Logic and tree automata

Abstract of the Ph.D. thesis

by

Szabolcs Iván

Supervisor

Zoltán Ésik

Doctoral School in Mathematics and Computer Science

Department of Foundations of Computer Science

University of Szeged, Szeged, Hungary

2008.

Introduction

The characterization of the expressive power of various logics on words, trees and other structures (the classical logics, e.g. first- or second-order logic as well as several kinds of temporal logics) has received a lot of attention. The classical case, when one deals with (finite or infinite) words, is mostly well understood.

Regarding the logic FO($<$), i.e., first-order logic equipped with the total ordering of the positions, a classical result of McNaughton and Papert [23] in conjunction with a result of Schützenberger [33] states that a language of finite words is first-order definable if and only if its syntactic monoid is a finite aperiodic monoid, or equivalently, its minimal automaton is a finite counter-free automaton. Furthermore, this class of languages also coincides with the class of languages that are definable in the linear temporal logic LTL [21].

A similar aperiodicity condition characterizes the expressive power of first-order logic and of LTL on infinite words, cf. [24, 25]. It follows from these results that it is decidable whether a regular language of finite or infinite words is first-order definable. From the Krohn-Rhodes theorem [4], it is known that a finite monoid is aperiodic if and only if it belongs to the least pseudovariety of finite monoids containing a certain 3-element monoid U_2 which is closed under the wreath product, or equivalently, to the least pseudovariety containing a certain 2-element monoid U_1 which is closed under the block product [34].

When formulated on the domain of (say, finite, ranked and ordered) trees, most of the above questions are still open. The decidability status of the definability problem of first-order logic on finite trees, equipped with the descendant relation, or with the descendant relation and all the successor relations, denoted FO($<$) and FO($<, S_i$), respectively, has been a long standing open problem, cf. [20, 28, 29, 38]. The case of FO(S_i), i.e. when the logic is equipped only with the successor relations, has been solved recently in [2].

Aperiodicity can be generalized to trees in several different ways. One of them was studied in [20, 38] and shown to be a necessary but not a sufficient condition of first-order definability. In [11] we have defined an infinite hierarchy of aperiodicity notions and studied their relations to logical definability.

In [14], it was shown that a language of (ranked and ordered) finite trees is first-order definable if and only if its "syntactic preclone" belongs to the least pseudovariety of finitary preclones closed under the "block product" of preclones which is generated by the preclone canonically associated to a simple two-element algebra. Thus decidability of first-order definability reduces to the decidability of membership in that specific pseudovariety of finitary preclones. However the question whether membership in that pseudovariety of finitary preclones is decidable is left open in [14].

A problem related to first-order definability is the characterization of the expressive power of CTL and CTL* [1, 5, 32]: What tree languages can be defined in CTL, or in CTL*? These logics are both generalizations of LTL over words (although a third generalization, also called LTL is incomparable with CTL). It is shown in [18] that for certain types of trees first-order definability is equivalent to definability in CTL*. In [8, 9], a CTL-like logic FTL($\mathcal{L}$) was associated to each class $\mathcal{L}$ of (regular) languages of finite (ranked and ordered) trees. As a main result, it was shown that when any quotient of each language in $\mathcal{L}$ is definable in FTL($\mathcal{L}$), and when the "next modalities are expressible", then a tree language is definable in FTL($\mathcal{L}$) if and only if its minimal tree automaton belongs to the least pseudovariety of finite tree automata containing the minimal automata of the languages in $\mathcal{L}$ and the finite definite tree automata [6, 19], which is closed under the cascade product. Cascade products of finite tree automata were studied in [6, 10, 15, 16, 31]. This notion is closely related to the wreath product of clones defined in [39].

In [12] we removed from the above mentioned result the assumption that the next modalities are expressible. This was achieved by the introduction of a special case of the cascade product of tree automata that we called the Moore product. We have shown that under the assumption that any quotient of each language in $\mathcal{L}$ (where $\mathcal{L}$ is a class of regular tree languages) is definable in FTL($\mathcal{L}$), a tree language is definable in FTL($\mathcal{L}$) if and only if it is regular and its minimal tree automaton belongs to the least pseudovariety of finite tree automata containing the finite 1-definite tree automata and the minimal tree automata of the languages in $\mathcal{L}$ which is closed under the Moore product. An alternative formulation of this result is as follows. Let $\mathbf{K}$ denote a class of finite tree automata, and let FTL($\mathbf{K}$) denote the logic FTL($\mathcal{L}$), where $\mathcal{L}$ is the class of (regular) tree languages recognizable by the tree automata in $\mathbf{K}$. Then a tree language is definable in FTL($\mathbf{K}$) if and only if its minimal tree automaton belongs to the least pseudovariety of finite tree automata containing $\mathbf{K}$ and the finite 1-definite tree automata, which is closed under the Moore product. We call pseudovarieties of finite (connected) tree automata which are closed under the Moore product (connected) Moore pseudovarieties.

In [13], an application of the above theorem has been elaborated; we provided polynomial time decidable characterizations of several small connected Moore pseudovarieties. As a byproduct of these results, we derived decidability of the CTL fragments CTL(EF^+) and CTL(EF^*) equipped respectively only with the strict and non-strict version of the EF-modality of CTL. The decidability of the expressive power of CTL(EF^+) was already established by Bojańczyk and Walukiewicz in [3] using different methods; the decidability of the expressive power of the latter fragment was left open there.

Using a different approach, Z. Wu [41] also proved that CTL(EF^*) has a decidable definability problem. His approach is based on an Ehrenfeucht-Fraïssé type game, see e.g. [22]. One (yet unpublished) contribution of this thesis is that we defined for each class $\mathcal{L}$ of tree languages and number $n \geq 0$ of rounds a two-player game called the n-round $\mathcal{L}$-game, played on a pair of trees between two competing players, Spoiler and Duplicator. This game has the following property: two trees, s and t satisfy the same set of FTL($\mathcal{L}$)-formulas of depth at most n (denoted $s \equiv_{\mathcal{L}}^{n} t$) if and only if Duplicator has a winning strategy in the n-round $\mathcal{L}$-game, played on the pair (s, t) of trees. Standard arguments in finite model theory show that when $\mathcal{L}$ is a finite class of tree languages, then $\equiv_{\mathcal{L}}^{n}$ is an equivalence relation of finite index for each n. Thus, for any finite class $\mathcal{L}$ of tree languages it holds that a tree language L is definable in FTL($\mathcal{L}$) if and only if there exists some number n such that whenever s is a tree contained in L and t is a tree not in L, then Spoiler has a winning strategy for the n-round $\mathcal{L}$-game, played on the pair (s, t).

