NEW APPLICATIONS OF THE WREATH PRODUCT OF FOREST ALGEBRAS *

HOWARD STRAUBING[1]

Abstract. We give several new applications of the wreath product of forest algebras to the study of logics on trees. These include new simplified proofs of necessary conditions for definability in CTL and first-order logic with the ancestor relation; a sequence of identities satisfied by all forest languages definable in PDL; and new examples of languages outside CTL, along with an application to the question of what properties are definable in both CTL and LTL.

Mathematics Subject Classification. 03D05, 68Q70.

1. INTRODUCTION

The present paper is part of an ongoing effort to understand what properties of finite labeled trees are expressible in first-order logic and related logics (*e.g.,* CTL, CTL^*, PDL). Recently, some progress has been made in approaching such problems through algebraic means. The idea is that one associates to a given regular tree language L a finite algebra (called the *syntactic forest algebra* of L) in much the same way that the syntactic monoid is associated to a regular language of words. Definability of L in a given logic is often reflected in computable properties of the associated algebra. This approach has led to effective characterizations of the properties definable in a number of logics (*e.g.,* the temporal logics EF and EX [3,7], first-order logic with successor [1], boolean combinations of Σ_1-languages [6], among others). For those logics for which such an effective test is still lacking, the algebraic method has led to the formulation of necessary

[1] Computer Science Department, Boston College, Chestnut Hill, 02467 Massachusetts, USA. straubin@cs.bc.edu

conditions for definability [7] and consequent proofs that certain properties cannot be expressed in a logic.

The results of Bojańczyk *et al.* [7], in particular, use the wreath product of forest algebras as the principal tool. Here we continue this line of work, providing a number of novel examples of how these algebraic tools can be applied. Our principal results are the following.

- We give simple new proofs of effective necessary conditions for definability in CTL and first-order logic with the ancestor relation. These appear in [7] as the absence of certain kinds of 'vertical confusion'. Our proof takes advantage of some natural expansion operations on forest algebras, which are of independent interest.
- We provide a sequence of identities that must be ultimately satisfied by the syntactic forest algebra of any language in CTL^* or PDL. These identities are a kind of generalized distributive law. We apply these results to show that there are languages in EF of arbitrarily high operator complexity within PDL, as well as a new proof that certain languages are not definable in $FO[\prec]$. While the non-definability results are known, the formulation in terms of identities is new.
- We consider the question of the overlap between LTL and CTL, previously studied by Maidl [11] and Bojańczyk [5]. We show that certain forest languages are outside CTL by methods that appear to be fundamentally new. We apply these to characterize the finite monoids M with the property that if L is any regular language of words recognized by M, then the set of forests with a maximal path in L is definable in CTL.

Our results, especially those concerning the generalized distributive laws and the overlap of CTL and LTL, suggest a number of open problems, which we discuss in the conclusion.

The paper illustrates the advantages of studying problems about logical expressibility in an algebraic setting. Once the machinery of forest algebras is in place, the applications to logic follow in a straightforward manner. The algebra is thus a source of new insights as well as a valuable complement to more traditional model-theoretic methods like Ehrenfeucht–Fraïssé games.

In Section 2 we provide a brief review of the algebraic model. We also take this opportunity to correct an error in the proof of a fundamental fact about division of finite forest algebras that appeared in [7]. Section 3 gives the definitions of the logics we consider in this paper. Because we deal with unranked finite forests, rather than infinite trees, the definitions of languages like CTL look somewhat different in this setting from the traditional formulations.

In Sections $4-6$ we provide our applications. Section 4 shows how the 'vertical confusion' criteria of [7] can be formulated and proved in a simple, natural fashion by using certain expansions of forest algebras. In Section 5 we establish identities that must be satisfied by forest languages in PDL, and use them to prove some non-definability and hierarchy results. In Section 6 we take up the question of the overlap between CTL and LTL. The concluding section lists some open problems.
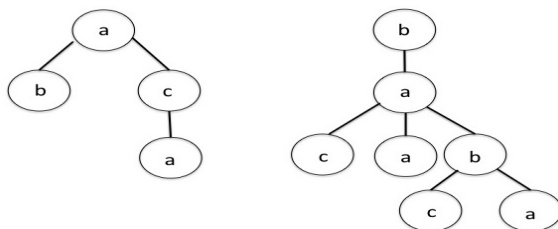
FIGURE 1. The forest $a(b + ca) + ba(c + a + b(c + a))$.

2. PRELIMINARIES

There are by now many basic surveys and a few books on the algebraic theory of automata over words, including the syntactic monoid, variety theory, and the ideal structure of finite semigroups. We refer the reader to Pin [13,14].

Forest algebras are much newer. Most of the basic material on these algebras and our particular approach to logics on trees can be found in Bojaǹczyk *et al.*, [3,7]. Here we repeat some of the principal definitions.

2.1. TREES AND FORESTS

Let A be a finite alphabet. We define *trees* and *forests* over A by mutual recursion: If s is a forest and $a \in A$, then as is a tree. If $t_1, \ldots, t_n$ is a finite sequence of trees, then $t_1 + \ldots + t_n$ is a forest. The recursion gets started with the empty sequence of trees, whose sum we denote by 0. Thus a forest is a formal expression like

$$a(b0 + ca0) + ba(c0 + a0 + b(c0 + a0)),$$

where $a, b, c \in A$. We usually drop the 0's when we write such expressions. We draw the forest in the obvious fashion as in Figure 1.

We then adopt the standard tree terminology, and write about *nodes, root* nodes and *leaf* nodes, *children, parents, ancestors and descendants* of nodes, the *subtree rooted at* a node, and the *forest of a node,* which consists of all the strict descendants of a node. We write t_x for the subtree rooted at the node x, and f_x for the forest of x. The set of all forests over A, is denoted H_A. H_A forms a monoid, with 0 as the identity element. Observe that the operation + is noncommutative.

2.2. CONTEXTS

If we replace one leaf of a forest by a *hole,* denoted □, we obtain a *context.* Figure 2 shows several contexts, both as diagrams and as formal expressions.

We denote the set of all contexts over A by V_A. Let $p, q \in V_A$, and $x \in H_A$. We obtain a new context pq upon replacing the hole in p by q, and a new forest ps upon replacing the hole in p by s. With these operations we have
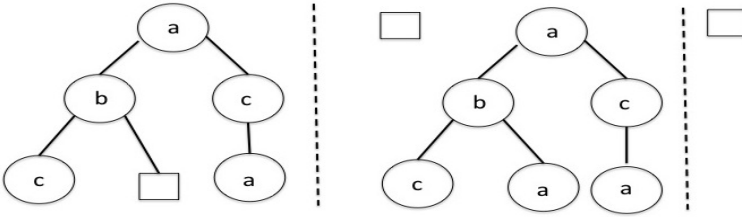
FIGURE 2. The contexts $a(b(c + \Box) + ca)$, $\Box + a(b(c + a) + ca)$, and $\Box$.

$$(pq)r = p(qr),$$

$$\Box p = p\Box = p,$$

$$p(qs) = (pq)s,$$

$$\Box s = s,$$

for any $p, q, r \in V_A$; $s \in H_A$. Thus V_A forms a monoid with respect to substitution, with $\Box$ as the identity, and substitution of a forest in a context defines a left action of the monoid V_A on H_A. (We emphasize that this is an action of a monoid on the *set H_A*: the operation in H_A plays no role here).

2.3. FOREST ALGEBRAS

The pair (H_A, V_A) is an instance of a special kind of algebraic structure, called a *forest algebra,* first introduced by Bojanczyk and Walukiewicz in [3]. In general, a forest algebra is a pair (H, V) satisfying the following properties:

- H is a monoid. The operation in H is written additively, and its identity element is accordingly denoted 0.
- V is a monoid. The operation in V is written multiplicatively, so its identity is usually denoted 1. (In some specific instances, as in the forest algebra (H_A, V_A), the identity is denoted $\Box$).
- There is a left action of the monoid V on the set H. This means

$$(v_1 v_2)h = v_1(v_2 h), 1 \cdot h = h,$$

for all $v_1, v_2 \in V, h \in H$.
- The action is *faithful.* That is, if $v_1 h = v_2 h$ for all $h \in H$, then $v_1 = v_2$.
- Let $g \in H$. Then there exist elements of V, which we denote $1 + g$ and $g + 1$, such that for all $h \in H$,

$$(g + 1)h = g + h, (1 + g)h = h + g.$$

In the case of (H_A, V_A), these elements are contexts of the form $\Box + s$, and $s + \Box$, where s is a forest, (See, for example, the second context in Fig. 2). Since $((g+1)v)h = (g+1)vh = g + vh$ for $v \in V$, we denote $(g+1)v$ by $g + v$, and similarly write $v + g$ for $(1 + g)v$.

We call H the *horizontal monoid* of (H, V), and V the *vertical monoid*. A homomorphism $\alpha : (H_1, V_1) \to (H_2, V_2)$ is actually a pair of monoid homomorphisms

$$\alpha_H : H_1 \to H_2, \alpha_V : V_1 \to V_2$$

that also respects the left action, in other words,

$$\alpha_V(v)\alpha_H(h) = \alpha_H(vh)$$

for all $v \in V, h \in H$. Usually we drop the subscripts H and V and simply write α for both components. Observe that if α maps onto V_2, then it maps onto H_2 as well, since we then have for any $h \in H_2$,

$$h = (1 + h) \cdot 0 = \alpha_V(v) \cdot 0 = \alpha_V(v)\alpha_H(0) = \alpha_H(v \cdot 0)$$

for some $v \in V_1$. Similarly, if two homomorphisms agree on the vertical monoid of the domain, they agree on the horizontal monoid.

(H_A, V_A) is called the *free forest algebra* on A, and is also denoted A^Δ. The free forest algebra has the following universal property: If (H, V) is a forest algebra, and $f : A \to V$ is any function, then there is a unique homomorphism $\alpha : A^\Delta \to (H, V)$ such that $\alpha(a\Box) = f(a)$ for all $a \in A$.

We now discuss a critical relation on forest algebras called *division.* Let (H_1, V_1), (H_2, V_2) be forest algebras, and let H' be a submonoid of H_2, V' a submonoid of V_2, such that V' contains all contexts $1 + h, h + 1$ with $h \in H'$ and $V'H' \subseteq H'$. We call (H', V') a *subalgebra* of (H_2, V_2), although strictly speaking it may fail to be a forest algebra, since V' might not act faithfully on H'. We say (H_1, V_1) *divides* (H_2, V_2) and write $(H_1, V_1) \prec (H_2, V_2)$, if there is such a subalgebra, together with an onto homomorphism $\alpha : (H', V') \to (H_1, V_1)$.

The theorem below gives an equivalent characterization of division. This is very close to, and in fact is slightly stronger than, a similar result (Lem. 4.2) of [7]. It is worth giving a careful proof, since there is a gap in the argument in [7], which is not entirely trivial to fill.

