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 satisyfing 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.

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

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 $I_h h' = h + h'$.

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_D^Δ. We can extend π to a morphism $\Sigma^\Delta \to \Sigma_D^\Delta$. 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 $\Sigma_D'^\Delta$ where Σ' is an extended alphabet. The Derived Category Theorem 15 implies that $\mathfrak{F}$ divides $\Sigma_D'^\Delta \wr \Sigma_D^\Delta$. □

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:

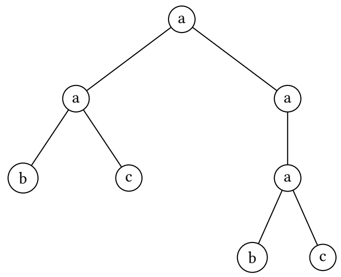


Figure 1. Illustration for Example 11

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.

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.

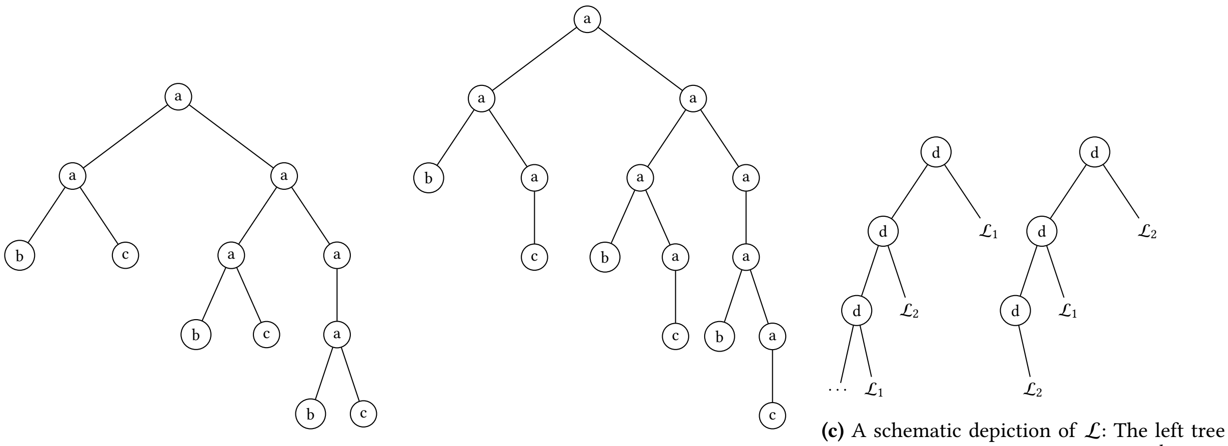
In this section, we will review the definition of the Derived Forest Category and the Derived Category Theorem from [18]. The Derived Forest Category is a category with some structure added to it, and the relation between forest categories and categories is akin to the relation between forest algebras 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*.



(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.

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
$$\left(h' \xleftarrow{p} h\right) + \left(h_2 \xleftarrow{h_1}\right) = (h' + h_2) \xleftarrow{p+s} h$$
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. $K_{x \xleftarrow{c}} + K_{z \xleftarrow{f} y} \subseteq K_{x \xleftarrow{c} + y z \xleftarrow{f}}$
   e. $K_{z \xleftarrow{f} y} + K_{x \xleftarrow{c}} \subseteq K_{z \xleftarrow{f} y + x \xleftarrow{c}}$
2. (Injectivity)
   a. If $y \xleftarrow{c} x$ and $y \xleftarrow{c'} x$ are distinct arrows, then $K_{y \xleftarrow{c} x} \cap K_{y \xleftarrow{c'} x} = \emptyset$

b. If $y \overset{c}{\longleftarrow}$ 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 $\phi_\mathfrak{F} : \Sigma^\Delta \rightarrow \mathfrak{F}$, 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 $\phi_X : \Sigma^\Delta \rightarrow \mathfrak{F}_1 \wr \mathfrak{F}_2 \wr \mathfrak{F}_3$ 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 $h' \overset{v}{\longleftarrow} h$:

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

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. □

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 $f, f' \in H_\Sigma$.

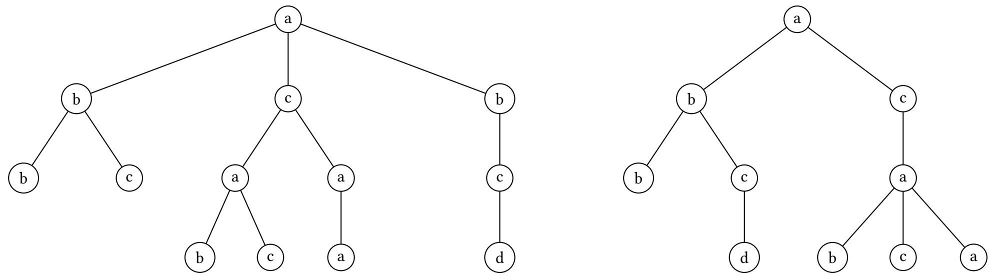
1. No two distinct sibling nodes in $\Psi(f)$ are labeled with the same symbol.
2. $\pi(f) = \pi(\Psi(f))$.
3. $\pi(f) = \pi(f')$ if and only if $\Psi(f) = \Psi(f')$.

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.



(a) **(b)**

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.

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.