\dots, s_i, \dots\}$$
$$\tau_{i+1} \text{ has the form } s_i \leftarrow \dots \{\dots\}$$

Given a path $w \in \Sigma^*$, we set Traces$_{\mathfrak{F}}(p)$ to be the set of traces τ where $|w| = |\tau|$ and the transition symbol of rule τ_i is equal to the symbol w_i, for all i.

We're ready to define the approximators $\mathcal{P}$:

Definition 37 (Approximators). Let $\rho \in \mathit{Rules}(\mathfrak{F})$. We say that a trail p in a forest *satisfies the conditions* for a rule ρ if there is a trace $\zeta \in$ Traces$_{\mathfrak{F}_i}$ such that (1) $|\zeta| = |p|$, (2) $\zeta_0 = \rho$, and (3) for each $j \in \{1, ..., |p|\}$, having the j-th node in p as its root is in $\Delta(\zeta_j)$.

Now let $R \subseteq \mathit{Rules}(\mathfrak{F})$. Then $\mathcal{P}(R) \subseteq H_\Sigma$ is the language of nonempty forests where every trail satisfies the conditions for some $\rho \in R$.

Let us first note that Condition (D) of the Main Lemma is an instant consequence: As $\mathcal{P}$ is defined in terms of trails, it is enough to check membership of each singleton forest.

D.3 Proving Condition (C) in the Main Lemma

In order to prove Condition (C), which states recognition by wreath products of finite distributive algebras, we will need the operation Ψ from Definition 20. Using this construction, we can first establish the following result for $\mathcal{P}(R)$:

Proposition 38. For each R, the language $\mathcal{P}(R)$ is recognized by a wreath product of two distributive finite forest algebras.

Proof. We first want to show that, for any $\rho \in Rules(\mathfrak{F})$, the language $\Delta(\rho)$ is recognized by a finite distributive forest algebra.

Recall Proposition 6. In view of this, the first two conditions in the definition of $\Delta(\rho)$ can certainly be represented by a finite distributive forest algebra.

Now, let us consider the third condition. $\Pi(s)$ is a regular language of words (Proposition 26). To show that $\Pi(\langle S_1, S_2\rangle)$ is also regular, we can make use of Ψ (Definition 20). For $i = 1, 2$, let $\mathcal{L}(S_i)$ be the language of forests $f_1 + ... + f_n$, where $f_j \in \mathcal{L}(\rho_j)$ with $\rho_j \in S_i$, for all j. Given that $\mathfrak{F}$ is finite, the languages $\mathcal{L}(S_1)$, $\mathcal{L}(S_2)$ are regular forest languages. By definition of $\langle\cdot,\cdot\rangle$, we have $\langle S_1, S_2\rangle = \pi(\mathcal{L}(S_1)) \cap \pi(\mathcal{L}(S_2))$. By Proposition 21, we have $\pi(\Psi(\mathcal{L}(S_i))) = \pi(\mathcal{L}(S_i))$. Thus, we can rewrite $\langle S_1, S_2\rangle = \pi(\Psi(\mathcal{L}(S_1))) \cap \pi(\Psi(\mathcal{L}(S_2)))$. Now observe that, again due to Proposition 21, we can permute π outside of Boolean operations when they apply to images of Ψ. In our case, $\pi(\Psi(\mathcal{L}(S_1))) \cap \pi(\Psi(\mathcal{L}(S_2))) = \pi(\Psi(\mathcal{L}(S_1)) \cap \Psi(\mathcal{L}(S_2)))$. We can therefore write $\Pi(\langle S_1, S_2\rangle)$ as $\Pi(\Psi(\mathcal{L}(S_1)) \cap \Psi(\mathcal{L}(S_2)))$, which is a regular word language by Proposition 26 and Proposition 22. Thus, Proposition 6 also allows us to encode the third condition in a finite distributive forest algebra.

Since $\mathfrak{F}$ is a finite forest algebra, there is a finite distributive algebra recognizing all languages $\Delta(\rho)$ ($\rho \in Rules(\mathfrak{F})$). Checking whether a sequence of rules is a valid trace only requires testing pairs of adjacent symbols. Therefore, $\mathrm{Traces}_{\mathfrak{F}}$ is a regular word language. Using the characterization of wreath products in terms of sequential composition (Theorem 4.2 of [6]), it follows that $\mathcal{P}(R)$ is recognized by a wreath product of two finite distributive algebras. □

We can now prove condition (C) from the Main Lemma:

Proposition 39. For any $r_1, ... r_k \in R$, the language

$$\{f = f_1 + ... + f_k : f_i \in \mathcal{P}(\{r_i\})\}$$

is recognized by a wreath product of three finite distributive forest algebras.

We show this as an instance of a general fact:

Proposition 40. Let $\mathcal{L}_1, ..., \mathcal{L}_n$ be languages closed under addition ($f, f' \in \mathcal{L}_i$ implies $f + f' \in \mathcal{L}_i$) and under taking nonempty subsets of member forests – that is, for any $f \in \mathcal{L}_i$ and $\emptyset \subsetneq f' \subseteq f$ we have $f' \in \mathcal{L}_i$.

Let $\mathfrak{F} := (H, V)$ be a forest algebra recognizing $\mathcal{L}_1, ..., \mathcal{L}_n$ via morphism ϕ. Then there is a finite distributive algebra $\mathfrak{G}$ such that $\mathfrak{G} \wr \mathfrak{F}$ recognizes the language $\mathcal{L}_1 + ... + \mathcal{L}_n$.

By definition of $\mathcal{P}$, each $\mathcal{P}(R)$ is closed under taking nonempty subsets of member forests: Since $\mathcal{P}(R)$ is defined in terms of traces, a forest f is in $\mathcal{P}(R)$ if and only if all the trees in f are in $\mathcal{P}(R)$. Therefore, the previous proposition is an instant consequence of this fact.

Proof. Let H_1 be $Pow(Pow(\{1, ..., n\}))$ with union as the monoid operation, and let $V_1 \subseteq H_1^{H_1}$ consist of (1) the set of constant functions $H_1 \to H_1$, and (2) the set of functions $f_K : L \mapsto K \cup L$ ($K, L \subseteq Pow(\{1, ..., n\})$). V_1 is a monoid, with function composition as the operation, and $f_\emptyset$ as the identity element. V_1 acts on H_1 by $v \cdot h := v(h)$. For $h \in H_1$, we can set I_h to be f_h. Thus, (H_1, V_1) is a finite forest algebra.