Theorem 2.1. *Let A be a finite alphabet, and ψ a homomorphism from A^Δ onto a forest algebra (H_1, V_1). Then $(H_1, V_1) \prec (H_2, V_2)$ if and only if there exists a submonoid K of H_2, an onto monoid homomorphism $\Phi : K \to H_1$, and for each $a \in A$, an element $\hat{a}$ of V_2 such that $\hat{a}K \subseteq K$, and $\Phi(\hat{a}h) = \psi(a\Box)\Phi(h)$ for all $h \in K$.*

Proof. One direction is trivial: If $(H_1, V_1) \prec (H_2, V_2)$, then the horizontal part of the forest algebra homomorphism Φ from a subalgebra (K, W) of (H_2, V_2) onto (H_1, V_1) provides us with the monoid homomorphism $\Phi : K \to H_1$. Further, for

each $a \in A$, there is an element $\hat{a} \in W$ such that $\Phi(\hat{a}) = \psi(a\square)$, and the desired property follows from the definition of forest algebra homomorphisms.

Conversely, suppose we have a homomorphism $\Phi : K \to H_1$, and a map $a \mapsto \hat{a}$ as in the statement of the theorem. By the universal property of free forest algebras, this map determines a unique homomorphism $\alpha : A^\Delta \to (H_2, V_2)$. We claim that for all $s \in H_A$, $\alpha(s) \in K$, and $\Phi\alpha(s) = \psi(s)$. In particular, $\alpha(s) = \alpha(s')$ implies $\psi(s) = \psi(s')$.

Assuming the claim is true, let us see how it implies $(H_1, V_1) \prec (H_2, V_2)$. We set

$$H' = \{\alpha(p \cdot 0) : p \in V_A\},$$

$$V' = \{\alpha(p) : p \in V_A\}.$$

By the claim, H' is contained in K. H' is a submonoid of K, since

$$\alpha(p \cdot 0) + \alpha(q \cdot 0) = \alpha(p \cdot 0 + q \cdot 0)$$
$$= \alpha((p + q \cdot 0) \cdot 0)$$
$$\in H',$$

and $0 = \alpha(1 \cdot 0) \in H'$. $V'H' \subseteq H'$, because $\alpha(p)\alpha(q \cdot 0) = \alpha((pq) \cdot 0)$. If $h \in H'$, then $h = \alpha(p \cdot 0)$ for some $p \in V_A$, and thus $1 + h = \alpha(\square + p \cdot 0) \in V'$, and likewise $h + 1 \in V'$. Thus (H', V') is a subalgebra, in the sense described above. We now define a pair of maps, both denoted θ, from H' to H_1 and from V' to V_1. If $h = \alpha(p \cdot 0)$ for some $p \in V_A$, then we set $\theta(h) = \psi(p \cdot 0)$. By our claim above, this is well-defined. θ is a homomorphism from H' to H_1, because if $s = \alpha(p \cdot 0)$, $t = \alpha(q \cdot 0)$, we have

$$\theta(s + t) = \theta(\alpha(p \cdot 0) + \alpha(q \cdot 0))$$
$$= \theta(\alpha((p + q \cdot 0) \cdot 0))$$
$$= \psi((p + q \cdot 0) \cdot 0)$$
$$= \psi(p \cdot 0 + q \cdot 0)$$
$$= \psi(p \cdot 0) + \psi(q \cdot 0)$$
$$= \theta(s) + \theta(t).$$

If $v \in V'$ and $v = \alpha(p)$, then we set $\theta(v) = \psi(p)$. We must show that this is well-defined: If $\alpha(p) = \alpha(q)$ and $h \in H_1$, then $h = \psi(s)$ for some $s \in H_A$. Thus $\alpha(ps) = \alpha(qs)$, so our claim gives

$$\psi(p)h = \psi(ps) = \psi(qs) = \psi(q)h,$$

and since h was arbitrary, faithfulness gives $\psi(p) = \psi(q)$. It is straightforward to verify that θ is a monoid homomorphism from V' to V_1, and that it preserves the left action and maps onto V_1. Thus $(H_1, V_1) \prec (H_2, V_2)$.

It remains to establish the claim. We prove by induction on the number of nodes in a forest $s \in H_A$ that $\Phi(\alpha(s)) = \psi(s)$. This is true if s is the empty forest 0.

If s is nonempty, it can either be written as $s_1 + s_2$, where s_1 and s_2 both have fewer nodes than s, or as as', where $a \in A$ and $s' \in H_A$ has fewer nodes than s. In the first case, $\Phi\alpha(s_1 + s_2) = \psi(s_1 + s_2)$ follows immediately from the inductive hypothesis and the fact that Φ, α and ψ are homomorphisms. In the second case, the inductive hypothesis and the property of $\hat{a}$ give

$$
\begin{aligned}
\Phi(\alpha(as')) &= \Phi(\alpha(a)\alpha(s')) \\
&= \Phi(\hat{a}\alpha(s')) \\
&= \psi(a\square)\Phi(\alpha(s')) \\
&= \psi(a\square)\psi(s') \\
&= \psi(as'). \qquad\qquad\square
\end{aligned}
$$

As mentioned above, there is a very similar theorem in [7] (essentially the case where the alphabet A is identical to V and ψ extends the identity map on A). In the proof of that result, it is claimed without justification that if $\alpha_i : A^\Delta \to (H_i, V_i)$ for $i = 1, 2$, are forest algebra homomorphisms, and if for all $p, q \in V_A$, $\alpha_1(p) = \alpha_1(q)$ implies $\alpha_2(p) = \alpha_2(q)$, then $(H_1, V_1) \prec (H_2, V_2)$. While the analogous property for semigroup or group homomorphisms is trivial, in the forest algebra setting it requires a careful argument.

2.4. FOREST LANGUAGES AND SYNTACTIC FOREST ALGEBRA

A set $L \subseteq H_A$ is called a *forest language.* Let $\alpha : A^\Delta \to (H, V)$ be a homomorphism, and let $X \subseteq H$. We say that $L = \alpha^{-1}(X)$ is *recognized* by α, and also that it is recognized by (H, V). If L is recognized by a finite forest algebra, then we say that L is a *regular forest language.* This coincides with the usual notions of regularity for unranked trees (see, for example, Libkin [10]): While we deal with forests rather than trees, L is a regular forest language in our formulation if and only if aL is a regular tree language for all $a \in A$.

If $L \subseteq H_A$ then we define an equivalence relation $\sim_L$ on H_A, called the *syntactic congruence* of L, by setting $s_1 \sim_L s_2$ if the sets $\{p \in V_A : ps_i \in L\}$ for $i = 1, 2$ are equal. This equivalence is compatible with the addition in H_A, and contexts in V_A act on equivalence classes in a well-defined manner. If $p, p' \in V_A$ we define $p \sim_L p'$ if p, p' induce the same map on equivalence classes; *i.e.,* $ps \sim_L p's$ for all $s \in H_A$. We thus obtain a homomorphism $\eta_L : A^\Delta \to (H_L, V_L)$, where $H_L = H_A/\sim_L$, and $V_L = V_A/\sim_L$. The equivalence $\sim_L$ was defined on contexts, precisely so that contexts that act identically on classes of forests are equivalent. Thus the action of V_L on H_L is faithful, so (H_L, V_L) is a forest algebra. (H_L, V_L) and η_L and are called, respectively, the *syntactic forest algebra* and *syntactic morphism* of L. We have the following fundamental theorem, an analogue of well-known properties of the syntactic monoids of word languages:

Theorem 2.2. *Let $L \subseteq H_A$. A homomorphism $\alpha : A^\Delta \to (H, V)$ recognizes L if and only if for all $s, t \in H_A$, $\alpha(s) = \alpha(t)$ implies $s \sim_L t$.*
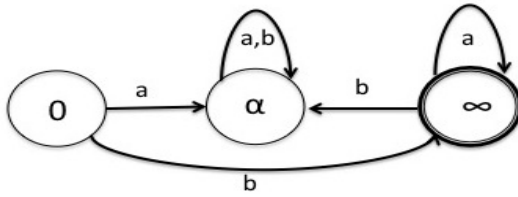
FIGURE 3. A forest algebra diagram. Here $\alpha + \alpha = \alpha$.

In particular, (H_L, V_L) divides any forest algebra recognizing L, and (H_L, V_L) is finite if and only if L is a regular forest language.

State diagrams for forest algebras. This paper contains a number of diagrams illustrating the syntactic forest algebras of various languages, and more generally homomorphisms α from A^Δ onto a finite forest algebra (H, V). Here we describe some of the conventions we use in these diagrams. Nodes in the diagram represent elements of H. We draw an arrow labeled a from h to h' if and only if $\alpha(a\square)h = h'$. The resulting labeled digraph, together with the addition in H, completely determine both α and the algebra (H, V). In the case where $\alpha = \eta_L$ is the syntactic morphism of a forest language L, L itself is a union of $\sim_L$ classes, and we will indicate a class in L by a doubled circular boundary on the corresponding node. Very often, we can simplify the presentation by stipulating some additional properties: For instance, most of the examples in this paper will have H idempotent and commutative. In this case, we can label the nodes in such a manner that the addition in H is easily inferred from the label. For example, the set of forests over $\{a, b\}$ in which some maximal path is in a^*b has its syntactic forest algebra given by the diagram in Figure 3. Here we need to stipulate one additional piece of information not implicit in the transitions or in the node labels 0 and ∞: namely that $\alpha + \alpha = \alpha$.

Wreath products. The wreath product is an operation on transformation monoids that extends in a straightforward manner to forest algebras. Given forest algebras $(H_1, V_1), (H_2, V_2)$ we define a new forest algebra

$$(H_1, V_1) \circ (H_2, V_2) = (H_1 \times H_2, V_1 \times V_2^{H_1}).$$

The addition in $H_1 \times H_2$ is simply the direct product, and the action is given by

$$(v, f) \cdot (h_1, h_2) = (vh_1, f(h_1)h_2).$$

It is easy to verify (see [7]) that $(H_1, V_1) \circ (H_2, V_2)$ is indeed a forest algebra. The wreath product of forest algebras is associative.

3. LOGICS FOR FOREST LANGUAGES

3.1. FIRST-ORDER AND GENERALIZED TEMPORAL LOGICS FOR FORESTS.

We describe some predicate and temporal logics for forests. While these are by and large well known, they are often encountered in the setting of infinite trees, rather than finite forests, where the definitions look a bit different.

First-order logic. Let A be a finite alphabet. Variables in first-order formulas represent nodes in a forest. The atomic formulas are of two kinds: $Q_a x$, where $a \in A$, means that node x is labeled a, while $x \prec y$ means node x is a (not necessarily strict) ancestor of node y. We can thus interpret sentences (formulas with no free variables) in forests. We write $s \models \phi$ if the sentence ϕ is true when interpreted in $s \in H_A$, and denote by L_ϕ the set of all $s \in H_A$ such that $s \models \phi$. We also say that the sentence ϕ *defines* the language L_ϕ.