Preliminaries

A *rank type* R is a finite nonempty set of nonnegative integers; elements of R are called *arities*. A *signature* Σ of rank type R is a union $\bigcup_{n \in R} \Sigma_n$ of finite, pairwise disjoint, nonempty sets of symbols. We fix once and for all a countably infinite set $X = \{x_1, x_2, \dots\}$ of *variables*, assumed to be disjoint from any signature. The set $\{x_1, \dots, x_n\}$ is denoted X_n.

Given a signature Σ of rank type R, and an integer $n \geq 0$, the set $T_\Sigma(X_n)$ of ΣX_n*-trees* is the least set satisfying the following conditions:

1. any variable $x \in X_n$ is a ΣX_n-tree;
2. if $0 \leq k \in R$ is an arity, $\sigma \in \Sigma_k$ is a symbol and $t_1, \dots, t_k$ are ΣX_n-trees, then $\sigma(t_1, \dots, t_k)$ is a ΣX_n-tree.

A tree $t \in T_\Sigma(X_n) - X_n$ which is not a variable is called *proper.* T_Σ is written for $T_\Sigma(X_0)$, the set of variable-free Σ-trees, whereas the set CT_Σ of Σ*-contexts* is the subset of $T_\Sigma(X_1)$ which contains the trees in which x_1 occurs exactly once.

When $t \in T_\Sigma(X_n)$ is a tree and $\underline{t} = (t_1, \dots, t_n)$ is an n-tuple of ΣX_m-trees, then $t(\underline{t})$ denotes the tree resulting from t by substituting t_i for each occurrence of x_i, $i = 1, \dots, n$. If $t \in T_\Sigma(X_1)$, we write tt_1 for $t(t_1)$. When $t = t_1 t_2$, we say that t_2 is a *subtree* of t.

$\mathrm{Root}(t)$ denotes the root symbol of the tree t.

A (Σ-)*tree language* is any set of variable-free Σ-trees. When $\zeta \in CT_\Sigma$ is a Σ-context and $L \subseteq T_\Sigma$ is a tree language, the *quotient* of L with respect to ζ is the language $\zeta^{-1}(L) = \{t : \zeta t \in L\}$.

Given a signature Σ and a set A such that A, Σ and X are pairwise disjoint, the set $T_{\Sigma,A}(X_n)$ of $\Sigma A X_n$*-polynomial symbols* for any $n \geq 0$ is the least set satisfying the following conditions:

1. any $x \in X_n$ is a $\Sigma A X_n$-polynomial symbol;
2. any $a \in A$ is a $\Sigma A X_n$-polynomial symbol;
3. when $0 \leq k \in R$ is an arity, $\sigma \in \Sigma_k$ is a symbol and $p_1, \dots, p_k$ are $\Sigma A X_n$-polynomial symbols, then $\sigma(p_1, \dots, p_k)$ is also a $\Sigma A X_n$-polynomial symbol.

Polynomial symbols different from the variables are called *proper.* $T_{\Sigma,A}$ denotes the set of the variable-free ΣA-polynomial symbols, whereas the set $CT_{\Sigma,A}$ of ΣA*-polynomial contexts* consists of the $\Sigma A X_1$-polynomial symbols in which x_1 occurs exactly once. $\mathrm{Root}(p)$ denotes the root symbol of the polynomial symbol p.

When Σ and Δ are signatures having the same rank type R, and $h : \Sigma \to \Delta$ is a rank-preserving map, then h induces a *literal tree homomorphism* from $T_\Sigma(X)$ to $T_\Delta(X)$, also denoted h: we get $h(s)$ from $s \in T_\Sigma(X)$ by relabeling each node of s labeled by some symbol $\sigma \in \Sigma$ to the Δ-symbol $h(\sigma)$. When a node is labeled by a variable, its label does not change.

Given a signature Σ of some rank type R, a *(Σ-)tree automaton* $\mathbb{A} = (A, \Sigma)$ consists of a nonempty *state set* A and to each symbol $\sigma \in \Sigma_n$ an *elementary operation* $\sigma^{\mathbb{A}} : A^n \to A$ is associated. When A is finite, $\mathbb{A}$ is also called finite.

When $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Sigma)$ are Σ-tree automata with $B \subseteq A$ such that all elementary operations $\sigma^{\mathbb{B}}$ of $\mathbb{B}$ are the restrictions of the corresponding elementary operation $\sigma^{\mathbb{A}}$, we say that $\mathbb{B}$ is a *subautomaton* of $\mathbb{A}$.

When $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Sigma)$ are tree automata over the same signature Σ, then their *direct product* is the Σ-tree automaton $\mathbb{A} \times \mathbb{B} = (A \times B, \Sigma)$ having the state set $A \times B$ such that the elementary operations are interpreted componentwise.

The Δ-tree automaton $\mathbb{B} = (B, \Delta)$ is a *renaming* of the Σ-tree automaton $\mathbb{A} = (A, \Sigma)$ (here Σ and Δ both have the same rank type) if $B = A$ and each elementary operation of $\mathbb{B}$ is also an elementary operation of $\mathbb{A}$.

When $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Sigma)$ are Σ-tree automata, we call a mapping $h : A \to B$ a *homomorphism* if it is compatible with all elementary operations. The tree automaton $\mathbb{B}$ is called a *quotient* of the tree automaton $\mathbb{A}$ if there exists a surjective homomorphism from $\mathbb{A}$ to $\mathbb{B}$. When $\mathbb{B}$ is a quotient of some subautomaton of $\mathbb{A}$, $\mathbb{B}$ is called a *divisor* of $\mathbb{A}$.

Given a Σ-tree automaton $\mathbb{A} = (A, \Sigma)$, each ΣX_m-tree t induces a *term function* $t^{\mathbb{A}} : A^m \to A$, defined as usual. Term functions induced by proper trees are called *proper term functions*. Also, any $\Sigma A X_m$-polynomial symbol p induces a *polynomial function* $p^{\mathbb{A}} : A^m \to A$. Polynomial functions of $\mathbb{A}$ induced by (proper) ΣA-polynomial contexts are called (proper) translations of $\mathbb{A}$.

Thus, when $\mathbb{A} = (A, \Sigma)$ is a Σ-tree automaton, then each tree $t \in T_\Sigma$ induces a constant function $t^{\mathbb{A}}$. We identify this constant function with its value and write $t^{\mathbb{A}} = a$ when t induces the constant function with value a. Then, each set $A' \subseteq A$ determines a tree language $L_{\mathbb{A},A'}$ defined as $\{t \in T_\Sigma : t^{\mathbb{A}} \in A'\}$. We say that a tree language $L \subseteq T_\Sigma$ is *recognizable* by the tree automaton $\mathbb{A} = (A, \Sigma)$ if $L = L_{\mathbb{A},A'}$ for some set $A' \subseteq A$. A tree language is called *regular* if it is recognizable by some finite tree automaton. It is well known that for any tree language L there exists a *minimal tree automaton* $\mathbb{A}_L$, unique up to isomorphism, by which L is recognizable and which divides every tree automaton $\mathbb{B}$ by which L is recognizable.