Let's verify that (H_1, V_1) is distributive: Consider $v[h_1 + h_2]$. If v is a constant function, this is certainly equal to $vh_1 + vh_2$. Now consider the case where $v = f_K$. Then $v[h_1 + h_2] = K \cup h_1 \cup h_2 = (K \cup h_1) \cup (K \cup h_2) = v[h_1] + v[h_2]$. We have shown that (H_1, V_1) is distributive.

We claim that $(H_1, V_1) \wr \mathfrak{F}$ recognizes the language $\mathcal{L}_1 + ... + \mathcal{L}_n$.

For each context type of the form $v = \alpha[X] \in V_\Sigma$ – that is, consisting only of a variable and a parent node labeled α, define a function $f_\alpha : H \to V_1$ given by

$$f_\alpha(h)(h') \equiv \{\{i : \alpha[\phi_{\mathfrak{F}}^{-1}(h)] \cap \mathcal{L}_i \neq \emptyset\}\}$$

(independent of h', given that $f_\alpha(h)$ is a constant function). We can set $\eta(\alpha[X]) := (f_\alpha, \phi(\alpha))$ and can extend this map to a morphism $\eta : \Sigma^\Delta \to (H_1, V_1) \wr \mathfrak{F}$.

For a forest f, which we can write as a sum of trees $t_1 + ... + t_n$, we have $\eta(f) = \{\{i : t_j \in \mathcal{L}_i\} : j = 1, ..., n\}$. Therefore, $\mathcal{L}_1 + .. + \mathcal{L}_n$ is the set of trees f where (1) each element of $\eta(f)$ is nonempty, (2) each $i = 1, ..., n$ occurs in some element of $\eta(f)$.

Thus, $(H_1, V_1) \wr \mathfrak{F}$ recognizes $\mathcal{L}_1 + ... + \mathcal{L}_n$ via η. □

D.4 Correctness of Approximators

We now show Condition (A) of the Main Lemma, which is to establish that the distributive approximators $\Delta(\rho)$ and the approximators $\mathcal{P}(R)$ are 'big' enough to include the intended languages:

Proposition 41. Let $\rho \in Rules(\mathfrak{F})$. Let t be a tree such that $\{t\} \in \mathcal{L}(\rho)$. Then $t \in \Delta(\rho)$.

Proof. Directly from the definition of $\Delta(\rho)$. □

Proposition 42 (Condition A of the Main Lemma)**.** For $R \subseteq Rules(\mathfrak{F}_i)$ and any $r_1, ..., r_n \in R$ ($n \geq 1$), we have

$$\mathcal{L}(r_1) + ... + \mathcal{L}(r_n) \subseteq \mathcal{P}(R)$$

Proof. We check this by going through the definition of $\mathcal{P}$.

Let $f \in \mathcal{L}(r_1) + ... + \mathcal{L}(r_n)$. Let p be a maximal trail (we only allow nonempty trails in the definition of 'trail') in f. Let $\tau \in (\mathrm{Rules}(\mathfrak{F}))^{|p|}$ be the sequence of rules where τ_i is the rule that the tree rooted at the i-th element of p evaluates to. That is, $\tau \in \mathrm{Traces}_{\mathfrak{F}}(p)$. Certainly, p must be running through some tree t in f. There is a rule $\rho = r_j \in R$ such that $t \in \mathcal{L}(\rho)$. By choice of τ, we have $\tau_0 = \rho \in R$. Again, by choice of τ, the tree rooted in the i-th element of p is in $\mathcal{L}(\tau_i)$. By the preceding proposition, it is also in $\Delta(\tau_i)$.

By the definition of $\mathcal{P}$, the claim follows. □

D.5 Right-to-Left Inclusion for Condition (B)

Up to this point, we have proven conditions (A), (C), (D) from the Main Lemma. From Proposition 42, we can derive one of the two inclusions of condition (B) in the Main Lemma, which will be the goal of this section. Recall Ψ from Definition 20.

Proposition 43. Let f be a forest. If $f \in \mathcal{P}(R)$, then $\Psi f \in \mathcal{P}(R)$.

Proof. Consider a trail in Ψf and some arbitrary trail consisting of the same symbol sequence in f. For each node v in Ψf along the trail, look at the corresponding node v' along the trail selected in f. For each node v, let $f_v, f_{v'}$ be the forests below v, v', respectively. Then f_v is in the upward closure of $f_{v'}$ – that is, we can obtain f_v

from $f_{v'}$ by adding nodes below existing nodes. From the definition of Δ, one verifies easily that $\Delta(\rho)$ is upward-closed for each ρ. Also, membership in $\Delta(\rho)$ only depends on the set of paths. Therefore, $f_{v'} \in \Delta(\rho)$ implies $f_v \in \Delta(\rho)$. Then the claim follows from the definition of $\mathcal{P}(R)$. □

Now, we prove one of the two directions of the main lemma:

Proposition 44 (Right-to-Left Direction of the Main Lemma).

$$\Pi\left(Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S)\right) \supset \Pi\left\langle R, S\right\rangle \tag{6}$$

Proof. Let $w \in \Pi\left\langle R, S\right\rangle$. Then there is $f \in \left\langle R, S\right\rangle$ with $w \in f$. Stated differently, there are f_1, f_2 such that $\pi(f_1) = \pi(f_2) = f$, height $f_i \leq n$, $f_1 \in \mathcal{L}(r_1) + ...\mathcal{L}(r_k)$ and $f_2 \in \mathcal{L}(s_1) + ... + \mathcal{L}(s_j)$ ($r_m \in R, s_m \in S$), and $w \in \pi(f_i)$.

From Proposition 42, we know $f_1 \in \mathcal{P}(R)$, $f_2 \in \mathcal{P}(S)$. From Proposition 43, we conclude $\Psi f_1 \in \mathcal{P}(R)$, $\Psi f_2 \in \mathcal{P}(R)$.

Since $\pi(f_1) = \pi(f_2)$, we actually have $\Psi f_1 = \Psi f_2$ (Proposition 21). Thus, $\Psi f_i \in \mathcal{P}(R) \cap \mathcal{P}(S)$. Given height $f_i \leq n$, we have $\Psi f_i \in (Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S))$. In view of $w \in \pi(f) = \pi(\Psi f_i)$, we get $w \in \Pi(Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S))$. □

D.6 Left-to-Right Inclusion at Small Heights

The goal of this section is to the prove the left-to-right inclusion in condition (B) in the Main Lemma in the case when $n \leq N$.