For example, the set of all forests in which some maximal path belongs to $a^* b$ is defined by the sentence

$$\exists x (Q_b x \wedge \forall y (x \prec y \rightarrow x = y) \wedge \forall y (y \prec x \rightarrow x = y \vee Q_a y)).$$

The subformula $\forall y (x \prec y \rightarrow x = y)$ says that x is a leaf node, and the subformula $\forall y (y \prec x \rightarrow x = y \vee Q_a y)$ says that every strict ancestor of x is labeled a.

We denote both the family of languages defined by such sentences, as well as the family of sentences itself, by $FO[\prec]$.

Generalized temporal logic. Our temporal formulas come in two flavors–*tree formulas* and *forest formulas*. The syntax and semantics of both kinds of formulas are defined by mutual recursion. The syntax is given by these rules:

- If $a \in A$, then a is a tree formula.
- Every forest formula is a tree formula.
- $\top$ and $\bot$ are forest formulas.
- If ϕ, ψ are tree formulas (respectively, forest formulas), then so are $\phi \vee \psi$, $\phi \wedge \psi$, and $\neg \phi$.
- Let $\psi_1, \ldots, \psi_{k-1}$ be a collection of tree formulas. We define new tree formulas $\phi_1, \ldots, \phi_k$ by setting

$$\phi_1 = \psi_1,$$

$$\phi_i = \psi_i \wedge \bigwedge_{j < i} \neg \psi_j,$$

for $1 < i < k$, and

$$\psi_k = \bigwedge_{j < k} \neg \psi_j.$$

A family $\Phi = \{\phi_1, \ldots, \phi_k\}$ constructed in this way is said to be *unambiguous*. We treat such an unambiguous collection of formulas as a finite alphabet, and take a regular language of *words* $K \subseteq \Phi^*$. Then $\mathsf{E}K$ is a forest formula.

We interpret forest formulas in forests, and tree formulas in trees, and therefore we have two notions, $\models_{\text{forest}}$ and $\models_{\text{tree}}$. The rules defining these semantic relations parallel those given above for the syntax.

- Let $a \in A$, and t a tree over A. Then $t \models_{\text{tree}} a$ if and only if $t = as$, for some $s \in H_A$. (That is, the root node of t is labeled a.)
- Let ϕ be a forest formula, $a \in A$, and $s \in H_A$. Then $as \models_{\text{tree}} \phi$ if and only if $s \models_{\text{forest}} \phi$. (In other words, a tree t tree-satisfies ϕ if and only if the forest f_x of its root node x forest-satisfies ϕ.)
- For all $s \in H_A$, $s \models_{\text{forest}} \top$ and $s \not\models_{\text{forest}} \bot$.
- Boolean connectives $\vee, \wedge, \neg$ have their usual meanings for both tree-satisfaction and forest-satisfaction.
- Let Φ be an unambiguous family of tree formulas, and let $K \subseteq \Phi^*$. Let $s \in H_A$. By the way the formulas of Φ were constructed, we can label each node x of s by the unique $\phi \in \Phi$ such that $t_x \models_{\text{tree}} \phi$. Then $s \models_{\text{forest}} \mathsf{E}K$ if and only if there is a path of nodes $x_1 \ldots x_r$ beginning at a root of s (but not necessarily extending all the way to a leaf) whose sequence of labels $\phi_1 \ldots \phi_r$ belongs to K.

If ϕ is a forest formula, then we denote by L_ϕ the set of all forests $s \in H_A$ such that $s \models_{\text{forest}} \phi$.

We give some examples of how this formalism is used.

Example 3.1. Let $A = \{a, b\}$. The unambiguous set Φ constructed from the single tree formula a is then just $\{a, \neg a\}$, and since we are working over $\{a, b\}$, this is equivalent to taking $\Phi = \{a, b\}$. Let K be the word language a^*b. Then $\mathsf{E}K$ is satisfied by the forests that contain a node labeled b. Observe that this is not the same thing as the set of forests in which there is a maximal path in a^*b, which we saw in an earlier example. We abbreviate the formula $\mathsf{E}K$ by $\mathsf{EF}b$.

Example 3.2. More generally, if ψ is any tree formula, then the forest formula $\mathsf{E}((\neg\psi)^*\psi)$ defines the set of all forests in which some node tree-satisfies ψ. We denote this formula $\mathsf{EF}\psi$.

Example 3.3. Again let $A = \{a, b\}$. The forest formula $\mathsf{emp} = \neg(\mathsf{EF}a \vee \mathsf{EF}b)$ is satisfied only by the empty forest. (We can similarly define emp for any finite input alphabet A.) Let ψ_1 be the tree formula a, and ψ_2 the tree formula $b \wedge \mathsf{emp}$. The unambiguous family Φ constructed from $\{\psi_1, \psi_2\}$ is therefore $\{\psi_1, \psi_2, \neg\psi_1 \wedge \neg\psi_2\}$. Observe that the tree rooted at a node x tree-satisfies ψ_2 if and only if x is a leaf labeled b. The set of forests with a maximal path in a^*b is thus defined by the forest formula $\mathsf{E}(\psi_1^*\psi_2)$.

Example 3.4. Let ψ be any tree formula, so that $\Phi = \{\psi, \neg\psi\}$ is the unambiguous family constructed from $\{\psi\}$. The forest formula $\mathsf{E}\psi(\psi \cup \neg\psi)^*$ thus defines the set of forests in which the tree rooted at some root node satisfies ψ. (Alternatively, we could simply write this as $\mathsf{E}\psi$.) We abbreviate this formula $\mathsf{EX}\psi$. So, for instance,

in the simplest case $A = \{a, b\}$, the forest formula $\mathsf{EX}a$ defines the set of forests in which some root node is labeled a, and the tree formula $\mathsf{EX}a$ is satisfied by those trees in which some child of the root is labeled a.

This extended temporal logic has been variously called 'chain logic' or 'propositional dynamic logic', although these terms were originally introduced in other contexts. We will use PDL here to maintain consistency with the notation of [7]. If we restrict application of the operator $\mathsf{E}K$ to word languages K that are themselves definable by first-order sentences over $<$, we obtain the logic CTL^*.

Let ϕ, ψ be any tree formulas. The unambiguous family Φ obtained from $\{\phi, \psi\}$ is $\{\phi, \psi \wedge \neg\phi, \neg\psi \wedge \neg\phi\}$. We write $\mathsf{E}\psi\mathsf{U}\phi$ as an abbreviation for the forest formula $\mathsf{E}(\psi \wedge \neg\phi)^*\phi$. If $s \in H_A$ then $s \models_{\mathrm{forest}} \mathsf{E}\psi\mathsf{U}\phi$ if and only if there is a node x such that $t_x \models_{\mathrm{tree}} \phi$, and for all ancestors y of x, $t_y \models_{\mathrm{tree}} \psi$. In other words, there is a path on which 'ψ holds until ϕ'.

CTL denotes the family of languages definable by formulas in which application of the $\mathsf{E}K$ operator is restricted to have this form. We also denote by CTL the family of such formulas. Note that we can recover both the EX and EF operators in CTL, since $\mathsf{EX}\psi$ is equivalent to $\mathsf{E}\bot\mathsf{U}\psi$, and $\mathsf{EF}\psi$ is equivalent to $\mathsf{E}(\neg\psi)\mathsf{U}\psi$.

The family of forests over $\{a, b\}$ with a maximal path in a^*b is defined by the CTL formula $\mathsf{E}a\mathsf{U}(b \wedge \mathsf{emp})$. Using a similar trick, we can show how to obtain a kind of universal quantification within CTL: Usually CTL is defined for infinite trees, and traditional treatments include another operator, that allows one to express 'there exists a maximal path that does not satisfy 'ψ until ϕ'' [9]. To see how to do this in our present framework for finite forests, we may assume that ψ is equivalent to $\psi \wedge \neg\phi$, as this does not change the meaning of 'ψ until ϕ', and set ρ to be $\neg\psi \wedge \neg\phi$. A maximal path that does not satisfy 'ψ until ϕ' either has the property that every node satisfies ψ, or there is a node satisfying ρ such that every strict ancestor of the node satisfies ψ. Thus the existence of such a path is given by the formula

$$\mathsf{E}\psi\mathsf{U}(\psi \wedge \mathsf{emp}) \vee \mathsf{E}\psi\mathsf{U}\rho.$$

There is an important subtlety here. Consider the property 'there are two consecutive b's', which is the same as 'there is a maximal path with two consecutive b's'. One might expect the negation trick just described to allow us to express 'there exists a maximal path without two consecutive b's'. However when write the original property as

$$\mathsf{EF}(b \wedge \mathsf{EX}b),$$

and apply the negation trick, we get the formula

$$\mathsf{E}\neg(b \wedge \mathsf{EX}b)\mathsf{U}(\neg(b \wedge \mathsf{EX}b) \wedge \mathsf{emp}),$$

which simplifies to
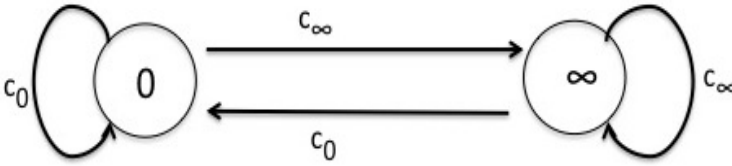
$$\mathsf{E}\neg(b \wedge \mathsf{EX}b)\mathsf{Uemp}.$$

FIGURE 4. The forest algebra $\mathcal{U}_2$.

This says that there is a maximal path along which every node is either a, or is labeled b but has no child labeled b. Observe that this is *not* the same thing as 'there is a maximal path without two consecutive b's'. For example, the forest $b(b + a)$ has a maximal path without two consecutive b's, but does not satisfy the formula. As we shall see later on (Thm. 6.1), the property 'there exists a maximal path without two consecutive b's' is not expressible in CTL.

3.2. WREATH PRODUCT CHARACTERIZATION OF LOGICALLY DEFINED FAMILIES

The principal result of [7] is a characterization of the languages definable in these logics in terms of iterated wreath products of certain basic forest algebras. We state the relevant parts of this theorem below.

We say that a finite forest algebra (H, V) is *horizontally idempotent and commutative* if H is an idempotent and commutative monoid. (H, V) is *distributive* if $v(h_1 + h_2) = vh_1 + vh_2$ holds for all $v \in V$; $h_1, h_2 \in H$. (H, V) is *aperiodic* if V is an aperiodic monoid (*i.e.,* V contains no nontrivial groups) and (H, V) is *flat* if H is commutative and $V = \{1 + h : h \in H\}$.

The forest algebra $\mathcal{U}_2$ is given by Figure 4.

The conventions about the use of the symbols 0 and ∞ imply that $\mathcal{U}_2$ is horizontally idempotent and commutative. The diagram only shows generators c_0, c_∞ of the vertical monoid, but it is easy to verify that apart from the identity, these are the only elements, since $c_x c_y = c_x$ for any $x \in \{0, \infty\}$, and $1 + 0 = 1, 1 + \infty = c_\infty$. $\mathcal{U}_2$ is distributive and aperiodic, but not flat. (Recall that $1 + 0$ and $1 + \infty$ are instances of the operation $(v, h) \mapsto v + h$, and represent elements of the vertical monoid).