A nonempty class $\mathbf{V}$ of finite tree automata is called a *pseudovariety of finite tree automata* if it is closed under taking direct products, homomorphic images, subautomata and renamings. For any class $\mathbf{K}$ of finite tree automata there exists a least pseudovariety $\langle \mathbf{K} \rangle$ containing $\mathbf{K}$. The class $\mathbf{K}$ of finite tree automata *generates* the pseudovariety $\langle \mathbf{K} \rangle$ of finite tree automata. When $\mathbf{K} = \{\mathbb{A}_1, \ldots, \mathbb{A}_n\}$ is a finite class, we write simply $\langle \mathbb{A}_1, \ldots, \mathbb{A}_n \rangle$ for $\langle \mathbf{K} \rangle$.

Since our logics will be defined on variable-free trees, in connection with logical definability it will suffice to deal with *connected* tree automata only. We assume that the rank type R contains 0 and at least one positive integer. The *connected part* of an automaton $\mathbb{A} = (A, \Sigma)$ is its subautomaton (A', Σ), where the set $A' = \{t^{\mathbb{A}} : t \in T_\Sigma\}$ consists of the accessible states of $\mathbb{A}$. The tree automaton $\mathbb{A}$ is *connected* if each state a of $\mathbb{A}$ is accessible. We define the *connected direct product* of two connected tree automata as the connected part of their direct product; *connected renaming* is defined analogously. A *pseudovariety of finite connected tree automata* is a class $\mathbf{V}^c$ of finite connected tree automata, which is closed under taking connected direct products, homomorphic images and connected renamings.

A strongly related notion is that of literal varieties. A nonempty class $\mathcal{L}$ of regular tree languages is a *literal variety of tree languages* if it is closed under taking quotients, Boolean operations and inverse literal tree homomorphisms. There exists an Eilenberg connection [4] between literal varieties of tree languages and pseudovarieties of finite connected tree automata: the mapping that associates to each pseudovariety $\mathbf{V}^c$ of finite connected tree automata the class $\mathcal{L}_{\mathbf{V}^c}$ which contains the tree languages that are recognizable by some member of $\mathbf{V}$ is an order isomorphism between the class of pseudovarieties of finite connected tree automata and the class of literal varieties of tree languages.

The logic FTL($\mathcal{L}$) and Moore pseudovarieties

In the first part of the thesis (Chapter 2), following [9], we defined the branching time temporal logic FTL($\mathcal{L}$) for each class $\mathcal{L}$ of tree languages. We provided an algebraic characterization of these logics when $\mathcal{L}$ consists of regular tree languages and possesses a natural property. We also showed an application of the characterization theorem and provided another, game-based description of the logic FTL($\mathcal{L}$) for any finite class $\mathcal{L}$.

The logic FTL($\mathcal{L}$)

The branching time future temporal logic FTL, introduced in [9], is defined as follows.

Syntax. Let Σ be a signature. The set of FTL-*formulas* over Σ is defined as the least set satisfying the following conditions:

1. For each $\sigma \in \Sigma$, p_σ is an (atomic) formula (of depth 0).

2. When φ_1 and φ_2 are formulas (having maximal depth d), $(\neg\varphi_1)$ and $(\varphi_1 \vee \varphi_2)$ are also formulas (of depth d).

3. If Δ is a signature, $L \subseteq T_\Delta$ is a tree language and for each $\delta \in \Delta$, φ_δ is a formula (having maximal depth d), then $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is also a formula (of depth $d+1$).

Semantics. Suppose that φ is a formula over Σ and $t \in T_\Sigma$ is a tree. We say that t *satisfies* φ, in notation $t \models \varphi$, if one of the following holds:

1. $\varphi = p_\sigma$ for some $\sigma \in \Sigma$ and $\mathrm{Root}(t) = \sigma$;

2. Boolean connectives are treated as usual;

3. $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ and the *characteristic tree* $\widehat{t} \in T_\Delta$ of t determined by the family $(\varphi_\delta)_{\delta\in\Delta}$ belongs to L. Here $\widehat{t}$ is a Δ-relabeling of t: a vertex v of t which is labeled by some $\sigma \in \Sigma_n$ is relabeled to $\delta \in \Delta_n$ if and only if one of the following holds:

- either $t|_v \models \varphi_\delta$ and δ is the first such element of Δ_n;
- or $t|_v \not\models \varphi_{\delta'}$ for any $\delta' \in \Delta_n$, and δ is the last element of Δ_n.

(We assume here that each signature Σ comes with a lexicographical ordering. However, it is known from [9] that the particular ordering is not important.)

An FTL-formula φ over Σ *defines* the tree language $L_\varphi = \{t \in T_\Sigma : t \models \varphi\}$.

We consider subsets of formulas associated to classes of tree languages. When $\mathcal{L}$ is a class of tree languages, we let FTL($\mathcal{L}$) denote the collection of formulas all of whose subformulas of the form $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ are such that L belongs to $\mathcal{L}$. We define $\mathbf{FTL}(\mathcal{L})$ to be the class of all languages definable by some FTL($\mathcal{L}$)-formula.

We define the equivalence relation $\equiv^n_{\mathcal{L}}$ on T_Σ for any signature Σ and integer $n \geq 0$: let $s \equiv^n_{\mathcal{L}} t$ hold for the trees $s, t \in T_\Sigma$ if s and t satisfy the same set of FTL($\mathcal{L}$)-formulas (over Σ) having depth at most n.

It was shown in [9] that the operator **FTL** preserves regularity and is a closure operator on classes of regular languages. Also, for any class $\mathcal{L}$ of regular tree languages, $\mathbf{FTL}(\mathcal{L})$ is closed under the Boolean operations and inverse literal homomorphisms. Moreover, $\mathbf{FTL}(\mathcal{L})$ is a literal variety of tree languages if and only if for each quotient L of a language in $\mathcal{L}$ it holds that $L \in \mathbf{FTL}(\mathcal{L})$.

