31/04

Zoltán Ésik
(Rovira i Virgili University)

An Algebraic Characterization of Temporal Logics on Trees

An Algebraic Characterization of Temporal Logics on Trees

Z. Ésik[1]
Dept. of Computer Science, University of Szeged
P.O.B. 652, 6701 Szeged, Hungary
and
Research Group on Mathematical Linguistics, Rovira i Virgili University
Pl. Imperial Tarraco 1, 43005 Tarragona, Spain

October 2004

[1]Supported in part by a grant from the National Foundation of Hungary for Scientific Research.

Abstract

In Chapter 1, we associate a modal operator with each language belonging to a given class of regular tree languages and use the cascade product of tree automata to give an algebraic characterization of the expressive power of the resulting logic. In Chapter 2, we provide a similar treatment of unordered trees. In Chapter 3, we give an effective characterization of the expressive power of a temporal logic on finite trees related to a fragment of CTL.

Chapter 1

1.1 Introduction

The cascade product and its semigroup theoretic variants have been very useful and powerful tools in the characterization of the expressive power of several logics over finite words, including first-order logic and its extension with modular counting, cf. McNaughton and Papert [16], Straubing, Therien and Thomas [23] and Straubing [22], linear temporal logic and the until hierarchy, Cohen, Perrin and Pin [6] and Therien and Wilke [25], and modular temporal logic, Bazirambawo, McKenzie and Therien [3], to mention a few references. In this paper, our aim is to show that the cascade product of tree automata has the same potential in the characterization of the expressive power of various CTL-like temporal logics on finite trees. We associate a modal operator with each language belonging to a given class of regular tree languages and use the cascade product of tree automata to give an algebraic characterization of the expressive power of the resulting logic. From our general results, we deduce algebraic characterizations of the expressive power of several specific logics on finite trees related to CTL.

Some notation When n is a natural number, we denote the set $\{1, \ldots, n\}$ by $[n]$. Thus, $[0]$ is another name for the empty set. When A is a set, A^* denotes the set of all finite words over A including the empty word.

1.2 Algebras

A *rank type* R is a finite nonempty set of nonnegative inyegers. To avoid trivial situations, we assume that each rank type contains a positive integer. A *ranked alphabet* Σ of rank type R is a disjoint union of finite sets Σ_n, $n \geq 0$, such that for each n, $\Sigma_n \neq \emptyset$ iff $n \in R$. The elements of Σ_n are called *letters* or *symbols* of rank n, or when $n = 0$, *constant symbols.*

Suppose that Σ is a ranked alphabet of rank type R. A Σ-*algebra* $\mathbb{A}$ consists of a nonempty set A, called the *carrier* of $\mathbb{A}$, and an operation $\sigma_{\mathbb{A}} : A^n \to A$, for each $\sigma \in \Sigma_n$, $n \geq 0$, called the *interpretation of* Σ. Homomorphisms, subalgebras, congruences, direct products, etc. are defined as usual, see, e.g., Grätzer [13]. Suppose that $\mathbb{A} = (A, (\sigma_{\mathbb{A}})_{\sigma \in \Sigma})$ is a Σ-algebra and $\mathbb{B} = (B, (\delta_{\mathbb{B}})_{\delta \in \Delta})$ is a Δ-algebra, where Σ and Δ are of the same rank type. We call $\mathbb{A}$ a *renaming* of $\mathbb{B}$ if $A = B$ and for each $\sigma \in \Sigma_n$, $n \geq 0$ there is a symbol $\delta \in \Delta_n$ with $\sigma_{\mathbb{A}} = \delta_{\mathbb{B}}$.

We will take the liberty of writing just σ for $\sigma_{\mathbb{A}}$ whenever the algebra $\mathbb{A}$ is clear from the context. We call the algebra $\mathbb{A}$ *finite* if its carrier is finite. Sometimes we do not specify the carrier of an algebra explicitly and follow the practice that if $\mathbb{A}, \mathbb{B}$ etc. denote algebras, then $A, B \ldots$ denote the corresponding carriers.

1.3 Trees and Tree Automata

Suppose that Σ is a ranked alphabet. Let $x_1, x_2, \ldots$ be a fixed countable sequence of variables, and for each $n \geq 0$, let X_n denote the set $\{x_1, \ldots, x_n\}$. The set $T_\Sigma(X_n)$ of *n-ary* Σ*-trees* is defined as the least set containing Σ_0 and X_n (which are assumed to be disjoint) such that whenever $t_1, \ldots, t_m$ are in $T_\Sigma(X_n)$ and $\sigma \in \Sigma_m$, then $\sigma(t_1, \ldots, t_m)$ is also in $T_\Sigma(X_n)$. When $n = 0$, we write just T_Σ. The elements of T_Σ are called *ground trees*. Note that T_Σ is nonempty iff Σ_0 is nonempty. Sometimes it is convenient to represent an n-ary tree as a directed graph which is a rooted tree equipped with a labeling function that maps vertices to letters in $\Sigma \cup X_n$ such that the outgoing edges of each vertex are linearly ordered. Moreover, a vertex is labeled in $\Sigma_0 \cup X_n$ iff it is a leaf, i.e., it has no successor, and is labeled in Σ_m for some $m > 0$ iff it has m immediate successors. The label of a vertex v in a tree will be denoted $t(v)$. The notion of *subtree of a tree* t *rooted at a vertex* v, denoted t_v, is defined as usual. The *immediate subtrees* of a tree are those rooted at the immediate successors of the root. The *depth* of a vertex v in a tree t is the length of the unique path from the root to v, where the depth of the root is 0. The depth of a tree is the length of the longest path in the tree.

Trees $c \in T_\Sigma(X_1)$ with a single leaf labeled x_1 are called *contexts*. A *primitive context* is a context of the form $\sigma(t_1, \ldots, t_{i-1}, x_1, t_{i+1}, \ldots, t_n)$, where $\sigma \in \Sigma_n$, $n > 0$, $i \in [n]$, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \in T_\Sigma$. Thus, a primitive context is a context such that the leaf labeled x_1 occurs at depth 1. We let CT_Σ denote the set of all contexts in $T_\Sigma(X_1)$.

Suppose that $t \in T_\Sigma(X_n)$ and $t_1, \ldots, t_n \in T_\Sigma(X_m)$ are trees. Then the tree resulting from t by substituting, for each $i \in [n]$, a copy of t_i for each occurrence of x_i in t, is denoted $t(t_1, \ldots, t_n)$. Note that this tree is in $T_\Sigma(X_m)$. The formal definition goes by induction on the structure of t. If $t = \sigma \in \Sigma_0$, then $t(t_1, \ldots, t_n) = \sigma$, and if $t = x_i$ for some $i \in [n]$, then $t(t_1, \ldots, t_n) = t_i$. Last, if $t = \sigma(s_1, \ldots, s_k)$ with $\sigma \in \Sigma_k$, $s_1, \ldots, s_k \in T_\Sigma(X_n)$, $k > 0$, then $t(t_1, \ldots, t_n) = \sigma(s_1(t_1, \ldots, t_n), \ldots, s_k(t_1, \ldots, t_n))$.

If $\mathbb{A}$ is a Σ-algebra with carrier A and $t \in T_\Sigma(X_n)$, then t *induces* a function $A^n \to A$, denoted $t_\mathbb{A}$, or just t. The definition is standard, see, e.g., Gécseg and Steinby [12] and Grätzer [13]. When $n = 0$, we identify $t_\mathbb{A}$ with an element of A.

To simplify the treatment, our temporal logics will be tailored so that only sets of ground trees will be definable. Accordingly, a *tree language* over Σ is a set $L \subseteq T_\Sigma$ of ground Σ-trees. In order to avoid trivial situations, when we speak of tree languages, we will always assume that the underlying rank type contains 0, so that each ranked alphabet contains constant symbols.

Suppose that Σ is a ranked alphabet of rank type R with $0 \in R$. A *Σ-tree automaton* is a Σ-algebra that contains no proper subalgebras. A tree automaton is finite if it is a finite algebra. A *homomorphism of Σ-tree automata* is a Σ-algebra homomorphism. Note that if $\mathbb{A}$ and $\mathbb{B}$ are Σ-tree automata, then there is at most one homomorphism $\mathbb{A} \to \mathbb{B}$. Moreover, any homomorphism of tree automata is a surjective function.

Let $\mathbb{A} = (A, (\sigma_\mathbb{A})_{\sigma\in\Sigma})$ be a tree automaton. The language *accepted* or *recognized by $\mathbb{A}$ with final states $F \subseteq A$* is defined by

$$L(\mathbb{A}, F) \quad = \quad \{t \in T_\Sigma : t_\mathbb{A} \in F\}.$$

A tree language $L \subseteq T_\Sigma$ is recognizable by the tree automaton $\mathbb{A}$ if $L = L(\mathbb{A}, F)$ for some $F \subseteq A$. A tree language $L \subseteq T_\Sigma$ is *regular* if it is recognizable by a finite tree automaton.

Each tree language $L \subseteq T_\Sigma$ is recognizable by a canonical tree automaton (unique up to isomorphism), the *minimal tree automaton $\mathbb{A}_L$ of L*. It has the universal property that whenever L is recognizable by a tree automaton $\mathbb{A}$ then there is a (necessarily unique) homomorphism $\mathbb{A} \to \mathbb{A}_L$. Thus, a language L is regular iff its minimal tree automaton is finite. It is known that a Σ-tree automaton $\mathbb{A} = (A, (\sigma_\mathbb{A})_{\sigma\in\Sigma})$ is isomorphic to the minimal tree automaton of $L \subseteq T_\Sigma$ iff $L = L(\mathbb{A}, F)$ for some (necessarily unique) $F \subseteq A$, and for any $a, b \in A$ with $a \neq b$ there is a context c with $c(a) \in F$ and $c(b) \notin F$, or $c(a) \notin F$ and $c(b) \in F$. Thus, the languages recognizable by $\mathbb{A}_L$ are unions of $\sim_L$-equivalence classes, where the relation $\sim_L$ on T_Σ is defined by

$$t \sim_L t' \quad \Leftrightarrow \quad \forall c \in CT_\Sigma \ (c(t) \in L \Leftrightarrow c(t') \in L).$$

Thus, L is regular iff $\sim_L$ is of finite index. For the reader's convenience, we include a proof of the following well-known fact.

Lemma 1.3.1 *Suppose that $L \subseteq T_\Sigma$ is regular. Then every language recognizable by $\mathbb{A}_L$ is a boolean combination of quotients of L.*

Proof. We know that the languages recognizable by $\mathbb{A}_L$ are unions of $\sim_L$-equivalence classes. But each $\sim_L$-equivalence class $[t]$ can be written as

$$\bigcap_{t\in c^{-1}L} c^{-1}L \setminus \bigcup_{t\notin c^{-1}L} c^{-1}L.$$

Since L is regular, the intersection and the union in the above formula are finite, since each language $c^{-1}L$ is recognizable by $\mathbb{A}_L$. Thus, every $\sim_L$-equivalence class $[t]$ and every language recognizable by $\mathbb{A}_L$ is the boolean combination of quotients of L. □

Suppose that a rank type R with $0 \in R$ is fixed. By a *class $\mathcal{L}$ of tree languages* we mean a collection of tree languages in T_Σ for each ranked alphabet Σ (of rank type R). A class of regular tree languages consists of regular languages.

Let Σ and Δ be two ranked alphabets of the same rank type. Given a tree $t \in T_\Sigma(X_n)$, a *relabeling* of t over Δ is obtained by changing the label of each vertex of t labeled in Σ_m to some symbol in Δ_m, for each $m \geq 0$. Labels in X_n do not change. Different occurrences of the same letter in Σ may be replaced by different letters. A related notion is that of a *literal tree homomorphism.* Suppose that h is a rank preserving function $\Sigma \to \Delta$. Then for each n, h determines a literal tree homomorphism $T_\Sigma(X_n) \to T_\Delta(X_n)$, also denoted h. The image of a tree $t \in T_\Sigma(X_n)$ is obtained from t by relabeling each vertex labeled $\sigma \in \Sigma$ by the letter $h(\sigma)$ (of the same rank). It is known that the class of regular tree languages is closed under literal homomorphisms and inverse literal homomorphisms. Thus, if $L \subseteq T_\Delta$ is regular and h is a literal homomorphism as described above, then $h^{-1}(L) = \{t \in T_\Sigma : h(t) \in L\}$ is regular.

In addition to relabelings and literal tree homomorphisms, we will make use of quotients. Suppose that $c \in CT_\Sigma$ is and $L \subseteq T_\Sigma$. The *quotient of L with respect to c* is defined as the tree language

$$c^{-1}L \quad = \quad \{t \in T_\Sigma : c(t) \in L\}.$$

It is known that the class of regular tree languages is closed under quotients, i.e., if $L \subseteq T_\Sigma$ is regular and $c \in CT_\Sigma$, then $c^{-1}L$ is regular. Moreover, a tree language $L \subseteq T_\Sigma$ is regular iff it has a finite number of different quotients, i.e., when the set $\{c^{-1}L : c \in CT_\Sigma\}$ is finite. It is clear that a class $\mathcal{L}$ of tree languages is closed under quotients iff it is closed under quotients with respect to primitive contexts. Other operations under which the class of regular languages is closed include the boolean operations.

For the above facts and more results on tree automata and tree languages, refer to any standard text such as Gécseg and Steinby [12].

1.4 Extended Temporal Logic

We now define our temporal logics on trees. We assume that a rank type R with $0 \in R$ is fixed and only coincide ranked alphabets of rank type R. Further, we assume that each ranked alphabet comes with a fixed lexicographic order.

Syntax. For a ranked alphabet Σ, the set of *formulas over* Σ *is* the least set containing the letters p_σ, for all $\sigma \in \Sigma$, closed with respect to the boolean connectives $\vee$ (disjunction) and $\neg$ (negation), as well as the following construct. Suppose that $L \subseteq T_\Delta$ and that for each $\delta \in \Delta$, φ_δ is a formula over Σ. Then

$$L(\delta \mapsto \varphi_\delta)_{\delta \in \Delta} \tag{1.1}$$

is a formula over Σ. The notion of *subformula* of a formula is defined as usual.

Semantics. Suppose that φ is a formula over Σ and $t \in T_\Sigma$. We say that t *satisfies* φ, in notation $t \models \varphi$, if

- $\varphi = p_\sigma$, for some $\sigma \in \Sigma_n$, and the root of t is labeled σ, i.e., $t = \sigma(t_1, \ldots, t_n)$, for some $t_1, \ldots, t_n$, or
- $\varphi = \varphi' \vee \varphi''$ and $t \models \varphi'$ or $t \models \varphi''$, or
- $\varphi = \neg\varphi'$ and it is not the case that $t \models \varphi'$, or
- $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$, and the *characteristic tree* $\widehat{t} \in T_\Delta$ determined by t and the family $(\varphi_\delta)_{\delta \in \Delta}$ belongs to L. Here, $\widehat{t}$ has the same underlying digraph as t, and a vertex v is labeled $\delta \in \Delta_n$ in $\widehat{t}$ iff v is labeled by some $\sigma \in \Sigma_n$ in the tree t, moreover, δ is the first in lexicographic order on Δ_n such that the subtree of t rooted at v satisfies φ_δ, i.e., $t_v \models \varphi_\delta$. If no such letter exists, then δ is the last in the lexicographic order on Δ_n.

For any formula φ of over Σ, we let L_φ denote the *language defined by* φ:

$$L_\varphi \quad = \quad \{t \in T_\Sigma : t \models \varphi\}.$$

We say that formulas φ and ψ over Σ are *equivalent* if $L_\varphi = L_\psi$. Throughout the paper we will use the boolean connectives $\wedge$ (conjunction) and $\rightarrow$ (implication) as abbreviations. Moreover, for any ranked alphabet Σ and $n \in R$, we define $\mathtt{tt}_n = \bigvee_{\sigma \in \Sigma_n} p_\sigma$ and $\mathtt{ff}_n = \neg\mathtt{tt}_n$. Thus, $t \models \mathtt{tt}_n$ iff the root of t is labeled in Σ_n. We further let $\mathtt{tt} = p_\sigma \vee \neg p_\sigma$, where σ is any letter in Σ and $\mathtt{ff} = \neg\mathtt{tt}$.

We will consider subsets of formulas associated with classes of tree languages. When $\mathcal{L}$ is a class of tree languages, we let $\mathrm{FTL}(\mathcal{L})$ denote the collection of formulas all of whose subformulas of the form (1.1) above are such that L belongs to $\mathcal{L}$. We define $\mathbf{FTL}(\mathcal{L})$ to be the class of all languages definable by formulas in $\mathrm{FTL}(\mathcal{L})$. It is clear that for each formula $L(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$ in $\mathrm{FTL}(\mathcal{L})$ over an alphabet Σ there is an equivalent formula $L(\delta \mapsto \varphi'_\delta)_{\delta \in \Delta}$ in $\mathrm{FTL}(\mathcal{L})$ over Σ such that the subformulas φ'_δ satisfy the following condition: There exist no $t \in T_\Sigma$ and distinct letters $\delta, \delta' \in \Delta_n$, for some n, such that $t \models \varphi'_\delta \wedge \varphi'_{\delta'}$. Indeed, when the lexicographic order on Δ_n is $\delta_1 < \ldots < \delta_k$, then we may define

$$\varphi'_{\delta_i} \quad = \quad \varphi_{\delta_i} \wedge \bigwedge_{j<i} \neg\varphi_{\delta_j},$$

for all $i \in [k]$. Alternatively, we may define

$$\varphi'_{\delta_i} \quad = \quad \mathtt{tt}_n \wedge \varphi_{\delta_i} \wedge \bigwedge_{j<i} \neg\varphi_{\delta_j},$$

for all $i < k$ as above, and

$$\varphi'_{\delta_k} \quad = \quad \mathtt{tt}_n \wedge \bigwedge_{j<k} \neg\varphi_{\delta_j}.$$

Thus, the modal formulas in FTL($\mathcal{L}$) over Σ associated with a language $L \subseteq T_\Delta$ in $\mathcal{L}$ may equivalently be written as $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, where the family $(\varphi_\delta)_{\delta\in\Delta}$ satisfies the following condition: For each tree $t \in T_\Sigma$ there is exactly one δ with $t \models \varphi_\delta$, and if the root of t is labeled in Σ_n, then this unique δ belongs to Δ_n. Below we will call such families $(\varphi_\delta)_{\delta\in\Delta}$ *deterministic*. Accordingly, we will sometimes write modal formulas over Σ associated with a language $L \subseteq T_\Delta$ as $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, where $(\varphi_\delta)_{\delta\in\Delta}$ is a deterministic family of formulas over Σ. When $(\varphi_\delta)_{\delta\in\Delta}$ is a deterministic family, we have $t \models L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ iff there exists a relabeling $\widehat{t}$ of t in L such that for all vertices v, $t_v \models \varphi_{\widehat{t}(v)}$. We call a formula φ deterministic if for every subformula of φ of the form $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, the family $(\varphi_\delta)_{\delta\in\Delta}$ is deterministic. As shown above, for each $\varphi \in$ FTL($\mathcal{L}$) there is a deterministic formula in FTL($\mathcal{L}$) which is equivalent to φ.

Remark 1.4.1 When $R = \{0, 1\}$, our logics FTL($\mathcal{L}$) are closely related to the extended propositional linear temporal logics introduced in Wolper [26].

Example 1.4.2 For each alphabet Σ, **FTL**($\emptyset$) consists of those languages that are unions of languages of the form $\{\sigma(t_1, \ldots, t_n) : t_1, \ldots, t_n \in T_\Sigma\}$, where $\sigma \in \Sigma_n$, $n \geq 0$.

Example 1.4.3 The *boolean ranked alphabet* Bool has exactly two symbols of rank n, for each $n \in R$, the symbols $\uparrow_n$ and $\downarrow_n$. Below we assume that the lexicographic order on Bool satisfies $\uparrow_n < \downarrow_n$, for each $n \in R$.

For each $i \in [\max(R)]$, let L_{X_i} denote the regular tree language of all trees in T_{Bool} of depth ≥ 1 such that the root has n immediate successors for some $n \geq i$, and the ith immediate successor of the root is labeled by $\uparrow_m$, for some m. Then the modal operator corresponding to L_{X_i} is a sort of next modality: When $(\varphi_\delta)_{\delta\in\mathrm{Bool}}$ is a family of formulas over Σ and $t \in T_\Sigma$, then $t \models L_{\mathrm{X}_i}(\delta \mapsto \varphi_\delta)_{\delta\in\mathrm{Bool}}$ iff the root of t is labeled by some symbol in Σ_n with $i \leq n$, and the ith immediate subtree satisfies $\varphi_{\uparrow_m}$, where m denotes the rank of the symbol labeling the root of this subtree. Let $L_{\mathrm{X}} = \cup_{i\in[\max(R)]} L_{\mathrm{X}_i}$. Then $t \models L_{\mathrm{X}}(\delta \mapsto \varphi_\delta)_{\delta\in\mathrm{Bool}}$ iff t is of depth ≥ 1 and the subtree of t rooted at some immediate successor of the root vertex satisfies $\varphi_{\uparrow_m}$ for that m for which the successor is labeled in Σ_m. Thus, when φ is a fixed formula over Σ and $\varphi_{\uparrow_n} = \varphi$, for all $n \in R$, then $t \models L_{\mathrm{X}}(\delta \mapsto \varphi_\delta)_{\delta\in\mathrm{Bool}}$ for a tree t in T_Σ iff the depth of t is at least 1 and the subtree rooted at some immediate successor of the root of t satisfies φ. We may denote this formula by $\mathrm{X}_i\varphi$. Conversely, if $(\varphi_\delta)_{\delta\in\mathrm{Bool}}$ is any family of formulas over Σ, then $L_{\mathrm{X}_i}(\delta \mapsto \varphi_\delta)_{\delta\in\mathrm{Bool}}$ may be expressed as $\mathrm{X}_i(\bigwedge_{n\in R}(\mathtt{tt}_n \to \varphi_{\uparrow_n})$.

Next, let $L_{\text{EF}} \subseteq T_{\text{Bool}}$ denote the regular language of those trees in T_{Bool} having at least one vertex labeled in $\{\uparrow_n : n \in R\}$. Then for any $(\varphi_\delta)_{\delta \in \text{Bool}}$ and t as above, $t \models L_{\text{EF}}(\delta \mapsto \varphi_\delta)_{\delta \in \text{Bool}}$ iff the subtree rooted at some vertex labeled in Σ_n, for some n, satisfies $\varphi_{\uparrow_n}$. Thus, the modal operator corresponding to this language L_{EF} is closely related to the EF modality of CTL, cf. [19]. In the same way, the CTL-modalities AG, EG, AF are closely related to the modal operators associated with the following languages, where we use the letter p to range over the maximal paths of a tree, while v ranges over vertices:

$$\begin{aligned} L_{\text{AG}} &= \{t \in T_{\text{Bool}} : \forall v\ t(v) \in \{\uparrow_n : n \in R\}\} \\ L_{\text{EG}} &= \{t \in T_{\text{Bool}} : \exists p \forall v \in p\ t(v) \in \{\uparrow_n : n \in R\}\} \\ L_{\text{AF}} &= \{t \in T_{\text{Bool}} : \forall p \exists v \in p\ t(v) \in \{\uparrow_n : n \in R\}\}. \end{aligned}$$

Example 1.4.4 Next we define tree languages such that the corresponding modal operators are closely related to the EU and AU modalities of CTL. For this reason, we consider the ranked alphabet Tern having three symbols for each $n \in R$, $\uparrow_n, \vee_n, \downarrow_n$, ordered as indicated. Below we will write $u < v$ for vertices u, v in the tree t to express that v is strictly below u, i.e., $u \neq v$ and v is a vertex of the subtree t_u rooted at u. Let

$$\begin{aligned} L_{\text{EU}} &= \{t \in T_{\text{Tern}} : \exists p \exists v \forall u < v\ (t(v) \in \{\uparrow_n : n \in R\} \wedge t(u) \in \{\vee_n : n \in R\})\} \\ L_{\text{AU}} &= \{t \in T_{\text{Tern}} : \forall p \exists v \forall u < v\ (t(v) \in \{\uparrow_n : n \in R\} \wedge t(u) \in \{\vee_n : n \in R\})\}, \end{aligned}$$

where p ranges over maximal paths as above. The modal operators corresponding to these languages are closely related to the EU and AU modalities of CTL.