We associate to various logics $\mathcal{L}$ a family $\mathcal{A}_\mathcal{L}$ of basic forest algebras, as follows:

- $\mathcal{A}_{PDL}$ consists of the horizontally idempotent and commutative, distributive forest algebras.
- $\mathcal{A}_{CTL^*}$ consists of the algebras in $\mathcal{A}_{PDL}$ that are also aperiodic.
- $\mathcal{A}_{FO[\prec]}$ consists of the algebras in $\mathcal{A}_{CTL^*}$ together with the flat aperiodic algebras.
- $\mathcal{A}_{CTL} = \{\mathcal{U}_2\}$.

Theorem 3.5. *Let A be a finite alphabet, and let $L \subseteq H_A$. Let $\mathcal{L}$ be any of the logics PDL, CTL^*, $FO[\prec], CTL$. Then L is definable in $\mathcal{L}$ if and only if it is recognized by a wreath product*

$$(H_1, V_1) \circ \ldots \circ (H_k, V_k),$$

where each (H_i, V_i) is in $\mathcal{A}_\mathcal{L}$.

(It should be pointed out that the statement of this theorem in [7] for the logic $FO[\prec]$ is slightly different: There, only one kind of base algebra for $FO[\prec]$ is given: these are called *aperiodic path algebras*. However, it is easier to prove this modified statement of the theorem, using identical techniques).

We give a refined version of this theorem for the logics PDL and CTL^*. Let $k > 0$. We define PDL_k to consist of those languages in PDL defined by formulas in which the depth of nesting of the $\mathsf{E}K$ operator is no more than k, and define CTL^*_k analogously.

Theorem 3.6. *Let A be a finite alphabet, and let $L \subseteq H_A$. Let $k > 0$. Then L is definable in PDL_k (respectively, CTL^*_k) if and only if it is recognized by a wreath product*

$$(H_1, V_1) \circ \ldots \circ (H_k, V_k),$$

where each (H_i, V_i) is in $\mathcal{A}_{PDL}$ (respectively $\mathcal{A}_{CTL^*}$).

Proof. We merely sketch how the argument in [7] needs to be modified. There it is shown that (a) If Φ is an unambiguous collection of tree formulas, and $K \subseteq \Phi^*$ a regular language, then $L_{\mathsf{E}K}$ is recognized by

$$(H_1, V_1) \circ (H_2, V_2),$$

where (H_2, V_2) is in $\mathcal{A}_{PDL}$ and (H_1, V_1) recognizes all the forest languages underlying Φ. (More precisely, if $\phi \in \Phi$, then ϕ can be written as a boolean combination of formulas $a \wedge \psi$, where ψ is a forest formula, and (H_1, V_1) recognizes all the L_ψ.) If further K is first-order definable, then $(H_2, V_2) \in \mathcal{A}_{CTL^*}$. ($b$) Conversely, suppose $L \subseteq H_A$ is recognized by a wreath product $(H_1, V_1) \circ (H_2, V_2)$, where (H_2, V_2) is in $\mathcal{A}_{PDL}$. Then L is a boolean combination of languages of the form $L_{\mathsf{E}K}$, where $K \subseteq \Phi^*$ and each $\phi \in \Phi$ is a boolean combination of $a \wedge \psi$, where $L_\psi \subseteq H_A$ is recognized by (H_1, V_1). In case $(H_2, V_2) \in \mathcal{A}_{CTL^*}$, the languages K can be chosen to be first-order definable.

The correspondence between operator depth and length of the wreath product now follows from these properties:

- If $L_1, \ldots, L_r$ are recognized by $(H_1, V_1), \ldots, (H_r, V_r)$, then all boolean combinations of the L_i are recognized by the direct product

$$\prod_{i=1}^{r} (H_i, V_i).$$

- Let (H_{ij}, V_{ij}), $1 \leq i \leq r$, $j = 1, 2$, be forest algebras. Then

$$\prod_{j=1}^{r} \big((H_{i1}, V_{i1}) \circ (H_{i2}, V_{i2})\big) \prec \big(\prod_{j=1}^{r}(H_{i1}, V_{i1})\big) \circ \big(\prod_{j=1}^{r}(H_{i2}, V_{i2})\big).$$

  In fact, the left-hand side is a subalgebra of the right-hand side. This is a well-known fact about the wreath product of transformation monoids that works without modification for forest algebras. See, for example, Eilenberg [8].

- If $(H_1, V_1), \ldots, (H_r, V_r)$ are in $\mathcal{A}_{PDL}$ (respectively, $\mathcal{A}_{CTL^*}$) then $\prod_{i=1}^{r}(H_i, V_i)$ is in $\mathcal{A}_{PDL}$ (respectively, $\mathcal{A}_{CTL^*}$). $\square$

4. APERIODICITY

If we forget about the additive structure of the horizontal monoids, a wreath product of forest algebras is just a wreath product of transformation monoids. As is well known, the wreath product preserves aperiodicity, and aperiodicity is preserved under homomorphic images and taking submonoids. It thus follows from Theorem 3.5 that for every L in $FO[\prec]$, V_L is aperiodic. Of course this holds for the subclasses CTL^* and CTL as well. We do not need anything as fancy as the wreath product to prove aperiodicity of the vertical monoids for these classes, but Theorem 3.5 provides some additional information that allows us to prove a stronger necessary condition based on aperiodicity.

Let (H, V) be a forest algebra. We can define the sum of two maps v, w from H to H by pointwise addition: for all $h \in H$,

$$(v + w)h = vh + wh.$$

We denote by $\hat{V}$ the smallest set of maps that contains V and that is closed under both addition and composition. $(H, \hat{V})$, which we also denote $\widehat{(H, V)}$ is thus a forest algebra with an additional additive structure on the vertical monoid: It is a *seminearring*, and we call it the *seminearring closure* of (H, V). (See Bojańczyk [2]). Just as elements of V are represented by contexts over A, elements of $\hat{V}$ are represented by *multicontexts*. A multicontext p may have several holes; p acts on a forest s by substituting s for each of the holes of p, giving a forest ps. See Figure 5.

The following lemma collects some elementary facts about the seminearring closure.

Lemma 4.1. *Let $(H_1, V_1), (H_2, V_2)$ be forest algebras.*
(a) If $(H_1, V_1) \prec (H_2, V_2)$, then $(H_1, \hat{V}_1) \prec (H_2, \hat{V}_2)$.
(b)
$$\widehat{(H_1, V_1) \circ (H_2, V_2)} \prec (H_1, \hat{V}_1) \circ (H_2, \hat{V}_2).$$

Proof. (a) Pick a finite alphabet A for which there is an onto homomorphism $\psi : A^{\Delta} \to (H_1, V_1)$. There then exist a homomorphism Φ from a submonoid H' of H_2 onto H_1, and an assignment $a \mapsto \hat{a}$ for each $a \in A$, that satisfy the conditions
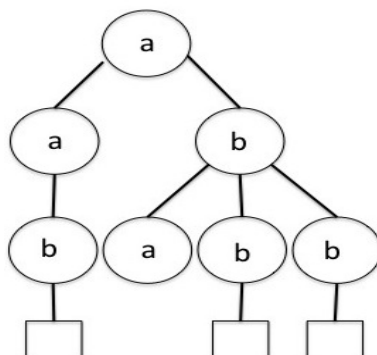
FIGURE 5. A multicontext.

in Theorem 2.1. By another application of this theorem, this time with $\hat{V}_1$ as the underlying alphabet, it will suffice to show that for every $v \in \hat{V}_1$ there exists $\hat{v} \in \hat{V}_2$ such that $\Phi(\hat{v}h) = v\Phi(h)$ for all $h \in H'$.

To do this, we extend the map ψ to multicontexts over A: Given a multicontext p we associate an element $\psi(p) \in \hat{V}_1$ by induction on the size of p:

$$\psi(p_1 + p_2) = \psi(p_1) + \psi(p_2), \psi(ap) = \psi(a)\psi(p).$$

We can similarly extend the map $a \mapsto \hat{a}$ to multicontexts, defining $\hat{p} \in \hat{V}_2$ by

$$\widehat{p_1 + p_2} = \hat{p}_1 + \hat{p}_2, \widehat{ap} = \hat{a}\hat{p}.$$

At each step in the construction of p, the following property is preserved: for any $h \in H'$,

$$\Phi(\hat{p}h) = \psi(p)\Phi(h).$$

Given $v \in \hat{V}_1$, we can represent it by a multicontext p_v such that $\psi(p_v) = v$. We then set $\hat{v} = \widehat{p_v}$, which gives the desired division.

(b) Let $(H, V) = (H_1, V_1) \circ (H_2, V_2)$. The vertical monoid W of $(H, \hat{V}_1) \circ (H, \hat{V}_2)$ is closed under composition, by definition of the wreath product, and it contains V. So it suffices to prove that W is closed under addition as well, since this will show that $(H_1, \hat{V}_1) \circ (H_2, \hat{V}_2)$ is a seminearring, and thus contains $\widehat{(H, V)}$. (This is a stronger result than the statement in the Lemma, since we obtain inclusion rather than just division).
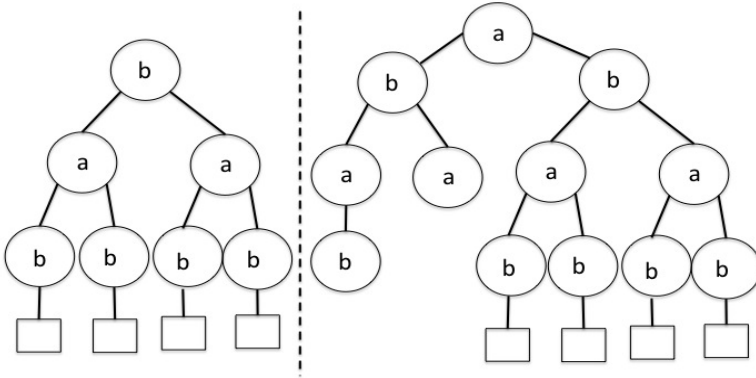
FIGURE 6. A pair of uniform multicontexts.

Let $(v, f), (w, g) \in W$, so that $v, w \in \widehat{V_1}$ and $f, g : H_1 \to \widehat{V_2}$. Then given $h_1 \in H_1, h_2 \in H_2$, we have

$$
\begin{aligned}
[(v, f) + (w, g)](h_1, h_2) &= (v, f)(h_1, h_2) + (w, g)(h_1, h_2) \\
&= (vh_1, f(h_1)h_2) + (wh_1, g(h_1)h_2) \\
&= (vh_1 + wh_1, f(h_1)h_2 + g(h_1)h_2) \\
&= ((v + w)h_1, (f(h_1) + g(h_1))h_2) \\
&= (v + w, F)(h_1, h_2),
\end{aligned}
$$

where $F : H_1 \to \widehat{V_2}$ is defined by $F(h) = f(h) + g(h)$ for all $h \in H_1$. Thus W is closed under addition, as claimed. □

We define another forest algebra $\widetilde{(H, V)} = (H, \tilde{V})$, intermediate between (H, V) and $(H, \hat{V})$, which we call the *uniform closure* of (H, V). $\tilde{V}$ is the smallest set containing V and closed under the composition and the operations

$$
v \mapsto v + \ldots + v.
$$

Note that since H is finite, there are only finitely many distinct sums of this form, so $\tilde{V}$ is finite and is effectively computable from V. Given a homomorphism $\psi : A^\Delta \to (H, V)$ we can represent elements of $\hat{V}$ by *uniform multicontexts* over A, which are obtained by applying these same operations to V_A.