We say that a path $w \in \pi(f)$ *belongs* to a trail p if $|w| = |p|$ and the node p_i is labeled w_i.

Lemma 45 (Technical Lemma). Let $n \leq N$, $\rho \in \mathit{Rules}(\mathfrak{F})$. Take t a tree. Let $w \in \pi(t)$, belonging to trail p. Assume there is a trace $\tau \in \mathit{Traces}_{\mathfrak{F}}(w)$ such that the tree rooted in π_i is in $\Delta(\tau_i)$, for each $i = 1, ..., |\pi|$.

Then there is a tree g such that

1. $g \in \mathcal{L}(\tau_1)$
2. $w \in \pi(g) \subseteq \pi(t)$

Proof. We do induction over the structure of the tree t.

The tree t has the form $\alpha[\{t_1, ..., t_k\}]$, where $k \geq 0$ indexes the children. (In the base case, $k = 0$ and there are no children). By induction hypothesis, we can assume the lemma holds for each of the trees t_i.

By assumption, $t \in \Delta(\tau_1)$. Since t has height $\leq N$, there is a tree $t' \in Z_N(\mathcal{L}(\tau_1))$ such that $\Pi(t') \subseteq \pi(t)$ according to the definition of $\Delta(\tau_1)$. Note that the root of t' must be α, since there are no empty trees.

If $|\tau| = 1$, set $g := t'$, and we have $g = t' \in \mathcal{L}(\tau_1)$ and $w = \alpha \in \pi(t') \subseteq \pi(t)$.

Now assume $|\tau| > 1$. Then one of the children t_i is rooted in τ_1, and $\alpha^{-1}w \in \pi(t_i)$, belonging to trail $p_{1\dots|p|}$. Also, $\tau' := \tau_{1\dots|p|}$ is a trace and the tree rooted in τ'_j is in $\Delta(\tau'_j)$ for each $j = 1, ..., |\tau'|$.

Thus, we can apply the induction hypothesis and obtain a tree g' such that $g' \in \mathcal{L}(\tau_2)$ and $\alpha^{-1}w \in \pi(g') \subseteq \pi(t_i)$.

Let $t'_1, ..., t'_l$ be the children of t' ($l \geq 0$, as the children set may be empty). Now set $g := \alpha[\{t'_1, ...t'_l, g'\}]$. Then $w \in \alpha\pi(g') \subseteq \pi(g)$. This shows (2).

Recall $g' \in \mathcal{L}(\tau_2)$. The rule τ_2 evaluates to a forest type q that occurs on the right-hand side of τ_1. Thus, since $t' \in \mathcal{L}(\tau_1)$, there is a child $t'_r \in \mathcal{L}(q)$. Due to horizonal idempotency, $(t'_r + g') \in \mathcal{L}(q)$. Thus, $\{t'_1, ...t'_l, g'\}$ evaluates to the same set of forest types $\subseteq H_{\mathfrak{F}}$ as $\{t'_1, ...t'_l\}$. Therefore, $g \in \mathcal{L}(\tau_1)$. This proves (1). □

Lemma 46. Let $n \leq N$ and $R \subseteq \mathit{Rules}(\mathfrak{F})$. Then

$$\pi Z_n(\mathcal{P}(R)) \subseteq \langle R, R\rangle$$

Proof. Take $f \in Z_n\mathcal{P}(R)$. Let $w \in \pi(f)$. By definition of $\mathcal{P}$, we obtain a trail π in f belonging to w and a trace τ such that $\tau_1 \in R$ and the tree rooted at π_i is in $\Delta(\tau_i)$, for $i = 1, ..., |\pi|$.

Let t_w be the trail rooted at π_1. From Lemma 45, we obtain a tree t'_w such that $t'_w \in \mathcal{L}(\tau_1)$ and $w \in \Pi(t'_w) \subseteq \pi(t)$. Define a forest $g := \{t'_w : w \in \pi(f)\}$. By construction, we have $\pi(f) = \pi(g)$, and $g \in \mathcal{L}(r_1) + ... + \mathcal{L}(r_k)$ ($r_j \in R$). Thus, $\pi(f) = \pi(g) \in \langle R, R\rangle$. □

Corollary 47. Let $n \leq N$ and $R, S \subseteq \mathit{Rules}(\mathfrak{F})$. Then

$$\Pi(Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S)) \subseteq \Pi Z_n\left\langle R, S\right\rangle$$

Proof. Let $w \in \Pi(Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S))$, so there is $f \in (Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S))$ with $w \in \pi(f)$. That is, $\pi(f) \in \pi(Z_n\mathcal{P}(R))$ and $\pi(f) \in \pi(Z_n\mathcal{P}(S))$. From Lemma 46, we have $\pi(f) \in \langle R, S\rangle$. Since $w \in \pi(f)$, w is in the path set of that language, which proves the claim. □

D.7 Left-to-Right Inclusion at Large Heights

To complete the Main Lemma, we prove the left-to-right inclusion of condition (B) for arbitrary heights n.

Definition 48 (Quotients). If f is a forest, we write $\alpha^{-1}f := \bigcup_{f':\alpha[f']\in f} f'$. Thus, $\alpha^{-1}f$ is also a forest, consisting of all the trees occurring below roots labeled α in members of f.

If $\mathcal{L}$ is a forest language, we write $\alpha^{-1}\mathcal{L} := \{\alpha^{-1}f : f \in \mathcal{L}\}$.

Lemma 49 (Second Direction at Arbitrary Heights). Let $R, S \subseteq \mathit{Rules}(\mathfrak{F})$, and $n \in \mathbb{N}$. Then

$$\Pi\left(Z_n\mathcal{P}(R) \cap Z_n\mathcal{P}(S)\right) = \Pi Z_n\left\langle R, S\right\rangle$$

Proposition 50. It suffices to prove Lemma 49 in the case where all rules in R, S have the same transition symbol α.

Proof. Let any sets $R, S \subseteq \mathit{Rules}(\mathfrak{F})$ be given. We can partition the set R into sets R_α ($\alpha \in \Sigma$) according to the transition symbol, and similarly for S.

Recall that our goal is to show the $\supseteq$ direction, as the $\subseteq$ direction is already known from Proposition 44.

