Logic and tree automata

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.

Contents

Acknowledgments

I thank Zoltán Ésik for the guidance and for his valuable time over the years; I can only hope that our joint work will be continued.

I am also thankful to Magnus Steinby for his hospitality during my ten-day visit to the University of Turku.

Introduction

This thesis is concerned with the definability problem of several classes of logics on finite trees, viewed mostly from the algebraic point of view.

The first chapter contains all the notations and preliminary facts we will use throughout the thesis, as well as a brief summary of the previous results from the involved field.

In the second chapter we recall the operator FTL introduced in [15], see also [14], Part I, which associates to each class $\mathcal{L}$ of tree languages a logic FTL($\mathcal{L}$); this logic is a branching time future temporal logic. This is done in Section 2.1. The definability problem of such a logic FTL($\mathcal{L}$) is the following: given a (regular) tree language, is it definable in FTL($\mathcal{L}$)? This problem is attacked by two different methods, neither of them yielding an immediate decidability result, but some kind of alternative characterization from a different point of view.

In Sections 2.2 and 2.3 an algebraic treatment is introduced. Here we define the Moore product of tree automata, which is a restricted form of the cascade product [43] and which turns out to be very suitable for our needs. In Section 2.2 the key concepts (such as the aforementioned Moore product, Moore pseudovarieties as well as the connected variants of these notions) and also definite tree automata are formalized. Afterwards, in Section 2.3 a characterization theorem (Theorem 2.3.3) is proved: we show that when $\mathcal{L}$ is a class of regular tree languages having the additional technical property that "all quotients of any $L \in \mathcal{L}$ are definable in FTL($\mathcal{L}$)", then a tree language L is definable in FTL($\mathcal{L}$) if and only if it is regular and its minimal automaton $\mathbb{A}_L$ is contained in the least class of finite tree automata that contains all the minimal automata of the members of $\mathcal{L}$ and a specific definite tree automaton $\mathbb{D}_0$, and is closed under renamings, and taking homomorphic images, subautomata and Moore products. Such classes are called Moore pseudovarieties of finite tree automata. These two sections contain results that are published in [18].

The characterization theorem does not immediately provide decidability results. Section 2.4 gives an application of the above characterization: we show that two fragments CTL(EF$^+$) and CTL(EF*) of the well-known and heavily researched temporal logic CTL [2, 45] both have a polynomial-time decidable definability problem, at least for finite trees.

This is achieved by identifying all the algebraic properties (that we call Moore properties) that a finite connected tree automaton $\mathbb{A}_L$ exactly has to possess in order to be contained in the Moore pseudovariety generated by the two automata $\mathbb{E}^+_{\mathrm{EF}}$ and $\mathbb{D}_0$ (in the case of CTL(EF$^+$)) or by the two automata $\mathbb{E}^*_{\mathrm{EF}}$ and $\mathbb{D}_0$ (in the case of CTL(EF*)). Here $\mathbb{E}^+_{\mathrm{EF}}$ and $\mathbb{E}^*_{\mathrm{EF}}$ respectively denote the minimal automata of the regular tree languages $\mathcal{L}_{\mathrm{EF}^+}$ and $\mathcal{L}_{\mathrm{EF}^*}$, that correspond to the strict, resp. non-strict variants of the temporal modality EF ("existence in the future"). By showing that these Moore pseudovarieties have a decidable membership problem we get as a byproduct that the above logics have a decidable definability problem. This section contains results that are published in [19].

The last section of Chapter 2, Section 2.5 provides another characterization of the logics FTL($\mathcal{L}$). This characterization is based on a game-theoretic approach: for any class $\mathcal{L}$ of tree languages and integer $n \geq 0$ we introduce an Ehrenfeucht-Fraïssé type game, called the n-round $\mathcal{L}$-game which is played on two trees, s and t, between two players, Spoiler and Duplicator. We say that Spoiler wins the n-round $\mathcal{L}$-game on a language L, if he wins the n-round $\mathcal{L}$-game on any pair (s,t) of trees with $s \in L$ and $t \notin L$. The main result of this section is that when $\mathcal{L}$ is a finite class of tree languages, then Spoiler wins the n-round $\mathcal{L}$-game on a language L for some $n \geq 0$ if and only if L is definable in the logic FTL($\mathcal{L}$). This section contains results that are not yet published.