Figure 6 shows a pair of uniform multicontexts. Observe that the multicontext on the right is obtained from the one on the left by composing with the context $a(b(ab + a) + 1)$. (In particular, this relaxes the requirement in [7] that in a uniform multicontext all the subtrees at each fixed level are identical).

The analogue of Lemma 4.1 for the uniform closure holds as well. The proof is identical: We simply substitute uniform multicontexts for arbitrary ones.

Lemma 4.2. *Let* $(H_1, V_1), (H_2, V_2)$ *be forest algebras.*
(a) If $(H_1, V_1) \prec (H_2, V_2)$, *then* $(H_1, \tilde{V}_1) \prec (H_2, \tilde{V}_2)$.
(b)

$$(H_1, \widetilde{V_1}) \circ (H_2, V_2) \prec (H_1, \tilde{V}_1) \circ (H_2, \tilde{V}_2).$$

The notions of seminearring closure and uniform closure lead to succinct statements and simple proofs of effective necessary conditions for definability in CTL and $FO[\prec]$. These were first proved in [7] in terms of absence of 'vertical confusion'.

Theorem 4.3. *Let* $L \subseteq H_A$.
(a) If L *is definable in* CTL, *then* $\widehat{V_L}$ *is aperiodic.*
(b) If L *is definable in* $FO[\prec]$, *then* $\widetilde{V_L}$ *is aperiodic.*

Proof. *(a)* Suppose L is definable in CTL. Then by Theorem 3.5 and Lemma 4.1,

$$(\widehat{H_L, V_L}) \prec \mathcal{U}_2 \widehat{\circ \ldots \circ} \mathcal{U}_2$$
$$\prec \widehat{\mathcal{U}_2} \circ \ldots \circ \widehat{\mathcal{U}_2}$$

But observe that the vertical monoid of $\widehat{\mathcal{U}_2}$ is already closed under addition, so that $\widehat{\mathcal{U}_2} = \mathcal{U}_2$. Thus (H_L, V_L) divides a wreath product of copies of $\mathcal{U}_2$. Since $\mathcal{U}_2$ is aperiodic, and aperiodicity is preserved under wreath product and division, V_L is aperiodic.

(b) The idea of the proof is the same. Theorem 3.5 and Lemma 4.2 imply

$$(\widetilde{H_L, V_L}) \prec (H_1, \widetilde{V_1}) \circ \ldots \circ (H_k, \widetilde{V_k}),$$

where each (H_i, V_i) is in $\mathcal{A}_{FO[\prec]}$. It remains to show that for each such algebra, $\widetilde{V}_i$ is aperiodic. We do this for each of the two kinds of algebras in $\mathcal{A}_{FO[\prec]}$: Suppose first that (H_i, V_i) is horizontally idempotent and commutative, distributive and aperiodic. Idempotence implies $v + \ldots + v = v$. Thus V_i is already closed under the operation $v \mapsto v + \ldots + v$, and so $\widetilde{V}_i = V_i$ is aperiodic. For the case where (H_i, V_i) is flat and aperiodic, we claim that every $v \in \widetilde{V}_i$ has the form

$$v : g \mapsto kg + h,$$

where $k \in \mathbf{Z}^+$ and $h \in H_i$. Certainly, each $v \in V_i$ has this form, since by flatness such a v maps g to $g + h$ for some $h \in H_i$. The set of such maps is also clearly closed under composition and the operations $v \mapsto v + \ldots + v$, which establishes the claim. Thus if $r > 0$, v^r maps $g \in H$ to

$$k^r g + (1 + k + \ldots + k^{r-1}) \cdot h.$$

Since H is aperiodic, there exists $M > 0$ such that $M'f = Mf$ for all $M' > M$, $f \in H$. Thus $v^r g = v^{r+1} g$ for sufficiently large r. Since every $v \in \widetilde{V}_i$ satifies $v^r = v^{r+1}$, $\widetilde{V}_i$ is aperiodic. $\qquad\square$
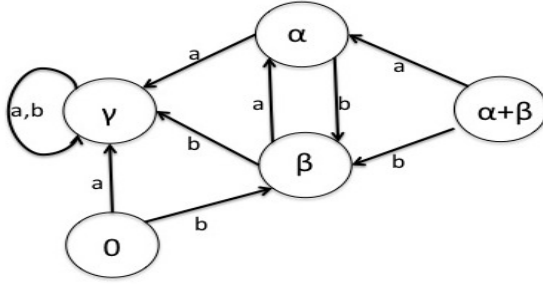
FIGURE 7. The syntactic forest algebra of $\mathsf{E}'(ab)^+$.

We recall the applications of this theorem from [7]. If $L \subseteq A^*$ is a regular word language, then we denote by $\mathsf{E}'L$ the set of forests that contain a maximal path whose sequence of labels is in L. (Note the difference between $\mathsf{E}'L$ and $\mathsf{E}L$ – the former concerns only paths that extend all the way to a leaf).

Proposition 4.4. $\mathsf{E}'(ab)^+$ *is not definable in CTL.*

Proof. Let $K = \mathsf{E}'(ab)^+$. It is easy to verify that H_K is idempotent and commutative, and the syntactic forest algebra is given by Figure 7, together with the rules $\alpha + \gamma = \alpha,\ \beta + \gamma = \beta$.
  Let $v \in \widehat{V_K}$ be the element represented by the multicontext $a\square + b\square$. Then

$$v\alpha = a\alpha + b\alpha = \gamma + \beta = \beta, v\beta = a\beta + b\beta = \gamma + \alpha = \alpha.$$

Thus v interchanges α and β, so $\widehat{V_K}$ contains a nontrivial group. By Theorem 4.3, K is not definable in CTL. $\square$

An example of an application of the second part of Theorem 4.3, also discussed in [7], is the set of L all binary trees in which some path has even length. Since these are unlabeled trees, we are working over a unary alphabet $\{a\}$. A simple computation shows that the uniform multicontext $a(\square + \square)$ interchanges two elements of H_L. Thus $\widetilde{V_L}$ contains a nontrivial group, so L is not definable in $FO[\prec]$. This example is interesting in light of the fact, discovered by Potthoff [15], that L is definable in first-order logic for ordered forests, in which there is an 'older sibling' as well as an 'ancestor' predicate.

5. IDENTITIES AND GENERALIZED DISTRIBUTIVITY

Let $\Xi = \{x_1, x_2, \ldots\}$ be a countable alphabet. An *identity* is a formal equation $s = t$, where $s, t \in H_\Xi$. A forest algebra (H, V) *satisfies* the identity $s = t$, written

$$(H, V) \models (s = t)$$

if and only if for every homomorphism $\psi : \Xi^\Delta \to (H, V)$, $\psi(s) = \psi(t)$.

Some of the conditions on forest algebras that we discussed earlier can be expressed in terms of identities. For example, the horizontally idempotent and commutative forest algebras are those that satisfy the identities

$$x_1 + x_2 = x_2 + x_1, x_1 + x_1 = x_1.$$

Distributive forest algebras are those that satisfy the identity

$$x_1(x_2 + x_3) = x_1x_2 + x_1x_3.$$

(As we proceed, we will informally drop the requirement that all variables in identities have the form x_i, and write identities as $x(y + z) = xy + xz$, $x + y = y + x$, *etc.*) Forest algebras (H, V) in which V is aperiodic are those that satisfy $x^n y = x^{n+1} y$ for some $n > 0$.

A *pseudovariety* of finite forest algebras is a family of algebras closed under finite direct products and division. The general theory of pseudovarieties of finite monoids and their associated identities, as presented, for example, in [8], extends to forest algebras. In particular, if

$$(s_i = t_i)_{i \geq 1}$$

is a sequence of identities, then the family of forest algebras that satisfy all but finitely many of the identities of the sequence forms a pseudovariety. The converse, which is far less obvious, is also true: All pseudovarieties arise in this way.

We now exhibit a particular sequence of identities that can be thought of as a kind of generalized distributive law:

$$s_1 = x_1x_2(y_1 + y_2), t_1 = x_1(x_2y_1 + x_2y_2).$$

For $n \geq 1$,

$$s_{n+1} = x_{2n+1}x_{2n+2}(s_n + t_n), t_{n+1} = x_{2n+1}(x_{2n+2}s_n + x_{2n+2}t_n).$$

Let $k > 0$, and let $\mathbf{PDL}_k$ denote the forest algebras that divide a wreath product of k members of $\mathcal{A}_{PDL}$. As we showed in Theorem 3.6, each $\mathbf{PDL}_k$ is a pseudovariety, and a forest language is definable in PDL_k if and only if its syntactic forest algebra is in $\mathbf{PDL}_k$.

Theorem 5.1. *Let $k > 0$. If $(H, V) \in \mathbf{PDL}_k$ and $x \in \Xi$, then $(H, V) \models (s_k = t_k)$.*

Proof. It suffices to show that if $(H_1, V_1) \models (s = t)$ and (H_2, V_2) is distributive and horizontally idempotent and commutative, then

$$(H_1, V_1) \circ (H_2, V_2) \models \big(x(s + t) = xs + xt\big).$$

When we substitute elements of the wreath product for the variables in s and t respectively, we obtain $(h_1, h_2), (k_1, k_2) \in H_1 \times H_2$. Since we suppose $(H_1, V_1) \models (s = t)$, we have $h_1 = k_1$. Let us denote their common value by h.

If we now substitute the vertical element $(v, f) \in V_1 \times V_2^{H_1}$ for s, we obtain (using distributivity in the right coordinate, and idempotence and commutativity in the left)

$$\begin{aligned}
x(s + t) &= (v, f)((h, h_2) + (h, k_2)) \\
&= (v, f)(h, h_2 + k_2) \\
&= (vh, f(h)(h_2 + k_2)) \\
&= (vh + vh, f(h)h_2 + f(h)k_2) \\
&= (vh, f(h)h_2) + (vh, f(h)k_2) \\
&= (v, f)(h, h_2) + (v, f)(h, k_2) \\
&= xs + xt. \qquad \square
\end{aligned}$$

We will give two applications of this identity-based approach to PDL. The first is a proof that the hierarchy based on operator depth in PDL is infinite. This differs from similar-looking hierarchy results concerning CTL^* in Shamir *et al.* [17] and Bojańczyk [4] in the exact way the hierarchy is formed. However, we do not really view our hierarchy result here as a new fact so much as an illustration of a new technique for proving such results.