Products of tree automata

Suppose $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are tree automata and γ is a family $(\gamma_n)_{n\in R}$ of functions, $\gamma_n : A^n \times \Sigma \to \Delta_n$ for each $n \in R$. Then the *cascade product* $\mathbb{A} \times_\gamma \mathbb{B}$ of $\mathbb{A}$ and $\mathbb{B}$ determined by γ is the tree automaton $(A \times B, \Sigma)$ with

$$\sigma^{\mathbb{A}\times_\gamma\mathbb{B}}\big((a_1, b_1), \ldots, (a_n, b_n)\big) \quad = \quad \big(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \delta^{\mathbb{B}}(b_1, \ldots, b_n)\big),$$

where $\delta = \gamma_n(a_1, \ldots, a_n, \sigma)$ for all $\sigma \in \Sigma_n$, $n \in R$ and $(a_1, b_1), \ldots, (a_n, b_n) \in A \times B$.

We call the above cascade product $\mathbb{A} \times_\gamma \mathbb{B}$ a *Moore product* if there exists a rank-preserving function $\alpha : A \times \Sigma \to \Delta$ (i.e. for any $\sigma \in \Sigma_n$, $n \in R$ and $a \in A$, $\alpha(a, \sigma) \in \Delta_n$ holds) with

$$\gamma_n(a_1, \ldots, a_n, \sigma) \quad = \quad \alpha(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \sigma)$$

for all $\sigma \in \Sigma_n$, $n \in R$ and $a_1, \ldots, a_n \in A$. Moreover, we call the above Moore product a *strict Moore product* if there is a function $\beta : A \times R \to \Delta$ with $\alpha(a, \sigma) = \beta(a, n)$, for all $a \in A$ and $\sigma \in \Sigma_n$.

Analogously to the connected direct product, the connected cascade product $\mathbb{A} \times_\gamma \mathbb{B}$ of the connected tree automata $\mathbb{A}$ and $\mathbb{B}$ determined by γ is the connected part of their cascade product determined by γ. Connected Moore and connected strict Moore products are defined similarly.

When $\mathbf{V}$ and $\mathbf{W}$ are pseudovarieties of finite tree automata, let $\mathbf{V} \times \mathbf{W}$ denote the pseudovariety of finite tree automata generated by all direct products $\mathbb{A} \times \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}$ and $\mathbb{B} \in \mathbf{W}$. Similarly, let $\mathbf{V} \times_M \mathbf{W}$, $\mathbf{V} \times_s \mathbf{W}$ and $\mathbf{V} \times_c \mathbf{W}$ respectively denote the pseudovariety of finite tree automata generated by all Moore, strict Moore and cascade products $\mathbb{A} \times_\alpha \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}$ and $\mathbb{B} \in \mathbf{W}$. These notions are also extended to pseudovarieties of finite *connected* tree automata: when $\mathbf{V}^c$ and $\mathbf{W}^c$ are pseudovarieties of finite connected tree automata, let $\mathbf{V}^c \times \mathbf{W}^c$ denote the pseudovariety of finite connected tree automata generated by all connected direct products $\mathbb{A} \times \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}^c$ and $\mathbb{B} \in \mathbf{W}^c$. The notations $\mathbf{V}^c \times_M \mathbf{W}^c$, $\mathbf{V}^c \times_s \mathbf{W}^c$ and $\mathbf{V}^c \times_c \mathbf{W}^c$ are defined analogously.

When $\mathbf{K}$ is any class of finite tree automata, let $\langle \mathbf{K} \rangle_s$, $\langle \mathbf{K} \rangle_M$, $\langle \mathbf{K} \rangle_c$ respectively denote the least pseudovariety of finite tree automata containing $\mathbf{K}$ which is closed under the strict Moore product, the Moore product, and the cascade product. We call these classes *strict Moore pseudovarieties*, *Moore pseudovarieties* and *cascade pseudovarieties*, respectively.

Analogously, when $\mathbf{K}^c$ is any class of finite connected tree automata, let $\langle \mathbf{K}^c \rangle_s$, $\langle \mathbf{K}^c \rangle_M$, $\langle \mathbf{K}^c \rangle_c$ respectively denote the least pseudovariety of connected finite tree automata containing $\mathbf{K}^c$ which is closed under the connected strict Moore product, the connected Moore product, and the connected cascade product. We call these pseudovarieties *connected strict Moore pseudovarieties*, *connected Moore pseudovarieties*, and *connected cascade pseudovarieties*, respectively.

Definite tree automata

We say that a Σ-tree automaton $\mathbb{A}$ is *k-definite*, for some $k \geq 1$, if for all ΣA-polynomial symbols p and q, if p and q agree up to depth $k-1$, then $p^{\mathbb{A}} = q^{\mathbb{A}}$. A tree automaton is *definite* if it is k-definite for some k. We let $\mathbf{D}_k$ denote the class of all finite k-definite tree automata and $\mathbf{D}$ the class of all finite definite tree automata. When $\mathbf{V}$ is a class of finite tree automata, let $\mathbf{V}^c$ denote the class of finite connected tree automata consisting of the connected parts of members of $\mathbf{V}$. Then, $\mathbf{D}_k^c$ is the class of all finite k-definite connected tree automata and $\mathbf{D}^c$ is the class of all finite definite connected tree automata.

The signature Bool contains exactly two symbols, $\uparrow_n$ and $\downarrow_n$ for each arity $n \in R$. The tree automaton $\mathbb{D}_0 = (\{0,1\}, \text{Bool})$ is defined as follows: for each $n \in R$, $\uparrow_n^{\mathbb{D}_0}$ is the constant function $\{0,1\}^n \to \{0,1\}$ with value 1, and similarly, $\downarrow_n^{\mathbb{D}_0}$ is the constant function $\{0,1\}^n \to \{0,1\}$ with value 0.

It was shown in [6] that $\mathbf{D} = \langle \mathbb{D}_0 \rangle_c$ is a cascade pseudovariety of finite tree automata generated by $\mathbb{D}_0$, thus $\langle \mathbf{D}_1 \rangle_c = \mathbf{D}$. But the class of all finite 1-definite tree automata is closed under the Moore product:

Proposition 2.1.11. $\mathbf{D}_1$ is a Moore pseudovariety of finite tree automata.

Correspondences between the products

We have shown several correspondences between the cascade, the Moore and the strict Moore products.