Example 1.4.5 Last, we consider a version of modular counting. Let $d > 1$ and r with $0 \leq r < d$ be given natural numbers. Let $L_{d,r}$ denote the set of all those trees in T_{Bool} such that the number of vertices labeled in $\{\uparrow_n : n \in R\}$ is congruent to r modulo d. If t is a tree in T_Σ and $(\varphi_\delta)_{\delta \in \text{Bool}}$ is a family of formulas over Σ, then $t \models L_{d,r}(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$ iff the number of vertices v labeled in Σ_n, $n \in R$ with $t_v \models \varphi_{\uparrow_n}$ is congruent to r modulo d.

1.5 Basic Results

In this section, we establish some elementary properties of the classes $\mathbf{FTL}(\mathcal{L})$, where $\mathcal{L}$ denotes a class of tree languages. We also study conditions on $\mathcal{L}$ and $\mathcal{L}'$ under which $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathcal{L}')$. We again assume that a rank type R with $0 \in R$ is fixed and that all considered ranked alphabets have rank type R.

Theorem 1.5.1 *For each class $\mathcal{L}$ of tree languages, $\mathbf{FTL}(\mathcal{L})$ contains $\mathcal{L}$ and is closed with respect to the boolean operations and inverse literal tree homomorphisms.*

Proof. It is obvious that $\mathbf{FTL}(\mathcal{L})$ is closed under the boolean operations. Moreover, each language $L \subseteq T_\Sigma$ in $\mathcal{L}$ is definable by the formula $L(\sigma \mapsto p_\sigma)_{\sigma\in\Sigma}$ in $\mathrm{FTL}(\mathcal{L})$. Assume now that $h : T_{\Sigma'} \to T_\Sigma$ is a literal tree homomorphism. We argue by induction on the structure of the formula φ over Σ in $\mathrm{FTL}(\mathcal{L})$ to show that $h^{-1}(L_\varphi)$ is definable by some formula ψ in $\mathrm{FTL}(\mathcal{L})$. When $\varphi = p_\sigma$, for some letter σ, then we define $\psi = \bigvee_{h(\sigma')=\sigma} p_{\sigma'}$. When σ is not in the range of h then this formula is ff. It is clear that $L_\psi = h^{-1}(L_\varphi)$. Suppose now that $\varphi = \varphi_1 \vee \varphi_2$ and that $L_{\psi_i} = h^{-1}(L_{\varphi_i})$, $i = 1, 2$. Then we define $\psi = \psi_1 \vee \psi_2$. When $\varphi = \neg\varphi_1$ and $L_{\psi_1} = h^{-1}(L_{\varphi_1})$, then let $\psi = \neg\psi_1$. In either case, we have $L_\psi = h^{-1}(L_\varphi)$. Finally, assume that $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, and that for each δ there is a formula ψ_δ in $\mathrm{FTL}(\mathcal{L})$ with $L_{\psi_\delta} = h^{-1}(L_{\varphi_\delta})$. Then define $\psi = L(\delta \mapsto \psi_\delta)_{\delta\in\Delta}$. Let $t \in T_{\Sigma'}$. Since for all $\delta \in \Delta$ and vertex v,

$$t_v \models \psi_\delta \Leftrightarrow h(t_v) \models \varphi_\delta \Leftrightarrow (h(t))_v \models \varphi_\delta,$$

the characteristic tree determined by t and the formulas $(\psi_\delta)_{\delta\in\Delta}$ is the same as that determined by $h(t)$ and the formulas φ_δ. It follows that $t \models \psi$ iff $h(t) \models \varphi$. □

To prove that **FTL** is a closure operator, we need:

Lemma 1.5.2 *Suppose that $(\varphi_\delta)_{\delta\in\Delta}$ is a deterministic family of formulas over Σ and $(\tau_\gamma)_{\gamma\in\Gamma}$ is a deterministic family of formulas over Δ. Let $t \in T_\Sigma$ and $\widehat{t}$ the characteristic tree determined by t and $(\varphi_\delta)_{\delta\in\Delta}$. Then let s be the characteristic tree determined by $\widehat{t}$ and the family $(\tau_\gamma)_{\gamma\in\Gamma}$. Then s is also the characteristic tree determined by t and the family $(L_{\gamma\mapsto\tau_\gamma}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta})_{\gamma\in\Gamma}$.*

Proof. First note that $(L_{\tau_\gamma}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta})_{\gamma\in\Gamma}$ is also a deterministic family. Given t, for every vertex v,

$$t_v \quad\models\quad L_{\tau_{s(v)}}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta},$$

since for every vertex w below v, $t_w \models \varphi_{\widehat{t}(w)}$, and since $\widehat{t}_v \in L_{\tau_{s(v)}}$. □

Next we show that **FTL** is a closure operator.

Theorem 1.5.3 **FTL** *is a closure operator on language classes.*

Proof. We have already seen that $\mathcal{L} \subseteq \mathbf{FTL}(\mathcal{L})$ holds for all $\mathcal{L}$. It is clear that $\mathbf{FTL}(\mathcal{L}_1) \subseteq \mathbf{FTL}(\mathcal{L}_2)$ whenever $\mathcal{L}_1 \subseteq \mathcal{L}_2$. Thus, to complete the proof, it suffices to show that for any class $\mathcal{L}$ of languages, $\mathbf{FTL}(\mathbf{FTL}(\mathcal{L})) = \mathbf{FTL}(\mathcal{L})$. The inclusion from right to left follows from Theorem 1.5.1. To prove that $\mathbf{FTL}(\mathbf{FTL}(\mathcal{L})) \subseteq \mathbf{FTL}(\mathcal{L})$, we argue by induction on the structure of the formula φ over Δ in $\mathrm{FTL}(\mathcal{L})$ to show that for every deterministic family $(\varphi_\delta)_{\delta\in\Delta}$ of formulas in $\mathrm{FTL}(\mathcal{L})$ over an alphabet Σ, the formula $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is expressible in $\mathrm{FTL}(\mathcal{L})$, i.e., there exists a formula in $\mathrm{FTL}(\mathcal{L})$ which is equivalent to it. Assume first that $\varphi = p_{\delta_0}$, for some $\delta_0 \in \Delta_{n_0}$. Then L_φ is the set of all trees in T_Δ whose root is labeled δ_0. It is clear that a tree $t \in T_\Sigma$ satisfies $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ iff t satisfies φ_{δ_0}, so that $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to φ_{δ_0}. In the induction step, assume first that $\varphi = \varphi_1 \vee \varphi_2$. Then $L_\varphi = L_{\varphi_1} \cup L_{\varphi_2}$ and thus $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to $L_{\varphi_1}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta} \vee L_{\varphi_2}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$. By induction, there exist ψ_1 and ψ_2 in FTL($\mathcal{L}$) such that $L_{\varphi_i}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to ψ_i, $i = 1, 2$. It follows that $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to $\psi_1 \vee \psi_2$ which is in FTL($\mathcal{L}$). Suppose next that $\varphi = \neg\varphi_1$, so that $L_\varphi = \overline{L_{\varphi_1}}$, the complement of L_{φ_1}. Then $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to $\neg(L_{\varphi_1}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta})$. It follows from the induction hypothesis that $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to a formula in FTL($\mathcal{L}$). Assume finally that $\varphi = K(\gamma \mapsto \tau_\gamma)_{\gamma\in\Gamma}$, where $K \subseteq T_\Gamma$, $K \in \mathcal{L}$ and the family $(\tau_\gamma)_{\gamma\in\Gamma}$ is deterministic. Let $\widehat{t}$ denote the characteristic tree determined by t and the family $(\varphi_\delta)_{\delta\in\Delta}$, and let s denote the characteristic tree determined by $\widehat{t}$ and the family $(\tau_\gamma)_{\gamma\in\Gamma}$. By Lemma 1.5.2, s is also the characteristic tree determined by t and $(L_{\gamma\mapsto\tau_\gamma}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta})_{\gamma\in\Gamma}$. Thus,

$$t \models K(\gamma \mapsto L_{\tau_\gamma}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta})_{\gamma\in\Gamma} \quad \Leftrightarrow \quad s \in K.$$

We have thus shown that $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to $K(\gamma \mapsto L_{\tau_\gamma}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta})_{\gamma\in\Gamma}$. By the induction hypothesis, for each γ there is a formula ψ_γ in FTL($\mathcal{L}$) which is equivalent to $L_{\tau_\gamma}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$. Thus, $L_\varphi(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is equivalent to $K(\gamma \mapsto \psi_\gamma)_{\gamma\in\Gamma}$. □

The language classes **FTL**($\mathcal{L}$) are not necessarily closed under quotients. However, we have:

Theorem 1.5.4 *The following conditions are equivalent for a class of tree languages $\mathcal{L}$:*

1. **FTL**($\mathcal{L}$) *is closed with respect to quotients.*
2. *Each quotient of any language in $\mathcal{L}$ belongs to* **FTL**($\mathcal{L}$).
3. *For each formula $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ in* FTL($\mathcal{L}$), *over any alphabet Σ, and for each context c over Δ there is a formula in* FTL($\mathcal{L}$) *which is equivalent to $(c^{-1}L)(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$.*
4. *For each formula $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ in* FTL($\mathcal{L}$), *over any alphabet Σ, and for each primitive context c over Δ there is a formula in* FTL($\mathcal{L}$) *which is equivalent to $(c^{-1}L)(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$.*

Proof. It is clear that the first condition implies the second and the third condition implies the fourth. Moreover, the second condition implies the third by Theorem 1.5.3. It remains to show that the fourth condition implies the first. Suppose that φ is a formula over Σ in FTL($\mathcal{L}$) and c is a primitive context over Σ. We show that $c^{-1}L_\varphi$ belongs to **FTL**($\mathcal{L}$). It follows by a straightforward induction that **FTL**($\mathcal{L}$) is closed under quotients with respect to any context.

When φ is p_σ, for a letter $\sigma \in \Sigma$, and the root of c is labeled by a letter other than σ, then $c^{-1}L_\varphi$ is $\emptyset$, which is definable by the formula $\mathbf{ff}$. When the root of c is labeled σ then $c^{-1}L_\varphi = T_\Sigma$ which is defined by the formula $\mathbf{tt}$. We continue by induction on the structure of φ. Suppose that $\varphi = \varphi_1 \vee \varphi_2$ or $\varphi = \neg\varphi_1$, and assume that $c^{-1}L_{\varphi_i}$ is defined by $\widetilde{\varphi}_i$ in FTL($\mathcal{L}$), $i = 1, 2$. Then $c^{-1}L_\varphi$ is defined by $\widetilde{\varphi}_1 \vee \widetilde{\varphi}_2$ or $\neg\widetilde{\varphi}_1$, respectively. Assume finally that φ is $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, where $L \subseteq T_\Delta$ and $(\varphi_\delta)_{\delta\in\Delta}$ is a deterministic family. Suppose that the root of c is labeled in Σ_{n_0}. Then for each $\delta_0 \in \Delta_{n_0}$, let c_{δ_0} denote the context over Δ obtained from c by relabeling its root by δ_0 and any other vertex u of c labeled by a letter in Σ_m, $m \geq 0$ by that letter $\delta \in \Delta_m$ such that the subtree of c rooted at u satisfies φ_δ. By the induction assumption, for any $\delta \in \Delta$ there exists a formula $\widetilde{\varphi}_\delta$ in FTL($\mathcal{L}$) defining $c^{-1}L_{\varphi_\delta}$. Moreover, by assumption, for each $\delta_0 \in \Delta_{n_0}$ there is a formula τ_{δ_0} in FTL($\mathcal{L}$) such that for all trees $t \in T_\Sigma$,

$$t \models \tau_{\delta_0} \quad \Leftrightarrow \quad t \models (c_{\delta_0}^{-1}L)(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}.$$

Then let

$$\widetilde{\varphi} \quad = \bigvee_{\delta_0\in\Delta_{n_0}} (\widetilde{\varphi}_{\delta_0} \wedge \tau_{\delta_0}).$$

We have, for all $t \in T_\Sigma$,

$$\begin{aligned}
t \models \widetilde{\varphi} \quad &\Leftrightarrow \quad \exists\delta_0 \in \Delta_{n_0}\ t \models \widetilde{\varphi}_{\delta_0} \wedge t \models \tau_{\delta_0} \\
&\Leftrightarrow \quad \exists\delta_0 \in \Delta_{n_0}\ c(t) \models \varphi_{\delta_0} \wedge t \models (c_{\delta_0}^{-1}L)(\delta \mapsto \varphi_\delta)_{\delta\in\Delta} \\
&\Leftrightarrow \quad \exists\delta_0 \in \Delta_{n_0}\ c(t) \models \varphi_{\delta_0} \wedge \exists s \in c_{\delta_0}^{-1}L\ \forall v\ t_v \models \varphi_{s(v)} \\
&\Leftrightarrow \quad \exists\delta_0 \in \Delta_{n_0}, s \in T_\Delta\ c_{\delta_0}(s) \in L \wedge c(t) \models \varphi_{\delta_0} \wedge \forall v\ t_v \models \varphi_{s(v)} \\
&\Leftrightarrow \quad c(t) \models L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta} \\
&\Leftrightarrow \quad c(t) \models \varphi.
\end{aligned}$$

This concludes the proof of Theorem 1.5.4. □

Corollary 1.5.5 1. *For any class $\mathcal{L}$ of tree languages,* $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathcal{L}')$, *where $\mathcal{L}'$ is the least class containing $\mathcal{L}$ closed with respect to the boolean operations and inverse literal morphisms.*

2. *For any class $\mathcal{L}$ of tree languages closed with respect to quotients, or such that the modal operators associated with the quotients of the languages in $\mathcal{L}$ are expressible in* FTL($\mathcal{L}$) *as in Theorem 1.5.4,* $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathcal{L}')$, *where $\mathcal{L}'$ is the least class containing $\mathcal{L}$ closed with respect to the boolean operations, quotients, and inverse literal morphisms.*

Suppose that $\mathbf{K}$ is a class of tree automata. We let $\mathcal{L}_\mathbf{K}$ denote the class of all tree languages recognizable by the tree automata in $\mathbf{K}$. Conversely, when $\mathcal{L}$ is a class of tree languages, let $\mathbf{K}_\mathcal{L}$ denote the class of all minimal tree automata of the languages in $\mathcal{L}$. For each class $\mathbf{K}$ of tree automata, we define FTL($\mathbf{K}$) = FTL($\mathcal{L}_\mathbf{K}$) and $\mathbf{FTL}(\mathbf{K}) = \mathbf{FTL}(\mathcal{L}_\mathbf{K})$.

Corollary 1.5.6 *Let $\mathcal{L}$ denote a class of regular tree languages. The following conditions are equivalent.*

1. $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathbf{K}_{\mathcal{L}})$.
2. *There exists some class* $\mathbf{K}$ *of finite tree automata with* $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathbf{K})$.
3. *There exists some class* $\mathcal{L}'$ *of regular tree languages closed with respect to quotients with* $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathcal{L}')$.
4. *Each quotient of any language in* $\mathcal{L}$ *belongs to* $\mathbf{FTL}(\mathcal{L})$.
5. $\mathbf{FTL}(\mathcal{L})$ *is closed with respect to quotients.*
6. *For each* L *in* $\mathcal{L}$*, the modalities associated with the quotients of* L *are expressible in* $\mathrm{FTL}(\mathcal{L})$ *as in Theorem 1.5.4.*

Proof. The last three conditions are equivalent by Theorem 1.5.4. The first condition clearly implies the second and the second the third which in turn implies the fourth, since for a class $\mathbf{K}$ of tree automata, $\mathcal{L}_{\mathbf{K}}$ is closed under quotients. It remains to show that the fourth condition implies the first. But by Lemma 1.3.1, when $\mathcal{L}$ consists of regular languages, every language recognizable by a tree automaton in $\mathbf{K}_{\mathcal{L}}$ is a boolean combination of quotients of some language in $\mathcal{L}$. It follows using Theorem 1.5.3 that $\mathbf{FTL}(\mathbf{K}_{\mathcal{L}}) \subseteq \mathbf{FTL}(\mathcal{L})$, while the reverse inclusion is obvious. $\square$

Example 1.5.7 Let L be any of the languages L_{X_i}, $i \in [\max(R)]$, L_{EF}, L_{EG}, L_{EU}, L_{AF}, L_{AG}, L_{AU}. Then each quotient of L is definable in $\mathrm{FTL}(\{L\})$. Thus, if $\mathcal{L}$ is any subcollection of these languages, then the equivalent conditions of Corollary 1.5.6 hold for $\mathcal{L}$.

1.6 A Variety Theorem

Several different concepts of varieties of regular tree languages with corresponding variety theorems have been proposed in the literature, cf. [1, 2, 20, 21, 9, 10]. The abundance of variety theorems is due to the fact that there exist several different reasonable notions of homomorphisms and quotients for trees, and the notion of syntactic algebra can be defined in several different frameworks: ordinary algebras, Almeida [1, 2], Steinby [20, 21], clones or Lawvere theories, Ésik [9], or preclones, Ésik and Weil [10]. Here we present yet another variety theorem that bears close connection to that given in Steinby [21].

In this section, all ranked alphabets are assumed to have a fixed common rank type R containing 0. Suppose that $\mathbb{A}$ and $\mathbb{B}$ are Σ-tree automata. Since tree automata are Σ-algebras, the direct product of $\mathbb{A}$ and $\mathbb{B}$ as an algebra is defined. However, the direct product may not be a tree automaton since it is not always generated by the constants. Therefore we define the *tree automaton direct product*, or *ta-direct product* of $\mathbb{A}$ and $\mathbb{B}$ as the smallest subalgebra of the usual direct product. The direct product of any finite number of tree automata is defined in the same way. We have already defined renamings of algebras. This notion gives rise to *tree automaton renamings*, or *ta-renamings.* Suppose that $\mathbb{A}$ is a Σ-tree automaton and the Δ-algebra $\mathbb{B}$ is a renaming of $\mathbb{A}$. (Δ is also of rank type R.) Then $\mathbb{B}$ has a least subalgebra which is called a ta-renaming of $\mathbb{A}$.

Recall that if $\mathbb{A}$ and $\mathbb{B}$ are Σ-tree automata and h is a homomorphism $\mathbb{A} \to \mathbb{B}$, then h is necessarily surjective. Thus we call $\mathbb{B}$ a *quotient* of $\mathbb{A}$.

For the purposes of this paper, we define a *(pseudo)variety of finite tree automata* to be any nonempty class of finite tree automata closed under the ta-direct product, ta-renaming and quotients. A *literal variety of tree languages* is any nonempty class of regular tree languages closed under the boolean operations, quotients and inverse literal tree homomorphisms. It is clear that both varieties of tree automata and literal varieties form (algebraic) lattices.

The relevance of literal varieties to the logics $\mathrm{FTL}(\mathcal{L})$ is justified by the following fact:

Corollary 1.6.1 *When $\mathcal{L}$ is a class of regular languages such that each quotient of any language in $\mathcal{L}$ belongs to $\mathbf{FTL}(\mathcal{L})$, then $\mathbf{FTL}(\mathcal{L})$ is a literal variety. Thus, when $\mathbf{K}$ is a class of finite tree automata, then $\mathbf{FTL}(\mathbf{K})$ is a literal variety.*

Proof. By Theorem 1.5.1 and Corollary 1.5.6. The fact that when $\mathcal{L}$ consists of regular languages then $\mathbf{FTL}(\mathcal{L})$ is a class of regular languages will be established independently in Corollary 1.9.9. Alternatively, one can embed $\mathrm{FTL}(\mathcal{L})$ into monadic second-order logic and use one direction of the main result of Thatcher and Wright [24] to the effect that every language definable in this logic is regular. □

Theorem 1.6.2 *The lattice of varieties of finite tree automata is isomorphic to the lattice of literal varieties of tree languages. An isomorphism is given by the assignment that maps each variety $\mathbf{V}$ of finite tree automata to the class $\mathcal{V} = \mathcal{L}_{\mathbf{V}}$ of those tree languages recognizable by the members of $\mathbf{V}$.*

Proof. If $\mathbf{V}$ is a variety of finite tree automata then the class $\mathcal{V} = \mathcal{L}_{\mathbf{V}}$ of regular tree languages is clearly nonempty and closed under the boolean operations (since $\mathbf{V}$ is closed under the direct product), quotients and inverse literal tree homomorphisms (since $\mathbf{V}$ is closed under renamings). We show that every literal variety $\mathcal{V}$ of tree languages corresponds to some variety of finite tree automata. Given $\mathcal{V}$, let $\mathbf{V}$ consist of those finite tree automata that only accept languages in $\mathcal{V}$. It is clear that $\mathbf{V}$ is nonempty. Since any language recognized by the ta-direct product of two finite tree automata is a boolean combination of languages recognized by the two tree automata, it follows that $\mathbf{V}$ is closed under the ta-direct product. Since $\mathcal{V}$ is closed under inverse literal homomorphisms, we have that $\mathbf{V}$ is closed under ta-renamings. Finally, since any language recognizable by a quotient of a tree automaton $\mathbb{A}$ is recognizable by $\mathbb{A}$, $\mathbf{V}$ is closed under quotients. Thus, $\mathbf{V}$ is a variety of finite tree automata. Let $\mathcal{W}$ denote the literal variety $\mathcal{L}_{\mathbf{V}}$ of all tree languages recognizable by the members of $\mathbf{V}$. We want to show that $\mathcal{V} = \mathcal{W}$. The inclusion $\mathcal{W} \subseteq \mathcal{V}$ is clear. Suppose now that $L \subseteq T_\Sigma$ is in $\mathcal{V}$ and consider the minimal tree automaton $\mathbb{A}_L$ of L. We know from Lemma 1.3.1 that every language recognizable by $\mathbb{A}_L$ is a boolean combination of quotients of L. It follows that every language recognizable by $\mathbb{A}_L$ is in $\mathcal{V}$, so that $\mathbb{A}_L \in \mathbf{V}$. Thus, since L is recognizable by $\mathbb{A}_L$, we have $L \in \mathcal{W}$. This proves that $\mathcal{V} \subseteq \mathcal{W}$.

Suppose that $\mathbf{V}$ is a variety of finite tree automata with corresponding literal variety $\mathcal{V}$. Let $\mathbf{W}$ denote the class of all finite tree automata that only accept languages in $\mathcal{V}$. By the above argument, we know that $\mathbf{W}$ is also a variety and is mapped to $\mathcal{V}$ under the correspondence given in the Theorem. It is clear that $\mathbf{V} \subseteq \mathbf{W}$. We want to show the reverse inclusion. So let $\mathbb{A}$ be a Σ-tree automaton in $\mathbf{W}$. For each $a \in A$, let $L_a \subseteq T_\Sigma$ denote the tree language accepted by $\mathbb{A}$ with unique final state a. Now each L_a is in $\mathcal{V}$ and thus recognizable by some tree automaton $\mathbb{B}_a$ in $\mathbf{V}$. Let $\mathbb{B}$ denote the direct product of the $\mathbb{B}_a$. Note that $\mathbb{B} \in \mathbf{V}$. We claim that $\mathbb{A}$ is a quotient of $\mathbb{B}$. For each element $b \in B$ there is a tree $t \in T_\Sigma$ with $b = t_{\mathbb{B}} = (t_{\mathbb{B}_a})_{a \in A}$. We map b to $h(b) = t_{\mathbb{A}}$. This map is well-defined, for if $t_{\mathbb{B}} = s_{\mathbb{B}}$, for $t, s \in T_\Sigma$, then for each a, $t_{\mathbb{B}_a} = s_{\mathbb{B}_a}$, so that $t \in L_a$ iff $s \in L_a$. This means that $t_{\mathbb{A}} = s_{\mathbb{A}}$. Since it is clear that h is a homomorphism, $\mathbb{A}$ is a quotient of $\mathbb{B}$, proving that $\mathbb{A} \in \mathbf{V}$.

To complete the proof, assume now that $\mathbf{V}_1$ and $\mathbf{V}_2$ are varieties of finite tree automata with corresponding literal varieties $\mathcal{V}_1$ and $\mathcal{V}_2$. If $\mathbf{V}_1 \subseteq \mathbf{V}_2$, then clearly $\mathcal{V}_1 \subseteq \mathcal{V}_2$. Assume that $\mathcal{V}_1 \subseteq \mathcal{V}_2$. Then since $\mathbf{V}_i$ consists of all finite tree automata that only accept languages in $\mathcal{V}_i$, $i = 1, 2$, it follows that $\mathbf{V}_1 \subseteq \mathbf{V}_2$. □

Remark 1.6.3 Suppose that $\mathbf{V}$ is a variety of finite tree automata and $\mathcal{V}$ is the corresponding literal variety. Then a tree language belongs to $\mathcal{V}$ iff its minimal tree automaton is in $\mathbf{V}$. Moreover, $\mathbf{V}$ is the least variety of finite automata containing the minimal automata of the languages in $\mathcal{V}$, and a tree automaton $\mathbb{A}$ is in $\mathbf{V}$ iff every language recognizable by $\mathbb{A}$ is in $\mathcal{V}$.