To this end, we define two sequences of temporal formulas $\{\alpha_n\}, \{\beta_n\}$, for $n \geq 1$, over the alphabet $\{a, b\}$. All the formulas use the temporal operator EF.

$$\alpha_1 : \mathsf{EF}a, \quad \beta_1 : \mathsf{EF}b,$$

$$\alpha_{n+1} : \mathsf{EF}(a \wedge \alpha_n \wedge \beta_n),$$

$$\beta_{n+1} : \mathsf{EF}(a \wedge \alpha_n) \wedge \mathsf{EF}(a \wedge \beta_n) \wedge \neg\alpha_{n+1}.$$

To understand these formulas, look at the case $n = 2$: α_2 says there is a node labeled a with strict descendants labeled a and b. β_2 says that there is no such node, but there is a node labeled a with a strict descendant labeled a and another node labeled a with a strict descendant labeled b. For example, the forest $ba(a+b)$ satisfies α_2, while $b(aa + ab)$ satisfies β_2.

Theorem 5.2. *Let $k > 0$. The syntactic forest algebras of L_{α_k} and L_{β_k} are in* $\mathbf{PDL}_k \backslash \mathbf{PDL}_{k-1}$.

Proof. Membership in $\mathbf{PDL}_k$ follows immediately from the nesting depth of the formulas α_k and β_k, and Theorem 3.6. To show non-membership in $\mathbf{PDL}_{k+1}$, we consider the following sequences $\{s'_n\}, \{t'_n\}$ of forests over $\{a, b\}$:

$$s'_1 = a, t'_1 = b,$$

$$s'_{n+1} = a(s'_n + t'_n), t'_{n+1} = as'_n + at'_n.$$

This is a specialization of the forests s_i, t_i introduced earlier for Theorem 5.1, we are just considering the special case where $x_i = 1$ for odd i and $x_i = a$ for even i.

Theorem 5.1 implies that for all $n > 1$, $s'_n = t'_n$ is an identity for $\mathbf{PDL}_{n-1}$. We also have, by a straighforward induction, that for all $n \geq 1$,

$$s'_n \models \alpha_n \wedge \neg\beta_n, t'_n \models \beta_n \wedge \neg\alpha_n.$$

These observations give us the desired result, for if, say, the syntactic forest algebra L_{α_n} were in $\mathbf{PDL}_{n-1}$, then s'_n and t'_n would be mapped by the syntactic morphism to the same element, and thus we would have s'_n, t'_n either both in L_{α_n}, or both outside it, a contradiction. The same applies, of course, to β_n. $\square$

An immediate corollary is

Corollary 5.3. *Let $n > 0$. Neither α_n nor β_n is equivalent to a PDL formula with nesting depth strictly less than n.*

Thus, while the EF operator is extremely weak in one sense, it can climb to all levels of the depth hierarchy within PDL.

As a second application of Theorem 5.1, we give a simple new proof of what has become a kind of standard counterexample. We consider forests over the alphabet $\{\wedge, \vee, 0, 1\}$. Each component of such a forest is a tree, and as long as the tree only has 0 and 1 as leaf labels, and only has $\wedge$ and $\vee$ as labels of interior nodes, it represents a well-formed boolean expression. We define L to be the set of all such forests in which every component is an expression tree that evaluates to True. For example $\vee(0 + 1) \in L$, but $\wedge(0 + 1) \notin L$.

Theorem 5.4. *L is not definable in $FO[\prec]$.*

Proof. Observe that H_L is idempotent and commutative. (This is the reason for defining L as the set of forests of valid expression trees, rather than just individual trees). In [7] it is proved that any language definable in $FO[\prec]$ that is horizontally idempotent and commutative is definable in PDL, and in fact in CTL^*. (See also Moller and Rabinovich [12]). We will consequently suppose that $(H_L, V_L) \in \mathbf{PDL}_n$ for some n and derive a contradiction. We define another specialization of our identities:

$$s'_0 = 0, t'_0 = 1,$$

and

$$s'_{n+1} = \vee \wedge (s'_n + t'_n), t'_{n+1} = \vee(\wedge s'_n + \wedge t'_n).$$

If we interpret these forests as left- and right-hand sides of identities, then Theorem 5.1 implies that $s'_n = t'_n$ is an identity for $\mathbf{PDL}_n$. But note that for each n, s'_n is a false boolean expression, and t'_n a true one, so s'_n and t'_n have different values under the syntactic morphism of L. Thus L cannot be in $\mathbf{PDL}_n$ for any n. $\square$

6. The overlap of CTL and LTL

Theorems 4.3 and 5.1 provide necessary conditions for definability in various logics. It would be very nice if any of these conditions turned out to be sufficient. We will discuss this question in more detail in the conclusion. But we can say right away that the aperiodicity conditions for CTL and $FO[\prec]$ given in Theorem 4.3 are *not* sufficient: Theorem 5.4, concerning the language L of true boolean expression forests, provides a counterexample in both cases, since one can check that $\widehat{V_L}$ is aperiodic.

Still, one might hope to rescue a weaker sufficiency result from the aperiodicity condition for CTL. Our counterexample was a language of the form $\mathsf{E}'L$, where L is a word language. Is it possible that for all such languages aperiodicity of $\widehat{V_{\mathsf{E}'L}}$ implies that $\mathsf{E}'L$ is definable in CTL?

If a language of the form $\mathsf{E}'L$ is in CTL, then L must be aperiodic, and consequently definable in LTL. We are thus addressing what, in other guises, is referred to as the 'common fragment' of LTL and CTL (Maidl [11], Bojańczyk [5]). For which regular languages L is $\mathsf{E}'L$ definable in CTL?

We give a partial solution to this problem by characterizing the monoids M such that $\mathsf{E}'L$ is in CTL for every $L \subseteq A^*$ recognized by M. To do so, we will first exhibit a language of the form $\mathsf{E}'L$ that is not definable CTL but for which $\widehat{V_{\mathsf{E}'L}}$ is nonetheless aperiodic. This shows both the extent to which the methods already exhibited for proving nondefinability can fail, and how a more careful analysis of the wreath product can provide the sought-after result.

6.1. Another language not in CTL

Let $A = \{a, b\}$. For the remainder of this subsection, let $L \subseteq H_A$ consist of all nonempty forests in which every maximal path has two consecutive occurrences of b.

Theorem 6.1. *L is not definable in CTL.*

Proof. We compute the syntactic forest algebra (H_L, V_L) of L. If $s \in L$ and $p \in V_A$, then $ps \in L$ if and only if every maximal path in p, except possibly the one leading to the hole, contains two consecutive occurrences of b. Thus all elements of L are equivalent under the syntactic congruence, so L forms a single congruence class. We will provisionally denote this class by $0'$: we'll explain this choice of notation later.

Suppose $s \notin L$, but that every root node of s is labeled b. Let $p \in V_A$. Then $ps \in L$ if and only if the following conditions hold:

  Every maximal path in p, except the path of ancestors of the hole, contains two consecutive b's.
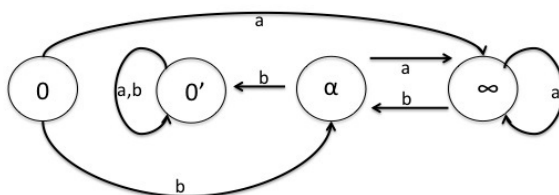 The parent of the hole is labeled b, or the path of ancestors of the hole contains two consecutive b's.

FIGURE 8. The syntactic forest algebra of L. Addition is idempotent and commutative, with $\alpha + 0' = \alpha$.
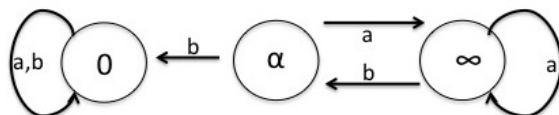

FIGURE 9. The subalgebra $\mathcal{K}$.

Thus for all such s, $ps \in L$ depends only on p, so all such forests are congruent. We denote this congruence class by α.

Suppose $s \neq 0$, but is not of either of the kinds described above. Then $ps \in L$ if and only if every maximal path in p, including the path to the hole, contains two consecutive b's. So all of these are congruent, and we denote their congruence class by ∞. This notation is justified by the fact that if s belongs to this class, then so does $s + s'$ for any forest s'.

Finally, the empty forest is not equivalent to any of these: Let $s \in L$. Then $0 + s \in L$, but this is not true if 0 is replaced by a member of the classes α or ∞. And 0 of course does not belong to the class L. We denote this class 0. Easily, $\{0, 0', \alpha, \infty\}$ is idempotent and commutative under addition, and $\alpha + 0' = \alpha$.

Figure 8 is a state transition diagram of this syntactic forest algebra, showing the transitions induced by the letters.

Observe that if we remove the state 0, we obtain a forest algebra in which $0'$ is the additive identity. We will work with this smaller algebra, which we call $\mathcal{K}$, and hereinafter denote its identity as 0 rather than $0'$. See Figure 9

By Theorem 3.5, L is definable in CTL if and only if (H_L, V_L) divides an iterated wreath product of copies of the forest algebra $\mathcal{U}_2$ (see Fig. 4). If this were true, then the subalgebra $\mathcal{K}$ would also divide such a wreath product. We will show that, in fact, this cannot occur.

The proof is by induction of the number of factors in the wreath product. Obviously $\mathcal{K} \nprec \mathcal{U}_2$, because the horizontal monoid of right-hand side has two elements, and the left-hand side three. For the inductive step, we will show that if X is any forest algebra such that $\mathcal{K} \prec \mathcal{U}_2 \circ X$, then $\mathcal{K} \prec X$.

If $\mathcal{K} \prec \mathcal{U}_2 \circ X$, then there exist a homomorphism Φ from a submonoid H' of the horizontal monoid of $\mathcal{U}_2 \circ X$, and vertical elements $\hat{a}$, $\hat{b}$ of the wreath product satisfying the conditions in Theorem 2.1. Let us denote the horizontal monoid of X by $H(X)$, and its vertical monoid by $V(X)$. We write

$$\hat{a} = (c, f), \hat{b} = (d, g),$$

where $f, g : \{0, \infty\} \to V(X)$, and $c, d \in \{1, c_0, c_\infty\}$. We choose elements $(a_q, x_q) \in \{0, \infty\} \times H(X)$ for $q \in \{0, \alpha, \infty\}$ such that $\Phi(a_q, x_q) = q$ for each q.