So let $w \in \Pi Z_n\left(\left(Z_n\mathcal{P}\left(R\right) \cap Z_n\mathcal{L}\left(S\right)\right)\right)$. Then there is a tree t such that $\{t\} \in \left(\left(Z_n\mathcal{P}\left(R\right) \cap Z_n\mathcal{L}\left(S\right)\right)\right)$ and $w \in \pi(t)$. Let $\alpha := w_1$ be the root symbol of t. In view of the definition of $\mathcal{P}$ and Δ, we can restrict the statement to rules with this symbol: $\{t\} \in (Z_n\mathcal{P}(R^\alpha) \cap Z_n\mathcal{P}\left(S^\alpha\right))$. Now let us presume that the lemma has already been shown for the pair R^α, S^α. Applying this, we get

$$w \in \Pi Z_n\left(\mathcal{P}\left(R^\alpha\right) \cap \mathcal{P}\left(S^\alpha\right)\right) = \Pi Z_n\left\langle R^\alpha, S^\alpha\right\rangle \subseteq \Pi Z_n\left\langle R, S\right\rangle$$

□

The remainder of this section will be devoted to completing the proof of Lemma 49 under the assumption that all rules in R, S have the same transition symbol α. This will then complete condition (B) of the Main Lemma.

We only need to show $\subseteq$, the other direction following from Prop 44. We do induction over n and prove this for all sets of rules simultaneously. We have already shown this for $n \leq N$ in Lemma 47, which serves as the inductive base. So, for the inductive step, let $n > N$. During the inductive step, we will assume that sets R, S of rules are given where all rules have the same transition symbol α. Applying the previous proposition then completes the inductive step for all sets R, S.

Giving names to the two sides in the Lemma 49, the goal is to prove equality between the path set of the language

$$\phi := (Z_n \mathcal{P}(R) \cap Z_n \mathcal{P}(S)) \tag{7}$$

and the (finite) path set

$$g := \Pi Z_n \langle R, S \rangle \tag{8}$$

Note that ϕ is a finite (due to Z_n) forest language, while g is a finite set of words. That is, the goal of this section is to prove

$$\Pi\phi = g$$

We have $\Pi\phi \supseteq g$ by Lemma 44.

We write $R = \{R_1, ..., R_{n_1}\}$, and $S = \{R_1, ..., R_{n_2}\}$. We write (recall Definition 29)

$$\begin{aligned} R_i &= \left\langle s_0^i, \alpha, \{s_l^i : l = 1, ..., n_{R_i}\}\right\rangle \\ S_i &= \left\langle t_0^i, \alpha, \{t_l^i : l = 1, ..., n_{S_i}\}\right\rangle \end{aligned} \tag{9}$$

When $s \in H_{\mathfrak{F}}$, define

$$\rho(s) := \{R \in Rules(\mathfrak{F}) : \pi^{(1)}(R) = s\} \tag{10}$$

To apply the induction hypothesis, we consider

$$\widehat{\phi} := Z_{n-1}\mathcal{P}\left(\bigcup_{i=1}^{n_1}\bigcup_l \rho(s_l^i)\right) \cap Z_{n-1}\mathcal{P}\left(\bigcup_{j=1}^{n_2}\bigcup_l \rho(t_l^j)\right) \tag{11}$$

Let us reflect what this means: Instead of the rules in R, we take all the rules that evaluate to a forest type that appears on the right-hand side of a rule in R. (The same for S.) The language $\mathcal{P}\left(\bigcup_{i=1}^{n_1}\bigcup_l \rho(s_l^i)\right)$ is a basic approximation to the quotient of $\mathcal{P}(R)$ by the root symbol α. Note that, as removing the root symbol reduces height, we're using height $n-1$ in the definition of $\widehat{\phi}$. Therefore, the object $\widehat{\phi}$ is a basic approximation to the quotient of ϕ by the root symbol α. Since it has smaller height, we can apply the induction hypothesis. However, it might be larger than the quotient of ϕ by α, so we'll need to do extra work before we can use the induction hypothesis to say something about f. This is the motivating idea behind the remainder of the proof, where we 'shrink' $\widehat{\phi}$ until its pathset is equal to that of the quotient of ϕ by the root symbol α.

Here it might be useful to remark that $\mathcal{P}$ and $\cup$ do not commute. For instance, $\mathcal{P}(\{\rho_1\}) \cup \mathcal{P}(\{\rho_2\}) \subseteq \mathcal{P}(\{\rho_1, \rho_2\})$, but the converse usually does not hold. The reason is that $\mathcal{P}$ looks at each trail separately and cannot 'keep track' of which tree 'belongs' to which rule.

Returning to the proof, since height$(\widehat{\phi}) = n - 1 < n$, we can plug $\widehat{\phi}$ into the induction hypothesis, and obtain

$$\Pi\widehat{\phi} =^{(IH)} \Pi Z_{n-1}\left\langle\left(\bigcup_{i=1}^{n_1}\bigcup_l \rho(s_l^i)\right), \left(\bigcup_{j=1}^{n_2}\bigcup_l \rho(t_l^j)\right)\right\rangle \tag{12}$$

As the next proposition shows, we can view $\Pi\widehat{\phi}$ as an outer approximation to $\alpha^{-1}g$ (recall Definition 48) and $\alpha^{-1}\Pi\phi$:

Proposition 51.

$$\alpha^{-1}g \subseteq \alpha^{-1}\Pi\phi \subseteq \Pi\widehat{\phi}$$

Note that α^{-1} refers to quotients of word languages here, not forest languages.

$$\alpha^{-1}\phi \subseteq \widehat{\phi}$$

Note that α^{-1} refers to the quotient of a forest language here.