Corollary 2.2.15. For any Moore pseudovariety $\mathbf{V}$ of finite tree automata,

$$\langle \mathbf{D}_1 \cup \mathbf{V} \rangle_M = \mathbf{D}_1 \times \mathbf{V}.$$

COROLLARY 2.2.17. For any pseudovariety $\mathbf{V}$ of finite tree automata,

$$\mathbf{D}_1 \subseteq \mathbf{V} \quad \text{implies} \quad \langle \mathbf{V} \rangle_M = \langle \mathbf{V} \rangle_s.$$

COROLLARY 2.2.28. For any class $\mathbf{K}$ of finite tree automata,

$$\langle \mathbf{D}_2 \cup \mathbf{K} \rangle_M \quad = \quad \langle \mathbf{D} \cup \mathbf{K} \rangle_M \ = \ \langle \mathbf{D} \cup \mathbf{K} \rangle_c.$$

COROLLARY 2.2.29. The following conditions are equivalent for any class $\mathbf{V}$ of finite tree automata.

1. $\mathbf{V}$ is a Moore pseudovariety containing $\mathbf{D}_2$.
2. $\mathbf{V}$ is a Moore pseudovariety containing $\mathbf{D}$.
3. $\mathbf{V}$ is a cascade pseudovariety containing $\mathbf{D}$.

Definability and membership

Following [9], we also associate a logic $\mathrm{FTL}(\mathbf{K}^c)$ to each class $\mathbf{K}^c$ of finite connected tree automata. First let $\mathcal{L}_{\mathbf{K}^c}$ denote the class of regular tree languages recognizable by the members of $\mathbf{K}^c$. Then we let $\mathrm{FTL}(\mathbf{K}^c)$ be the logic $\mathrm{FTL}(\mathcal{L}_{\mathbf{K}^c})$ and define $\mathbf{FTL}(\mathbf{K}^c) = \mathbf{FTL}(\mathcal{L}_{\mathbf{K}^c})$. In [9] it was shown that for any class $\mathcal{L}$ of regular tree languages, $\mathbf{FTL}(\mathcal{L})$ is a literal variety if and only if $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathbf{K}^c)$ for some class $\mathbf{K}^c$ of finite connected tree automata. We proved the following characterization theorem:

THEOREM 2.3.4. For any class $\mathbf{K}^c$ of finite connected tree automata, a language L belongs to $\mathbf{FTL}(\mathbf{K}^c)$ if and only if its minimal tree automaton $\mathbb{A}_L$ belongs to $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$ if and only if L is recognizable by an automaton in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$.

COROLLARY 2.3.5. Assume that $\mathcal{L}$ is a class of regular tree languages such that any quotient of each language in $\mathcal{L}$ belongs to $\mathbf{FTL}(\mathcal{L})$. Then a language L is in $\mathbf{FTL}(\mathcal{L})$ if and only if $\mathbb{A}_L$ is contained in the least connected (strict) Moore pseudovariety of finite connected tree automata containing $\mathbf{D}_1^c$ and the minimal tree automata of the languages in $\mathcal{L}$.

COROLLARY 2.3.5. For every connected Moore pseudovariety $\mathbf{V}^c$ of finite connected tree automata containing $\mathbf{D}_1^c$ it holds that $\mathcal{L}_{\mathbf{V}^c} = \mathbf{FTL}(\mathbf{V}^c)$. Moreover, the map $\mathbf{V}^c \mapsto \mathbf{FTL}(\mathbf{V}^c)$ is an order isomorphism between the lattice of all connected Moore pseudovarieties of finite connected tree automata containing $\mathbf{D}_1^c$ and the lattice of all literal varieties of regular tree languages $\mathcal{V}$ with $\mathbf{FTL}(\mathcal{V}) = \mathcal{V}$.

Application

We call a property $\mathcal{P}$ of tree automata a *Moore property* if the class of all finite tree automata having $\mathcal{P}$ forms a Moore pseudovariety.

We say that a Σ-tree automaton $\mathbb{A}$ is *commutative* if it satisfies all equations

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

for all $\sigma \in \Sigma_n$, $n \in R$, $n > 0$ and for all permutations π of the set $\{1, \ldots, n\}$.

We say that a Σ-tree automaton $\mathbb{A}$ is *stutter invariant* if

$$\sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, \sigma^{\mathbb{A}}(a_1, \ldots, a_n)) \quad = \quad \sigma^{\mathbb{A}}(a_1, \ldots, a_n)$$

for all $\sigma \in \Sigma_n$, $n > 0$, $a_1, \ldots, a_n \in A$.

When $\mathbb{A}$ is a Σ-tree automaton, $\preceq_{\mathbb{A}}$ denotes the *accessibility relation* of $\mathbb{A}$, that is, $a \preceq_{\mathbb{A}} b$ holds for $a, b \in A$ if for some ΣA-polynomial context p we have $p^{\mathbb{A}}(a) = b$. Clearly, $\preceq_{\mathbb{A}}$ is a reflexive and transitive relation. The relation $\sim_{\mathbb{A}}$ over A is defined by $a \sim_{\mathbb{A}} b$ if $a \preceq_{\mathbb{A}} b$ and $b \preceq_{\mathbb{A}} a$ both hold; then, $\sim_{\mathbb{A}}$ is an equivalence relation.

A tree automaton $\mathbb{A}$ is called *monotone* if $\preceq_{\mathbb{A}}$ is a partial order.

We call a tree automaton $\mathbb{A} = (A, \Sigma)$ *maximal dependent* if for any function symbol $\sigma \in \Sigma_n$ and $a_1, \ldots, a_{n-1}, a_n, a'_n \in A$ with $a_n \preceq_{\mathbb{A}} a_i$ and $a'_n \preceq_{\mathbb{A}} a_j$ for some $1 \leq i, j \leq n-1$, we have $\sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, a_n) = \sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, a'_n)$.

We call a tree automaton $\mathbb{A} = (A, \Sigma)$ *component dependent* if for any function symbol $\sigma \in \Sigma_n$ and states $a_1 \sim_{\mathbb{A}} a'_1, \ldots, a_n \sim_{\mathbb{A}} a'_n \in A$ it holds that $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) = \sigma^{\mathbb{A}}(a'_1, \ldots, a'_n)$.

We call a tree automaton $\mathbb{A} = (A, \Sigma)$ *componentwise unique* if for any $a, b \in A$ and proper ΣA-polynomial contexts p, q such that $\mathrm{Root}(p) = \mathrm{Root}(q)$, $p^{\mathbb{A}}(a) = b$ and $q^{\mathbb{A}}(b) = a$, it holds that $a = b$.

We proved that each of the above defined properties are Moore properties; let **Com**, **Stu**, **Mon**, **MaxDep**, **CompDep** and **CompUnique** denote the class of all finite commutative, stutter invariant, monotone, maximal dependent, component dependent and componentwise unique automata, respectively. Thus, each of these classes is a Moore pseudovariety.