We first consider the case where $a_0 = a_\alpha = a_\infty = \infty$. Observe that the set

$$H'' = \{x \in H(X) : (\infty, x) \in \mathrm{dom}\Phi\}$$

is a subsemigroup of $H(X)$, and the map $\Psi : H'' \to \{0, \alpha, \infty\}$ defined by

$$\Psi(x) = \Phi(\infty, x)$$

is a homomorphism. By our assumption that $a_q = \infty$ for all q, we also have that Ψ is onto. We define

$$\bar{a} = f(\infty) + x_0, \bar{b} = g(\infty) + x_0 \in V(X).$$

If $x \in H''$, then we have

$$
\begin{aligned}
(\infty, \bar{a} \cdot x) &= (c \cdot \infty + \infty, f(\infty) \cdot x + x_0) \\
&= (c, f) \cdot (\infty, x) + (\infty, x_0) \\
&= \hat{a} \cdot (\infty, x) + (\infty, x_0) \\
&\in \mathrm{dom}\Phi,
\end{aligned}
$$

since $\hat{a} \cdot (\mathrm{dom}\Phi) \subseteq \mathrm{dom}\Phi$, and $\mathrm{dom}\Phi$ is closed under addition. Thus $\bar{a} \cdot (\mathrm{dom}\Psi) \subseteq \mathrm{dom}\Psi$, and likewise $\bar{b} \cdot (\mathrm{dom}\Psi) \subseteq \mathrm{dom}\Psi$. We also have, for any $x \in H''$,

$$
\begin{aligned}
\Psi(\bar{a} \cdot x) &= \Phi(\infty, \bar{a} \cdot x) \\
&= \Phi(c \cdot \infty + \infty, f(\infty) \cdot x + x_0) \\
&= \Phi(\hat{a}(\infty, x) + (\infty, x_0)) \\
&= \Phi(\hat{a} \cdot (\infty, x)) + \Phi(\infty, x_0) \\
&= a\Phi(\infty, x) + 0 \\
&= a\Phi(\infty, x) \\
&= a\Psi(x).
\end{aligned}
$$

Similarly $\Psi(\bar{b} \cdot x) = b\Psi(x)$. Thus $\mathcal{K} \prec X$.

We thus suppose that the a_q cannot all be chosen equal to ∞. If $a_0 = \infty$, then we find

$$\Phi(\infty, x_\alpha + x_0) = \Phi(a_\alpha, x_\alpha) + \Phi(\infty, x_0) = \alpha + 0 = \alpha,$$

and likewise $\Phi(\infty, x_\infty + x_0) = \infty$. Thus all three elements have preimages in $\{\infty\} \times H(X)$, and we are back in the previous case. So we can suppose that no element of the form (∞, x) can be mapped by Φ to 0. If $c = c_\infty$, we find

$$\Phi(\infty, f(0) \cdot x_0) = \Phi(\hat{a}(0, x_0)) = a \cdot 0 = 0,$$

which means that we could have chosen $a_0 = \infty$. As we have shown this is impossible, we must have $c \in \{1, c_0\}$. The same argument with $\hat{b}$ in place of $\hat{a}$ shows $d \in \{1, c_0\}$.

If $a_\alpha = \infty$, then

$$\Phi(d \cdot \infty, g(\infty) \cdot x_\alpha) = \Phi(\hat{b}(\infty, x_\alpha)) = b \cdot \alpha = 0,$$

so $d \cdot \infty$ cannot be ∞, and thus $d = c_0$. This gives

$$\begin{aligned}
\Phi(0, g(a_\infty) \cdot x_\infty) &= \Phi(c_0 \cdot a_\infty, g(a_\infty) \cdot x_\infty) \\
&= \Phi(\hat{b}(a_\infty, x_\infty)) \\
&= b \cdot \infty \\
&= \alpha,
\end{aligned}$$

so we could have chosen our preimage of α with $a_\alpha = 0$. This gives

$$\begin{aligned}
\Phi(0, f(0) \cdot x_\alpha) &= \Phi(c \cdot 0, f(0) \cdot x_\alpha) \\
&= \Phi(\hat{a}(0, x_\alpha)) \\
&= a \cdot \alpha \\
&= \infty
\end{aligned}$$

so we also could have chosen our preimage of ∞ with $a_\infty = 0$. So we may now suppose $(0, x_0), (0, x_\alpha), (0, x_\infty)$ are mapped by Φ to $0, \alpha, \infty$ respectively. We now set $\bar{a} = f(0), \bar{b} = g(0)$, and

$$H'' = \{x \in H(X) : (0, x) \in \mathrm{dom}\Phi\}.$$

As before, we have H'' closed under addition, and the map $\Psi : x \mapsto \Phi(0, x)$ a homomorphism from H'' onto $\{0, \alpha, \infty\}$. If $(0, x) \in \mathrm{dom}\Phi$, then we have

$$(0, \bar{a} \cdot x) = (c \cdot 0, f(0) \cdot x) = \hat{a}(0, x) \in \mathrm{dom}\Phi,$$

so that $\bar{a}(\mathrm{dom}\Psi) \subseteq \mathrm{dom}\Psi$, and similarly for $\bar{b}$. Finally, we have for $x \in H''$,

$$\begin{aligned}
\Psi(\bar{a} \cdot x) &= \Phi(0, \bar{a} \cdot x) \\
&= \Phi(c \cdot 0, f(0) \cdot x) \\
&= \Phi(\hat{a}(0, x)) \\
&= a\Phi(0, x) \\
&= a\Psi(x),
\end{aligned}$$

and similarly $\Psi(\bar{b} \cdot x) = b\Psi(x)$. So $\mathcal{K} \prec X$ in this case as well. $\qquad\square$

The complement of L is the language $\mathsf{E}'T$, where $T \subseteq \{a, b\}^*$ consists of all forests in which some maximal path does not contain two consecutive b's. It follows from Theorem 6.1 that this language, as well, is not definable in CTL. This provides a negative answer to the question raised earlier whether every forest language of the form $\mathsf{E}'K$ for some $K \subseteq \{a, b\}^*$ whose associated seminearring is aperiodic, is definable in CTL. To complete the analysis, we need to make sure that $(\widehat{H_L, V_L})$ is indeed aperiodic.

Proposition 6.2. $(\widehat{H_L, V_L})$ *is aperiodic.*

Proof. We will show that no multicontext p can induce a nontrivial cycle in the horizontal elements of H_L. Obviously, 0 cannot be an element of any such cycle. We have a number of cases to consider, depending on the form of p.

If p has a hole at every leaf node, then $p \cdot 0' = 0'$. This means that $0'$ cannot be a member of any cycle induced by p. We need to rule out the possibility that p interchanges α and ∞. If p has a root node labeled a, then we can not have $p \cdot h = \alpha$ for any $h \in H_L$. If all root nodes are labeled b, then we can never have $p \cdot h = \infty$.

So the only possibility for a cycle is one where p has some leaf nodes that are not holes. If p has a root node labeled a, then as above, we can not have $p \cdot h = \alpha$. Thus p would have to interchange $0'$ and ∞. Similarly, if p has all root nodes labeled b, then p would have to interchange $0'$ and α. We can show that neither of these is possible as a consequence of the following fact, which can be proved by an easy induction on the number of nodes of p: For every multicontext p, if $p \cdot \alpha = 0'$ or $p \cdot \infty = 0'$, then $p \cdot 0' = 0'$. □

6.2. MONOIDS ASSOCIATED WITH THE COMMON FRAGMENT OF LTL AND CTL

The ideal result in this vein would be an algorithm that takes as input an automaton recognizing a word language $L \subset A^*$ and outputs whether or not $\mathsf{E}'L$ is in CTL. This problem is open, and, as we discuss later in the conclusion, it may prove to be very difficult.

In general, the structure of the syntactic monoid of L is not sufficient to answer this question. This is shown by Theorem 6.1: If L is given by the regular expression $(a + b)^*bb(a + b)^*$–that is, if L consists of strings over $\{a, b\}$ that contain two consecutive occurrences of b, then $\mathsf{E}'L$ is in CTL. But this is not true for the complement of L, which has the same syntactic monoid.

Here we give necessary and sufficient conditions on a finite monoid M so that for all $L \subseteq A^*$ recognized by M, $\mathsf{E}'L \in CTL$. We first recall some basic facts about the ideal structure of finite semigroups: Two elements s, t of a finite monoid M are $\mathcal{J}$-equivalent if $MsM = MtM$. Equivalence classes of this relation are called $\mathcal{J}$-classes. If a $\mathcal{J}$-class contains an idempotent the class is said to be *regular*. Generally speaking, the product of two elements in a $\mathcal{J}$-class J need not belong to the $\mathcal{J}$-class, and if $st \notin J$, we must have $MstM \cap J = \emptyset$—in other words, we can never get back to J once the product of two elements takes us outside of J. The *Rees*
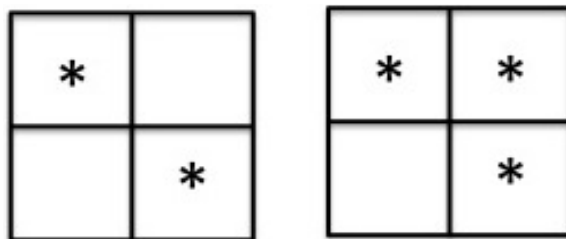
FIGURE 10. The nontrivial $\mathcal{J}$-classes in $M((ab)^+)$ (left) and $M((a+b)^*bb(a+b)^*)$ (right).

Matrix Theorem, restricted to finite aperiodic monoids, states that for each regular $\mathcal{J}$-class J there is a pair of finite sets X, Y and a map $P : Y \times X \to \{0, 1\}$, such that J is isomorphic to $X \times Y$ with multiplication given by

$$(x, y) \cdot (x', y') = (x, y'), \quad \text{if} \quad P(y, x') = 1,$$

with the product undefined if $P(y, x') = 0$. This latter case is the situation where the product of the elements of J corresponding to (x, y) and (x', y') is not in J.

The function P has an additional property: For each $y \in Y$ there is at least one $x \in X$ such that $P(y, x) = 1$, and similarly for each $x \in X$ there is at least one $y \in Y$ such that $P(y, x) = 1$. Observe that $P(y, x) = 1$ if and only if (x, y) is an idempotent. We can thus depict a regular $\mathcal{J}$-class in a finite aperiodic monoid by what has come to be called an *egg-box diagram*: This is a rectangular grid with rows indexed by X, columns by Y, and with an asterisk in each cell corresponding to an idempotent.

The syntactic monoids of $(ab)^+$ and $(a+b)^*bb(a+b)^*$ have very similar structures. Both contain six elements: 1, 0, and a regular $\mathcal{J}$-class with a 2×2 Rees matrix representation. In the case of $(ab)^+$ this $\mathcal{J}$-class contains two idempotents (the syntactic congruence classes of ab and ba) and two non-idempotents (the syntactic congruence classes of a and b). In the case of $(a+b)^*bb(a+b)^*$ the class of a is also idempotent. Egg-box diagrams of these $\mathcal{J}$-classes are shown in Figure 10.

The class **DA** consists of finite aperiodic monoids M in which every regular $\mathcal{J}$-class is closed under multiplication (equivalently, every element of every regular $\mathcal{J}$-class is idempotent). **DA** forms a pseudovariety of finite monoids. This variety, which arises in an surprisingly large number of contexts, was first identified by Schützenberger [16], who proved the following property of the languages recognized by monoids in **DA**.

Theorem 6.3. *Let $L \subseteq A^*$. If $M(L) \in \mathbf{DA}$ then L is a finite union of languages of the form*

$$A_0^* a_1 A_1^* \dots a_n A_n^*,$$

where for $i = 1, \dots, n$, $a_i \in A$, and for $i = 0, \dots, n$, $A_i \subseteq A$.