Example 1.6.4 The least literal variety of regular languages contains for each ranked alphabet Σ (of rank type R) just the languages $\emptyset$ and T_Σ. The corresponding variety of finite tree automata is the class of all *trivial*, i.e., singleton tree automata. The greatest literal variety is the class of all regular languages. The corresponding variety of finite tree automata is the class of all finite tree automata.

Example 1.6.5 For a nonnegative integer k, a tree language $L \subseteq T_\Sigma$ is called *k-definite* if the membership of a tree $t \in T_\Sigma$ in L only depends on the cut-off of t at depth k, i.e., on that part of the tree determined by the vertices of depth *strictly less* than k. By extension, a tree language is *definite* if it is k-definite for some k. For each k, let $\mathcal{D}_k$ denote the class of k-definite tree languages, and let $\mathcal{D}$ denote the class of definite tree languages, so that $\mathcal{D} = \bigcup_{k\geq 0} \mathcal{D}_k$. Note that for every ranked alphabet Σ, the only languages over Σ contained in $\mathcal{D}_0$ are $\emptyset$ and T_Σ. Any 1-definite language over Σ is a union of languages of the form $T_\sigma = \{\sigma(t_1, \ldots, t_n) : t_1, \ldots, t_n \in T_\Sigma\}$, where $\sigma \in \Sigma_n$, $n \geq 0$. In general, any k-definite tree language is a union of languages of the form $T_t = \{t(t_1, \ldots, t_n) : t_1, \ldots, t_n \in T_\Sigma\}$, where $t \in T_\Sigma(X_n), n \geq 0$ is of depth $< k$, moreover, each x_i occurs at most (or exactly) once in t and each leaf labeled in X_n is of depth k.

Definite tree languages were introduced by Heuter in [14] and subsequently studied by Nivat and Podelski [17] and Ésik in [8]. It is shown in these papers (though stated in different form) that $\mathcal{D}$ and each $\mathcal{D}_k$ is a literal variety of tree languages. The variety of finite tree automata corresponding to $\mathcal{D}_k$ can be described as follows. Call a Σ-algebra *k-definite* if it satisfies all equations (in the sense of Universal Algebra, cf. Grätzer [13]) $t = s$ such that the trees $t, s \in T_\Sigma(X_n)$, $n \geq 0$ agree up to depth k, i.e., s and t have equal cut-offs at depth k. (Actually, it suffices to require this condition for trees of depth $\leq k$.) A *definite algebra* is an algebra which is k-definite for some k. A *definite tree automaton* (*k-definite tree automaton*) is a tree automaton which is a definite algebra (k-definite algebra, resp.). We let $\mathbf{D}$ ($\mathbf{D}_k$, resp.) denote the class of all finite definite (k-definite, resp.) tree automata. For each k, $\mathbf{D}_k$ is the variety of tree automata corresponding to $\mathcal{D}_k$, moreover, $\mathbf{D}$ is the variety corresponding to $\mathcal{D}$. See also [9].

It is clear that there exists an algorithm to decide whether a finite algebra is k-definite. It follows that each $\mathcal{D}_k$ is decidable: Given a regular tree language L (by a finite tree automaton equipped with a set of final states), there is an effective procedure to test whether or not L is k-definite. In Heuter [14], it is shown that $\mathcal{D}$ is also decidable, see also Nivat and Podelski [17] and Ésik [8].

1.7 The Cascade Product

Let R be a rank type kept fixed in this section. All ranked sets will be assumed to be of rank type R.

Let $\mathbb{A}$ be a Σ-algebra, $\mathbb{B}$ a Δ-algebra, and α a family of functions $\alpha_n : A^n \times \Sigma_n \to \Delta_n$, $n \in R$. The *cascade product* $\mathbb{A} \times_\alpha \mathbb{B}$ determined by α is the Σ-algebra with carrier $A \times B$ and operations

$$\sigma((a_1, b_1), \ldots, (a_n, b_n)) \quad = \quad (\sigma(a_1, \ldots, a_n), \delta(b_1, \ldots, b_n)),$$

where $\delta = \alpha_n(a_1, \ldots, a_n, \sigma)$, for all $((a_1, b_1), \ldots, (a_n, b_n)) \in A \times B$, $\sigma \in \Sigma_n$,

$n \in R$. When $0 \in R$ and $\mathbb{A}$ and $\mathbb{B}$ are tree automata, the *ta-cascade product* of $\mathbb{A}$ and $\mathbb{B}$ determined by α is the least subalgebra of the above cascade product. We use the same notation $\mathbb{A} \times_\alpha \mathbb{B}$ to denote a ta-cascade product of tree automata $\mathbb{A}$ and $\mathbb{B}$. Moreover, below we will sometimes write just cascade product for the ta-cascade product, direct product for the ta-direct product, etc. The direct product is clearly a special case of the cascade product.

The cascade product of algebras (or tree automata) can be extended to several factors: $\mathbb{A}_1 \times_{\alpha_1} \mathbb{A}_2 \times_{\alpha_2} \ldots \times_{\alpha_{n-1}} \mathbb{A}_n$. Here, when $\mathbb{A}_i$ is a Σ_i-algebra, then α_i is a family of functions

$$(A_1 \times \ldots \times A_{i-1})^m \times (\Sigma_1)_m \quad \rightarrow \quad (\Sigma_i)_m, \quad m \in R.$$

Note that $\mathbb{A}_1 \times_{\alpha_1} \mathbb{A}_2 \times_{\alpha_2} \ldots \times_{\alpha_{n-1}} \mathbb{A}_n$ is a Σ_1-algebra (Σ_1-tree automaton).

Suppose that $\mathbb{A}$ is a Σ-algebra and α is a family of functions $A^n \times \Sigma_n \rightarrow \Delta_n$. Then we call the pair $(\mathbb{A}, \alpha)$ a *tree transducer*. For each $n \geq 0$, the tree transducer $(\mathbb{A}, \alpha)$ induces a mapping $f : T_\Sigma \rightarrow T_\Delta$, called the *relabeling induced by* $(\mathbb{A}, \alpha)$. Given a tree $t \in T_\Sigma(X_n)$, $f(t)$ is defined as follows. When $t = \sigma \in \Sigma_0$, then $f(t) = \alpha_0(\sigma)$. Suppose now that $t = \sigma(t_1, \ldots, t_m)$, where $m > 0$, $\sigma \in \Sigma_m$ and $t_1, \ldots, t_m \in T_\Sigma$. Then $f(t) = \delta(f(t_1), \ldots, f(t_m))$, where $\delta = \alpha_m((t_1)_\mathbb{A}, \ldots, (t_m)_\mathbb{A}, \sigma)$. More generally, when $t \in T_\Sigma(X_n)$ and $a_1, \ldots, a_n \in A$, $n \geq 0$, we define $f_{(a_1,\ldots,a_n)}(t)$ as follows: When $t = x_i$ with $i \in [n]$, then $f_{(a_1,\ldots,a_n)}(t) = x_i$, and when $t = \sigma \in \Sigma_0$, then $f_{(a_1,\ldots,a_n)}(t) = \alpha_0(\sigma)$. Moreover, when $t = \sigma(t_1, \ldots, t_m)$, where $m > 0$, $\sigma \in \Sigma_m$ and $t_1, \ldots, t_m \in T_\Sigma(X_n)$, then $f_{(a_1,\ldots,a_n)}(t) = \delta(f_{(a_1,\ldots,a_n)}(t_1), \ldots, f_{(a_1,\ldots,a_n)}(t_m))$, where

$$\delta = \alpha_m((t_1)_\mathbb{A}(a_1, \ldots, a_n), \ldots, (t_m)_\mathbb{A}(a_1, \ldots, a_n), \sigma).$$

Below we will write $\alpha(t)$ for $f(t)$ and $\alpha_{(a_1,\ldots,a_n)}(t)$ for $f_{(a_1,\ldots,a_n)}(t)$.

Proposition 1.7.1 *Suppose that $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ is a cascade product of the Σ-algebra $\mathbb{A}$ and the Δ-algebra $\mathbb{B}$. Then for any tree $t \in T_\Sigma$, $t_\mathbb{C} = (t_\mathbb{A}, s_\mathbb{B})$, where $s = \alpha(t)$ is the image of t with respect to the relabeling induced by $(\mathbb{A}, \alpha)$. More generally, for every $t \in T_\Sigma(X_n)$ and $(a_i, b_i) \in A \times B$, $i \in [n]$, $t_\mathbb{C}((a_1, b_1), \ldots, (a_n, b_n)) = (t_\mathbb{A}(a_1, \ldots, a_n), s_\mathbb{B}(b_1, \ldots, b_n))$, where $s = \alpha_{(a_1,\ldots,a_n)}(t)$. A similar fact holds for the ta-cascade product.*

Proof. By a straightforward induction on the structure of t. □

By a *closed variety of finite algebras* we mean a nonempty class of finite algebras (of the same rank type R) closed with respect to the cascade product, renamings, subalgebras, and homomorphic images. Similarly, a *closed variety of finite tree automata* is any nonempty class of finite tree automata closed with respect to the ta-cascade product, ta-renaming and quotients. It is clear that any closed variety of tree automata is a variety. For any class $\mathbf{K}$ of finite tree automata, we let $\widehat{\mathbf{K}}$ denote the least closed variety of finite tree automata containing $\mathbf{K}$. Moreover, when $\mathbf{V}$ and $\mathbf{W}$ are closed varieties of finite tree automata, then we define $\mathbf{V} \vee \mathbf{W}$ as the least closed variety containing both $\mathbf{V}$ and $\mathbf{W}$, i.e., $\mathbf{V} \vee \mathbf{W} = \widehat{\mathbf{V} \cup \mathbf{W}}$.

Remark 1.7.2 Suppose that $\mathbf{K}$ is a class of finite algebras that contains all of the trivial algebras. Then $\mathbf{K}$ is a closed variety iff it is closed under the cascade product, subalgebras, and homomorphic images. Similarly, if $\mathbf{K}$ is a class of finite tree automata containing the trivial (singleton) tree-automata, then $\mathbf{K}$ is a closed variety of finite tree automata iff it is closed under the cascade product and quotients.

Remark 1.7.3 Suppose that $\mathbf{K}$ is a nonempty class of finite algebras. It is known (cf., e.g., Ésik [8]) that the least closed variety containing $\mathbf{K}$ consists of all homomorphic images of subalgebras of cascade products $\mathbb{A}_1 \times_{\alpha_1} \mathbb{A}_2 \times_{\alpha_2} \ldots \times_{\alpha_{n-1}} \mathbb{A}_n$, where $\mathbb{A}_1$ is a renaming of an algebra in $\mathbf{K}$ and each $\mathbb{A}_i$ with $i > 1$ is in $\mathbf{K}$, or when $\mathbf{K}$ contains all the trivial algebras, each $\mathbb{A}_i$ is in $\mathbf{K}$. A similar fact holds for finite tree automata.

An example of a closed variety of finite algebras is the class of all finite definite algebras. Let $\mathbb{D}_0(R)$, or just $\mathbb{D}_0$ denote the Bool-algebra (i.e., Σ-algebra with $\Sigma = \text{Bool}$) with carrier $\{0,1\}$ and constant valued operations

$$\begin{aligned} \downarrow_n (a_1, \ldots, a_n) &= 0 \\ \uparrow_n (a_1, \ldots, a_n) &= 1, \quad n \in R. \end{aligned}$$

Note that when $0 \in R$, then $\mathbb{D}_0$ is a tree automaton. The following result was proved in Ésik [8].

Theorem 1.7.4 *The class of all finite definite algebras is the least closed variety containing the algebra* $\mathbb{D}_0$.

It follows that when $0 \in R$, then the class $\mathbf{D}$ of finite definite tree automata is a closed variety of finite tree automata and is generated by $\mathbb{D}_0$.

1.8 Definite Tree Languages, Revisited

In this section, all ranked alphabets are assumed to have a fixed rank type R with $0 \in R$.

We say that *the* next *modalities are expressible in the logic* FTL($\mathcal{L}$) if for all alphabets Σ and integers i with $i \in [\max(R)]$, and for every formula φ in FTL($\mathcal{L}$) over Σ, there exists a formula $\mathrm{X}_i\varphi$ in FTL($\mathcal{L}$) such that for all trees $t \in T_\Sigma$, $t \models \mathrm{X}_i\varphi$ iff t is of the form $\sigma(t_1, \ldots, t_n)$, where $n \geq i$, and $t_i \models \varphi$. More generally, there is a canonical way to assign a word w in $[\max(R)]^*$ to every vertex of a tree $t \in T_\Sigma$, the "address" of v (see, e.g., Nivat and Podelski

[17]). Given a word $w \in [\max(R)]^*$, we say that the modality X_w is expressible in FTL($\mathcal{L}$) if for every formula φ in FTL($\mathcal{L}$) over any alphabet Σ there exists a formula $\mathrm{X}_w\varphi$ in FTL($\mathcal{L}$) such that for all trees $t \in T_\Sigma$, $t \models \mathrm{X}_w\varphi$ iff t has a vertex v at the address w and the subtree t_v rooted at this vertex satisfies φ. It is clear that for all words w, w' and for all formulas φ, $\mathrm{X}_w\mathrm{X}_{w'}\varphi$ is equivalent to the formula $\mathrm{X}_{ww'}\varphi$.

The following fact is clear.

Proposition 1.8.1 *The following conditions are equivalent for a logic* FTL($\mathcal{L}$)*:*

1. *The next modalities are expressible in* FTL($\mathcal{L}$).
2. *For every w, the modality X_w is expressible in* FTL($\mathcal{L}$).
3. *For each $i \in [\max(R)]$,* **FTL**($\mathcal{L}$) *contains the language L_{X_i}.*

Proof. Since for each i, the formula $\mathrm{X}_i(\bigvee_{n\in R} \uparrow_n)$ defines L_{X_i} over the ranked alphabet Bool, the first condition implies the third. Moreover, since $\mathrm{X}_i\varphi$ is expressible as $L_{\mathrm{X}_i}(\delta \mapsto \psi_\delta)_{\delta\in\mathrm{Bool}}$, where $\psi_{\uparrow_n} = \varphi$ and $\psi_{\downarrow_n} = \neg\varphi$ for all $n \in R$, the third condition implies the first. □

The languages L_{X_i} were defined in Example 1.4.3. Let $\mathcal{L}_\mathrm{X} = \{L_{\mathrm{X}_i} : i \in [\max(R)]\}$. Below we denote the logic FTL($\mathcal{L}_\mathrm{X}$) by CTL(X), and the tree language class **FTL**($\mathcal{L}_\mathrm{X}$) by **CTL**(X).

Proposition 1.8.2 **CTL**(X) $= \mathcal{D}$.

Proof. We know that each definite language over Σ is a finite union of languages of the form T_t defined in Example 1.6.5. The property that a tree belongs to T_t is clearly expressible using the X_ws. For the reverse inclusion, one argues by induction on the structure of the formula φ over Σ in CTL(X) to show that $L_\varphi \in \mathcal{D}$. The base of the induction is clear. In the induction step, the case of boolean connectives is covered by the fact that $\mathcal{D}$ is a literal variety and is thus closed under the boolean operations. Finally, one proves that if $(\varphi_\delta)_{\delta\in\mathrm{Bool}}$ define definite languages, then so does the formula $\varphi = L_{\mathrm{X}_i}(\delta \mapsto \varphi_\delta)_{\delta\in\mathrm{Bool}}$, for each i. Indeed, in this case L_φ is the collection of all trees whose root is labeled by a symbol of rank $\geq i$ such that the ith immediate subtree satisfies $\varphi_{\uparrow_n}$, where the root of this subtree is labeled in Σ_n. Now if $L_{\varphi_{\uparrow_n}}$ is k-definite, then L_φ is $(k+1)$-definite. □

Corollary 1.8.3 *The following conditions are equivalent for a logic* FTL($\mathcal{L}$)*:*

1. *The next modalities are expressible in* FTL($\mathcal{L}$).
2. *For every w, the modality X_w is expressible in* FTL($\mathcal{L}$).

3. $\mathcal{L}_X \subseteq \mathbf{FTL}(\mathcal{L})$.

4. $\mathcal{D}_2 \subseteq \mathbf{FTL}(\mathcal{L})$.

5. $\mathcal{D} \subseteq \mathbf{FTL}(\mathcal{L})$.

Corollary 1.8.4 $\mathbf{FTL}(\mathcal{L}) = \mathcal{D}$ *iff* $\mathcal{L} \subseteq \mathcal{D}$ *and* $\mathcal{L}_X \subseteq \mathbf{FTL}(\mathcal{L})$.

1.9 Main Results

In this section, all ranked sets are assumed to have a fixed rank type R with $0 \in R$. In the next two proposition, let $\mathcal{L}$ denote a class of tree languages.

Proposition 1.9.1 *Suppose that* $\mathbb{A}$ *and* $\mathbb{B}$ *are finite tree automata and* $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ *is a ta-cascade product of* $\mathbb{A}$ *and* $\mathbb{B}$. *If every language recognizable by* $\mathbb{A}$ *or* $\mathbb{B}$ *belongs to* $\mathbf{FTL}(\mathcal{L})$, *and if the next modality is expressible in* $\mathrm{FTL}(\mathcal{L})$, *then every language recognizable by* $\mathbb{C}$ *also belongs to* $\mathbf{FTL}(\mathcal{L})$.

Proof. Suppose that $\mathbb{A}$ is a Σ-tree automaton and $\mathbb{B}$ is a Δ-tree automaton, so that $\mathbb{C}$ is a Σ-tree automaton. Let h denote the unique homomorphism $T_\Sigma \to \mathbb{C}$. It suffices to show that for each $(a, b) \in C$, the language $h^{-1}((a, b))$ belongs to $\mathbf{FTL}(\mathcal{L})$.

For every $t \in T_\Sigma$, $t_\mathbb{C} = (t_\mathbb{A}, s_\mathbb{B})$, where $s = \alpha(t)$ is the image of t under the relabeling induced by the tree transducer $(\mathbb{A}, \alpha)$. (See Proposition 1.7.1.) Thus,

$$h^{-1}((a,b)) \quad = \quad \{t : t_\mathbb{A} = a \wedge s_\mathbb{B} = b\}.$$

By assumption, for each $a \in A$ there exists a formula τ_a over Σ in $\mathrm{FTL}(\mathcal{L})$ defining the set of trees $h^{-1}(\pi^{-1}(a))$, where π denotes the projection $C \to A$, $(a, b) \mapsto a$. For each $b \in B$, let $T_b = \{s \in T_\Delta : s_\mathbb{B} = b\}$. We construct a (deterministic) family of formulas $(\varphi_\delta)_{\delta \in \Delta}$ over Σ such that for each tree $t \in T_\Sigma$, the characteristic tree determined by t and this family is exactly $\alpha(t)$. For each $\sigma \in \Sigma_n$, we define:

$$\varphi_\delta \quad = \quad \bigvee_{\alpha_n(a_1,\ldots,a_n,\sigma)=\delta} p_\sigma \wedge X_1\tau_{a_1} \wedge \ldots \wedge X_n\tau_{a_n}.$$

Given $(a, b) \in C$, let

$$\varphi \quad = \quad \tau_a \wedge T_b(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}.$$

Then we have

$$\begin{aligned} L_\varphi \quad &= \quad \{t \in T_\Sigma : t_\mathbb{A} = a \wedge \alpha(t) \in T_b\} \\ &= \quad \{t \in T_\Sigma : t_\mathbb{C} = (a, b)\}. \end{aligned}$$

Since by assumption T_b is definable in FTL($\mathcal{L}$) and the next modalities are expressible, it follows from Theorem 1.5.3 that there is a formula in FTL($\mathcal{L}$) which is equivalent to φ. $\square$

Proposition 1.9.2 *Suppose that $\varphi = K(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is a formula over Σ in* FTL($\mathcal{L}$)*, where $(\varphi_\delta)_{\delta\in\Delta}$ is a deterministic family. Suppose that K is recognizable by $\mathbb{B}$ and that each L_{φ_δ} is recognizable by $\mathbb{A}$, where $\mathbb{A}$ and $\mathbb{B}$ are possibly infinite tree automata. Then $L_\varphi \subseteq T_\Sigma$ is recognizable by a ta-cascade product of $\mathbb{A}$ and $\mathbb{B}$.*

Proof. Let h denote the unique homomorphism $T_\Sigma \to \mathbb{A}$ and h_K the unique homomorphism $T_\Delta \to \mathbb{B}$. For each $\delta \in \Delta$, let F_δ denote the set $h(L_{\varphi_\delta})$. Since $(\varphi_\delta)_{\delta\in\Delta}$ is a deterministic family, the sets F_δ are pairwise disjoint. For each $\sigma \in \Sigma_n$ and $a_1,\ldots,a_n \in A$, $n \in R$, define $\alpha_n(a_1,\ldots,a_n,\sigma) = \delta \in \Delta_n$ iff $\sigma_A(a_1,\ldots,a_n) \in F_\delta$. By the above remark, there is at most one such δ. To see that there is at least one, take $t_i \in T_\Sigma$ with $(t_i)_\mathbb{A} = a_i$, $i \in [n]$. Then let $t = \sigma(t_1,\ldots,t_n)$. There exists some $\delta \in \Delta_n$ with $t \models \varphi_\delta$. Therefore $\sigma_\mathbb{A}(a_1,\ldots,a_n) = t_\mathbb{A} \in F_\delta$. Now it follows that for every $t \in T_\Sigma$, the characteristic tree determined by t and the family $(\varphi_\delta)_{\delta\in\Delta}$ is exactly $\alpha(t)$, the image of t under the relabeling induced by $(\mathbb{A},\alpha)$. It follows that L_φ is recognized by $\mathbb{C}$ with set of final states $\{(a,b) : b \in h_K(K)\}$. $\square$

Theorem 1.9.3 *For any class $\mathbf{K}$ of finite tree automata, every language in the class $\mathbf{FTL}(\mathbf{K})$ is recognizable by some tree automaton in $\widehat{\mathbf{K}} \vee \mathbf{D}$.*

Proof. Let φ denote a deterministic formula over Σ in FTL($\mathbf{K}$). We show that L_φ is recognizable by some automaton in $\widehat{\mathbf{K}} \vee \mathbf{D}$. When φ is p_σ, for some $\sigma \in \Sigma$, then L_φ is 1-definite and thus recognizable by some automaton in $\mathbf{D}_1 \subseteq \mathbf{D}$. We continue by induction on the structure of φ. Assume that $\varphi = \varphi_1 \vee \varphi_2$ such that L_{φ_i} is recognizable by $\mathbb{A}_i$ in $\widehat{\mathbf{K}} \vee \mathbf{D}$, $i = 1,2$. Then L_φ is recognizable by the ta-direct product $\mathbb{A}_1 \times \mathbb{A}_2$ which is also in $\widehat{\mathbf{K}} \vee \mathbf{D}$. When $\varphi = \neg\varphi_1$, where L_{φ_1} is recognizable by $\mathbb{A}_1$ above, then L_φ is also recognizable by $\mathbb{A}_1$. Finally, when $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ and each L_{φ_δ} is recognizable by some tree automaton in $\widehat{\mathbf{K}} \vee \mathbf{D}$, then it follows by Proposition 1.9.2 that L_φ is recognizable by some tree automaton in $\widehat{\mathbf{K}} \vee \mathbf{D}$. (Note that since $\widehat{\mathbf{K}} \vee \mathbf{D}$ is closed with respect to the direct product, we may assume without loss of generality that each L_{φ_δ} is recognizable by the same tree automaton $\mathbb{A}$ in $\widehat{\mathbf{K}} \vee \mathbf{D}$). $\square$

Remark 1.9.4 In the above proof, we did not use the assumption that $\mathbf{K}$ consists of finite automata. Our argument gives that for any class $\mathbf{K}$ of possibly infinite tree automata, every language in $\mathbf{FTL}(\mathbf{K})$ is recognizable by some tree automaton in the least class of tree automata containing $\mathbf{K}$ and $\mathbf{D}$, closed with respect to the ta-cascade product.