Proof. We already know the first inequality of the first claim since we know $g \subseteq \Pi\phi$ from Proposition 44. The second inclusion of the first claim follows immediately from the second claim. Let's now prove the second claim:

Let $f \in \alpha^{-1}\phi$. By definition of quotients of forest languages (Definition 48), there is $f' \in \phi$ and $t_1, ..., t_k \in f'$ such that $t_i = \alpha[f_i]$ $(i = 1, ..., k)$ and $f_1 + ... + f_k = f$. Now we want to appeal to the definition of $\mathcal{P}$ (Definition 37). So let p be a trail in f, so it is also a trail in some f_i. This can be extended to a trail vp in t_i (where v is the root of t'). Since $t_i \in \phi$, we know $\{t_i\} \in \mathcal{P}(R)$. By definition of $\mathcal{P}$, there is a trace $\tau \in \text{Traces}_{\mathfrak{F}}$ such that (1) $|\tau| = |p| + 1$, (2) $\tau_1 \in R$, say $\tau_1 = R_k$, (3) for each $j \in \{1, ..., |p| + 1\}$, the tree rooted at $(vp)_j$ is in $\Delta(\tau_j)$. Now by the definition of $\text{Traces}_{\mathfrak{F}}$ and since $\tau_1 = R_k$, we know $\tau_2 \in \bigcup_l \rho(s_l^k) \subseteq \bigcup_{i=1}^{n_1}\bigcup_l \rho(s_l^i)$. So let $\tau' := \tau_{2...|\tau|} \in \text{Traces}_{\mathfrak{F}}$. Then we have (1) $|\tau'| = |p|$, (2) $\tau'_1 \in \bigcup_{i=1}^{n_1}\bigcup_l \rho(s_l^i)$, (3) for each $j \in \{1, ..., |p|\}$, the tree rooted at p_j is in $\Delta(\tau'_j)$. By definition of $\mathcal{P}$, we know $f \in \mathcal{P}\left(\bigcup_{i=1}^{n_1}\bigcup_l \rho(s_l^i)\right)$.

The proof for the S-part is analogous. Together, we get $f \in \widehat{\phi}$. □

We now want to prove that $\Pi\phi = g$, which as explained above is exactly the statement of the Main Lemma to be proven. We construct monotonically decreasing sequences $\widehat{\phi}_k \subseteq H_\Sigma^D$ and $W_k^1 \subseteq R$, $W_k^2 \subseteq S$ $(k = 0, 1, ...)$. Our proof strategy will be to 'sandwich' $\Pi\phi$ between this decreasing sequence and g.

To make notation nicer, we will write (recall (11)):

$$S_k^1 := \bigcup_{i \in W_k^1}\bigcup_l \rho(s_l^i)$$

$$S_k^2 := \bigcup_{i \in W_k^2}\bigcup_l \rho(t_l^i)$$

Definition 52. Here is the inductive definition:

- $\widehat{\phi}_0 := \widehat{\phi}$, $W_0^1 := R$, $W_0^2 := s$
- We are in the $k + 1$-th step.
  1. *Case I*: For each $\rho \in W_k^1 \cup W_k^2$, there is a forest $h \in \mathcal{L}(\rho)$ such that $\pi(h) \subseteq \left(\alpha\Pi\widehat{\phi}_k \cup \{\epsilon\}\right)$.
     We take
     $$W_{k+1}^s := W_k^s$$
     for $s = 1, 2$.
  2. *Case II*: Case I is not satisfied. That is, there is $\rho \in W_k^z$ $(z \in \{1, 2\})$, such that, for each $h \in \mathcal{L}(\rho)$ we have $\pi(h) \nsubseteq \alpha\Pi\widehat{\phi}_k \cup \{\epsilon\}$.
     For definiteness, we take $z = 1$. So $\rho \in W_k^1$. Set
     $$W_{k+1}^1 := W_k^1 - \{\rho\}, \quad W_{k+1}^2 := W_k^2$$
     The construction is symmetric if instead $z = 2$.

  This concludes the inductive definition of W_k^1, W_k^2. In either of the two cases, we set

  $$\widehat{\phi}_{k+1} := \widehat{\phi}_k \cap \left(\mathcal{P}(S_k^1) \cap \mathcal{P}(S_k^2)\right)$$

Since $|W_k^1| + |W_k^2|$ gets strictly smaller whenever we are in Case II, we can reach Case II only finitely often (If the sets happen to become empty, Case I becomes trivially true.), and the sequences W_k^1, W_k^2 become stationary.

Having defined the sequences $\widehat{\phi}_k, W_k^i$, we first note the following:

Proposition 53. 1. For all k, we have

$$\widehat{\phi}_{k+1} = \left(Z_{n-1}\mathcal{P}(S^1_k) \cap Z_{n-1}\mathcal{P}(S^2_k)\right)$$

2. $\Pi\widehat{\phi}_{k+1} = \Pi Z_{n-1}\left(\left\langle S^1_k, S^2_k\right\rangle\right)$

Proof. 1. is immediate from the definition.
2. follows from (a) and the induction hypothesis, analogous to (12).

□

Recall that in Proposition 51 we established that $\widehat{\phi}$ extends $\alpha^{-1}\phi$. We now show that this remains true for all $\widehat{\phi}_k$. This is the place where the choice of N and $\Delta(\rho)$ come in crucially.

Proposition 54.

$$\forall k : \alpha^{-1}\phi \subseteq \widehat{\phi}_k \tag{13}$$

Proof. By induction over k. In the case of $k = 0$, we established this in Proposition 51.

In the inductive step, we want to show $\widehat{\phi}_k \supset \alpha^{-1}\phi$ and already know $\widehat{\phi}_{k-1} \supset \alpha^{-1}\phi$. Assume $\widehat{\phi}_k \not\supset \alpha^{-1}\phi$. So there is a forest

$$f \in \left(\alpha^{-1}\phi - \widehat{\phi}_k\right) = \left((\alpha^{-1}\phi \cap \widehat{\phi}_{k-1}) - \widehat{\phi}_k\right) \tag{14}$$

We know that f is in $\widehat{\phi}_{k-1} - \widehat{\phi}_k$, which by Proposition 53 is equal to

$$Z_{n-1}\left[(\mathcal{P}(S^1_{k-2}) \cap \mathcal{P}(S^2_{k-2})) - (\mathcal{P}(S^1_{k-1}) \cap \mathcal{P}(S^2_{k-1}))\right]$$

We have $f \in \alpha^{-1}\phi$. This means there is a forest $f_0 \in \phi$ such that f is the collection of the children of the root nodes in f_0 (all root nodes in ϕ have label α).

Recall that, by definition of $\mathcal{P}(R)$ (Definition 37), it is the language of forests where each trail satisfies the conditions for some $\rho \in R$.