Section 3 is devoted to different notions of aperiodicity in finite tree automata. For finite semigroups, aperiodicity is a well understood concept: a finite semigroup S is aperiodic if there exists an $n \geq 1$ such that $s^n = s^{n+1}$ holds for each element $s \in S$. This form of aperiodicity can be straightforwardly extended to languages of nonempty words: a regular word language L is aperiodic if and only if its syntactic semigroup is aperiodic, i.e. when there exists an $n \geq 1$ such that $uv^n w \in L \Leftrightarrow uv^{n+1}w \in L$ for any triple (u,v,w) of words. Aperiodicity nicely correlates with both first-order definability and LTL-definability in this classical case, see [30, 32, 46]. However, when one tries to extend this notion to trees, by defining a tree language to be aperiodic if and only if its syntactic semigroup [54] is aperiodic (this is further referred as "context aperiodicity"), the correlation fades: the class of all aperiodic regular tree languages properly contains the class of first-order definable tree languages.

Thus, in this chapter we define more restrictive variants of aperiodicity: the notion of n-aperiodicity, for each $n \geq 1$, and that of strong aperiodicity, which is the limit of the n-aperiodicity notions: a tree automaton is defined to be strongly aperiodic if it is n-aperiodic for every $n \geq 1$. These notions form a hierarchy: the outermost class defined by this hierarchy is the class of context aperiodic finite tree automata, which contains the class of 1-aperiodic finite tree automata; for each $n \geq 1$, the class of n-aperiodic finite tree automata contains the class of $(n+1)$-aperiodic finite tree automata, and the smallest aperiodicity class is that of strongly aperiodic finite tree automata which contains all the finite definite tree automata.

It turns out that in the classical word case (which corresponds to the rank type $\{0, 1\}$ or $\{1\}$, i.e. when the signature consists of unary symbols and optionally constants), this hierarchy collapses to two levels: in that setting, the two outermost classes (that of context aperiodic and 1-aperiodic finite automata) coincide, and this class strictly contains the class of 2-aperiodic finite automata, which coincides with all the inner classes down to the class of finite definite automata. Interestingly, for any non-classical rank type (that contains at least one integer greater than one) we prove that the hierarchy does not collapse: the class of context aperiodic finite tree automata is strictly greater than the class of 1-aperiodic finite tree automata; for any n, the class of n-aperiodic finite tree automata strictly contains the class of all $(n + 1)$-aperiodic finite tree automata, and there also exist strongly aperiodic finite tree automata that are not definite. These claims are proved in Section 3.3.

We also examine the complexity of the membership problem of these classes, which are in fact generalized cascade pseudovarieties of finite tree automata (as shown in Section 3.2), i.e. they are closed under renamings, taking subautomata, homomorphic images and a generalized version of the cascade product, with the exception of the class of context aperiodic finite tree automata, that forms a cascade pseudovariety. From the fact that deciding aperiodicity is a PSPACE-complete problem already in the word case [4, 7], we conclude that deciding n-aperiodicity is PSPACE-hard for any n. We also show that the problem is solvable in exponential time for each n. On the other hand, we give a polynomial time algorithm for deciding strong aperiodicity. This is done in Section 3.4.

In Section 3.5 we relate the above notions of aperiodicity to logic. It turns out that the class of 1-aperiodic tree languages (a regular tree language is called n-aperiodic if its minimal tree automaton is 1-aperiodic) lies strictly between the class of CTL-definable tree languages and the class of context aperiodic tree languages, whereas the class of first-order definable tree languages is incomparable with the class of 1-aperiodic tree languages. Section 3.6 is devoted to a generalization of the aperiodicity notions, motivated by the celebrated Krohn-Rhodes decomposition theory. Finally, in Section 3.7 we slightly modify the definition of our aperiodicity notions, by involving polynomial functions instead of term functions in the conditions. It turns out that, based on polynomial functions, the aperiodicity hierarchy collapses at level 2. As a last conclusion we show that the polynomial variant of 1-aperiodicity is incomparable with the original version of n-aperiodicity for any $n > 1$. The results of Chapter 3 were published in [17].