Theorem 1.9.5 *Suppose that the next modalities are expressible in* FTL($\mathbf{K}$)*, where* $\mathbf{K}$ *is a class of finite tree automata. Then a language* L *belongs to* $\mathbf{FTL}(\mathbf{K})$ *iff its minimal tree automaton* $\mathbb{A}_L$ *belongs to* $\widehat{\mathbf{K}} \vee \mathbf{D}$ *iff* L *is recognizable by an automaton in* $\widehat{\mathbf{K}} \vee \mathbf{D}$.

Proof. We know from Corollary 1.8.3 that $\mathbf{FTL}(\mathbf{K})$ contains the definite tree languages and thus $\mathbf{FTL}(\mathbf{K}) = \mathbf{FTL}(\mathbf{K} \cup \mathbf{D})$, by Theorem 1.5.3. Let us define the *rank* of $\mathbb{A} \in \widehat{\mathbf{K}} \vee \mathbf{D}$ to be the smallest number of ta-cascade product and quotient operations needed to generate $\mathbb{A}$ from $\mathbf{K} \cup \mathbf{D}$. (Recall Remark 1.7.2.) We prove by induction on the rank of $\mathbb{A}$ that every language recognizable by $\mathbb{A}$ is in $\mathbf{FTL}(\mathbf{K} \cup \mathbf{D})$. When the rank is 0 we have $\mathbb{A} \in \mathbf{K} \cup \mathbf{D}$ and the result is immediate. When the rank of $\mathbb{A}$ is positive, then $\mathbb{A}$ is either a quotient of a tree automaton $\mathbb{B}$ in $\widehat{\mathbf{K}} \vee \mathbf{D}$ of smaller rank, or $\mathbb{A}$ is a ta-cascade product of some tree automata in $\widehat{\mathbf{K}} \vee \mathbf{D}$ of smaller rank. In the first case, every language recognizable by $\mathbb{A}$ is recognizable by $\mathbb{B}$. In the second case, the result follows from Proposition 1.9.1. Conversely, by Theorem 1.9.3, every language in $\mathbf{FTL}(\mathbf{K})$ is recognizable by some tree automaton in $\widehat{\mathbf{K}} \vee \mathbf{D}$. □

By combining Theorem 1.7.4 with the above result, we have:

Corollary 1.9.6 *Suppose that the next modalities are expressible in* FTL($\mathbf{K}$)*, where* $\mathbf{K}$ *is a class of finite tree automata. Then a language* L *belongs to* $\mathbf{FTL}(\mathbf{K})$ *iff its minimal tree automaton* $\mathbb{A}_L$ *belongs to* $\widehat{\mathbf{K} \cup \{\mathbb{D}_0\}}$ *iff* L *is recognizable by an automaton in* $\widehat{\mathbf{K} \cup \{\mathbb{D}_0\}}$.

Example 1.9.7 The assumption in the above result that the next modalities be expressible in FTL($\mathbf{K}$) is important. Indeed, let $\mathbf{K} = \emptyset$. Then $\mathbf{FTL}(\mathbf{K})$ is the class $\mathcal{D}_1$ of all 1-definite tree languages, while $\widehat{\mathbf{K}} \vee \mathbf{D} = \mathbf{D}$ which corresponds to the literal variety $\mathcal{D}$ of all definite tree languages that properly contains $\mathcal{D}_1$.

Corollary 1.9.8 *Suppose that* $\mathcal{L}$ *is a class of regular languages such that each quotient of any language in* $\mathcal{L}$ *belongs to* $\mathbf{FTL}(\mathcal{L})$ *and the next modalities are expressible in* FTL($\mathcal{L}$). *Then a language* L *belongs to* $\mathbf{FTL}(\mathbf{K})$ *iff its minimal tree automaton* $\mathbb{A}_L$ *belongs to* $\widehat{\mathbf{K}_{\mathcal{L}} \cup \{\mathbb{D}_0\}}$ *iff* L *is recognizable by some automaton in* $\widehat{\mathbf{K}_{\mathcal{L}} \cup \{\mathbb{D}_0\}}$.

Corollary 1.9.9 *For each class* $\mathcal{L}$ *of regular tree languages,* $\mathbf{FTL}(\mathcal{L})$ *consists of regular languages.*

Call a nonempty class of regular tree languages $\mathcal{L}$ *closed* if $\mathbf{FTL}(\mathcal{L}) \subseteq \mathcal{L}$ and if $\mathcal{L}$ is closed with respect to quotients. By Theorems 1.5.1 and 1.5.4, every closed class is a literal variety of tree languages. Moreover, by Corollary 1.5.6, $\mathcal{L}$ is closed iff $\mathcal{L} = \mathbf{FTL}(\mathcal{L}')$ for a class $\mathcal{L}'$ of regular tree languages closed with respect to quotients iff $\mathcal{L} = \mathbf{FTL}(\mathbf{K})$ for a class $\mathbf{K}$ of finite tree automata.

Recall that by Theorem 1.6.2, the assignment

$$\begin{aligned} \mathbf{V} \quad \mapsto \quad \mathcal{L}_{\mathbf{V}} &= \{L : L \textit{ is recognizable by some } \mathbb{A} \in \mathbf{V}\} \\ &= \{L : \mathbb{A}_L \in \mathbf{V}\} \end{aligned}$$

defines an order isomorphism between varieties $\mathbf{V}$ of finite tree automata and literal varieties $\mathcal{V}$ of tree languages. The inverse assignment maps a literal variety $\mathcal{V}$ to the class of those finite tree automata $\mathbb{A}$ such that every language recognizable by $\mathbb{A}$ belongs to $\mathcal{V}$.

Theorem 1.9.10 *If $\mathbf{V}$ is a closed variety of finite tree automata containing $\mathbf{D}$, then $\mathcal{L}_{\mathbf{V}} = \mathbf{FTL}(\mathbf{V})$. Moreover, the assignment $\mathbf{V} \mapsto \mathbf{FTL}(\mathbf{V})$ defines an order isomorphism between closed varieties $\mathbf{V}$ of finite tree automata containing $\mathbf{D}$ and closed classes $\mathcal{L}$ of regular tree languages containing $\mathcal{D}$.*

Proof. If $\mathbf{V}$ is a closed variety containing $\mathbf{D}$, then by Theorem 1.9.5 and Corollary 1.8.3, $\mathcal{L}_{\mathbf{V}} = \mathbf{FTL}(\mathbf{V})$. Moreover, $\mathbf{FTL}(\mathbf{V})$ contains $\mathcal{D}$. By the Variety Theorem, we have $\mathbf{V}_1 \subseteq \mathbf{V}_2$ iff $\mathcal{L}_{\mathbf{V}_1} \subseteq \mathcal{L}_{\mathbf{V}_2}$. Finally, the map is surjective, for if $\mathcal{L}$ is a closed class of regular tree languages containing the definite tree languages, then $\mathcal{L} = \mathcal{L}_{\mathbf{V}}$ for some variety $\mathbf{V}$ of finite tree automata containing $\mathbf{D}$. By Proposition 1.9.1, $\mathbf{V}$ is closed with respect to the ta-cascade product, and $\mathcal{L} = \mathbf{FTL}(\mathbf{V})$. □

1.10 Some Applications

Recall that $\mathcal{L}_{\mathrm{X}}$ denotes the set of languages $\{L_{\mathrm{X}_i},\ i \in [\max(R)]\}$. Recall the definitions of the languages L_{EF}, L_{EG}, L_{EU}. The minimal tree automata of the latter three languages can be described as follows. The minimal automaton of L_{EF} is $\mathbb{E}_F(R)$, which has two elements, $0, 1$, and operations

$$\begin{aligned} \uparrow_n (b_1, \ldots, b_n) &= 1 \\ \downarrow_n (b_1, \ldots, b_n) &= b_1 \vee \ldots \vee b_n, \end{aligned}$$

for all $b_1, \ldots, b_n \in \{0, 1\}$, $n \in R$. The minimal automaton $\mathbb{E}_G(R)$ of L_{EG} also has two elements, $0, 1$. The operations are:

$$\begin{aligned} \uparrow_n (b_1, \ldots, b_n) &= \begin{cases} 1 & \text{if } n = 0 \\ b_1 \vee \ldots \vee b_n & \text{if } n > 0 \end{cases} \\ \downarrow_n (b_1, \ldots, b_n) &= 0, \end{aligned}$$

for all $b_1, \ldots, b_n \in \{0, 1\}$, $n \in R$. Finally, the minimal automaton $\mathbb{E}_U(R)$ of L_{EU} is defined on the set $\{0, 1\}$ by

$$\begin{aligned} \uparrow_n (b_1, \ldots, b_n) &= 1 \\ \vee_n (b_1, \ldots, b_n) &= b_1 \vee \ldots \vee b_n \\ \downarrow_n (b_1, \ldots, b_n) &= 0, \end{aligned}$$

for all $b_1, \ldots, b_n \in \{0,1\}$, $n \in R$. Below we will just write $\mathbb{E}_F, \mathbb{E}_G, \mathbb{E}_U$ for the automata $\mathbb{E}_F(R), \mathbb{E}_G(R), \mathbb{E}_U(R)$ whenever R is understood. We define

$$\begin{aligned}
\mathbf{CTL}(\mathrm{X}, \mathrm{EF}) &= \mathbf{FTL}(\mathcal{L}_X \cup \{L_{\mathrm{EF}}\}) \\
\mathbf{CTL}(\mathrm{X}, \mathrm{EG}) &= \mathbf{FTL}(\mathcal{L}_X \cup \{L_{\mathrm{EG}}\}) \\
\mathbf{CTL}(\mathrm{X}, \mathrm{EF}, \mathrm{EG}) &= \mathbf{FTL}(\mathcal{L}_X \cup \{L_{\mathrm{EF}}, L_{\mathrm{EG}}\}) \\
\mathbf{CTL} &= \mathbf{FTL}(\mathcal{L}_X \cup \{L_{\mathrm{EU}}\}).
\end{aligned}$$

By Example 1.5.7 and Corollary 1.8.3, all assumptions of Corollary 1.9.8 apply to the sets of languages used in the above definitions. Note that the minimal automaton of L_{EF} is a renaming of a reduct of the minimal automaton of L_{EU}, and the minimal automaton of L_{EG} is a renaming of a reduct of the minimal automaton of L_{EU}. Thus, $\mathbf{CTL} = \mathbf{FTL}(\mathcal{L}_X \cup \{L_{\mathrm{EF}}, L_{\mathrm{EG}}, L_{\mathrm{EU}}\})$.

Theorem 1.10.1 *1. For $Y \in \{F, G\}$, a tree language belongs to* $\mathbf{CTL}(\mathrm{X}, \mathrm{EY})$ *iff its minimal tree automaton is in* $\widehat{\{\mathbb{E}_Y, \mathbb{D}_0\}}$.

2. A tree language belongs to $\mathbf{CTL}(\mathrm{X}, \mathrm{EF}, \mathrm{EG})$ *iff its minimal tree automaton is in* $\widehat{\{\mathbb{E}_F, \mathbb{E}_G, \mathbb{D}_0\}}$.

3. A tree language belongs to $\mathbf{CTL}$ *iff its minimal automaton belongs to* $\widehat{\{\mathbb{E}_U\}}$.

Proof. The first two statements follow from Corollary 1.9.6. The third statement follows from Corollary 1.9.6, Theorem 1.7.4, and the fact that $\mathbb{D}_0$ is isomorphic to the reduct of $\mathbb{E}_U$ obtained by forgetting about the $\vee$-operation. □

Remark 1.10.2 The logic defining the class $\mathbf{CTL}(\mathrm{X}, \mathrm{EF}, \mathrm{EG})$ is closely related to the logic introduced in Ben-Ari, Manna and Pnueli in [4].

The minimal automata of L_{AG}, L_{AF} and L_{AU} are respectively isomorphic to the minimal automata of L_{EF}, L_{EG} and L_{EU}. Thus, $\mathbf{CTL}(\mathrm{X}, \mathrm{EF}) = \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{AG}}\})$, $\mathbf{CTL}(\mathrm{X}, \mathrm{EG}) = \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{AF}}\})$, $\mathbf{CTL}(\mathrm{X}, \mathrm{EF}, \mathrm{EG}) = \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{AG}}, L_{\mathrm{AF}}\})$. Moreover, $\mathbf{CTL} = \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{EF}}, L_{\mathrm{EG}}, L_{\mathrm{EU}}, L_{\mathrm{AF}}, L_{\mathrm{AG}}, L_{\mathrm{AU}}\}) = \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{AF}}, L_{\mathrm{AG}}, L_{\mathrm{AU}}\}) = \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{AU}}\})$.

Remark 1.10.3 Suppose that $R = \{0, 1\}$. To each finite alphabet A let us associate the ranked alphabet Σ_A of rank type R with $(\Sigma_A)_1 = A$ and $(\Sigma_A)_0 = \{\#\}$. We may identify each word $u \in A^*$ with a term in T_{Σ_A}. Using this identification, the logic CTL essentially becomes LTL, propositional linear temporal logic, cf. [19]. It follows from the above algebraic characterization of the class $\mathbf{CTL}$ that a language $L \subseteq A^*$ is definable in LTL iff its minimal automaton belongs to the least class of ordinary automata containing the "binary identity-reset automaton", closed under the cascade composition, subautomata, and homomorphic images. This fact was proven by Cohen, Perrin and Pin in [6] (using the wreath product instead of the cascade composition). In fact, our methods and results generalize those of [6]. The binary identity-reset automaton has two states, $0, 1$, and three input letters inducing the two constant functions and the identity function on $\{0, 1\}$, respectively. Using the Krohn-Rhodes Decomposition Theorem [7], it then follows that a language $L \subseteq A^*$ is definable in LTL iff its syntactic monoid is aperiodic. Thus, by the characterization the expressive power of first-order logic on finite words in McNaughton and Papert [16], one derives Kamp's theorem [15] to the effect that first-order logic on finite words is equivalent to propositional linear temporal logic, see also Gabbay, Pnueli, Shelah and Stavi [11].

Recall from Example 1.4.5 the definition of the languages $L_{d,r}$, where $d > 1$ and $0 \leq r < d$. The minimal tree automaton $\mathbb{M}_d$ of $L_{d,r}$ has d elements, $0, \ldots, d-1$, and operations

$$\begin{aligned} \uparrow_n (r_1, \ldots, r_n) &= (r_1 + \ldots + r_n + 1) \mod d \\ \downarrow_n (r_1, \ldots, r_n) &= (r_1 + \ldots + r_n) \mod d, \end{aligned}$$

for all $r_1, \ldots, r_n \in \{0, \ldots, d-1\}$ and $n \in R$. For each d, let $\mathcal{L}_d = \{L_{d,r} : 0 \leq r < d\}$, and let $\mathcal{L}_{\text{mod}} = \bigcup_{d>1} \mathcal{L}_d$. Define

$$\begin{aligned} \mathbf{CTL} + \mathbf{MOD}(d) &= \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{EU}}\} \cup \mathcal{L}_d) \\ \mathbf{CTL} + \mathbf{MOD} &= \mathbf{FTL}(\mathcal{L}_{\mathrm{X}} \cup \{L_{\mathrm{EU}}\} \cup \mathcal{L}_{\text{mod}}). \end{aligned}$$

Using Theorem 1.9.5, we obtain:

Theorem 1.10.4 *1. For every $d > 1$, a tree language belongs to* $\mathbf{CTL} + \mathbf{MOD}(d)$ *iff its minimal tree automaton is in* $\widehat{\{\mathbb{E}_U, \mathbb{M}_d\}}$.

2. A tree language belongs to $\mathbf{CTL} + \mathbf{MOD}$ *iff its minimal tree automaton is in the least closed variety containing* $\mathbb{E}_U$ *and the tree automata* $\mathbb{M}_d$*,* $d > 1$.

1.11 Conclusion

We have associated a modal operator with each language L of finite trees, and a logic FTL($\mathcal{L}$) with each class $\mathcal{L}$ of languages of finite trees. We have shown that several natural modal operators can be captured by suitably chosen languages. Then, for certain classes $\mathcal{L}$ of regular tree languages, we reduced the problem of the characterization of the expressive power of the logic FTL($\mathcal{L}$) to an algebraic problem thus making it possible to study the expressive power of the logics involved by the powerful methods of algebra. This approach has been very fruitful for logics on finite and ω-words. Our general results have many immediate applications and we have presented some.

In order to transform the obtained concrete algebraic characterizations (e.g., that in Theorem 1.10.1) into decision procedures, one has to develop a structure theory of finite algebras. In Part 3, we will use Theorem 1.10.1 to derive an effective characterization of the language class $\mathbf{CTL}(\mathrm{X}, \mathrm{EF})$. This characterization complements the results recently obtained by Bojanczyk and Walukiewicz in [5]. In Part 2, we will extend our general results to finite trees such that the outgoing edges of a vertex are not ordered.

Chapter 2

2.1 Introduction

In Chapter 1, we considered temporal logics on trees as defined in tree automata theory, cf. Gécseg and Steinby [12]. Such trees are *ordered*, since the outgoing edges of each vertex are equipped with a linear order. However, the tree models of the usual temporal logics such as CTL are *unordered*. In Chapter 2, we consider temporal logics on finite unordered trees. In our main result, we provide an algebraic characterization of the expressive power of a wide class of temporal logics on finite unordered trees.

When $a_1, \ldots, a_n$ is a finite family of elements of a set A, then we let $\{\{a_1, \ldots, a_n\}\}$ denote the multiset over A, where each $a \in A$ appears with multiplicity $\sum_{a_i=a} 1$, the total number of occurrences of a in the family.

2.2 Unordered Trees

In this section, all ranked alphabets have a fixed common rank type R. We call a Σ-algebra $\mathbb{A}$ *commutative* if it satisfies all equations

$$\sigma(x_1, \ldots, x_n) \quad = \quad \sigma(x_{\pi(1)}, \ldots, x_{\pi(n)}),$$

for all $\sigma \in \Sigma_n$, $n > 0$, and for all permutations $\pi : [n] \to [n]$. When $0 \in R$, a *commutative Σ-tree automaton* is a Σ-tree automaton which is a commutative algebra. Homomorphisms of commutative Σ-algebras (Σ-tree automata, respectively) are Σ-algebra homomorphisms. Note that for each Σ, the class of all commutative Σ-algebras is a Birkhoff variety, cf. Grätzer [13].

We say that a tree language $L \subseteq T_\Sigma$ is *closed under permutations*, or *permutation closed*, if for each $t = t_0(\sigma(t_1. \ldots, t_n))$ in L, where $t = t_0 \in T_\Sigma(X_1)$ (or $t_0 \in CT_\Sigma$), $\sigma \in \Sigma_n$, $n > 0$, and $t_1, \ldots, t_n \in T_\Sigma$, and for all permutations $\pi : [n] \to [n]$, if $t \in L$ then $t_0(\sigma(t_{\pi(1)}, \ldots, t_{\pi(n)})) \in L$. The following fact is clear.

Proposition 2.2.1 *Suppose that $0 \in R$ and $L \subseteq T_\Sigma$. Then the following are equivalent.*

1. *L is recognizable by a commutative Σ-tree automaton.*
2. *The minimal automaton $\mathbb{A}_L$ is commutative.*
3. *L is closed under permutations.*

Similarly, the following conditions are also equivalent.

1. *L is recognizable by a finite commutative Σ-tree automaton.*
2. *The minimal automaton $\mathbb{A}_L$ is finite and commutative.*
3. *L is regular and closed under permutations.*

If a language is closed under permutations, then it can represented by a set of *unordered trees*, defined below.

Suppose that Σ is a ranked alphabet (of rank type R) and $n \geq 0$. An *n-ary unordered Σ-tree*, or *n-ary unordered tree over* Σ, is either a letter $\sigma \in \Sigma_0$, or a variable x_i in X_n, or an ordered pair $(\sigma, \{\{t_1, \ldots, t_m\}\})$ consisting of a letter $\sigma \in \Sigma_m$, $m > 0$ and a multiset $\{\{t_1, \ldots, t_m\}\}$ of n-ary unordered trees $t_1, \ldots, t_m$ over Σ, denoted $\sigma\{\{t_1, \ldots, t_m\}\}$. We let $U_\Sigma(X_n)$ denote the set of all n-ary unordered Σ-trees. We may turn $U_\Sigma(X_n)$ into a Σ-algebra, $\mathbf{U}_\Sigma(X_n)$, by defining $\sigma_{\mathbf{U}_\Sigma(X_n)}(t_1, \ldots, t_m) = \sigma\{\{t_1, \ldots, t_m\}\}$, for all $\sigma \in \Sigma_m$, $m \geq 0$ and $t_1, \ldots, t_m \in U_\Sigma(X_n)$. When $m = 0$, this tree is σ. When $n = 0$, we just write U_Σ and $\mathbf{U}_\Sigma$. It is clear that the algebra $\mathbf{U}_\Sigma$ is initial in the Birkhoff variety (cf. Grätzer [13]) of all Σ-algebras satisfying the commutativity laws defined above. Similarly, for each $n \geq 0$, $\mathbf{U}_\Sigma(X_n)$ is freely generated by X_n in the Birkhoff variety of all Σ-algebras satisfying the commutativity laws. Since $\mathbf{T}_\Sigma(X_n)$ is freely generated by X_n in the class of all Σ-algebras, there is a unique homomorphism $\mathbf{T}_\Sigma(X_n) \to \mathbf{U}_\Sigma(X_n)$ which is the identity function on X_n that we denote in this section by h_Σ. Note that h_Σ is surjective. (The integer n does not appear in the notation). Thus, if $t \in T_\Sigma$ then $h_\Sigma(t) \in U_\Sigma$.

Each $t \in U_\Sigma(X_n)$ may be represented by a directed graph which is a rooted tree and is equipped with a labeling function consistently mapping the set of vertices to $\Sigma \cup X_n$. But contrary to the case of (ordered) Σ-trees, the outgoing edges of a vertex are not ordered. When $t \in T_\Sigma(X_n)$, $h_\Sigma(t)$ is obtained from t by forgetting about the order on the outgoing edges of the vertices. For unordered trees, the notions of subtree, immediate subtree, successor of a vertex, etc. are defined as for ordered trees. The subtree of a tree $t \in U_\Sigma(X_n)$ rooted at vertex v is denoted t_v.

Suppose that $0 \in R$ and Σ is a ranked alphabet. A subset of U_Σ is called an *unordered tree language*. A *class of unordered tree languages* is any collection

$\mathcal{L}$ of unordered tree languages in U_Σ for all ranked alphabets Σ (of rank type R). When $\mathcal{L}$ is a class of (ordered) tree languages, then for each Σ, the class $h(\mathcal{L})$ contains those unordered tree languages over Σ of the form $h_\Sigma(L)$, where $L \subseteq T_\Sigma$ is in $\mathcal{L}$. Conversely, if $\mathcal{L}$ is a class of unordered tree languages, then for any Σ, $h^{-1}(\mathcal{L})$ contains the languages $h_\Sigma^{-1}(L)$, for all $L \subseteq T_\Sigma$, $L \in \mathcal{L}$. Note that $h(h^{-1}(\mathcal{L})) = \mathcal{L}$.

The following facts are clear.

Proposition 2.2.2 *For each $L \subseteq T_\Sigma$, $h_\Sigma^{-1}(h_\Sigma(L))$ is the least permutation closed tree language containing L. Thus, L is permutation closed iff $L = h_\Sigma^{-1}(h_\Sigma(L))$.*

Proposition 2.2.3 *The permutation closed tree languages in T_Σ form a boolean algebra isomorphic to the boolean algebra of unordered tree languages in U_Σ, an isomorphism is given by the assignment $L \mapsto h_\Sigma(L)$, for all permutation closed $L \subseteq T_\Sigma$. The inverse of this isomorphism is given by the map $L \mapsto h_\Sigma^{-1}(L)$, $L \subseteq U_\Sigma$.*

Proposition 2.2.4 *The lattice of all classes of permutation closed tree languages is isomorphic to the lattice of all classes of unordered tree languages, an isomorphism being the map $\mathcal{L} \mapsto h(\mathcal{L})$, where $\mathcal{L}$ is a class of permutation closed ordered tree languages. The inverse of this isomorphism maps a class $\mathcal{L}$ of unordered tree languages to $h^{-1}(\mathcal{L})$.*

Using Proposition 2.2.3, we have:

Proposition 2.2.5 *For each class $\mathcal{L}$ of unordered tree languages, $\mathcal{L}$ is closed under the boolean operations iff $h^{-1}(\mathcal{L})$ is closed.*

Next we treat inverse literal homomorphisms on unordered tree languages. Suppose that Σ, Δ are ranked alphabets (of rank type R). It is clear how to extend any rank preserving function $k : \Delta \to \Sigma$ to a function $U_\Delta \to U_\Sigma$, called a *literal tree homomorphism.* When $L \subseteq U_\Sigma$ and k is a literal homomorphism $U_\Delta \to U_\Sigma$, we call $k^{-1}(L)$ the inverse image of L under the literal homomorphism k. Recall from Chapter 1 that each rank preserving function $k : \Delta \to \Sigma$ also induces a literal homomorphism $T_\Delta \to T_\Sigma$ of ordered trees, denoted by the same letter.