From the definition of $\mathcal{P}$ and $\phi \in \mathcal{P}(R)$, we know that each maximal trail in f_0 satisfies the conditions for some rule in R. Also, there is a trail p' in f that does not satisfy the condition for any forest type in S^1_{k-1}. This trail extends to a trail p in f_0 that does not satisfy the condition for any rule in W^1_{k-1} (+). The trail p satisfies the conditions for some rule $\rho \in R$, since $p \in \phi$ and $\phi \in \mathcal{P}(R)$. So the trail p satisfies the conditions for some rules in $R - W^1_{k-1}$ but for no rule in W^1_{k-1}. From (+), we know $\rho \in R - W^1_{k-1}$. So ρ was removed at some prior stage $k' < k - 1$ of the construction.

So, in stage k', Case II was reached, and for every $h \in \mathcal{L}(\rho)$, we have $\pi(h) \not\subseteq \alpha\Pi\widehat{\phi}_{k'} \cup \{\epsilon\}$. Then there is a forest type s^q_l from the third component of ρ such that, for any $h \in \mathcal{L}(s^q_l)$, we have $h \not\subseteq \Pi\widehat{\phi}_{k'}$.

From Proposition 53.2, we have

$$\Pi\widehat{\phi}_{k'} = \Pi Z_{n-1}\left\langle S^1_{k'-1}, S^2_{k'-1}\right\rangle \subseteq \Pi\left\langle S^1_{k'-1}, S^2_{k'-1}\right\rangle \tag{15}$$

Now take $I := \Pi(\mathcal{L}(s^q_l)) - \Pi\left\langle S^1_{k'-1}, S^2_{k'-1}\right\rangle$. By (15), (*) $I \cap \Pi\widehat{\phi}_{k'} = \emptyset$. At this point, we can apply Lemma 34: Since $n \geq N$, for each $h \in \mathcal{L}(s^q_l)$, we have $\pi(h) \cap I \neq \emptyset$. Therefore, for any $h \in \mathcal{L}(\rho)$, we have $\pi(h) \cap (\alpha \circ I) \neq \emptyset$. From the third point in the definition of Δ (Definition 35), we conclude: (†) For each $h \in \Delta(\rho)$, we have $\pi(h) \cap (\alpha \circ I) \neq \emptyset$.

We know p satisfies the conditions for ρ, so $\pi(f_0) \in \Delta(\rho)$. Since $f_0 \in \phi$, we get $\pi(\phi) \cap (\alpha \circ I) \neq \emptyset$ from (†). As we have $\widehat{\phi}_{k-1} \supset \alpha^{-1}\phi$ from the inductive hypothesis, we obtain $\pi(\widehat{\phi}_{k-1}) \cap (\alpha \circ I) \neq \emptyset$. On the other hand, $k' < k - 1$ and thus $\widehat{\phi}_{k'} \supset \widehat{\phi}_{k-1}$. This is a contradiction to $(*)$. Thus, the initial assumption was incorrect. □

Putting together what we have shown so far, we get a chain of inequalities:

$$\alpha^{-1}g \subseteq \alpha^{-1}\Pi\phi \subseteq \cdots \subseteq \Pi\widehat{\phi}_3 \subseteq \Pi\widehat{\phi}_2 \subseteq \Pi\widehat{\phi}_1 \subseteq \Pi\widehat{\phi} \tag{16}$$

In order to sandwich $\alpha^{-1}\Pi\phi$ between $\alpha^{-1}g$ and the decreasing sequence $\Pi\widehat{\phi}_k$, we show that the sequence ultimately takes on the value $\alpha^{-1}g$. First, we show

Proposition 55. For k such that Case II is never reached for $k' \geq k - 1$, we have

$$\alpha(\Pi\widehat{\phi}_k) \subseteq g \tag{17}$$

where $\alpha(\Pi\widehat{\phi}_k) := \{\alpha w : w \in \Pi\widehat{\phi}_k\}$

Since Case II is reached only finitely many often, there in particular is such a k.

Proof. By assumption, for each $\rho \in W^1_k$, there is $\phi \in \mathcal{L}(\rho)$ such that $\Pi(\phi) \subseteq \Pi\widehat{\phi}_k$. Also, for each $\rho' \in W^2_k$, there is $\phi \in \mathcal{L}(\rho')$ such that $\Pi(\phi) \subseteq \Pi\widehat{\phi}_k$.

Therefore, for each $\rho \in W^1_k$ and each ρ', there is $\phi \in \mathcal{L}(s^\rho_l)$ such that $\Pi(\phi) \subseteq \alpha^{-1}\Pi\widehat{\phi}_k$. And same for W^2_k, t^i_l. (†)

By Proposition 53 (2), we know

$$\alpha\Pi\widehat{\phi}_k = \alpha\Pi Z_{n-1}\left\langle\left(\bigcup_{\rho\in W^1_k}\bigcup_l s^\rho_l\right), \left(\bigcup_{\rho'\in W^2_k}\bigcup_j t^{\rho'}_j\right)\right\rangle \tag{18}$$

Now let $d \in Z_{n-1}\left\langle\bigcup_{\rho\in W^1_k}\bigcup_l s^\rho_l, \bigcup_{\rho'\in W^2_k}\bigcup_j t^{\rho'}_j\right\rangle$, so there are forest types $\sigma_j := s^\rho_l$ ($\rho \in W^1_k$) such that $d = \pi(f_1 + \ldots + f_k)$ with $f_j \in \mathcal{L}(\sigma_j)$. Using (†), for each $\rho \in W^1_k$ and each j, there is $\phi_{\rho,j} \in \mathcal{L}(s^\rho_l)$ such that $\Pi(\phi) \subseteq \alpha^{-1}\Pi\widehat{\phi}_k$. If we set $d' := d \cup \pi(\sum_{\rho,j}\phi_{\rho,j})$, then we have $d \subseteq d'$, but also

$$d' \in Z_{n-1}\left(\left(\sum_{\rho\in W^1_k}\sum_l \pi\mathcal{L}\left(s^\rho_l\right)\right) \cap \left(\sum_{\rho'\in W^2_k}\sum_j \pi\mathcal{L}\left(t^{\rho'}_j\right)\right)\right)$$

Thus, $d \subseteq d' \subseteq \Pi Z_{n-1}\left(\sum_{\rho\in W^1_k}\sum_l \pi\mathcal{L}\left(s^\rho_l\right) \cap \sum_{\rho'\in W^2_k}\sum_j \pi\mathcal{L}\left(t^{\rho'}_j\right)\right)$

Putting this together with (18), we obtain:

$$\alpha\Pi\widehat{\phi}_k = \alpha\Pi\left(\sum_{\rho\in W^1_k}\sum_j \pi Z_{n-1}\mathcal{L}\left(s^\rho_j\right) \cap \sum_{\rho'\in W^2_k}\sum_j \pi Z_{n-1}\mathcal{L}\left(t^{\rho'}_j\right)\right)$$