Chapter 1

Preliminaries

1.1 Brief summary of earlier results

This section briefly summarizes the known results about connections between logical definability and algebra related to the thesis.

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.

For instance, considering monadic second-order logic (MSO) on words, equipped either with the successor relation or a total ordering on the positions of the word, exactly the regular languages are definable, see [6, 10], or [49].

Regarding the less expressive logic FO($<$), i.e., first-order logic equipped with the total ordering of the positions, a classical result of McNaughton and Papert [32] in conjunction with a result of Schützenberger [46] 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 [30].

A similar aperiodicity condition characterizes the expressive power of first-order logic and LTL on infinite words, cf. [35, 36]. 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 [9], 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 [49].

The logic FO(+1), first-order logic equipped with the successor relation (but without the ordering) on the positions corresponds exactly to the class **LTT** of locally threshold testable languages [30]; this is again a decidable property, see [38, 51, 53].

When formulated on the domain of (say, finite, ranked and ordered) trees, most of the above questions are still open. It still holds that in MSO, exactly the regular tree languages are definable [42, 50] (when the logic is equipped with the successor relations). Also, the logic FO(+1) still corresponds to the class of locally threshold testable languages, see [31]; but just as in the word case, this characterization is by no means effective. The question has been further studied in [3] where a decidable characterization has been found.

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. [29, 40, 41, 54].

Aperiodicity can be generalized to trees in several different ways. One of them was studied in [29, 54] and shown to be a necessary but not a sufficient condition of first-order definability. In [17] we have defined an infinite hierarchy of aperiodicity notions and studied their relations to logical definability.

In [20], 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 [20].

A problem related to first-order definability is the characterization of the expressive power of CTL and CTL* [2, 11, 45]: 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 [27] that for certain types of trees first-order definability is equivalent to definability in CTL*. In [14, 15], 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 [12, 28], which is closed under the cascade product. (Any such tree language is clearly regular, or recognizable, cf. [25].)

Cascade products of finite tree automata were studied e.g. in [12, 16, 21, 24, 43]. This notion is closely related to the wreath product of clones defined in [55].

In [18] 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 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.

In [19], an application of the above theorem has been elaborated; we provided polynomial time decidable characterizations of several small Moore pseudovarieties of finite connected tree automata. 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 [5] using different methods; the decidability of the expressive power of the latter fragment was left open there.

Using a different approach, Z. Wu [58] also proved that CTL(EF*) has a decidable definability problem. His approach is based on an Ehrenfeucht-Fraïssé type game. A celebrated result (see e.g. [31]) states that first-order logic over relational structures over an arbitrary finite relational signature can be reformulated in terms of a game between two players, Spoiler and Duplicator. It has been shown that a property $\mathcal{P}$ of structures is definable in first-order logic if and only if there exists a number n of rounds such that when $\mathcal{A}$ is a structure having property $\mathcal{P}$ and $\mathcal{B}$ is a structure not having $\mathcal{P}$, then Spoiler has a winning strategy in the so-called n-round Ehrenfeucht-Fraïssé game on the pair $(\mathcal{A}, \mathcal{B})$ of structures. One (yet unpublished) contribution of this thesis is that we define for each class $\mathcal{L}$ of tree languages and number n of rounds a two-player game, called the n-round $\mathcal{L}$-game with the following property: two trees, s and t satisfy the same set of FTL($\mathcal{L}$)-formulas of depth at most n (denoted $s \equiv^n_{\mathcal{L}} t$) 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 (see e.g. [31]) show that when $\mathcal{L}$ is a finite class of tree languages, then $\equiv^n_{\mathcal{L}}$ 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 of rounds such that whenever s is a tree contained in L and t is a tree not in L, then Spoiler has a winning strategy in the n-round $\mathcal{L}$-game, played on the pair (s, t).