Proposition 2.2.6 *For each language $L \subseteq U_\Sigma$ and for any rank preserving function $k : \Delta \to \Sigma$, $h_\Delta^{-1}(k^{-1}(L)) = k^{-1}(h_\Sigma^{-1}(L))$.*

Proof. Immediate from the fact that for all $t \in T_\Delta$, $h_\Sigma(k(t)) = k(h_\Delta(t))$. □

Corollary 2.2.7 *For any class $\mathcal{L}$ of unordered tree languages, $\mathcal{L}$ is closed under inverse literal homomorphisms iff so is $h^{-1}(\mathcal{L})$.*

Last, we consider quotients. Suppose that Σ is a ranked alphabet and $t \in U_\Sigma(X_1)$ contains exactly one vertex labeled x_1. Then for any language $L \subseteq U_\Sigma$, we define the *quotient of L with respect to t* to be the language $t^{-1}L = \{s \in U_\Sigma : t(s) \in L\}$. Here, $t(s)$ is the tree obtained from t by substituting s for the vertex of t labeled x_1.

Lemma 2.2.8 *For any $L \subseteq U_\Sigma$, $t \in U_\Sigma(X_1)$ with a single occurrence of x_1, and for any $s \in h_\Sigma^{-1}(t)$, $h_\Sigma^{-1}(t^{-1}L) = s^{-1}(h_\Sigma^{-1}(L))$. Thus, $t^{-1}L = h_\Sigma(s^{-1}(h_\Sigma^{-1}(L)))$.*

Proof. Use the fact that for any tree $t' \in T_\Sigma$, we have $h_\Sigma(s(t')) = h_\Sigma(s)(h_\Sigma(t'))$. Thus,

$$\begin{aligned} t' \in h_\Sigma^{-1}(t^{-1}(L)) &\Leftrightarrow h_\Sigma(t') \in t^{-1}(L) \\ &\Leftrightarrow t(h_\Sigma(t')) \in L \\ &\Leftrightarrow h_\Sigma(s)(h_\Sigma(t')) \in L \\ &\Leftrightarrow h_\Sigma(s(t')) \in L \\ &\Leftrightarrow s(t') \in h_\Sigma^{-1}(L) \\ &\Leftrightarrow t' \in s^{-1}(h_\Sigma^{-1}(L)). \quad \square \end{aligned}$$

Corollary 2.2.9 *For any $L \subseteq U_\Sigma$, $t \in U_\Sigma(X_1)$ with a single occurrence of x_1, and for any $s_1, s_2 \in h_\Sigma^{-1}(t)$, $s_1^{-1}(h_\Sigma^{-1}(L)) = s_2^{-1}(h_\Sigma^{-1}(L))$.*

Corollary 2.2.10 *Suppose that $\mathcal{L}$ is a class of unordered tree languages. Let $\mathcal{K}$ denote the class of all quotients of the languages in $\mathcal{L}$, and $\mathcal{K}'$ the class of all quotients of the languages in $h^{-1}(\mathcal{L})$. Then $h^{-1}(\mathcal{K}) = \mathcal{K}'$, so that $\mathcal{K} = h(\mathcal{K}')$.*

By the above Corollary, if $\mathcal{L}$ and $\mathcal{L}'$ are two classes of unordered tree languages, then $\mathcal{L}'$ contains all quotients of the languages in $\mathcal{L}$ iff $h^{-1}(\mathcal{L}')$ contains all quotients of the languages in $h^{-1}(\mathcal{L})$.

Corollary 2.2.11 *A class $\mathcal{L}$ of unordered tree languages is closed with respect to quotients iff $h^{-1}(\mathcal{L})$ is closed with respect to quotients.*

2.3 Logics

Suppose that a rank type R with $0 \in R$ is fixed. We assume that each ranked alphabet is linearly ordered.

Given a class of unordered tree languages, we define the logic $\mathrm{FTL}(\mathcal{L})$ whose formulas over a ranked alphabet Σ are the letters p_σ, for $\sigma \in \Sigma$, boolean combinations $\neg\varphi$ and $\varphi \vee \psi$, where φ and ψ are already formulas, and the formulas $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, where $L \subseteq U_\Delta$ is in $\mathcal{L}$ and each φ_δ is a formula in $\mathrm{FTL}(\mathcal{L})$.

Given an *unordered* tree t and a formula φ over Σ in $\mathrm{FTL}(\mathcal{L})$, we define the satisfaction relation $t \models \varphi$ in the same way as for ordered trees. In particular, when $\varphi = L(\delta \mapsto \varphi_\delta)$, then $t \models \varphi$ iff the *characteristic tree* $\widehat{t} \in U_\Delta$ determined by t and $(\varphi_\delta)_{\delta\in\Delta}$ is in L. The characteristic tree is defined in the same way as in the ordered case. Let φ be a formula over Σ in $\mathrm{FTL}(\mathcal{L})$. The *language defined by* φ is the set $L_\varphi = \{t \in U_\Sigma : t \models \varphi\}$. We let $\mathbf{FTL}(\mathcal{L})$ denote the class of all unordered tree languages definable by the formulas in $\mathrm{FTL}(\mathcal{L})$.

Proposition 2.3.1 *Suppose that $\mathcal{L}$ is a class of unordered tree languages. Then $h^{-1}(\mathbf{FTL}(\mathcal{L})) = \mathbf{FTL}(h^{-1}(\mathcal{L}))$, so that $\mathbf{FTL}(\mathcal{L}) = h(\mathbf{FTL}(h^{-1}(\mathcal{L})))$.*

Proof. Let φ be a formula over Σ in $\mathrm{FTL}(\mathcal{L})$. We argue by induction on the structure of φ to define a formula $h^{-1}(\varphi)$ in $\mathrm{FTL}(h^{-1}(\mathcal{L}))$ that defines the language $h_\Sigma^{-1}(L_\varphi)$. When $\varphi = p_\sigma$ with $\sigma \in \Sigma_0$, let $h^{-1}(\varphi) = p_\sigma$. Suppose now that $\varphi = \varphi_1 \vee \varphi_1$ or $\varphi = \neg\varphi_1$. In the first case, let $h^{-1}(\varphi) = h^{-1}(\varphi_1) \vee h^{-1}(\varphi_2)$, and in the second, let $h^{-1}(\varphi) = \neg h^{-1}(\varphi_1)$. Last, suppose that $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$. Then we define $h^{-1}(\varphi) = h_\Delta^{-1}(L)(\delta \mapsto h^{-1}(\varphi_\delta))_{\delta\in\Delta}$. The fact that $L_{h^{-1}(\varphi)} = h_\Sigma^{-1}(L_\varphi)$ follows by noting that, by the induction hypothesis, for each vertex v of a tree $t \in T_\Sigma$ labeled in Σ_n, $n \geq 0$, and for each $\delta \in \Delta_n$, $t_v \models h^{-1}(\varphi_\delta)$ iff $h_\Sigma(t_v) \models \varphi_\delta$. Thus, if s denotes the characteristic tree determined by t and the family $(h^{-1}(\varphi_\delta))_{\delta\in\Delta}$, then $h_\Delta(s)$ is the characteristic tree determined by $h_\Sigma(t)$ and $(\varphi_\delta)_{\delta\in\Delta}$. We have $s \in h^{-1}(L)$ iff $h(s) \in L$, so that $t \models h^{-1}(\varphi)$ iff $h_\Sigma(t) \models \varphi$. This proves that $h^{-1}(\mathbf{FTL}(\mathcal{L})) \subseteq \mathbf{FTL}(h^{-1}(\mathcal{L}))$. Note that $h^{-1}(\varphi)$ is obtained by replacing each language L in a subformula $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ of φ by $h_\Delta^{-1}(L)$. It is clear that every formula in $\mathrm{FTL}(h^{-1}(\mathcal{L}))$ arises in this way: Given a formula $\varphi \in \mathrm{FTL}(h^{-1}(\mathcal{L}))$, let $h(\varphi)$ be the formula obtained from φ by replacing each language $h_\Delta^{-1}(L)$ occurring in a subformula of φ by L, then $\varphi = h^{-1}(h(\varphi))$. It follows now that $\mathbf{FTL}(h^{-1}(\mathcal{L})) \subseteq h^{-1}(\mathbf{FTL}(\mathcal{L}))$. □

2.4 Closure Properties

The simple observations of the preceding sections allow us to derive the closure properties of our logics on unordered trees from the corresponding closure properties of ordered trees. Let R be a fixed rank type with $0 \in R$. In this section, all ranked alphabets will be of rank type R.

Theorem 2.4.1 *The operator* **FTL** *is a closure operator on unordered tree language classes.*

Proof. We only prove that for all classes $\mathcal{L}$ of unordered tree languages, it holds that $\mathbf{FTL}(\mathbf{FTL}(\mathcal{L})) = \mathbf{FTL}(\mathcal{L})$. Using Proposition 2.3.1, this equality holds iff $h^{-1}(\mathbf{FTL}(\mathbf{FTL}(\mathcal{L}))) = h^{-1}(\mathbf{FTL}(\mathcal{L}))$ iff $\mathbf{FTL}(\mathbf{FTL}(h^{-1}(\mathcal{L}))) = \mathbf{FTL}(h^{-1}(\mathcal{L}))$. But the last condition holds by Theorem 5.3 in Chapter 1.

□

Theorem 2.4.2 *For each class $\mathcal{L}$ of unordered tree languages, $\mathbf{FTL}(\mathcal{L})$ is closed with respect to the boolean operations and inverse literal homomorphisms.*

Proof. From Propositions 2.3.1, 2.2.5, Corollary 2.2.7, and Theorem 5.1 in Chapter 1. □

Suppose that $\mathcal{L}$ is a class of unordered tree languages and $L \subseteq U_\Delta$. We say that *the modal operator associated with L is expressible in* $\mathrm{FTL}(\mathcal{L})$ if for any family of formulas $(\varphi_\delta)_{\delta\in\Delta}$ in $\mathrm{FTL}(\mathcal{L})$ over some ranked alphabet Σ there exists an $\mathrm{FTL}(\mathcal{L})$-formula equivalent to $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$.

Theorem 2.4.3 *Suppose that $\mathcal{L}$ is a class of unordered tree languages. Then the following conditions are equivalent.*

1. *Each quotient of any language in $\mathcal{L}$ is in $\mathbf{FTL}(\mathcal{L})$.*
2. *$\mathbf{FTL}(\mathcal{L})$ is closed with respect to quotients.*
3. *For each $L \in \mathcal{L}$, $L \subseteq U_\Sigma$, and for each $t \in U_\Sigma(X_1)$ with exactly one occurrence of x_1, the modal operator associated with $t^{-1}L$ is expressible in $\mathrm{FTL}(\mathcal{L})$.*
4. *Each quotient of any language in $h^{-1}(\mathcal{L})$ is in $\mathbf{FTL}(h^{-1}(\mathcal{L}))$.*
5. *$\mathbf{FTL}(h^{-1}(\mathcal{L}))$ is closed with respect to quotients.*
6. *For each $L \in h^{-1}(\mathcal{L})$, $L \subseteq T_\Sigma$, and for each $t \in T_\Sigma(X_1)$ with exactly one occurrence of x_1, the modal operator associated with $t^{-1}L$ is expressible in $\mathrm{FTL}(h^{-1}(\mathcal{L}))$.*

Proof. By Proposition 2.3.1 and Corollary 2.2.10, the first condition is equivalent to the fourth and the second condition is equivalent to the fifth condition. Moreover, by the proof of Proposition 2.3.1, the third condition is equivalent to the sixth. Finally, the last three conditions are equivalent by Theorem 5.4 in Chapter 1. □

Below we will say that *quotients are expressible in* $\mathrm{FTL}(\mathcal{L})$ if for each $L \in \mathcal{L}$, $L \subseteq U_\Sigma$, and for each $t \in U_\Sigma(X_1)$ with exactly one occurrence of x_1, the modal operator associated with $t^{-1}L$ is expressible in $\mathrm{FTL}(\mathcal{L})$.

2.5 Regular Languages

Suppose that R is a fixed rank type containing 0. We will consider ranked alphabets of rank type R.

We say that an unordered tree language $L \subseteq U_\Sigma$ is *recognizable* by a commutative Σ-tree automaton $\mathbb{A}$ if $k^{-1}(k(L)) = L$ holds for the unique homomorphism $k : U_\Sigma \to \mathbb{A}$.

Proposition 2.5.1 *A language $L \subseteq U_\Sigma$ is recognizable by $\mathbb{A}$ iff $h_\Sigma^{-1}(L)$ is recognizable by $\mathbb{A}$.*

It follows that for each $L \subseteq U_\Sigma$ and commutative Σ-tree automaton $\mathbb{A}$, L is recognziable by $\mathbb{A}$ iff $\mathbb{A}$ is a quotient of $\mathbb{A}_{h_\Sigma^{-1}(L)}$, the minimal tree automaton of $h_\Sigma^{-1}(L)$. We call $\mathbb{A}_{h_\Sigma^{-1}(L)}$ the *minimal tree automaton* of L.

Corollary 2.5.2 *A language $L \subseteq U_\Sigma$ is recognizable by a finite commutative tree automaton iff its minimal tree automaton is finite.*

We call such unordered tree languages *regular*, or *recognizable*.

Corollary 2.5.3 *A language $L \subseteq U_\Sigma$ is regular iff $h_\Sigma^{-1}(L)$ is regular.*

Corollary 2.5.4 *The class of regular unordered tree languages is closed under the boolean operations, quotients, and inverse literal homomorphisms.*

Proof. We know that the class of regular ordered tree languages is closed under these operations. The rest follows from Proposition 2.2.5, Corollary 2.2.7 and Corollary 2.2.11. □

Corollary 2.5.5 *The lattice of all classes of permutation closed ordered regular tree languages is isomorphic to the lattice of all classes of unordered regular tree languages, an isomorphism being the map $\mathcal{L} \mapsto h(\mathcal{L})$, for all classes $\mathcal{L}$ of permutation closed ordered regular tree languages. The inverse of this isomorphism maps a class $\mathcal{L}$ of unordered regular tree languages to $h^{-1}(\mathcal{L})$.*

Proof. From Corollary 2.5.3 and Proposition 2.2.4. □

2.6 Varieties

In this section, we again fix a rank type R and assume that all ranked alphabets are of rank type R. Moreover, we assume that $0 \in R$.

By the Variety Theorem of Chapter 1, there is an order isomorphism between varieties of finite tree automata and literal varieties of (regular) ordered tree languages. It is easy to see that the variety **Com** of finite tree automata corresponds to the literal variety $\mathcal{C}$om consisting of all regular tree languages closed under permutations. Thus, under this correspondence, varieties included in **Com** are mapped to literal varieties included in $\mathcal{C}$om, i.e., to literal varieties of permutation closed tree languages. Below we will call a variety included in **Com** a *variety of finite commutative tree automata*, and a literal variety included in $\mathcal{C}$om a *commutative literal (ordered) tree language variety.*

We also define literal varieties of unordered tree languages. We say that class $\mathcal{L}$ of regular unordered tree languages is a *literal variety of unordered tree languages* if it is nonempty, closed under the boolean operations, inverse literal homomorphisms, and quotients. In this section, our aim is to establish a Variety Theorem that relates literal varieties of unordered tree languages to varieties of finite commutative tree automata.

Proposition 2.6.1 *A class $\mathcal{L}$ of unordered regular languages is a literal variety iff $h^{-1}(\mathcal{L})$ is a (commutative) literal variety. Moreover, the lattice of all commutative literal varieties of ordered tree languages is isomorphic to the lattice of all literal varieties of unordered tree languages, an isomorphism being the map $\mathcal{L} \mapsto h(\mathcal{L})$, for all commutative literal varieties $\mathcal{L}$ of ordered tree languages. The inverse of this isomorphism maps a literal variety $\mathcal{L}$ of unordered tree languages to $h^{-1}(\mathcal{L})$.*

Proof. This follows from Corollary 2.5.5, Proposition 2.2.5, Corollary 2.2.7 and Corollary 2.2.11. □

Theorem 2.6.2 *For each variety $\mathbf{V}$ of finite commutative tree automata, let $\mathcal{V}_u$ denote the class of all unordered tree languages recognizable by the members of $\mathbf{V}$ (or equivalently, whose minimal automata are in $\mathbf{V}$). Then the assignment $\mathbf{V} \mapsto \mathcal{V}_u$ defines an order isomorphism between varieties of finite commutative tree automata and literal varieties of unordered tree languages.*

Proof. For every commutative variety $\mathbf{V}$ of finite tree automata, let $\mathcal{V}$ denote the commutative literal variety of ordered tree languages corresponding to $\mathbf{V}$. By Theorem 9.10 in Chapter 1, the assignment $\mathbf{V} \mapsto \mathcal{V}$ is an order isomorphism from the lattice of varieties of finite commutative tree automata onto the lattice of commutative literal varieties of tree languages. To complete the proof, note that by Proposition 2.6.1, the lattice of commutative literal varieties of ordered tree languages is isomorphic to the lattice of literal varieties of unordered tree languages, and that an isomorphism is given by the mapping $\mathcal{V} \mapsto h(\mathcal{V})$, for all commutative literal varieties $\mathcal{V}$ of ordered tree languages. The composite of the two isomorphisms is the required isomorphism. □

Below we will write $\mathcal{L}^u_{\mathbf{V}}$ for the literal variety $\mathcal{V}_u$ corresponding to $\mathbf{V}$.

2.7 Commutative Cascade Product

In this section, let R denote a fixed rank type which may or may not contain 0. In this section we will consider both algebras and tree automata (of rank type R). Whenever we mention tree automata, we assume implicitely that $0 \in R$.

The variety **Com** of finite commutative tree automata is not closed under the cascade product. However, it is closed under the *commutative cascade product* defined below.

Suppose that $\mathbb{A}$ is a Σ-algebra and $\mathbb{B}$ is a Δ-algebra, where Σ and Δ are of rank type R. Moreover, suppose that for each $n \in R$, α_n is a mapping $nA \times \Sigma_n \to \Delta_n$, where nA denotes the set of all multisets $\{\{a_1, \ldots, a_n\}\}$ of elements of A. Then the commutative cascade product $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ determined by the family $\alpha = (\alpha_n)_{n \in R}$ is the following Σ-algebra. The carrier of $\mathbb{C}$ is the set $C = A \times B$. Moreover, for each $\sigma \in \Sigma_n$, $n \geq 0$, and for each $(a_i, b_i) \in C$, $i \in [n]$,

$$\sigma_{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) \quad = \quad (\sigma_{\mathbb{A}}(a_1, \ldots, a_n), \delta_{\mathbb{B}}(b_1, \ldots, b_n)),$$

where $\delta = \alpha_n(\{\{a_1, \ldots, a_n\}\}, \sigma)$. Note that each commutative cascade product may be ragarded as a cascade product. Conversely, a cascade product $\mathbb{A} \times_\alpha \mathbb{B}$ of a Σ-algebra $\mathbb{A}$ and a Δ-algebra $\mathbb{B}$ such that for each $n \in R$ the function $\alpha_n(a_1, \ldots, a_n, \sigma)$ depends only on Σ and the multiset $\{\{a_1, \ldots, a_n\}\}$ may be regarded as a commutative cascade product of $\mathbb{A}$ and $\mathbb{B}$. The commutative cascade product can be generalized to several factors. When $\mathbb{A}_i$ is a Σ_i-algebra for each $i \in [n]$, $n \geq 1$, and for each $j \in [n-1]$, α_j is a family of functions

$$m(A_1 \times \ldots \times A_j) \times (\Sigma_1)_m \quad \to \quad (\Sigma_{j+1})_m, \quad m \in R,$$

then the commutative cascade product of the $\mathbb{A}_i$ determined by the functions α_j is denoted $\mathbb{A}_1 \times_{\alpha_1} \ldots \times_{\alpha_{n-1}} \mathbb{A}_n$. Any such commutative cascade product may also be specified by functions $(A_1 \times \ldots \times A_j)^m \times (\Sigma_1)_m \to (\Sigma_{j+1})_m$, $m \in R$, subject to certain conditions.

When $0 \in R$, a *commutative (ta-)cascade product* of tree automata $\mathbb{A}$ and $\mathbb{B}$ is the least subalgebra of a commutative cascade product of $\mathbb{A}$ and $\mathbb{B}$, and similarly for commutative ta-cascade products $\mathbb{A}_1 \times_{\alpha_1} \ldots \times_{\alpha_{n-1}} \mathbb{A}_n$, where each $\mathbb{A}_i$ is a tree automaton.

Proposition 2.7.1 *Any commutative cascade product of commutative algebras is commutative.*

Proof. Suppose that $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ is a commutative cascade product of the commutative Σ-algebra $\mathbb{A}$ and the commutative Δ-algebra $\mathbb{B}$. Then for all

$(a_i, b_i) \in C$, $i \in [n]$, and for all permutations $\pi : [n] \to [n]$,

$$\begin{aligned}\sigma_{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) &= (\sigma_{\mathbb{A}}(a_1, \ldots, a_n), \delta_{\mathbb{B}}(b_1, \ldots, b_n)) \\ &= (\sigma_{\mathbb{A}}(a_{\pi(1)}, \ldots, a_{\pi(n)}), \delta_{\mathbb{B}}(b_{\pi(1)}, \ldots, b_{\pi(n)})) \\ &= \sigma_{\mathbb{C}}((a_{\pi(1)}, b_{\pi(1)}), \ldots, (a_{\pi(n)}, b_{\pi(n)})),\end{aligned}$$

where $\delta = \alpha_n(\{\{a_1, \ldots, a_n\}\}, \sigma) = \alpha_n(\{\{a_{\pi(1)}, \ldots, a_{\pi(n)}\}\}, \sigma)$. □

Thus, any commutative ta-cascade product of commutative tree automata is a commutative tree automaton. We call a nonempty class of finite commutative algebras a *commutative closed variety* if it is closed under the commutative cascade product, renaming, subalgebras and quotients. (Note that any renaming of a commutative algebra is commutative.) Similarly, a nonempty class of finite tree automata is a *commutative closed variety of finite tree automata* if it is closed under the commutative cascade product, renaming and quotients. Note that any commutative closed variety of finite algebras or finite tree automata is closed under the direct product and is thus a variety. By the above Proposition, **Com** is a commutative closed variety of finite tree automata, and in fact the largest one. Similarly, the class of all finite commutative algebras is the largest closed variety of finite algebras.

Remark 2.7.2 Note that a commutative closed variety of finite algebras or finite tree automata may not be a closed variety as defined in Chapter 1, since it may not necessarily closed under the cascade product.

Remark 2.7.3 Suppose that **K** is a class of finite commutative algebras. Then the least commutative closed variety of finite algebras containing **K** is the class of all quotients of subalgebras of commutative cascade products $\mathbb{A}_1 \times_{\alpha_1} \ldots \times_{\alpha_n} \mathbb{A}_n$ of algebras in **K**. A similar fact is true for commutative closed varieties of finite tree automata.

We want to show that any commutative closed variety of finite algebras is the intersection of a closed variety with the class of all finite commutative algebras, and similarly for finite tree automata. In our argument, we will make use of Propositions 2.7.4 and 2.7.5.