Considering how paths of forests are built from paths of their children, we can rewrite this as

$$\left(\alpha\Pi\widehat{\phi}_k\right) \cup \{\epsilon\} = \Pi\left(\sum_{\rho\in W^1_k}\pi Z_n\mathcal{L}(\rho) \cap \sum_{\rho'\in W^2_k}\pi Z_n\mathcal{L}(\rho')\right)$$

In view of the definition of $\langle\cdot,\cdot\rangle$, the right-hand side is a subset of $\Pi Z_n\left\langle W^1_k, W^2_k\right\rangle$. We therefore get

$$\alpha\Pi\widehat{\phi}_k \subseteq \Pi Z_n\left\langle W^1_k, W^2_k\right\rangle$$

Now $W^1_k \subseteq R$, $W^2_k \subseteq S$, and thus

$$\alpha\Pi\widehat{\phi}_k \subseteq \Pi Z_n\langle R, S\rangle$$

The expression on the right side is equal to g by definition of g (8).

Taken together, we have shown $\alpha\Pi\widehat{\phi}_k \subseteq g$. □

As a converse to the last observation, we have:

Proposition 56.

$$g \subseteq \Pi\phi \subseteq \Big(\{\epsilon\} \cup (\alpha\Pi\widehat{\phi}_k)\Big)$$

Proof. We have shown the first inclusion, $g \subseteq \Pi\phi$, previously (Proposition 44). Let's consider the second one, $\Pi\phi \subseteq \Big(\{\epsilon\} \cup (\alpha\Pi\widehat{\phi}_k)\Big)$. We already know $\alpha^{-1}\phi \subseteq \widehat{\phi}_k$ (Lemma 54). This entails $\Pi(\alpha^{-1}\phi) \subseteq \Pi\widehat{\phi}_k$. Since every tree occurring in elements of ϕ has α as its root node symbol, we have $\Pi(\phi) = \{\epsilon\} \cup \alpha\Pi(\alpha^{-1}\phi)$. Thus, we conclude $\Pi\phi \subseteq \{\epsilon\} \cup (\alpha\Pi\widehat{\phi}_k)$. □

From the last two Propositions, for large k, we get a chain of inclusions

$$\alpha\Pi\widehat{\phi}_k \subseteq g \subseteq \Pi\phi \subseteq \Big(\{\epsilon\} \cup (\alpha\Pi\widehat{\phi}_k)\Big) \tag{19}$$

Considering $\epsilon \in g$ (any pathset must contain ϵ), we deduce

$$\Big(\{\epsilon\} \cup (\alpha\Pi\widehat{\phi}_k)\Big) = g$$

and, looking at the inclusion chain (19) again

$$g = \Pi\phi$$

This concludes, first, the inductive step, and thus the entire proof of the Main Lemma.

E Proof of Proposition 10

Let $\mathfrak{F}$ be two-distributive, and let $\phi : \Sigma^\Delta \to \mathfrak{F}$ be a morphism. In Definition 25, we defined H_Σ^D as the image of π and thus a set of pathsets. However, we can also view it as a monoid of forest types: Let $\sim$ be the congruence on H_Σ defined by $f \sim f' \Leftrightarrow \pi(f) = \pi(f')$. This extends to a congruence on V_Σ. The quotient of $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$ by this congruence is an infinite forest algebra whose forest types are precisely the elements of H_Σ^D. It is also not hard to show that this infinite forest algebra is equal to the quotient of Σ^Δ by the congruence induced by $v[h + h'] = vh + vh'$, and is therefore distributive. We will call this infinite distributive forest algebra $\Sigma_D^\Delta = (H_\Sigma^D, V_\Sigma^D)$. Its forest types are equivalence classes of forests that have the same path set, and can thus be represented as finite path sets. We can extend π to a forest algebra morphism $\Sigma^\Delta \to \Sigma_D^\Delta$, which is the canonical projection for the congruence $\sim$.

The proof of the Proposition now closely follows the proof of the main theorem. Analogous to that proof, let us consider $D_{\phi,\pi}$, which has *infinitely* many objects. Note that the proof of the Derived Category Theorem (Theorem 15) given by [18] applies even when the involved algebras and categories are infinite. We want to show that $D_{\phi,\pi}$ is locally distributive. Let h, h' be objects, let $f_1, f_2 \in \mathrm{HArr}(h)$, and let $v \in \mathrm{Arr}(h, h')$. By the definition of the Derived Category, we can write f_1 as $h \xleftarrow{h_1}$ and f_2 as $h \xleftarrow{h_2}$. Also, we can write v as $h' \xleftarrow{p} h$, with $p \in V_{\mathfrak{F}}$. We want to prove the equality from Definition 18. Note that $h_1, h_2 \in H_{\mathfrak{F}}$. In view of the construction of the half-arrows in the derived category, there are forests $t_1, t_2 \in H_\Sigma$ such that $\phi(t_i) = h_i$ and $\pi(t_i) = h$ for $i = 1, 2$. Let $\alpha' \in \phi^{-1}(p)$. Since $\mathfrak{F}$ is 2-distributive, we have $\phi(\alpha'[t_1 + t_2]) = \phi(\alpha' t_1 + \alpha' t_2)$. Applying ϕ, this means

$$p(h_1 + h_2) = p(h_1) + p(h_2)$$

In the derived category, this translates to

$$v(f_1 + f_2)) = vf_1 + vf_2$$

or, in arrow-based notation,

$$h' \xleftarrow{p} (h \xleftarrow{h_1} + h \xleftarrow{h_2}) = h' \xleftarrow{p} h \xleftarrow{h_1} + h' \xleftarrow{p} h \xleftarrow{h_2}$$

Thus, $D_{\phi,\pi}$ is locally distributive.