(As we have stated this theorem, the converse is false, but in fact Schützenberger gave a precise characterization of languages recognized by **DA**, which entails an additional unambiguity condition on the concatenation product and union. We will not require this).

Here is our principal result concerning languages of the form $\mathsf{E}'L$ definable in CTL.

Theorem 6.4. *Let M be a finite monoid, and A a finite alphabet with at least two letters. The following are equivalent:*

- *For every $L \subseteq A^*$ recognized by M, $\mathsf{E}'L$ is in CTL.*
- *$M \in \mathbf{DA}$.*

Proof. First suppose that $M \in \mathbf{DA}$, and let L be recognized by M. Then $M(L) \prec M$. Since **DA** is closed under division, $M(L) \in \mathbf{DA}$, and thus by Theorem 6.3, L is a union of products of the form $A_0^* a_1 A_1^* \ldots a_n A_n^*$, where for $i = 1, \ldots, n$, $a_i \in A$, and for $i = 0, \ldots, n$, $A_i \subseteq A$. Easily for any $L_1, L_2 \subseteq A^*$, $\mathsf{E}'(L_1 \cup L_2) = \mathsf{E}'L_1 \cup \mathsf{E}'L_2$. Thus it suffices to show that $\mathsf{E}'L \in CTL$ when L has the form $A_0^* a_1 A_1^* \ldots a_n A_n^*$. We can prove this by induction on the length n of the product. First, if $A' \subseteq A$, then $\mathsf{E}'(A')^*$ is defined by the formula

$$\mathsf{emp} \vee \mathsf{E}\phi\mathsf{U}(\phi \wedge \mathsf{emp}),$$

where ϕ is $\bigvee_{a \in A'} a$. Now, assuming we have a CTL formula ψ defining the set of forests with a maximal path in $A_k^* a_{k+1} \ldots a_n A_n^*$, we obtain the following formula for $A_{k-1}^* a_k A_k^* a_{k+1} \ldots a_n A_n^*$:

$$\mathsf{E}\left(\bigvee_{a \in A_{k-1}} a\right) \mathsf{U}(a_k \wedge \psi).$$

For the converse, assume that $M \notin \mathbf{DA}$. We will show that there are word languages L recognized by M such that $\mathsf{E}'L$ is not in CTL.

If M contains a group, then it recognizes a word language L such that $M(L)$ contains a group. There is thus a word v and a sequence of words $(u_0, \ldots, u_{k-1})$ such that v cycles these words in $M(L)$: That is, vu_i is equivalent to $u_{(i+1) \bmod k}$ under the syntactic congruence of L, but the u_i are not all congruent to one another. In particular, u_0 and u_1 are not congruent, so there exist words x, y such that $xu_0y \in L$ and $xu_1y \notin L$, or vice-versa. Now consider the sequence of words $(u_0y, u_1y, \ldots, u_{k-1}y)$ as *forests,* and compare them modulo the syntactic congruence of $\mathsf{E}'L$. We then have that u_0y and u_1y are not equivalent. However, we do have vu_iy equivalent to $vu_{(i+1) \bmod k}y$. Thus v induces a nontrivial cycle in the syntactic forest algebra of $\mathsf{E}'L$. This algebra is thus not aperiodic, so it cannot be definable in CTL.

We may therefore assume that M is aperiodic. Thus M has regular $\mathcal{J}$-class J that contains a non-idempotent element. Consider a Rees matrix representation

of J, and let (x, y) be a non-idempotent element. There must be $x' \neq x$ and $y' \neq y$ such that (x, y') and (x', y) are both idempotent. Let $a, b \in A$ be distinct letters, and define a homomorphism $\phi : A^* \to M$ that maps b to (x, y), a to (x', y'), and all other letters of A to $(x, y)^2$. Consider the language $L = \phi^{-1}(J)$ recognized by M.

If $w \in L$, then $w \in \{a, b\}^*$, because otherwise $w \in M(x, y)^2 M$, which is outside of J. If (x', y') is not idempotent, then w cannot contain any factor of the form aa, or bb, because both $P(y, x)$ and $P(y', x')$ are 0. Thus L consists precisely of the words in which a and b strictly alternate, and so $\mathsf{E}'L$ is not definable in CTL, a consequence of Proposition 4.4. (That Proposition concerns the language $\mathsf{E}'(ab)^+$, but the same proof applies to $\mathsf{E}'L$.) If (x', y') is idempotent, $P(y', x') = 1$, so $w \in L$ if and only if w contains no factor of the form bb. But now Theorem 6.1 implies that $\mathsf{E}'L$ is not definable in CTL. $\hfill\square$

7. Conclusion

Much of this paper has been devoted to showing how standard notions from the algebraic theory of regular word languages–sequences of identities, preservation of aperiodicity under wreath products and quotients, ideal structure of semigroups–can be brought to bear on questions of logical definability of properties of trees, both producing new insights and simplifying existing arguments. However, there is one very significant gap: We have only been able to demonstrate algebraic necessary conditions for definability in various logics, by deriving properties of the wreath product. What is missing is a mechanism for obtaining wreath product decompositions of forest algebras, which will be required in order to use this theory to obtain sufficient conditions. Such a decomposition is demonstrated in [7], but only in the rather simple case of a characterization of the logic EF.

Let us examine our results in this light.

Seminearring closure and aperiodicity. As we saw, aperiodicity of the seminearring closure of a forest algebra is a necessary, but not a sufficient condiition for membership in CTL. Still, this notion might be relevant for the problem of obtaining an effective characterization of CTL. Let us begin with a seminearring (H, V) in which H is idempotent and commutative, and V is aperiodic. By the Krohn–Rhodes Theorem, (H, V) *considered as a transformation monoid* divides an iterated wreath product of copies of $\mathcal{U}_2$, again considered as transformation monoids–that is, ignoring the additive structure. The problem of characterizing membership in CTL is essentially that of finding when such a transformation monoid division preserves the additive structure, and is thus a forest algebra division.

We should expect that, considered as a transformation monoid, (H, V) might admit a shorter decomposition as a wreath product of copies of $\mathcal{U}_2$ than it has as a forest algebra. It would be interesting to see examples of languages in CTL whose syntactic forest algebras have this property. (If we knew for some reason that such a length discrepancy never arises, then we would have a proof of decidability of membership in CTL, albeit with a very inefficient algorithm).

Generalized distributivity. What is the exact relation between the generalized distributive idenitities and definability in PDL? The dream theorem here would state that if a finite forest algebra is horizontally idempotent and commutative and satisfies $s_n = t_n$ for some n, then it divides a wreath product of distributive, horizontally idempotent and commutative forest algebras.

This seems like too much to hope for, and it is probably more prudent to search for a counterexample. But the identities give us a new way to think about the problem. If generalized distributivity is not sufficient, then what do we need to add to these identities to permit the decomposition?

The overlap of CTL and LTL. The work of Bojaǹczyk [5], building on earlier results of Maidl [11], provides an effective characterization of the languages $L \subseteq A^*$ such that $\mathsf{E}'L$ is definable in CTL using only positive applications of the operator $\mathsf{E}\phi\mathsf{U}\psi$: that is, this operator is never used within the scope of a negation. L is definable in this way if and only if it has depth $3/2$ in the concatenation hierarchy; that is, L is a union of languages of the form

$$A_0^* a_1 A_1^* \ldots a_n A_n^*,$$

where for $i = 1, \ldots, n$, $a_i \in A$, and for $i = 0, \ldots, n$, $A_i \subseteq A$. One can decide effectively whether a given regular language L is of depth $3/2$.

We have already seen that every language in **DA** has this form, and we have argued above that for every such language L, $\mathsf{E}'L$ is definable in CTL. However, there are languages of depth $3/2$ that are outside **DA**. Bojańczyk provides an example of a language L that does not have depth $3/2$, but for which $\mathsf{E}'L$ is in CTL.

The question of definability of $\mathsf{E}'L$ in CTL is thus very complex. We conjecture that every L for which $\mathsf{E}'L$ belongs to CTL has depth no more than 2 in the concatenation hierarchy; these are boolean combinations of languages of depth $3/2$. (Bojaǹczyk's example has depth 2). Even if this turns out to be true, it is far from settling the general question: We have already seen that there are languages L of depth 2 for which $\mathsf{E}'L$ is not in CTL. Moreover, in contrast to depth $3/2$, the problem of effectively determining if a given regular language has depth 2 has been open for many years.

References

[1] M. Benedikt and L. Segoufin, Regular tree languages definable in fo and in fomod. *ACM Trans. Comput. Logic* **11** (2009) 1–4.
[2] M. Bojanczyk, Algebra for trees, in *AutoMathA Handbook*, edited by J.-E. Pin (2012). Preprint.
[3] M. Bojanczyk and I. Walukiewicz, Forest algebras, in *Logic and Automata: History and Perspectives*, edited by E. Graedel, J. Flum and T. Wilke. Amsterdam University Press (2008).
[4] M. Bojańczyk, *Decidable Properties of Tree Languages*. Ph.D. thesis, University of Warsaw (2004).

[5] M. Bojanczyk, The common fragment of actl and ltl. In *FoSSaCS*, vol. 4962 of *Lect. Notes Comput Sci.*, edited by R.M. Amadio (2008) 172–185.

[6] M. Bojanczyk, L. Segoufin and H. Straubing, Piecewise testable tree languages. To appear in *Logical Methods Comput. Sci.* (2012).

[7] M. Bojanczyk, H. Straubing and I. Walukiewicz, Wreath products of forest algebras, with applications to tree logics. To appear in *Logical Methods Comput. Sci.* (2012).

[8] S. Eilenberg, Automata, Languages, and Machines, vol. B. *Pure and Applied Mathematics*. New York, Academic Press (1976).

[9] E. Allen Emerson, Temporal and modal logic, in *Handbook Of Theoretical Computer Science*. Elsevier (1995) 995–1072.

[10] L. Libkin, Elements Of Finite Model Theory. Texts in *Theoretical Computer Science*. Springer (2004).

[11] M. Maidl, The common fragment of ctl and ltl. In *FOCS*. IEEE Computer Society (2000) 643–652.

[12] F. Moller and A. M. Rabinovich, On the expressive power of ctl. In *LICS*. IEEE Computer Society (1999) 360–368.

[13] J.E. Pin, *Varieties of formal languages*. North Oxford Academic (1986).

[14] J.-E. Pin, Syntactic semigroups, vol. 1 in *Handbook of Formal Languages*. Springer (1997).

[15] A. Potthoff, First-order logic on finite trees. *Lect. Notes Comput. Sci.* **915** (1995) 125–139.

[16] M. Schützenberger, Sur le produit de concatenation non ambigu. *Semigroup Forum* **13** (1976) 47–75. `Doi: 10.1007/BF02194921`.

[17] S. Shamir, O. Kupferman and E. Shamir, Branching-depth hierarchies. *Electr. Notes Theor. Comput. Sci.* **39** (2000) 65–78.