Proposition 2.7.4 *Suppose that $\mathbb{A}, \mathbb{B}$ are Σ-algebras such that $\mathbb{B}$ is commutative. Suppose that for each n, A^n is equipped with a linear order. Define the Σ-algebra $\mathbb{A}'$ on the set A as follows: For each $\sigma \in \Sigma_n$ and $a_1, \ldots, a_n \in A$, $n \geq 0$, $\sigma_{\mathbb{A}'}(a_1, \ldots, a_n) = \sigma_{\mathbb{A}}(a_{\pi(1)}, \ldots, a_{\pi(n)})$, where π is that permutation $[n] \to [n]$ for which $(a_{\pi(1)}, \ldots, a_{\pi(n)})$ is the least in the linear order among the vectors $(b_1, \ldots, b_n) \in A^n$ with $\{\{b_1, \ldots, b_n\}\} = \{\{a_1, \ldots, a_n\}\}$. If $\mathbb{B}$ is a quotient of a subalgebra of $\mathbb{A}$ then it is also a quotient of a subalgebra of $\mathbb{A}'$.*

Proof. Suppose that $\mathbb{C}$ is a subalgebra of $\mathbb{A}$ and f is a homomorphism $\mathbb{C} \to \mathbb{B}$. Then the carrier C of $\mathbb{C}$ determines a subalgebra $\mathbb{C}'$ of $\mathbb{A}'$. Moreover, f is a homomorphism $\mathbb{C}' \to \mathbb{B}$. Indeed, suppose that $\sigma \in \Sigma_n$ and $a_1, \ldots, a_n \in C$, $n \geq 0$. Let π denote the permutation $[n] \to [n]$ described above. Then we have

$$\sigma_{\mathbb{C}'}(a_1, \ldots, a_n) \quad = \quad \sigma_{\mathbb{C}}(a_{\pi(1)}, \ldots, a_{\pi(n)}) \in C,$$

and

$$\begin{aligned} f(\sigma_{\mathbb{C}'}(a_1, \ldots, a_n)) &= f(\sigma_{\mathbb{C}}(a_{\pi(1)}, \ldots, a_{\pi(n)})) \\ &= \sigma_{\mathbb{B}}(f(a_{\pi(1)}), \ldots, f(a_{\pi(n)})) \\ &= \sigma_{\mathbb{B}}(f(a_1), \ldots, f(a_n)), \end{aligned}$$

where the last line follows from the commutativity of $\mathbb{B}$. □

Proposition 2.7.5 *Suppose that $\mathbb{A}_i$ is a Σ_i-algebra for $i \in [n]$, and consider a cascade product $\mathbb{A} = \mathbb{A}_1 \times_{\alpha_1} \times \ldots \times_{\alpha_{n-1}} \mathbb{A}_n$. If a commutative Σ_1-algebra $\mathbb{B}$ is a homomorphic image of a subalgebra of $\mathbb{A}$, then there exists a commutative cascade product $\mathbb{A}' = \mathbb{A}_1 \times_{\alpha'_1} \times \ldots \times_{\alpha'_{n-1}} \mathbb{A}_n$ such that $\mathbb{B}$ is a homomorphic image of a subalgebra of $\mathbb{A}'$.*

Proof. Let us equip each A_i with a linear order $\leq_i$ and let us order each $A_1 \times \ldots \times A_j$, $j \in [n]$ lexicographically by $(a_1, \ldots, a_j) \leq (b_1, \ldots, b_j)$ iff $(a_1, \ldots, a_j) = (b_1, \ldots, b_j)$ or there exists some $i \in [j]$ such that $a_1 = b_1, \ldots, a_{i-1} = b_{i-1}$ and $a_i < b_i$. We define the commutative cascade product $\mathbb{A}' = \mathbb{A}_1 \times_{\alpha'_1} \times \ldots \times_{\alpha'_{n-1}} \mathbb{A}_n$ by specifying the functions α'_{jm}, $j \in [n-1], m \in R$ as functions

$$(A_1 \times \ldots \times A_j)^m \times (\Sigma_1)_m \quad \to \quad (\Sigma_{j+1})_m.$$

Given $(a_{11}, \ldots, a_{j1}), \ldots, (a_{1m}, \ldots, a_{jm})$ in $A_1 \times \ldots \times A_j$ and $\sigma \in (\Sigma_1)_m$, define

$$\begin{aligned} &\alpha'_{jm}((a_{11}, \ldots, a_{j1}), \ldots, (a_{1m}, \ldots, a_{jm}), \sigma) = \\ &= \alpha_{jm}((a_{1\pi(1)}, \ldots, a_{j\pi(1)}), \ldots, (a_{1\pi(m)}, \ldots, a_{j\pi(m)}), \sigma), \end{aligned}$$

where the permutation $\pi : [m] \to [m]$ satisfies $(a_{1\pi(1)}, \ldots, a_{j\pi(1)}) \leq \ldots \leq (a_{1\pi(m)}, \ldots, a_{j\pi(m)})$. The fact that $\mathbb{A}'$ also contains a subalgebra that can be mapped homomorphically onto $\mathbb{B}$ follows from Proposition 2.7.4. To see this, let us order each $(A_1 \times \ldots \times A_n)^m$, $m \in R$ by lexicographically extending the order on $A_1 \times \ldots \times A_n$. Then, with respect to this order, for each $((a_{11}, \ldots, a_{n1}), \ldots, (a_{1m}, \ldots, a_{nm}))$ in $(A_1 \times \ldots \times A_n)^m$ and $\sigma \in (\Sigma_1)_m$, $\sigma_{\mathbb{A}}((a_{11}, \ldots, a_{n1}), \ldots, (a_{1m}, \ldots, a_{nm}))$ and $\sigma_{\mathbb{A}'}((a_{11}, \ldots, a_{n1}), \ldots, (a_{1m}, \ldots, a_{nm}))$ are related exactly as in Proposition 2.7.4. The proof is completed by applying Proposition 2.7.4. □

Theorem 2.7.6 *Suppose that $\mathbf{K}$ is a class of commutative finite algebras. Then the least commutative closed variety containing $\mathbf{K}$ is the class of all commutative algebras in the least closed variety containing $\mathbf{K}$.*

Proof. Let $\mathbf{V}$ denote the least commutative closed variety containing $\mathbf{K}$, and let $\mathbf{W}$ denote the least closed variety containing $\mathbf{K}$. Since $\mathbf{V} \subseteq \mathbf{W}$ and $\mathbf{V}$ is included in the variety of all finite commutative algebras, $\mathbf{V}$ is included in the intersection of $\mathbf{W}$ with the variety of all finite commutative algebras. To prove the reverse inclusion, assume that $\mathbb{A}$ is commutative and belongs to $\mathbf{W}$. Then $\mathbb{A}$ is a quotient of a subalgebra of a cascade product of some algebras $\mathbb{A}_i$ in $\mathbf{V}$, $i \in [n]$, $n \geq 1$. By the previous proposition, $\mathbb{A}$ is a quotient of a subalgebra of a commutative cascade product of the $\mathbb{A}_i$. Since $\mathbf{V}$ is closed under the commutative cascade product, it follows that $\mathbb{A} \in \mathbf{V}$. □

Corollary 2.7.7 *Suppose that* $0 \in R$ *and* $\mathbf{K}$ *is a class of commutative finite tree automata. Then the least commutative closed variety of finite tree automata containing* $\mathbf{K}$ *is the class of all commutative tree automata in the least closed variety of finite tree automata containing* $\mathbf{K}$.

Corollary 2.7.8 *A class* $\mathbf{K}$ *of finite algebras is a commutative closed variety iff there exists a closed variety* $\mathbf{W}$ *of finite algebras such that* $\mathbf{V}$ *is the class of all commutative algebras in* $\mathbf{W}$. *Similarly, when* $0 \in R$, *then a class* $\mathbf{K}$ *of finite tree automata is a commutative closed variety iff there exists a closed variety* $\mathbf{W}$ *of finite tree automata such that* $\mathbf{V} = \mathbf{W} \cap \mathbf{Com}$.

2.8 Commutative Definite Languages

In this section we assume that R is a fixed rank type with $0 \in R$.

Let $\mathcal{L}$ denote a class of unordered tree languages. We say that *the next modalities are expressible in* FTL($\mathcal{L}$) if for each $i \in [\max(R)]$ and formula $\varphi \in$ FTL($\mathcal{L}$) over any ranked alphabet Σ, there exists a formula $X_{=i}\varphi$ such that for any tree $t \in U_\Sigma$, $t \models X_{=i}\varphi$ iff t has exactly i immediate subtrees satisfying φ. (Thus, the root of t is labeled in Σ_n for some $n \geq i$.)

Proposition 2.8.1 *Suppose that* $\mathcal{L}$ *is a class of unordered tree languages. The following conditions are equivalent.*

1. *The next modalities are expressible in* FTL($\mathcal{L}$).
2. *For each* $1 \leq i \leq \max(R)$ *and formula* $\varphi \in$ FTL($\mathcal{L}$) *over any ranked alphabet* Σ, *there exists a formula* $X_{<i}\varphi$ *such that for any tree* $t \in U_\Sigma$, $t \models X_{<i}\varphi$ *iff* t *has* $< i$ *immediate subtrees satisfying* φ.
3. *For each* $0 \leq i \leq \max(R) - 1$ *and formula* $\varphi \in$ FTL($\mathcal{L}$) *over any ranked alphabet* Σ, *there exists a formula* $X_{\leq i}\varphi$ *such that for any tree* $t \in U_\Sigma$, $t \models X_{\leq i}\varphi$ *iff* t *has* $\leq i$ *immediate subtrees satisfying* φ.

Proof. Assume first that the next modalities are expressible in FTL($\mathcal{L}$). Then the formula

$$X_{=0}\varphi \quad = \quad \neg(X_{=1}\varphi \vee \ldots \vee X_{=\max(R)}\varphi)$$

expresses that a tree has no immediate subtree satisfying φ. And for every $1 \leq i \leq \max(R)$, $X_{<i}\varphi$ can be expressed as $\bigvee_{j=0}^{i-1} X_{=j}\varphi$. This proves that the first condition implies the second. The fact that the second condition implies the third follows by noting that for each $0 \leq i \leq \max(R) - 1$ and φ, $X_{\leq i}\varphi$ is equivalent to $X_{<i+1}\varphi$. Last, assume that the third condition holds. Then for every $1 \leq i \leq \max(R)$, $X_{=i}\varphi$ can be expressed as $X_{\leq i}\varphi \wedge \neg(X_{\leq i-1}\varphi)$, for all $i < \max(R)$, and $\neg \mathrm{X}_{\leq i-1}\varphi$, if $i = \max(R)$. □

Proposition 2.8.2 *Suppose that $\mathcal{L}$ is a class of unordered tree languages such that the next modalities are expressible in* FTL($\mathcal{L}$). *Then for each $n \in R$, $n > 0$, and formulas $\varphi_1, \ldots, \varphi_n$ in* FTL($\mathcal{L}$) *over some ranked alphabet Σ there exists a formula* $\mathrm{X}\{\{\varphi_1, \ldots, \varphi_n\}\}$ *in* FTL($\mathcal{L}$) *over Σ, depending only on the multiset $\{\{\varphi_1, \ldots, \varphi_n\}\}$, such that for all trees $t \in U_\Sigma$, $t \models \mathrm{X}\{\{\varphi_1, \ldots, \varphi_n\}\}$ iff the root of t is labeled in Σ_n and its n immediate subtrees satisfy the formulas $\varphi_1, \ldots, \varphi_n$ in some order.*

Proof. That the root of a tree is labeled in Σ_n is expressible by the formula $\bigvee_{\sigma \in \Sigma_n} p_\sigma = \mathtt{tt}_n$. Since the boolean connectives are available in the language, we may as well assume that any two of the φ_i are either (syntactically) equal or inconsistent: no tree satisfies both of them. So let us assume that the sequence $\varphi_1, \ldots, \varphi_n$ contains m_1 copies of $\psi_1, \ldots, m_k$ copies of ψ_k, where $m_1, \ldots, m_k > 0$, $m_1 + \ldots + m + k = n$, and any two of the formulas ψ_j are inconsistent. Then the property formulated in the Proposition can be expressed as

$$\mathtt{tt}_n \wedge \bigwedge_{j \in [k]} X_{=m_j}\psi_j. \quad \square$$

Call a language $L \subseteq U_\Sigma$ *k-definite*, for some integer $k \geq 0$, if for all unordered trees s, t in U_Σ such that the cut off of s at depth k agrees with the cut off of t at depth k, it holds that $s \in L$ iff $t \in L$. Moreover, call $L \subseteq U_\Sigma$ *definite* if it is k-definite for some $k \geq 0$. Let $\mathcal{UD}$ denote the class of all definite unordered tree languages (of rank type R), and for each $k \geq 0$, let $\mathcal{UD}_k$ denote the class of all k-definite unordered tree languages. Thus, $\mathcal{UD} = \cup_{k \geq 0} \mathcal{UD}_k$.

For example, the following languages $L_{\mathrm{X}_{=i}} \subseteq U_{\mathrm{Bool}}$, $i \in [\max(R)]$ are 2-definite: A tree $t \in U_{\mathrm{Bool}}$ belongs to $L_{\mathrm{X}_{=i}}$ iff its root is labeled by in Bool_n for some $n \geq i$ and has exactly i immediate successors labeled in the set $\{\uparrow_m : m \in R\}$. Let $\mathcal{L}_{\mathrm{UX}}$ denote the collection of all these languages $L_{\mathrm{X}_{=i}}$.

Proposition 2.8.3 *The following conditions are equivalent for a class $\mathcal{L}$ of unordered tree languages.*

1. *The next modalities are expressible in* $\mathrm{FTL}(\mathcal{L})$.
2. $\mathcal{L}_{\mathrm{UX}} \subseteq \mathbf{FTL}(\mathcal{L})$.
3. $\mathcal{UD}_2 \subseteq \mathbf{FTL}(\mathcal{L})$.
4. $\mathcal{UD} \subseteq \mathbf{FTL}(\mathcal{L})$.

Proof. The fourth condition clearly implies the third which in turn implies the second. The second condition is equivalent to the first, since a tree satisfies a formula $X_{=i}\varphi$ iff it satisfies $L_{X_{=i}}(\psi_\delta)_{\delta\in\mathrm{Bool}}$, where $\psi_\delta = \varphi$ if $\delta \in \{\uparrow_m : m \in R\}$ and $\psi_\delta = \neg\varphi$ otherwise. Moreover, for any family $(\varphi_\delta)_{\delta\in\Delta}$, $L_{X_{=i}}(\varphi_\delta)_{\delta\in\mathrm{Bool}}$ is expressible as $X_{=i}\psi$, where $\psi = \bigwedge_{n\in R}(p_{\uparrow_n} \to \varphi_{\uparrow_n})$.

Thus, it remains to show that the first condition implies the fourth. So suppose that the next modalities are expressible in $\mathrm{FTL}(\mathcal{L})$. We show by induction on k that $\mathcal{UD}_k \subseteq \mathbf{FTL}(\mathcal{L})$. When $k = 0$ this is clear, since for each Σ, $\mathcal{UD}_0$ contains two languages over Σ: $\emptyset$ and U_Σ. Suppose that $k > 0$. Then any language in $\mathcal{UD}_k$ is a finite union of languages $\sigma\{\{L_1, \ldots, L_n\}\}$ consisting of all trees whose root is labeled σ, for some $\sigma \in \Sigma_n$, $n \geq 0$, and whose immediate subtrees are, in some order, in the $(k-1)$-definite languages $L_1, \ldots, L_n$. By induction, each L_i is definable by some φ_i in $\mathrm{FTL}(\mathcal{L})$. Thus, $\sigma(L_1, \ldots, L_n)$ is definable by the formula $p_\sigma \wedge X\{\{\varphi_1 \ldots . \varphi_n\}\}$. The result now follows from Proposition 2.8.2. □

Recall from Chapter 1 that $\mathbf{D}$ denotes the closed variety of all finite definite tree automata, and for each $k \geq 0$, $\mathbf{D}_k$ is the variety of all finite k-definite tree automata. The corresponding literal varieties of ordered tree languages are respectively $\mathcal{D}$ and $\mathcal{D}_k$, $k \geq 0$. Recall that **Com** denotes the variety of finite commutative tree automata and $\mathcal{C}\mathrm{om}$ denotes the commutative literal variety of all permutation closed tree languages. Let us denote $\mathbf{CD} = \mathbf{Com} \cap \mathbf{D}$, $\mathcal{CD} = \mathcal{C}\mathrm{om} \cap \mathcal{D}$, and let $\mathbf{CD}_k = \mathbf{Com} \cap \mathbf{D}_k$, $\mathcal{CD}_k = \mathcal{C}\mathrm{om} \cap \mathcal{D}_k$ for all $k \geq 0$. It is clear that $\mathbf{CD}$, and each $\mathbf{CD}_k$, is a variety of finite commutative tree automata, and $\mathcal{CD}$ and $\mathcal{CD}_k$ are the corresponding commutative literal varieties of ordered tree languages. The following fact is clear.

Proposition 2.8.4 $h^{-1}(\mathcal{UD}) = \mathcal{CD}$ *and* $h^{-1}(\mathcal{UD}_k) = \mathcal{CD}_k$, *for all* $k \geq 0$.

Corollary 2.8.5 $\mathcal{UD}$ *is a literal variety of unordered tree languages, the literal variety corresponding to* $\mathbf{D}$. *Similarly, for each* $k \geq 0$, $\mathcal{UD}_k$ *is the literal variety of unordered tree languages corresponding to* $\mathbf{CD}_k$.

Corollary 2.8.6 *For each class* $\mathcal{L}$ *of unordered tree languages, the next modalities are expressible in* $\mathrm{FTL}(\mathcal{L})$ *iff the next modalities are expressible in* $\mathrm{FTL}(h^{-1}(\mathcal{L}))$.

Proof. By Proposition 2.8.3, the next modalities are expressible in $\mathrm{FTL}(\mathcal{L})$ iff $\mathcal{UD} \subseteq \mathbf{FTL}(\mathcal{L})$. By Proposition 2.8.4, this is further equivalent to the condition that $\mathcal{D} \subseteq \mathbf{FTL}(h^{-1}(\mathcal{L}))$. Last, by Corollary 8.3 in Chapter 1, this holds iff the next modalities are expressible in $\mathrm{FTL}(h^{-1}(\mathcal{L}))$. □

As in Chapter 1, let $\mathbb{D}_0$ denote the two element Bool-algebra on the set $\{0,1\}$ with the operations

$$\begin{aligned} \uparrow_n (a_1, \ldots, a_n) &= 1 \\ \downarrow_n (a_1, \ldots, a_n) &= 0, \quad n \in R. \end{aligned}$$

As an application of Corollary 2.7.7 we now show:

Proposition 2.8.7 **CD** *is a commutative closed variety of finite tree automata and is generated by* $\mathbb{D}_0$.

Proof. It was shown in [8] that **D** is the least closed variety of finite tree automata containing $\mathbb{D}_0$. Since $\mathbb{D}_0$ is commutative, it follows from Corollary 2.7.7 that **CD** is the commutative closed variety of finite tree automata generated by $\mathbb{D}_0$. □

2.9 Expressiveness

Our main results provide an algebraic charactrization of the expressive power of the logics $\mathrm{FTL}(\mathcal{L})$, where $\mathcal{L}$ is a class of regular unordered tree languages satisfying certain natural conditions.

Theorem 2.9.1 *Suppose that $\mathcal{L}$ is a class of regular unordered tree languages such that quotients and the next modalities are expressible in* $\mathrm{FTL}(\mathcal{L})$. *Then an unordered tree language $L \subseteq U_\Sigma$ is in $\mathbf{FTL}(\mathcal{L})$ iff its minimal automaton $\mathbb{A}_L$ belongs to the least commutative closed variety containing $\mathbb{D}_0$ and the minimal automata of the languages in $\mathcal{L}$.*

Proof. We know from Proposition 2.3.1 that $L \in \mathbf{FTL}(\mathcal{L})$ iff $h_\Sigma^{-1}(L) \in \mathbf{FTL}(h^{-1}(\mathcal{L}))$. By Corollary 2.8.6, since the next modalities are expressible in $\mathrm{FTL}(\mathcal{L})$, they are expressible in $\mathrm{FTL}(h^{-1}(\mathcal{L}))$. Moreover, since quotients are expressible in $\mathbf{FTL}(\mathcal{L})$, they are expressible in $\mathbf{FTL}(h^{-1}(\mathcal{L}))$. Thus, Corollary 9.6 in Chapter 1, $L \in \mathbf{FTL}(\mathcal{L})$ iff $\mathbb{A}_{h_\Sigma^{-1}(L)}$ belongs to the least closed variety of finite tree automata containing $\mathbb{D}_0$ and the minimal automata of the ordered tree languages in $h_\Sigma^{-1}(\mathcal{L})$. Note that for each L, the minimal automaton $\mathbb{A}_{h_\Sigma^{-1}(L)}$ of $h_\Sigma^{-1}(L)$ is just the minimal automaton $\mathbb{A}_L$ of L. Moreover, the minimal automaton of each $L \in \mathcal{L}$ is commutative as is the automaton $\mathbb{D}_0$. Thus, by Theorem 2.7.6, the class of commutative tree automata in the least closed variety containing $\mathbb{D}_0$ and the automata $\mathbb{A}_L$, $L \in \mathcal{L}$ is just the least commutative closed variety of finite tree automata containing these tree automata. In conclusion, $L \in \mathbf{FTL}(\mathcal{L})$

iff its minimal automaton $\mathbb{A}_L$ belongs to the least commutative closed variety containing $\mathbb{D}_0$ and the minimal automata of the languages in $\mathcal{L}$. □

Suppose that $\mathbf{K}$ is a class of finite commutative automata. Then we let $\mathcal{L}^u_{\mathbf{K}}$ denote the class of all unordered tree languages recognizable by the members of $\mathbf{K}$. We define $\mathrm{FTL}^u(\mathbf{K})$ to be the logic $\mathrm{FTL}(\mathcal{L}^u_{\mathbf{K}})$ and $\mathbf{FTL}^u(\mathbf{K}) = \mathbf{FTL}(\mathcal{L}_{\mathbf{K}})$.

Corollary 2.9.2 *Suppose that* $\mathbf{K}$ *is a class of finite commutative tree automata such that the next modalities are expressible in* $\mathbf{FTL}^u(\mathbf{K})$. *Then an unordered tree language* $L \subseteq U_\Sigma$ *is in* $\mathbf{FTL}^u(\mathbf{K})$ *iff its minimal automaton* $\mathbb{A}_L$ *belongs to the least commutative closed variety containing* $\mathbb{D}_0$ *and* $\mathbf{K}$.

Proof. It is clear that $\mathcal{L}_{\mathbf{K}}$ is closed under quotients and thus, by Theorem 2.4.3, quotients are expressible in $\mathrm{FTL}^u(\mathbf{K})$. The rest follows from Theorem 2.9.1. □

We call a class of regular unordered tree languages $\mathcal{L}$ *closed* if $\mathcal{L}$ is closed under quotients and if $\mathbf{FTL}(\mathcal{L}) \subseteq \mathcal{L}$.

Theorem 2.9.3 *Let* $\mathbf{V}$ *a commutative closed variety of finite tree automata containing* $\mathbf{CD}$. *Then* $\mathbf{FTL}^u(\mathbf{V}) = \mathcal{L}_{\mathbf{V}}$. *Moreover, the assignment* $\mathbf{V} \mapsto \mathbf{FTL}^u(\mathbf{V})$ *defines an order isomorphism between commutative closed varieties of finite tree automata containing* $\mathbf{CD}$ *and closed classes of unordered tree languages containing the* $\mathcal{CD}$.

Proof. Suppose that $\mathbf{V}$ is a commutative closed variety containing $\mathbf{CD}$. By Corollary 2.9.2 and Proposition 2.8.3, $\mathbf{FTL}^u(\mathbf{V})$ is the class of all unordered tree languages whose minimal automaton belongs to $\mathbf{V}$, i.e., $\mathbf{FTL}^u(\mathbf{V}) = \mathcal{L}^u_{\mathbf{V}}$. It is clear that $\mathcal{L}^u_{\mathbf{V}}$ contains $\mathcal{CD}$. The rest follows from the Variety Theorem, Theorem 2.6.2. □

2.10 Applications

In this section, we again assume that R is a rank type containing 0. The set of languages $\mathcal{L}_{\mathrm{UX}}$ was defined in Section 2.8. Recall the definitions of the languages L_{EF}, L_{EG}, L_{EU} from Chapter 1, and let $L^u_{\mathrm{EF}} = h_{\mathrm{Bool}}(L_{\mathrm{EF}})$, $L^u_{\mathrm{EG}} = h_{\mathrm{Bool}}(L_{\mathrm{EG}})$, $L^u_{\mathrm{EU}} = h_{\mathrm{Bool}}(L_{\mathrm{EU}})$. Since the languages L_{EF}, L_{EG}, L_{EU} are all permutation closed, the minimal automata for these languages are respectively the minimal automata for L^u_{EF}, L^u_{EG}, L^u_{EU}. Recall from Chapter 1 that these automata are denoted by $\mathbb{E}_F$, $\mathbb{E}_G$, and $\mathbb{E}_U$. Define

$$\begin{aligned}
\mathbf{CTL}^u(\mathrm{X},\mathrm{EF}) &= \mathbf{FTL}(\mathcal{L}_{\mathrm{UX}} \cup \{L^u_{\mathrm{EF}}\})\\
\mathbf{CTL}^u(\mathrm{X},\mathrm{EG}) &= \mathbf{FTL}(\mathcal{L}_{\mathrm{UX}} \cup \{L^u_{\mathrm{EG}}\})\\
\mathbf{CTL}^u(\mathrm{X},\mathrm{EF},\mathrm{EG}) &= \mathbf{FTL}(\mathcal{L}_{\mathrm{UX}} \cup \{L^u_{\mathrm{EF}}, L^u_{\mathrm{EG}}\})\\
\mathbf{CTL}^u &= \mathbf{FTL}(\mathcal{L}_{\mathrm{UX}} \cup \{L^u_{\mathrm{EU}}\}).
\end{aligned}$$

From Corollary 2.9.2, we immediately obtain:

Theorem 2.10.1 *1. For $Y \in \{F, G\}$, a tree language belongs to* $\mathbf{CTL}^u(\mathrm{X}, \mathrm{EY})$ *iff its minimal tree automaton is in the least commutative closed variety of finite tree automata containing* $\mathbb{D}_0$ *and* $\mathbb{E}_Y$.