It is not finite, so we cannot directly apply the Local-Global Theorem 19 here. If we inspect the proof of the Local-Global Theorem, we see that the first part still applies here: For any two distinct half-arrows in $D_{\phi,\pi}$, we get

$$(\dagger)\ \pi_{\Sigma'}(\mathrm{Val}^{-1}(h \xleftarrow{c})) \cap \pi_{\Sigma'}(\mathrm{Val}^{-1}(h' \xleftarrow{c'})) = \emptyset$$

where $\pi_{\Sigma'}$ maps forest-diagrams to sets of paths over the alphabet Σ' consisting of arrows and half-arrows of $D_{\phi,\pi}$. This alphabet is *infinite*, but this needn't concern us: Σ'^Δ is again a free forest algebra. Furthermore, $\Sigma_D'^\Delta$, the quotient by the congruence induced by $\pi_{\Sigma'}$, is an infinite distributive algebra. We now construct a division $D_{\phi,\pi} \prec \Sigma_D'^\Delta$ analogous to our reasoning in the proof of the Local-Global Theorem. We assign to each half-arrow $h \xleftarrow{c}$ in $D_{\phi,\pi}$ the set of half-arrows in $\Sigma_D'^\Delta$ that are in the image of $\mathrm{Val}^{-1}(h \xleftarrow{c})$ under $\pi_{\Sigma'}$, and similarly for arrows. As in that proof, this mapping preserves operations, and, due to (†), is injective.

In view of $D_{\phi,\pi} \prec \Sigma_D'^\Delta$, the Derived Category Theorem now implies that $\mathfrak{F}$ divides the wreath product of the two infinite distributive algebras $\Sigma_D'^\Delta$ and Σ_D^Δ.

F Proof for Example 12

To show that the syntactic algebra is 2-distributive, we first note that π separates $\mathcal{L}_1$ and $\mathcal{L}_2$: We say that f is *compatible with* $\mathcal{L}_1$ if

$$\max_n \left(a^n b \in \pi(f)\right) = \max_n \left(a^n c \in \pi(f)\right)$$

, while f is *compatible with* $\mathcal{L}_2$ if

$$1 + \max_n \left(a^n b \in \pi(f)\right) = \max_n \left(a^n c \in \pi(f)\right)$$

These two conditions define disjoint subsets of H_Σ, separate the two languages, and are definable through a morphism into an infinite distributive forest algebra. Now consider the following statement

1. there is some nonempty path
2. each maximal path has the form $d^n a^m (b/c)$, where
   a. the d's alternatingly have children that are compatible (modulo π) with $\mathcal{L}_1$ and compatible with $\mathcal{L}_2$
   b. In each path $d^n a^m b$, either (1) the b node has a c sibling and for each a node and for the last d node, what comes below it is compatible with $\mathcal{L}_1$, or (2) the b node has an ac sibling and for each a node and for the last d node, what comes below it is compatible with $\mathcal{L}_2$
   c. In each path $d^n a^m c$, either (1) the c node has a b sibling and for each a node and for the last d node, what comes below it is compatible with $\mathcal{L}_1$, or (2) the last a node exists and has a b sibling and for each a node before it and for the last d node, what comes below it is compatible with $\mathcal{L}_2$

This can be expressed by a morphism into a wreath product of two infinite distributive algebras. One can show that this defines $\mathcal{L}$:

Proof. First, it is straightforward that all forests in $\mathcal{L}$ satisfy this description. For the converse, we do induction over the size of forests satisfying the description. We want to show that, for any path where the forest below the last d node is compatible with $\mathcal{L}_1$, those of its children whose roots are labeled a actually form an element of $\mathcal{L}_1$. Similarly for $\mathcal{L}_2$. As the base case, we take those forests that have only d nodes.

Now let f be a forest satisfying this description, and let $\pi = v_1 ... v_n$ a maximal sequence of nodes from root to leaf, with v_1 a root node and v_n a leaf labeled b or c. We know there is at least one d node, unless f is an empty forest (which is covered by the base case). Let's assume that what comes below the last d node is compatible with $\mathcal{L}_1$. If v_n is labeled b, it has a c sibling, otherwise it is labeled c and has a b sibling. Thus, v_n plus this sibling are in $\mathcal{L}_1$.

We remove these two nodes. We then remove any a nodes that have only a nodes below them.

- Case 1: The last d node has no a child. Then we can apply the induction hypothesis or go directly to the base case.
  If this is not the case, what comes below the last d node must be compatible with either $\mathcal{L}_1$ or $\mathcal{L}_2$.
- Case 2: What comes below the last d node is still compatible with $\mathcal{L}_1$. Then we apply the induction hypothesis.
- Case 3: What comes below the last d node is now compatible with $\mathcal{L}_2$. We know that there is a path $a^{l+1}c$ branching off somewhere from our path, and it is in total at most as long as our path was, and the longest a^*b path is shorter by one. Contradiction, since what comes below the node at which the two branches parted must have been compatible with $\mathcal{L}_1$, which contradicts what we get when considering the conditions placed on this $a^{l+1}c$ path.

We have concluded that the a-children of the last d-node must indeed have formed an element of $\mathcal{L}_1$.

So now let's assume that what comes below the last d node is compatible with $\mathcal{L}_2$. If v_n is labeled c, v_{n-1} is an a node, and v_{n-1} has a b sibling. Let's remove these two nodes, and then all a nodes having only a nodes below them. Let's assume that what comes below the last d node is now compatible with $\mathcal{L}_1$. But then as previously we get a contradiction with conditions placed on the paths that are now longest.

In conclusion, we have shown that the description given above captures $\mathcal{L}$. □

Now let $\mathfrak{F}$ be the syntactic algebra with morphism ϕ and let $\mathfrak{G}$ be a finite distributive algebra with some morphism ψ. Consider the derived category $D_{\phi,\psi}$. Considering Proposition 6, no language recognized by a finite distributive forest algebra can separate $\mathcal{L}_1$ from $\mathcal{L}_2$. Thus, since $\mathfrak{G}$ is finite and distributive, there are forests $f_1 \in \mathcal{L}_1$, $f_2 \in \mathcal{L}_2$ with $\psi(f_1) = \psi(f_2)$. On the other hand, $d[f_1] + d[f_2] \in \mathcal{L}$, while $d[f_1 + f_2] \notin \mathcal{L}$. Thus,

$$\phi(d[f_1 + f_2]) \neq \phi(d[f_1] + d[f_2])$$

Since $\psi(f_1) = \psi(f_2)$, the derived category is not locally distributive (recall Definition 18). So the derived category does not divide any distributive forest algebra. Appealing to the second direction of the Derived Category Theorem as stated in Theorem 27, we see that, whenever $\mathfrak{G}' \wr \mathfrak{G}$ recognizes $\mathcal{L}$, the algebra $\mathfrak{G}'$ cannot be distributive.