The following relations hold:

Proposition 2.4.18.

$$\mathbf{CompDep} \cap \mathbf{Com} \cap \mathbf{MaxDep} \ \subseteq \ \mathbf{CompUnique}.$$

Corollary 2.4.39.

$$\mathbf{Mon} \times \mathbf{D}_1 \quad = \quad \mathbf{CompDep} \cap \mathbf{CompUnique}.$$

Let $L_{\mathrm{EF}^+} \subseteq T_{\mathrm{Bool}}$ denote the regular tree language of those trees in T_{Bool} having at least one non-root vertex labeled in $\{\uparrow_n : n \in R\}$ and let $L_{\mathrm{EF}^*} \subseteq T_{\mathrm{Bool}}$ consist of all trees in T_{Bool} with at least one vertex labeled in $\{\uparrow_n : n \in R\}$. Let Σ be a signature and φ a fixed formula over Σ. If $(\varphi_\delta)_{\delta \in \mathrm{Bool}}$ is such that $\varphi_{\uparrow_n} = \varphi$, for all $n \in R$, then $t \models L_{\mathrm{EF}^+}(\delta \mapsto \varphi_\delta)_{\delta \in \mathrm{Bool}}$ if and only if $s \models \varphi$ for some proper subtree s of t. And $t \models L_{\mathrm{EF}^*}(\delta \mapsto \varphi_\delta)_{\delta \in \mathrm{Bool}}$ if and only if some subtree of t satisfies φ. Thus, the modal operators corresponding to these languages are closely related to the strict and non-strict EF modalities of CTL, cf. [32].

Let $\mathbb{E}^+_{\mathrm{EF}}$ and $\mathbb{E}^*_{\mathrm{EF}}$ denote the minimal tree automata of L_{EF^+} and L_{EF^*}, respectively. We characterized the Moore pseudovarieties $\langle \mathbb{E}^+_{\mathrm{EF}} \rangle_M$, $\langle \mathbb{E}^*_{\mathrm{EF}} \rangle_M$, $\langle \mathbb{E}^+_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M$ and $\langle \mathbb{E}^*_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M$ in terms of the above (P-time decidable) properties:

Theorem 2.4.30.

$$\langle \mathbb{E}^{+}_{\mathrm{EF}} \rangle_M \quad = \quad \mathbf{Mon}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c.$$

Theorem 2.4.36.

$$\langle \mathbb{E}^{*}_{\mathrm{EF}} \rangle_M \quad = \quad \mathbf{Mon}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c \cap \mathbf{Stu}^c = \langle \mathbb{E}^{+}_{\mathrm{EF}} \rangle_M \cap \mathbf{Stu}^c.$$

Theorem 2.4.40. For any pseudovariety $\mathbf{V}$ of finite tree automata,

$$\mathbf{D}_1 \subseteq \mathbf{V} \subseteq \mathbf{Mon} \times \mathbf{D}_1 \quad \text{implies} \quad (\mathbf{Mon} \cap \mathbf{V}) \times \mathbf{D}_1 = \mathbf{V}.$$

Using the above facts, we characterized $\langle \mathbb{E}^{+}_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M$ and $\langle \mathbb{E}^{*}_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M$ effectively:

Corollary 2.4.41. The following equalities hold:

i) $\langle \mathbb{E}^{+}_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M = \langle \mathbb{E}^{+}_{\mathrm{EF}} \rangle_M \times \mathbf{D}_1^c = \mathbf{CompDep}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c$;

ii) $\langle \mathbb{E}^{*}_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M = \langle \mathbb{E}^{*}_{\mathrm{EF}} \rangle_M \times \mathbf{D}_1^c = \mathbf{CompDep}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c \cap \mathbf{Stu}^c$.

As a byproduct, we obtained a characterization of two fragments of the temporal logic CTL:

Corollary 2.4.42. The following hold for any tree language L:

i) L is in $\mathbf{CTL}(\mathrm{EF}^+)$ if and only if its minimal tree automaton $\mathbb{A}_L$ is contained in $\mathbf{CompDep}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c$;

ii) L is in $\mathbf{CTL}(\mathrm{EF}^*)$ if and only if its minimal tree automaton $\mathbb{A}_L$ is contained in $\mathbf{CompDep}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c \cap \mathbf{Stu}^c$.

Since membership in the above varieties is clearly decidable, it follows that it is decidable for a given regular tree language L (given by its minimal tree automaton) whether L is in $\mathbf{CTL}(\mathrm{EF}^+)$, or in $\mathbf{CTL}(\mathrm{EF}^*)$; moreover, both problems are decidable in polynomial time. For the logic $\mathbf{CTL}(\mathrm{EF}^+)$ this was already shown in [3] using different methods.

Ehrenfeucht-Fraïssé type games

We defined the so-called n-round $\mathcal{L}$-game for any class $\mathcal{L}$ of tree languages and number $n \geq 0$ of rounds and showed its correspondence to the logic FTL($\mathcal{L}$). The game is played on two trees, between two players, Spoiler and Duplicator.

Let $\mathcal{L}$ be a class of tree languages, $n \geq 0$ an integer, Σ a signature and let $t_0, t_1 \in T_\Sigma$ be trees. The *n-round $\mathcal{L}$-game* on (t_0, t_1) is played as follows.

1. If $\mathrm{Root}(t_0) \neq \mathrm{Root}(t_1)$, Spoiler wins. Otherwise, Step 2 follows.
2. If $n = 0$, Duplicator wins. Otherwise, Step 3 follows.

3. Spoiler chooses a language $L \in \mathcal{L}$ over a signature Δ, an index $i \in \{0, 1\}$, a Δ-relabeling $\widehat{t_i} \in L$ of t_i and a Δ-relabeling $\widehat{t_j} \notin L$ of t_j, where $j = 1 - i$ is the other index. If he cannot do so, Duplicator wins. Otherwise, Step 4 follows.

4. Duplicator chooses two nodes, x and y of the pair (t_0, t_1) of trees, having different labels according to the relabelings $\widehat{t_i}$. If he cannot do so, Spoiler wins. Otherwise, an $(n-1)$-round $\mathcal{L}$-game is played on the subtrees rooted in x and y. Whoever wins the subgame, wins also the game.

Clearly, for any class $\mathcal{L}$ of languages, number $n \geq 0$ of rounds and trees (s, t), one of the players has a winning strategy for the n-round $\mathcal{L}$-game on (s, t). We say that the player having a winning strategy *wins* the game. Let $s \sim^n_{\mathcal{L}} t$ denote that Duplicator wins the n-round $\mathcal{L}$-game on (s, t). We proved the following correspondence between $\sim^n_{\mathcal{L}}$ and the logic FTL($\mathcal{L}$):

Corollary 2.5.3. For any class $\mathcal{L}$ of tree languages and integer $n \geq 0$, the relations $\sim^n_{\mathcal{L}}$ and $\equiv^n_{\mathcal{L}}$ coincide.