2. *A tree language belongs to* $\mathbf{CTL}^u(\mathrm{X}, \mathrm{EF}, \mathrm{EG})$ *iff its minimal tree automaton is in the least commutative closed variety containing* $\mathbb{D}_0$, $\mathbb{E}_F$ *and* $\mathbb{E}_G$.

3. *A tree language belongs to* $\mathbf{CTL}^u$ *iff its minimal automaton belongs to the commutative closed variety generated by* $\mathbb{E}_U$.

Recall from Example 4.4 in Chapter 1 the definition of the languages $L_{d,r}$, where $d > 1$ and $0 \leq r < d$, and the definition of the corresponding minimal automata $\mathbb{M}_d$, $d > 1$. Note that each $\mathbb{M}_d$ is commutative. For each d, let $\mathcal{L}^u_d = \{h_{\mathrm{Bool}}(L_{d,r}) : 0 \leq r < d\}$, and let $\mathcal{L}^u_{\mathrm{mod}} = \bigcup_{d>1} \mathcal{L}^u_d$. Define

$$\begin{aligned} \mathbf{CTL}^u + \mathbf{MOD}(d) &= \mathbf{FTL}(\mathcal{L}_{\mathrm{UX}} \cup \{L^u_{\mathrm{EU}}\} \cup \mathcal{L}^u_d) \\ \mathbf{CTL}^u + \mathbf{MOD} &= \mathbf{FTL}(\mathcal{L}_{\mathrm{UX}} \cup \{L^u_{\mathrm{EU}}\} \cup \mathcal{L}^u_{\mathrm{mod}}). \end{aligned}$$

Using Corollary 2.9.2, we obtain:

Theorem 2.10.2 *1. For every $d > 1$, a tree language belongs to* $\mathbf{CTL}^u + \mathbf{MOD}(d)$ *iff its minimal tree automaton is in the commutative closed variety generated by* $\mathbb{E}_U$ *and* $\mathbb{M}_d$.

2. *A tree language belongs to* $\mathbf{CTL}^u + \mathbf{MOD}$ *iff its minimal tree automaton is in the least commutative closed variety containing* $\mathbb{E}_U$ *and the tree automata* $\mathbb{M}_d$, $d > 1$.

2.11 Idempotence

In this section, we consider yet another variant of temporal logics on trees. Call an algebra $\mathbb{A}$ of rank type R *idempotent* if it is commutative and satisfies the equations

$$\sigma(x_1, \ldots, x_m, x_1, \ldots, x_1) = \sigma(x_1, \ldots, x_m, x_2, \ldots, x_2),$$

for all $1 \leq m < n$ and $\sigma \in \Sigma_n$. In such algebras $\mathbb{A}$, the result of an operation only depends on the set of its arguments, i.e.,

$$\sigma(a_1, \ldots, a_n) = \sigma(b_1, \ldots, b_n)$$

whenever $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$ are equal subsets of A and $\sigma \in \Sigma_n$. We call a tree automaton idempotent if it is an idempotent algebra. In our next result, which provides a characterization of tree languages recognizable by idempotent tree automata, we make use of a congruence relation $\sim$ on U_Σ. For any $s, t \in U_\Sigma$, we define $s \sim t$ iff $s = t \in \Sigma_0$ or $s = \sigma\{\{s_1, \ldots, s_n\}\}$, $t = \sigma\{\{t_1, \ldots, t_n\}\}$, where $\sigma \in \Sigma_n$ and $s_i, t_i \in U_\Sigma$, for all $i \in [n]$, and for every $i \in [n]$ there is a $j \in [n]$ with $s_i \sim t_i$, and vice versa. It is clear that $\sim$ is the least congruence relation on $\mathbf{U}_\Sigma$ such that the quotient algebra $I_\Sigma = U_\Sigma / \sim$ is idempotent. Thus, I_Σ is the initial idempotent algebra. Using this fact, we immediately have:

Proposition 2.11.1 *A tree language $L \subseteq U_\Sigma$ is recognizable by an idempotent algebra iff it is saturated by $\sim$, i.e., $s \sim t$ and $s \in L$ implies that $t \in L$. Moreover, L is recognizable by a finite idempotent algebra iff it is regular and is saturated by $\sim$.*

Idempotent tree automata form a variety of finite tree automata included in the variety of finite commutative tree automata that we denote below by **Idem**. The corresponding literal variety of unordered tree languages will be denoted by $\mathcal{I}$dem: it consists of all regular tree languages that are saturated by $\sim$. We call $\mathcal{I}$dem the class of all *idempotent regular unordered tree languages.*

The variety **Idem** is not closed under the commutative cascade product, but it is closed under the *idempotent cascade product* defined as follows. Suppose that $\mathbb{A}$ is a Σ-algebra and $\mathbb{B}$ is a Δ-algebra, and consider a family of functions $\alpha_n : A^n \times \Sigma_n \to \Delta_n$, $n \in R$, such that $\alpha_n(a_1, \ldots, a_n, \sigma)$ only depends on σ and the set $\{a_1, \ldots, a_n\}$. Then the cascade product of $\mathbb{A} \times_\alpha \mathbb{B}$ determined by the family $\alpha = (\alpha_n)_{n \in R}$ is called an idempotent cascade product of $\mathbb{A}$ and $\mathbb{B}$. When $\mathbb{A}$ and $\mathbb{B}$ are tree automata, the idempotent ta-cascade product of $\mathbb{A}$ and $\mathbb{B}$ determined by α is the least subalgebra of the ordinary cascade product. It is easy to see that **Idem** is closed under the ta-cascade product. Below, we will just write idempotent cascade product for the idempotent ta-cascade product.

Suppose that $\mathbf{K}$ is a class of finite idempotent algebras. Say that the idempotent next modality is expressible in $\mathrm{FTL}^u(\mathbf{K})$ if for each formula φ in $\mathrm{FTL}^u(\mathbf{K})$ over any ranked set Σ there exists a formula $\mathrm{EX}\varphi$ in $\mathrm{FTL}^u(\mathbf{K})$ over Σ such that for any $t \in U_\Sigma$, $t \models \mathrm{EX}\varphi$ iff t has an immediate subtree satisfying φ. It is not difficult to see that this condition holds iff $\mathbf{FTL}^u(\mathbf{K})$ contains all *idempotent definite tree languages*, i.e., those commutative definite tree languages contained in $\mathcal{I}$dem.

Using the methods of the previous sections, we can prove the following results.

Theorem 2.11.2 *Suppose that $\mathbf{K}$ is a class of finite idempotent algebras such that the idempotent next modality is expressible in* $\mathrm{FTL}^u(\mathbf{K})$. *Then an idempotent tree language $L \subseteq U_\Sigma$ is in $\mathbf{FTL}^u(\mathbf{K})$ iff its minimal tree automaton belongs to the least variety of finite idempotent algebras containing $\mathbb{D}_0$ and $\mathbf{K}$, closed under the idempotent cascade product.*

Theorem 2.11.3 *Let* $\mathbf{V}$ *be a variety of finite tree automata containing the finite idempotent definite tree automata and contained in* **Idem**, *closed under the idempotent cascade product. Then* $\mathbf{FTL}^u(\mathbf{V}) = \mathcal{L}^u_{\mathbf{V}}$. *Moreover, the assignment* $\mathbf{V} \mapsto \mathbf{FTL}^u(\mathbf{V})$ *defines an order isomorphism between varieties of finite idempotent tree automata containing the finite idempotent definite tree automata closed under the idempotent cascade product, and closed classes of unordered tree languages contained in* $\mathcal{I}\mathrm{dem}$ *and containing the idempotent definite tree languages.*

Let $\mathbf{CTL}^i$ denote the class of all idempotent tree languages definable by the formulas of the logic $\mathrm{FTL}^u(\{\mathbb{E}_U\})$. As an application, we have:

Theorem 2.11.4 *An unordered tree language belongs to* $\mathbf{CTL}^i$ *iff its minimal tree automaton is in the least variety of finite (idempotent) tree automata containing* $\mathbb{E}_U$ *closed under the idempotent cascade product.*

Chapter 3

3.1 Introduction

In Chapter 1, we associated a temporal logic with each class of regular tree languages and gave an algebraic characterization of the expressive power of these logics under certain natural assumptions. Our characterization was based on the notion of the cascade product of finite algebras. In order to turn the obtained algebraic characterization into decision procedures, one has to develop a structure theory of finite algebras with respect to the cascade product. In this paper, we give an effective characterization of the expressive power of a simple temporal logic involving only the next and eventually modalities. Our result is based on the general results of Chapter 1 and on an analysis of the structure of finite algebras in the variety generated by certain two-element algebras, closed under the cascade product.

3.2 Preliminaries

Let R denote a rank type containing 0. We let R^- stand for the rank type $R - \{0\}$. Similarly, if Σ is a ranked alphabet of rank type R, then we let Σ^- denote the ranked alphabet of rank type R^- obtained from R by removing all symbols of rank 0. When $\mathbb{A}$ is a finite Σ-algebra of rank type R, then we let $\mathbb{A}^-$ denote the Σ^--algebra obtained from $\mathbb{A}$ by forgetting about the constants. By extension, if $\mathbf{K}$ is a class of finite algebras of rank type R, then $\mathbf{K}^- = \{\mathbb{A}^- : \mathbb{A} \in \mathbf{K}\}$ is a class of finite algebras of rank type R^-.

Conversely, if $\mathbf{K}$ is a class of finite algebras of rank type R^-, then $\mathbf{K}^+$ denotes the class of all *finite tree automata* of rank type R whose reducts obtained by forgetting about the constants belong to $\mathbf{K}$. We call a class $\mathbf{K}$ of finite tree automata of rank type R *strictly closed* if there is a closed variety $\mathbf{K}_0$ of finite algebras of rank type R^- such that $\mathbf{K} = \mathbf{K}_0^+$. Note that every strictly closed class of finite tree automata is a closed variety of finite tree automata. Thus, we will also call a strictly closed class of finite tree automata a *strictly closed* *variety.*

Remark 3.2.1 When $\mathbf{K}$ is strictly closed, there is a unique closed variety $\mathbf{K}_0$ of finite algebras of type R^- with $\mathbf{K} = \mathbf{K}_0^+$. In fact, $\mathbf{K}_0 = \mathbf{K}^-$.

When Σ is a ranked alphabet of rank type R and a is a letter not in Σ, then we let $\Sigma(a)$ denote the ranked alphabet obtained from Σ by adding a to Σ_0.

Proposition 3.2.2 *The following conditions are equivalent for a class* $\mathbf{K}$ *of finite tree automata of rank type* R.

1. $\mathbf{K}$ *is a strictly closed variety.*
2. $\mathbf{K}^-$ *is a closed variety of finite algebras of rank type* R^- *and* $\mathbf{K}$ *is closed under adding constants, i.e., whenever* $\mathbb{A}$ *is a* Σ*-tree automaton in* $\mathbf{K}$ *and* c *is in* A*, then the* $\Sigma(\overline{c})$*-tree automaton* $\mathbb{A}(c)$ *obtained from* $\mathbb{A}$ *by interpreting the (fresh) symbol* $\overline{c}$ *as* c *belongs to* $\mathbf{K}$.
3. $\mathbf{K}$ *is a closed variety of finite tree automata which is additionally closed under adding constants.*
4. *There is a class* $\mathbf{K}_0$ *of finite tree automata such that for each* Σ*-tree automaton* $\mathbb{A}$ *in* $\mathbf{K}$ *and for any* $c \in A$ *there is a letter* $\overline{c}$ *in* Σ_0 *whose interpretation is* c*, and such that* $\mathbf{K}$ *is the least closed variety of finite automata containing* $\mathbf{K}_0$.

Proof. The first two conditions are clearly equivalent. Assume now that $\mathbf{K}$ is a strictly closed variety of finite algebras of rank type R. We have already noted that $\mathbf{K}$ is a closed variety of finite tree automata. If $\mathbb{A} \in \mathbf{K}$ and $c \in A$, then $\mathbb{A}(c) \in \mathbf{K}$ since $\mathbb{A}(c)^- = \mathbb{A}^- \in \mathbf{K}^-$. Thus, the first condition implies the third. The fact that the third condition implies the fourth follows by letting $\mathbf{K}_0$ consist of those tree automata $\mathbb{A}$ in $\mathbf{K}$ such that each $c \in A$ is the interpretation of at least one constant symbol. Finally, to see that the fourth condition implies the first, one can show that if $\mathbf{K}_1$ denotes the closed variety of finite algebras of rank type R^- generated by $\mathbf{K}_0^-$, then $\mathbf{K} = \mathbf{K}_1^+$. □

The results of the paper can be best presented with the help of partial algebras. Suppose that Σ is a ranked alphabet of rank type R^-. A *partial* Σ*-algebra* $\mathbb{A}$ consists of a nonempty set A and a partial operation for each symbol in Σ, i.e., a partial function $\sigma_{\mathbb{A}} : A^n \to A$ for each $\sigma \in \Sigma_n$, $n > 0$. Note that every Σ-algebra is a partial algebra. The notions of homomorphism, subalgebras etc. can be extended to partial algebras in several different ways, cf., e.g., Grätzer [13]. Here we use these concepts as described below.

Suppose that $\mathbb{A} = (A, (\sigma_{\mathbb{A}})_{\sigma\in\Sigma})$ and $\mathbb{B} = (B, (\sigma_{\mathbb{B}})_{\sigma\in\Sigma})$ are partial Σ-algebras, where Σ is of rank type R^-. We say that $\mathbb{A}$ is a *partial subalgebra* of $\mathbb{B}$ if $A \subseteq B$ and for all $\sigma \in \Sigma_n$, $n > 0$, and $a_1, \ldots, a_n \in A$, if $\sigma_{\mathbb{A}}(a_1, \ldots, a_n)$ is defined then so is $\sigma_{\mathbb{B}}(a_1, \ldots, a_n)$ and $\sigma_{\mathbb{A}}(a_1, \ldots, a_n) = \sigma_{\mathbb{B}}(a_1, \ldots, a_n)$. When $\mathbb{A}$ is a partial subalgebra of $\mathbb{B}$ and $A = B$, we also say that $\mathbb{B}$ is an *extension* of $\mathbb{A}$. Moreover, we say that $\mathbb{A}$ is an *induced partial subalgebra* of $\mathbb{B}$ if $\mathbb{A}$ is a partial subalgebra of $\mathbb{B}$ and for each $\sigma \in \Sigma_n$, $n > 0$ and $a_1, \ldots, a_n \in A$, $\sigma_{\mathbb{A}}(a_1, \ldots, a_n)$ is defined iff $\sigma_{\mathbb{B}}(a_1, \ldots, a_n)$ is defined. Thus, when X is a nonempty subset of B, then X *induces a partial subalgebra of* $\mathbb{B}$ whose carrier is X and whose operations are the restrictions of the operations of $\mathbb{B}$ onto X. Note that an induced partial subalgebra of $\mathbb{B}$ is not necessarily closed under all operations of $\mathbb{B}$. A *homomorphism* $\mathbb{A} \to \mathbb{B}$ is a function $h : A \to B$ such that for all $\sigma \in \Sigma_n$, $n > 0$ and for all $a_1, \ldots, a_n \in A$, if $\sigma(a_1, \ldots, a_n)$ is defined then $\sigma(h(a_1), \ldots, h(a_n))$ is defined, and $h(\sigma(a_1, \ldots, a_n)) = \sigma(h(a_1), \ldots, h(a_n))$. We say that an equivalence relation $\sim$ on A is a *congruence* of $\mathbb{A}$ if for all $\sigma \in \Sigma_n$, $n > 0$, $a_1, \ldots, a_n, a'_1, \ldots, a'_n \in A$, if $a_i \sim a'_i$, for all $i \in [n]$, and $\sigma(a_1, \ldots, a_n)$ and $\sigma(a'_1, \ldots . a'_n)$ are both defined, then $\sigma(a_1, \ldots, a_n) \sim \sigma(a'_1, \ldots . a'_n)$. If $\sim$ is a congruence of $\mathbb{A}$, the factor algebra $\mathbb{A}/\sim$ is the partial algebra on the quotient set $A/\sim$ such that for all $\sigma \in \Sigma_n$, $n > 0$, and for all congruence classes $C_1, \ldots, C_n, C$, it holds that $\sigma(C_1, \ldots, C_n) = C$ iff $\sigma(c_1, \ldots, c_n) \in C$ holds in $\mathbb{A}$ for some $c_i \in C_i$, $i \in [n]$. Note that the quotient map $A \to A/\sim$ is a homomorphism $\mathbb{A} \to \mathbb{A}/\sim$.

Suppose that $\mathbb{A}$ is a partial Σ-algebra, where Σ is of rank type R^-. Let $a, b \in A$. We say that b is *accessible* from a if there is a tree $t \in T_\Sigma(X_{n+1})$, for some $n \geq 0$, such that $b = t(a, c_1, \ldots, c_n)$ for some $c_1, \ldots, c_n \in A$. A *transitivity class* of $\mathbb{A}$ is any maximal subset X of A with the property that for any $a, b \in X$, b is accessible from a. It is clear that each $a \in A$ is contained in a unique transitivity class. The transitivity classes are partially ordered as follows. Suppose that X and Y are transitivity classes. Then $X \leq Y$ iff there exist $a \in X$ and $b \in Y$ such that b is accessible from a iff for all $a \in X$ and $b \in Y$, b is accessible from a. Note that if $X_1, \ldots, X_n, X$ are transitivity classes, $a_1 \in X_1, \ldots, a_n \in X_n$, and $\sigma(a_1, \ldots, a_n) \in X$, for some $\sigma \in \Sigma_n$, then $X_i \leq X$ holds for all $i \in [n]$. In particular, any maximal transitivity class is closed with respect to all operations. Moreover, for any transitivity class X, the union of all transitivity classes $\geq X$ is closed with respect to all operations.

Lemma 3.2.3 *Suppose that $\mathbb{A}$ is a partial algebra of rank type R^- and ρ is a congruence relation of $\mathbb{A}$. Suppose that whenever $a \rho b$ holds for some $a, b \in A$, then a and b are in the same transitivity class. Then the transitivity classes of $\mathbb{A}/\rho$ are the sets X/ρ, where X is a transitivity class of $\mathbb{A}$.*

3.3 A Closed Variety of Finite Algebras

In Chapter 1, we defined the algebra (tree automaton) $\mathbb{E}_F(R)$ for each rank type R containing 0. By forgetting about the constants, we obtain the algebra $\mathbb{E}_F(R^-)$. Below, when the context permits, we will just write $\mathbb{E}_F$ for both $\mathbb{E}_F(R)$ and $\mathbb{E}_F(R^-)$.

Let $\mathbf{W}_p$ denote the class of finite partial Σ-algebras $\mathbb{A}$, for all ranked alphabets Σ of rank type R^-, with the following property: There exists an integer $k \geq 0$ such that for every $t \in T_\Sigma(X_{m+n})$, $m, n \geq 0$, such that the depth of each vertex labeled x_i with $i \in [m]$ is at least k, and for all transitivity classes X and $a_i, b_i \in X$, $i \in [m]$, and $c_j \in A$, $j \in [n]$, if both $t(a_1, \ldots, a_m, c_1, \ldots, c_n)$ and $t(b_1, \ldots, b_m, c_1, \ldots, c_n)$ exist and are in X, then $t(a_1, \ldots, a_m, c_1, \ldots, c_n) = t(b_1, \ldots, b_m, c_1, \ldots, c_n)$. When $\mathbb{A} \in \mathbf{W}_p$, the least such integer k will be called the *index* of $\mathbb{A}$. We let $\mathbf{W}$ denote the subclass of all (complete) algebras in $\mathbf{W}_p$.

Proposition 3.3.1 *$\mathbf{W}$ is a closed variety containing the finite definite algebras of rank type R^- and the algebra $\mathbb{E}_F(R^-)$.*

Proof. It is clear that $\mathbf{W}$ contains $\mathbb{E}_F$ and all finite definite algebras. Since $\mathbf{W}$ contains all trivial algebras and is clearly closed under subalgebras, it suffices to show that $\mathbf{W}$ is closed under the cascade product and homomorphic images. So suppose that $h : \mathbb{A} \to \mathbb{B}$ is a surjective homomorphism, where $\mathbb{A}$ is in $\mathbf{W}$. Suppose that $t \in T_\Sigma(X_{m+n})$, Y is a transitivity class of $\mathbb{B}$ and $a_i', b_i' \in Y$, $i \in [m]$, $c_j' \in B$, $j \in [n]$, such that $t(a_1', \ldots, a_m',, c_1', \ldots, c_n')$ and $t(b_1', \ldots, b_m', c_1', \ldots, c_n')$ are in Y and $t(a_1', \ldots, a_m',, c_1', \ldots, c_n') \neq t(b_1', \ldots, b_m', c_1', \ldots, c_n')$. Then let X denote a transitivity class of $\mathbb{A}$ such that $h(X)$ intersects Y which is maximal with this property with respect to the ordering of transitivity classes of $\mathbb{A}$. Then $h(X) = Y$. Let $a_i, b_i \in X$ and $c_j \in A$, $i \in [m]$, $j \in [n]$ with $h(a_i) = a_i'$, $h(b_i) = b_i'$ and $h(c_j) = c_j'$, for all $i \in [m]$ and $j \in [n]$. Using the maximality of X, we have that $t(a_1, \ldots, a_m, c_1, \ldots, c_n)$ and $t(b_1, \ldots, b_m, c_1, \ldots, c_n)$ are in X, moreover, since

$$
\begin{aligned}
h(t(a_1, \ldots, a_m, c_1, \ldots, c_n)) &= t(a_1', \ldots, a_m', c_1', \ldots, c_n') \\
&\neq t(b_1', \ldots, b_m',, c_1', \ldots, c_n') \\
&= h(t(b_1, \ldots, b_m, c_1, \ldots, c_n)),
\end{aligned}
$$

we have $t(a_1, \ldots, a_m, c_1, \ldots, c_n) \neq t(b_1, \ldots, b_m, c_1, \ldots, c_n)$. This shows that if k denotes the index of $\mathbb{A}$, then for every $t \in T_\Sigma(X_{m+n})$ such that the depth of each vertex labeled x_i with $i \in [m]$ is at least k, and for all transitivity classes Y of $\mathbb{B}$ and $a_i', b_i' \in Y$, $c_j' \in B$, $i \in [m]$, $j \in [n]$, either at least one of $t(a_1', \ldots, a_m', c_1', \ldots, c_n')$ and $t(b_1', \ldots, b_m', c_1', \ldots, c_n')$ is not in Y or $t(a_1', \ldots, a_m', c_1', \ldots, c_n') = t(b_1', \ldots, b_m', c_1', \ldots, c_n')$. Thus, $\mathbb{B} \in \mathbf{W}$.