Corollary 2.5.4. The following are equivalent for any finite class $\mathcal{L}$ of tree languages and any language L:

i) L is definable in FTL($\mathcal{L}$);

ii) there exists an integer $n \geq 0$ such that Spoiler wins the n-round $\mathcal{L}$-game on (s, t), whenever $s \in L$ and $t \notin L$ both hold.

Aperiodicity

In the second part of the thesis (Chapter 3) we introduced and studied several notions of aperiodicity of finite tree automata.

The notion of n-aperiodicity

Let $\mathbb{A} = (A, \Sigma)$ be a finite tree automaton. Extending the notion of term functions, we let each m-tuple $\underline{t} = (t_1, \ldots, t_m)$ of trees $t_i \in T_\Sigma(X_n)$ induce a vector-valued term function $\underline{t}^{\mathbb{A}} = \langle t_1^{\mathbb{A}}, \ldots, t_m^{\mathbb{A}} \rangle : A^n \to A^m$, which is the *target tupling* of the m functions $t_i^{\mathbb{A}} : A^n \to A$, $1 \leq i \leq m$. When each $t_i^{\mathbb{A}}$ is proper, $\underline{t}^{\mathbb{A}}$ is also called proper. It is clear that for each $n \geq 1$, the proper term functions $A^n \to A^n$ form a semigroup, denoted $S_n(\mathbb{A})$ which is finite if $\mathbb{A}$ is a finite tree automaton. In this semigroup, product is function composition. The subsemigroup of $S_1(\mathbb{A})$ consisting of the term functions induced by the proper contexts will be denoted $C(\mathbb{A})$ if the set of proper Σ-contexts is not empty; otherwise let $C(\mathbb{A})$ be a trivial semigroup.

Recall that a finite semigroup S is called aperiodic if for some $k > 0$, $s^k = s^{k+1}$ holds for every element s of S. We call a finite Σ-tree automaton $\mathbb{A}$ *n-aperiodic* for some $n \geq 1$ if the semigroup $S_n(\mathbb{A})$ is aperiodic. By extension, we call $\mathbb{A}$ *strongly aperiodic* if it is

n-aperiodic for each $n \geq 1$. We call a finite Σ-tree automaton $\mathbb{A}$ *context aperiodic* if $C(\mathbb{A})$ is aperiodic.

We proved the following characterization of strong aperiodicity:

Corollary 3.1.11. A finite tree automaton $\mathbb{A}$ is strongly aperiodic if and only if for each $n > 0$, no proper term function $A^n \to A^n$ has two or more different fixed points.

Generalized cascade product

Let $\mathbb{A}$ be a Σ-tree automaton, $\mathbb{B}$ a Δ-tree automaton and γ a family $(\gamma_n)_{n\in R}$ of functions, where for each $n \in R$, γ_n maps $A^n \times \Sigma$ to the set of all *proper* ΔX_n-trees. Then the *generalized cascade product* $\mathbb{A} \times_\gamma \mathbb{B}$ is defined as the tree automaton on the set $A \times B$ such that for any $\sigma \in \Sigma_n$ and $(a_1, b_1), \ldots, (a_n, b_n) \in A \times B$,

$$\sigma^{\mathbb{A}\times_\gamma\mathbb{B}}((a_1, b_1), \ldots, (a_n, b_n)) \quad = \quad (\sigma^{\mathbb{A}}(a_1, \ldots, a_n), t^{\mathbb{B}}(b_1, \ldots, b_n))$$

where $t = \gamma_n(a_1, \ldots, a_n, \sigma)$.

We say that a nonempty class of finite tree automata is a *generalized cascade pseudovariety* of finite tree automata if it is closed under taking subautomata, homomorphic images, renamings and the generalized cascade product.

We proved the following two algebraic closure properties:

Theorem 3.2.8. For any integer $n > 0$, $\mathbf{SAper}_n$ is a generalized cascade pseudovariety of finite tree automata. Hence, **SAper** is also a generalized cascade pseudovariety.

Theorem 3.2.9. **CAper** is a cascade pseudovariety of finite tree automata.

Strict containments and complexity

It is clear that

$$\mathbf{CAper} \supseteq \mathbf{SAper}_1 \supseteq \mathbf{SAper}_2 \supseteq \ldots \supseteq \mathbf{SAper} \supseteq \mathbf{D}$$

is a decreasing chain. It turned out that this hierarchy collapses if $R = \{0, 1\}$ or $R = \{1\}$.

Proposition 3.3.1. When $R = \{1\}$ or $R = \{0, 1\}$, it holds that

$$\mathbf{CAper} = \mathbf{SAper}_1 \supset \mathbf{SAper}_2 = \mathbf{SAper} = \mathbf{D}.$$

However, when R contains an integer $k > 1$, the hierarchy is proper:

Proposition 3.3.2. If R contains an integer $k > 1$ then $\mathbf{SAper}_1 \subset \mathbf{CAper}$.

Proposition 3.3.3. Suppose that R contains an integer $k > 1$. Then for each $n > 1$ there exists an $(n-1)$-aperiodic finite tree automaton which is not n-aperiodic.

Proposition 3.3.4. If R contains an integer $k > 1$, then $\mathbf{D} \subset \mathbf{SAper}$.

We also studied the complexity of the membership problems of the classes $\mathbf{SAper}_n$ and **SAper**. We obtained the following results:

THEOREM 3.4.3. It is decidable in polynomial time whether a given finite tree automaton $\mathbb{A}$ is strongly aperiodic.

THEOREM 3.4.4. For each fixed n, it is PSPACE-hard to decide, given a finite tree automaton $\mathbb{A}$, whether $\mathbb{A}$ is n-aperiodic.

Aperiodicity and logic

We related the aperiodicity notions to the temporal logic CTL and to first-order logic. The results were the following:

1. If a tree language is definable in CTL, then its minimal tree automaton is 1-aperiodic.
2. There exists a tree language definable in CTL whose minimal automaton is not 2-aperiodic.
3. There exists a tree language definable in FO($<$) whose minimal automaton is not 1-aperiodic.
4. There exists a regular tree language whose minimal automaton is 1-aperiodic but which is not definable in FO($<, S_i$).

Aperiodicity for polynomials

We also introduced a slight modification of the aperiodicity notions. When $\mathbb{A} = (A, \Sigma)$ is a finite tree automaton, and $m, n > 0$, each m-tuple $\underline{p} = (p_1, \ldots, p_m)$ of ΣAX_n-polynomial symbols induces a vector-valued polynomial function $\underline{p}^{\mathbb{A}} = \langle p_1^{\mathbb{A}}, \ldots, p_m^{\mathbb{A}} \rangle : A^n \to A^m$, which is the *target tupling* of the functions $p_1^{\mathbb{A}}, \ldots, p_m^{\mathbb{A}}$. When each p_i is proper, $\underline{p}^{\mathbb{A}}$ is also called proper. Clearly, for each $n > 0$ and finite tree automaton $\mathbb{A}$, the proper polynomial functions $A^n \to A^n$ form a semigroup $S_n^{(p)}(\mathbb{A})$, with function composition as product. For each $n > 0$ we define $\mathbf{SAper}_n^{(p)}$ as the class of finite tree automata $\mathbb{A}$ such that $S_n^{(p)}(\mathbb{A})$ is an aperiodic semigroup. $\mathbf{SAper}^{(p)}$ and $\mathbf{CAper}^{(p)}$ are defined analogously.

It turned out the hierarchy collapses in this polynomial setting:

THEOREM 3.7.4.

$$\mathbf{D} = \mathbf{SAper}^{(p)} = \mathbf{SAper}_2^{(p)} \subset \mathbf{SAper}_1^{(p)} \subset \mathbf{CAper}^{(p)}.$$

Since definiteness is decidable in polynomial time, this characterization of $\mathbf{SAper}^{(p)}$ is effective. Finally, we also studied the relationship of $\mathbf{SAper}_1^{(p)}$ to the other aperiodicity classes.

COROLLARY 3.7.7. The class $\mathbf{SAper}_1^{(p)}$ nontrivially intersects $\mathbf{SAper}$ and each member of the hierarchy $\mathbf{SAper}_n$ for $n \geq 2$.

References

[1] M. Ben-Ari, Z. Manna and A. Pnueli. The temporal logic of branching time. *Acta Informatica*, 20:207–226, 1983.

[2] M. Benedikt and L. Segoufin. Regular tree languages definable in FO. In *STACS 2005*, LNCS 2404, pages 327–339, Springer-Verlag, 2005.

[3] M. Bojańczyk and I. Walukiewicz. Characterising EF and EX tree logics. In *proc. CONCUR 2004*, LNCS 3170, pages 131–145, Springer-Verlag, 2004.

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

[5] E. A. Emerson and J. Y. Halpern. "Sometimes" and "not never" revisited: on branching versus linear time temporal logic. *J. Assoc. Comput. Mach.* 33:151–178, 1986.

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

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

[8] Z. Ésik. An algebraic characterization of temporal logics on finite trees. Parts I, II, III. In *1st International Conference on Algebraic Informatics, 2005*, pages 53–77, 79–99, 101–110, Aristotle Univ. Thessaloniki, Thessaloniki, 2005.

[9] Z. Ésik. Characterizing CTL-like logics on finite trees. *Theoretical Computer Science*, 356:136–152, 2006.

[10] Z. Ésik and F. Gécseg. Type independent varieties and metric equivalence of tree automata. *Fundamenta Informaticae*, 2:205-216, 1986.

[11] Z. Ésik and Sz. Iván. Aperiodicity in tree automata. In *2nd International Conference on Algebraic Informatics, 2007*, LNCS 4728, Springer, p. 189–207, 2007.

[12] Z. Ésik and Sz. Iván. Products of tree automata with an application to temporal logic. *Fundamenta Informaticae* 82:61–82, 2008. (to appear)

[13] Z. Ésik and Sz. Iván. Some varieties of finite tree automata related to restricted temporal logic. *Fundamenta Informaticae* 82:83–103, 2008. (to appear)

[14] Z. Ésik and P. Weil. On logically defined recognizable tree languages. In *FST&TCS 03, Mumbai*, LNCS 2914, pages 195–206, Springer-Verlag, 2003.

[15] Z. Ésik and P. Weil. Algebraic recognizability of regular tree languages. *Theoret. Comput. Sci.*, 340:291–321, 2005.

[16] F. Gécseg and B. Imreh. On isomorphic representations of monotone tree and nondeterministic tree automata. *Words, Semigroups and Transductions 2001*, pages 141–154, 2001.

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

[18] T. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *ICALP 1987, Karlsruhe*, pages 269–279, LNCS 267, Springer-Verlag, 1987.

[19] U. Heuter. Definite tree languages. *Bulletin of the EATCS*, 35:137–142. 1988.

[20] U. Heuter. First-order properties of trees, star-free expressions, and aperiodicity. *RAIRO Inform. Théor. Appl.*, 25:125–145, 1991.

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

[22] L. Libkin. *Elements of finite model theory.* Springer, 2004.

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

[24] D. Perrin and J.-E. Pin. First-order logic and star-free sets. *J. Comput. System Sci.*, 32:393–406, 1986.

[25] D. Perrin and J.-E. Pin. *Infinite Words*, Pure and Applied Mathematics Vol 141, Elsevier, 2004.

[26] J.-E. Pin. *Varieties of Formal Languages.* Plenum Publishing Corp., New York, 1986.

[27] J.-E. Pin. The expressive power of existential first-order sentences of B uchis sequential calculus. In *International Colloquium on Automata, Languages and Programming*, LNCS vol. 1099, p. 300–311, 1996.

[28] A. Potthoff. First order logic on finite trees. In *TAPSOFT '95*, pages 125–139, LNCS 915, Springer-Verlag, 1995.

[29] A. Potthoff. Modulo-counting quantifiers over finite trees. *Theoretical Computer Science*, 126:97–112, 1994.

[30] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. AMS*, 141:1–35, 1969.

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

[32] K. Schneider. *Verification of Reactive Systems.* Springer-Verlag, 2004.

[33] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control* 8:190–194, 1965.

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

[35] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2:57–82, 1968.

[36] D. Thérien and A. Weiss. Graph congruences and wreath products. *Journal of Pure and Applied Algebra*, 36(2):205 215, 1985.

[37] W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25:360–375, 1982.

[38] W. Thomas. Logical aspects in the study of tree languages. In *Ninth Colloquium on Trees in Algebra and Programming, Bordeaux, 1984*, pages 31–49, Cambridge Univ. Press, Cambridge, 1984.

[39] J. VanderWerf. Wreath products of algebras: generalizing the Krohn-Rhodes theorem to arbitrary algebras. *Semigroup Forum*, 52:93–100, 1996.

[40] Th. Wilke. Classifying discrete temporal properties. In *STACS 99, Trier*, pages 32–46, LNCS 1563, Springer-Verlag, Berlin, 1999.

[41] Zh. Wu. A note on the characterization of TL(EF). *Information Processing Letters*, 102:48–54, 2007.