To show that $\mathbf{W}$ is closed under the cascade product, suppose that $\mathbb{A}$ is a finite Σ-algebra in $\mathbf{W}$, $\mathbb{B}$ is a finite Δ-algebra in $\mathbf{W}$, and consider a cascade product $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$. Let k denote the sum of the indices of $\mathbb{A}$ and $\mathbb{B}$. Suppose that $t \in T_\Sigma(X_{m+n})$ such that the depth of each vertex labeled x_i with $i \in [m]$ is at least k and Z is a transitivity class of $\mathbb{C}$. We can decompose t as $r(s_1, \ldots, s_\ell, x_{m+1}, \ldots, x_{m+n})$, for some trees $r \in T_\Sigma(X_{\ell+n})$ and $s_j \in T_\Sigma(X_{m+n})$, $j \in [\ell]$, such that whenever a vertex in some s_j is labeled x_i with $i \in [m]$ then the depth of that vertex is greater than or equal to the index of $\mathbb{A}$, and the depth of each vertex of r labeled x_j with $j \in [\ell]$ is greater than or equal to the index of $\mathbb{B}$. It is easy to see using Proposition 7.1 in Chapter 1 that there exist a transitivity class X of $\mathbb{A}$ and a transitivity class Y of $\mathbb{B}$ with $Z \subseteq X \times Y$. Let $(a_i, a_i'), (b_i, b_i') \in Z$ and $(c_j, c_j') \in C$, for all $i \in [m]$ and $j \in [n]$. Assume that $t_{\mathbb{C}}((a_1, a_1'), \ldots, (a_m, a_m'), (c_1, c_1'), \ldots, (c_n, c_n'))$ and $t_{\mathbb{C}}((b_1, b_1'), \ldots, (b_m, b_m'), (c_1, c_1'), \ldots, (c_n, c_n'))$ are in Z. We want to show that these two elements are equal. Since for all $j \in [\ell]$, $(s_j)_{\mathbb{A}}(a_1, \ldots, a_m, c_1, \ldots, c_n)$ and $(s_j)_{\mathbb{A}}(b_1, \ldots, b_m, c_1, \ldots, c_n)$ are in X, we have $(s_j)_{\mathbb{A}}(a_1, \ldots, a_m, , c_1, \ldots, c_n) = (s_j)_{\mathbb{A}}(b_1, \ldots, b_m, , c_1, \ldots, c_n) = d_j$, $j \in [\ell]$. For each $j \in [\ell]$, let us define $s_j^a = \alpha_{(a_1, \ldots, a_m, c_1, \ldots, c_n)}(s_j)$ and $s_j^b = \alpha_{(b_1, \ldots, b_m, c_1, \ldots, c_n)}(s_j)$. Moreover, define $e_j = s_j^a(a_1', \ldots, a_m', c_1', \ldots, c_n')$ and $f_j = s_j^b(b_1', \ldots, b_m', c_1', \ldots, c_n')$, $j \in [\ell]$, and $\widehat{r} = \alpha_{(d_1, \ldots, d_m, c_1, \ldots, c_n)}(r)$. By Proposition 7.1 in Chapter 1,

$$\begin{aligned} &t_{\mathbb{C}}((a_1, , a_1'), \ldots, (a_m, a_m'), (c_1, c_1'), \ldots, (c_n, c_n')) = \\ &\quad = \quad (r_{\mathbb{A}}(d_1, \ldots, d_\ell, c_1, \ldots, c_n), \widehat{r}_{\mathbb{B}}(e_1, \ldots, e_\ell, c_1, \ldots, c_n)) \\ &t_{\mathbb{C}}((b_1, b_1'), \ldots, (b_m, b_m')(c_1, c_1'), \ldots, (c_n, c_n')) = \\ &\quad = \quad (r_{\mathbb{A}}(d_1, \ldots, d_\ell, c_1, \ldots, c_n), \widehat{r}_{\mathbb{B}}(f_1, \ldots, f_\ell, c_1, \ldots, c_n)). \end{aligned}$$

However, $e_j, f_j \in Y$, for all $j \in [\ell]$, and both $\widehat{r}_{\mathbb{B}}(e_1, \ldots, e_\ell, c_1, \ldots, c_n)$ and $\widehat{r}_{\mathbb{B}}(f_1, \ldots, f_\ell, c_1, \ldots, c_n)$ are in Y. But since $\mathbb{B}$ is in $\mathbf{W}$ and the index of $\mathbb{B}$ is less than or equal to the depth of any vertex of $\widehat{r}$ labeled x_j, for all $j \in [\ell]$, we have that

$$\widehat{r}_{\mathbb{B}}(e_1, \ldots, e_\ell, c_1, \ldots, c_n) \quad = \quad \widehat{r}_{\mathbb{B}}(f_1, \ldots, f_\ell, c_1, \ldots, c_n).$$

This proves that the elements $t_{\mathbb{C}}((a_1, a_1'), \ldots, (a_m, a_m'), (c_1, c_1'), \ldots, (c_n, c_n'))$ and $t_{\mathbb{C}}((b_1, b_1'), \ldots, (b_m, b_m'), (c_1, c_1'), \ldots, (c_n, c_n'))$ are equal. We have thus proved that $\mathbb{C}$ is in $\mathbf{W}$ with index less than or equal to k. □

Proposition 3.3.2 *Suppose that $\mathbb{A}$ is in $\mathbf{W}_p$. Then each nontrivial transitivity class X of $\mathbb{A}$ contains two different elements a, b such that for any $\sigma \in \Sigma_n$ and $c_i, c_i' \in A$ such that $c_i = c_i'$ or $\{c_i, c_i'\} = \{a, b\}$, for all $i \in [n]$, if $c = \sigma(c_1, \ldots, c_n)$ and $c' = \sigma(c_1', \ldots, c_n')$ are defined and belong to X, then $c = c'$.*

Proof. Suppose that X is a nontrivial transitivity class. There exists an integer k with the property that for all $t \in T_\Sigma(X_{m+n})$ with $m, n \geq 0$ such that each leaf of t labeled x_i with $i \in [m]$ is of depth $\geq k$ and for any $a_1', \ldots, a_m', b_1', \ldots, b_m' \in X$ and $d_1, \ldots, d_n \in A$, if $t(a_1', \ldots, a_m', , c_1, \ldots, c_n)$ and $t(b_1', \ldots, b_m', , c_1, \ldots, c_n)$ are both defined and belong to X, then the elements $t(a_1', \ldots, a_m', c_1, \ldots, c_n)$ and $t(b_1', \ldots, b_m', c_1, \ldots, c_n)$ are equal. Now let k_0 denote the least such integer. Since X has at least two elements, $k_0 > 0$. Moreover, there exists some $t_0 \in T_\Sigma(X_{m_0+n_0})$, $a_1', \ldots, a_{m_0}', b_1', \ldots, b_{m_0}' \in X$ and $d_1, \ldots, d_{n_0} \in A$ such that every leaf of t_0 labeled in $\{x_1, \ldots, x_{m_0}\}$ is of depth $\geq k_0 - 1$ and $a = t_0(a_1', \ldots, a_{m_0}', c_1, \ldots, c_{n_0})$ and $b = t_0(b_1', \ldots, b_{m_0}', c_1, \ldots, c_{n_0})$ are different elements of X. It is now clear that a and b satisfy the condition in the statement of the Proposition. □

Proposition 3.3.3 *A finite partial algebra of rank type R^- belongs to $\mathbf{W}_p$ iff it has an extension to an algebra in $\mathbf{W}$. Moreover, any partial algebra $\mathbb{A}$ in $\mathbf{W}_p$ has an extension $\mathbb{A}'$ in $\mathbf{W}$ such that the transitivity classes of $\mathbb{A}$ are the same as those of $\mathbb{A}'$.*

Proof. The sufficiency part is obvious. Suppose that $\mathbb{A}$ is in $\mathbf{W}_p$. We prove that $\mathbb{A}$ has an extension to an algebra $\mathbb{A}'$ in $\mathbf{W}$ having the same transitivity classes. Let $\#(\mathbb{A})$ denote the number of tuples $(\sigma, a_1, \ldots, a_m)$ with $\sigma \in \Sigma_m$, $m > 0$, $a_1, \ldots, a_m \in A$ such that $\sigma(a_1, \ldots, a_m)$ is undefined. We argue by induction on $\#(A)$. When this number is 0, our claim is trivial. Suppose that $\#(\mathbb{A}) > 0$. Let us extend the partial order on the transitivity classes to a linear order, and let $X_{\max}$ denote the greatest transitivity class with respect to this linear order. If there exist some $\sigma \in \Sigma_m$, $m > 0$ and $a_1, \ldots, a_m \in A - X_{\max}$ such that $\sigma(a_1, \ldots, a_m)$ is not defined in $\mathbb{A}$, then make it defined by any element of $X_{\max}$. The resulting partial algebra is also in $\mathbf{W}_p$, which by induction has an extension to an algebra in $\mathbf{W}$ with the same transitivity classes, and thus by the same transitivity classes that $\mathbb{A}$ has. Thus, we may suppose that whenever $\sigma(a_1, \ldots, a_m)$ is undefined in $\mathbb{A}$, then at least one of the a_i is in $X_{\max}$. Now if $X_{\max}$ is a singleton, it is clear how to extend $\mathbb{A}$ to an algebra in $\mathbf{W}$: Whenever $\sigma(a_1, \ldots, a_m)$ is undefined, make it defined by the unique element of $X_{\max}$. So in the rest of the proof we assume that $X_{\max}$ has at least 2 elements. By Proposition 3.3.2, there exist different elements $c, c' \in X_{\max}$ such that the relation that collapses c and c' and keeps all other elements intact is a congruence relation of $\mathbb{A}$. Let us denote this congruence relation by ρ. Moreover, for all $\sigma \in \Sigma_m$, $m > 0$ and congruence classes $C_1, \ldots, C_m$, if $\sigma(C_1, \ldots, C_m) = \{c, c'\}$ in the quotient partial algebra $\mathbb{A}/\rho$, then either $\sigma(C_1, \ldots, C_m) = \{c\}$ or $\sigma(C_1, \ldots, C_m) = \{c'\}$ holds in $\mathbb{A}$. Now $\mathbb{A}/\rho$ is also in $\mathbf{W}_p$ and, by the induction hypothesis, has an extension to an algebra $\mathbb{B}$ in $\mathbf{W}$ with the same transitivity classes. We use $\mathbb{B}$ to construct a suitable extension $\mathbb{A}'$ of $\mathbb{A}$. Let $\sigma \in \Sigma_m$, $m > 0$, $a_1, \ldots, a_m \in A$. If in $\mathbb{B}$, $C = \sigma(\rho(a_1), \ldots, \rho(a_m))$ is not the congruence class $\{c, c'\}$, then in $\mathbb{A}'$ we define $\sigma(a_1, \ldots, a_m)$ as the unique element of C. If $C = \{c, c'\}$, then we know that in $\mathbb{A}$, either $\sigma(\rho(a_1), \ldots, \rho(a_m)) \subseteq \{c\}$ or $\sigma(\rho(a_1), \ldots, \rho(a_m)) = \{c'\}$. In the first case, define $\sigma(a_1, \ldots, a_m) = c$, and in the second, define $\sigma(a_1, \ldots, a_m) = c'$. Note that ρ is also a congruence relation of $\mathbb{A}'$.

It is clear that $\mathbb{A}'$ is an extension of $\mathbb{A}$. Moreover, it is easy to see using Lemma 3.2.3 that the transitivity classes of $\mathbb{A}'$ are those of $\mathbb{A}$. We know that $\mathbb{B}$ is in $\mathbf{W}$. Let k denote the index of $\mathbb{B}$. We show that $\mathbb{A}'$ has index $\leq k+1$. To prove this, suppose that $t \in T_\Sigma(X_{m+n})$ is such that the depth of each vertex labeled in the set $\{x_1, \ldots, x_m\}$ is at least $k+1$. Write $t = \sigma(t_1, \ldots, t_\ell)$, where $\sigma \in \Sigma_\ell$, $\ell > 0$. Let $a_1, a_1', \ldots, a_m, a_m'$ be in the same transitivity class Y, and let $b_1, \ldots, b_n \in A$. Assume that in $\mathbb{A}'$, both $t(a_1, \ldots, a_m, b_1, \ldots, b_n)$ and $t(a_1', \ldots, a_m', b_1, \ldots, b_n)$ are in Y. Define $d_i = t_i(a_1, \ldots, a_m, b_1, \ldots, b_n)$ and $d_i' = t_i(a_1', \ldots, a_m', b_1, \ldots, b_n)$, for all $i \in [\ell]$. If for some i, none of the variables in $\{x_1, \ldots, x_m\}$ occurs in t_i, then clearly $d_i = d_i'$. If some of these variables does occur, then the transitivity class of d_i and of d_i' is Y, since it must be both below and above Y in the partial order of the transitivity classes. Since $\mathbb{B}$ is in $\mathbf{W}$ and has index k, it follows now that $\rho(d_i) = \rho(d_i')$, for all $i \in [\ell]$. Now by construction, $\rho(t_{\mathbb{A}'}(a_1, \ldots, a_m, b_1, \ldots, b_n)) = \sigma_{\mathbb{B}}(\rho(d_1), \ldots, \rho(d_\ell)) = \sigma_{\mathbb{B}}(\rho(d_1'), \ldots, \rho(d_\ell')) = \rho(t_{\mathbb{A}'}(a_1', \ldots, a_m', b_1, \ldots, b_n))$. Let C denote this equivalence class. If $C \neq \{c, c'\}$, then both $t_{\mathbb{A}'}(a_1, \ldots, a_m, b_1, \ldots, b_n)$ and $t_{\mathbb{A}'}(a_1', \ldots, a_m', b_1, \ldots, b_n)$ are equal to the unique element of C. If $C = \{c, c'\}$, so that $Y = X_{\max}$, then, by construction, either $t_{\mathbb{A}'}(a_1, \ldots, a_m, b_1, \ldots, b_n) = c = t_{\mathbb{A}'}(a_1', \ldots, a_m', b_1, \ldots, b_n)$ or $t_{\mathbb{A}'}(a_1, \ldots, a_m, b_1, \ldots, b_n) = c' = t_{\mathbb{A}'}(a_1', \ldots, a_m', b_1, \ldots, b_n)$ holds in $\mathbb{A}'$. □

Theorem 3.3.4 $\mathbf{W}$ *is the least closed variety of finite algebras of rank type* R^- *containing the definite algebras and the algebra* $\mathbb{E}_F(R^-)$.

Proof. Let $\mathbf{W}'$ denote the least closed variety of finite algebras of rank type R^- containing the definite algebras and the algebra $\mathbb{E}_F(R^-)$. By Proposition 3.3.1, we have that $\mathbf{W}' \subseteq \mathbf{W}$. Suppose now that $\mathbb{A}$ is in $\mathbf{W}$. We use induction on the number of elements of A to show that $\mathbb{A}$ belongs to $\mathbf{W}'$. The induction base when A has a single element is obvious. So suppose that A has at least two elements.

We know that the transitivity classes of $\mathbb{A}$ are partially ordered. Let us extend this partial order arbitrarily to a linear order and let $X_{\max}$ denote the greatest transitivity class of $\mathbb{A}$ with respect to this linear order.

Assume first that $X_{\max}$ has a single element, denoted $a_{\max}$. Then let $\mathbb{B}$ denote the induced partial subalgebra of $\mathbb{A}$ determined by the set $B = A - \{a_{\max}\}$. Clearly, $\mathbb{B} \in \mathbf{W}_p$. We know that $\mathbb{B}$ has an extension to an algebra $\mathbb{B}'$ in $\mathbf{W}$. Moreover, by the induction hypothesis, $\mathbb{B}'$ is in $\mathbf{W}'$. To prove that $\mathbb{A} \in \mathbf{W}'$, we show that $\mathbb{A}$ is a homomorphic image of a cascade product $\mathbb{C} = \mathbb{B}' \times_\alpha \mathbb{E}_F$. To this end, for each $n \in R^-$ define $\alpha_n : B^n \times \Sigma_n \to \{\uparrow_n, \downarrow_n\}$ as follows:

$$\alpha_n(a_1, \ldots, a_n, \sigma) \quad = \quad \begin{cases} \uparrow_n & \text{if } \sigma_{\mathbb{A}}(a_1, \ldots, a_n) = a_{\max} \\ \downarrow_n & \text{otherwise.} \end{cases}$$

It is clear that the function $(a, 0) \mapsto a$, $(a, 1) \mapsto a_{\max}$, $a \in B$, is a surjective homomorphism $\mathbb{C} \to \mathbb{A}$.

The second case is that $A_{\max}$ has two ore more elements. Then let $X = X_{\max}$ in Proposition 3.3.2. There exist different elements $a, b \in X_{\max}$ such that all conditions of Proposition 3.3.2 hold. In particular, the equivalence relation ρ that collapses a, b and keeps the other elements separated is a congruence ρ. Moreover, for all $\sigma \in \Sigma_n$ and elements $a_i, b_i \in A$ with $a_i \rho b_i$, $i \in [n]$, $\sigma(a_1, \ldots, a_n) = \sigma(b_1, \ldots, b_n)$. The quotient $\mathbb{A}/\rho$ is in $\mathbf{W}$. Thus, by the induction hypothesis, it is in $\mathbf{W}'$. Let $\mathbb{D}_0$ denote the 1-definite algebra on the set $\{0, 1\}$ with operations $\uparrow_n (d_1, \ldots, d_n) = 1$ and $\downarrow_n (d_1, \ldots, d_n) = 0$, for all $n \in R^-$ and $d_1, \ldots, d_n \in \{0, 1\}$. Then define the cascade product $\mathbb{A}/\rho \times_\alpha \mathbb{D}_0$ by

$$\alpha_n(\rho(a_1), \ldots, \rho(a_n), \sigma) \quad = \quad \begin{cases} \uparrow_n & \text{if } \sigma(a_1, \ldots, a_n) = a \\ \downarrow_n & \text{otherwise,} \end{cases} \qquad n \in R.$$

Then the set $\{(\{c\},0) : c \notin \{a,b\}\} \cup \{(\{a,b\},0),(\{a,b\},1)\}$ determines a subalgebra isomorphic to $\mathbb{A}$. Thus, $\mathbb{A} \in \mathbf{W}'$. □

3.4 An Effective Characterization of CTL(X, EF)

The temporal logic **CTL**(X, EF) was defined in Chapter 1. In this section, we combine results from Chapter 1 and the previous sections to derive an effective characterization of **CTL**(X, EF).

Theorem 3.4.1 *Suppose that Σ is a ranked alphabet of rank type R. A language $L \subseteq T_\Sigma$ is in* **FTL**(X, EF) *iff the minimal automaton of L is in $\mathbf{W}^+$ iff L can be accepted by a tree automaton in $\mathbf{W}^+$.*

Proof. From Theorem 10.1 in Chapter 1 and Theorem 3.3.4. □

Theorem 3.4.2 *There exists an algorithm to decide whether or not a regular tree language (given by a tree automaton with a specified set of final states) is in* **FTL**(X, EF).

Proof. By Theorem 3.4.1, all we have to show is that if $\mathbb{A}$ is in $\mathbf{W}$ and has index k, then k is less than $|A|^2$, But this follows by noting that given any tree $t \in T_\Sigma(X_{m+n})$, transitivity class C, $a_1,\ldots,a_m,b_1,\ldots,b_m \in C$ and $c_1,\ldots,c_n \in A$, if the depth of a leaf labeled x_i is greater than or equal to $|A|^2$, then there exist different vertices v_1 and v_2 along this path such that the subtrees rooted at these vertices evaluate to the same element a on the tuple $(a_1,\ldots,a_m,c_1,\ldots,c_n)$, and to the same element b on $(b_1,\ldots,b_m,c_1,\ldots,c_n)$. Assume that v_1 is closer to the root. Then we may replace the subtree rooted at v_1 with the subtree rooted in v_2 to obtain a tree $t' \in T_\Sigma(X_{m+m})$ with $t'(a_1,\ldots,a_m,c_1,\ldots,c_n) = t(a_1,\ldots,a_m,c_1,\ldots,c_n)$ and $t'(b_1,\ldots,b_m,c_1,\ldots,c_n) = t(b_1,\ldots,b_m,c_1,\ldots,c_n)$. By repeating this procedure, in the end we obtain a tree t' such that the above equalities hold and the depth of t' is less than $|A|^2$. Moreover, if

$$t(a_1,\ldots,a_m,c_1,\ldots,c_n) \neq t(b_1,\ldots,b_m,c_1,\ldots,c_n),$$

then also

$$t'(a_1,\ldots,a_m,c_1,\ldots,c_n) \neq t'(b_1,\ldots,b_m,c_1,\ldots,c_n). \quad \square$$

Remark 3.4.3 When $R = \{0,1\}$, our characterization of the expressive power of **CTL**(X, EF) agrees with that obtained by Cohen, Perrin and Pin in [6]. Moreover, in this case tree language a L is in **CTL**(X, EF) iff its minimal automaton $\mathbb{A}$ satisfies the following condition: Whenever $a,b \in A$, $a \neq b$ are in the same transitivity class of $\mathbb{A}^-$ and $p(a) = a$, $p(b) = b$ for some term $p \in T_{\Sigma^-}(X_1)$, then $p = x_1$. However, when R contains an integer > 1, this condition is *not* equivalent to the one in Theorem 3.4.1.

Corollary 3.4.4 *A tree language $L \subseteq U_\Sigma$ of rank type R is in $\mathbf{CTL}^u(\mathrm{X}, \mathrm{EF})$ iff its minimal automaton belongs to $\mathbf{W}^+$. It is decidable for a regular unordered tree language whether it belongs to $\mathbf{CTL}^u(\mathrm{X}, \mathrm{EF})$.*

Bibliography

[1] J. Almeida, On pseudovarieties, varieties of languages, filters of congruences, pseudoidentities and related topics, *Algebra Universalis*, 27(1990), 333–350.

[2] J. Almeida, *Finite Semigroups and Universal Algebra*, World Scientific, 1994.

[3] A. Baziramwabo, P. McKenzie and D. Therien, Modular temporal logic, in: *Proc. 1999 IEEE Conference LICS, Trento, Italy.*

[4] M. Ben-Ari, Z. Manna and A. Pnueli, The temporal logic of branching time, in: *Proc. Symp. Principles of Programming Languages*, 1981, 164–176.

[5] M. Bojanczyk and I. Walukiewicz, Characterising EF and EX tree logics, *proc. CONCUR 2004*, to appear.

[6] J. Cohen, D. Perrin and J.-E. Pin, On the expressive power of temporal logic, *J. Computer and System Sciences*, 46(1993), 271–294.

[7] S. Eilenberg, *Automata, Languages, and Machines*, vol. A and B, Academic Press, 1974 and 1976.

[8] Z. Ésik, Definite tree automata and their cascade compositions, *Publ. Math.*, 48(1996), 243–262.

[9] Z. Ésik, A variety theorem for trees and theories, *Publ. Math.*, 54(1999), 711–762.

[10] Z. Ésik and P. Weil, On logically defined recognizable tree languages, in: *Proc. FST&TCS 03, Mumbai*, LNCS 2914, Springer, 2003, 195–206.

[11] D. Gabbay, A. Pnueli, S. Shelah and J. Stavi, On the temporal analysis of fairness, in: *Proc. 7th ACM Symp. Principles of Programming Langauges*, ACM, 1980, 163–173.

[12] F. Gécseg and M. Steinby, *Tree Automata*, Akadémiai Kiadó, 1984.

[13] G. Grätzer, *Universal Algebra*, 2nd ed., Springer, 1979.

[14] U. Heuter, Definite tree languages, *Bulletin of the EATCS*, 35(1988), 137–142.

[15] J. A. Kamp, Tense logic and the theory of linear order, Ph. D. Thesis, UCLA, 1968.

[16] R. McNaughton and S. Papert, *Counter-Free Automata*, MIT Press, 1971.

[17] M. Nivat and A. Podelski, Definite tree languages (cont'd), *Bulletin of the EATCS*, 38(1989), 186–190.

[18] G. Ricci, Cascades of tree-automata and computations in universal algebras. *Math. Systems Theory*, 7(1973), 201–218.

[19] K. Schneider, *Verification of Reactive Systems*, Springer, 2004.

[20] M. Steinby, A theory of tree language varieties, in: *Tree Automata and Languages*, North-Holland, Amsterdam, 1992, 57–81.

[21] M. Steinby, General varieties of tree languages, *Theoret. Comput. Sci.*, 205(1998), 1–43.

[22] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity*, Birkhauser, 1994.

[23] H. Straubing, D. Therien and W. Thomas, Regular languages defined with generalized quantifiers, *Information and Computation*, 118(1995), 289–301.

[24] J. W. Thatcher and J. B. Wright, Generalized finite automata theory with an application to a decision problem of second-order logic. *Math. Systems Theory*, 2(1968), 57–81.

[25] D. Therien and Th. Wilke, Temporal logic and semidirect products: An effective characterization of the until hierarchy, *DIMACS TR-96-28*, 1996.

[26] P. Wolper, Temporal logic can be more expressive, *Information and Control*, 56(1983), 72–99.