1.2 Sets, words, trees and languages

When $n \geq 0$ is an integer, $[n]$ denotes the set $\{1, \ldots, n\}$. Thus, $[0]$ is another notation for the empty set $\emptyset$. $\mathbb{N}$ denotes the set of all nonnegative integers.

When S is a set, S^* denotes the set $\{s_1 \ldots s_n : n \in \mathbb{N}, s_1, \ldots, s_n \in S\}$ of *(finite) words* over S. This set contains the *empty word* that consists of zero symbols and is denoted ϵ. The *concatenation* (or simply *product*) $u \cdot v$ of the words $u = s_1 \ldots s_k$ and $v = s'_1 \ldots s'_n$ is the word $s_1 \ldots s_k s'_1 \ldots s'_n$. The symbol $\cdot$ is often omitted. It is clear that $(S^*, \cdot)$ is the free monoid generated by S, with the unit element ϵ. The *prefix relation* $\preceq_{S^*}$ over words is defined as $u \preceq_{S^*} v$ if $v = uu'$ for some word $u' \in S^*$. $u \prec_{S^*} v$ holds if $u \preceq_{S^*} v$ and $u \neq v$, i.e. u is a *proper* prefix of v. We will usually omit the subscript S^*.

A *rank type* R is a finite nonempty subset of $\mathbb{N}$. 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 $X = \{x_1, x_2, \ldots\}$ of *variables*, that is assumed to be disjoint from any signature. The subset $\{x_1, \ldots, x_n\}$ of X 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* (or ΣX_n-*terms*) is the least set satisfying the following conditions:

1. any variable $x \in X_n$ is a ΣX_n-tree;
2. any symbol $\sigma \in \Sigma_0$ is a ΣX_n-tree;
3. if $0 < k \in R$ is an arity, $\sigma \in \Sigma_k$ is a symbol and $t_1, \ldots, t_k$ are ΣX_n-trees, then $\sigma(t_1, \ldots, t_k)$ is a ΣX_n-tree.

It is clear that $T_\Sigma(X_n)$ is nonempty if and only if $n > 0$ or $0 \in R$ holds, and it is infinite if and only if it is nonempty and R contains at least one positive integer. A tree $t \in T_\Sigma(X_n) - X_n$ which is not a variable is called *proper*.

Symbols $\sigma \in \Sigma_0$ are called *constant symbols*. When σ is a constant symbol, we can view the ΣX_n-tree σ also as the tree $\sigma()$ (i.e. of the form $t = \sigma(t_1, \ldots, t_k)$ with $k = 0$ and $\sigma \in \Sigma_0$).

T_Σ is written for $T_\Sigma(X_0)$, the set of variable-free trees over Σ. These trees are also called Σ-*trees*. The set of Σ-*contexts* is the subset CT_Σ of $T_\Sigma(X_1)$ which contains the trees in which x_1 occurs exactly once. Contexts will usually be denoted by ζ, ξ etc, whereas trees in general by s, t etc. We write $T_\Sigma(X)$ for $\bigcup_{n\geq 0} T_\Sigma(X_n)$, the set of all ΣX-trees.

The *height* $\mathrm{hg}(t)$ of a tree $t \in T_\Sigma(X)$ is defined as usual:

1. when $t \in X \cup \Sigma_0$, then $\mathrm{hg}(t) = 0$;
2. when $t = \sigma(t_1, \ldots, t_n)$ with $\sigma \in \Sigma_n$ for some $n > 0$, then $\mathrm{hg}(t)$ is defined as $1 + \max\{\mathrm{hg}(t_i) : i \in [n]\}$.

When $t \in T_\Sigma(X_n)$ is a Σ-tree and $\underline{t} = (t_1, \ldots, t_n)$ is an n-tuple of ΣX_m-trees, then the tree $t(\underline{t}) \in T_\Sigma(X_m)$ is defined inductively as follows:

1. if $t = x_i$ for some $i \in [n]$, then $t(\underline{t}) = t_i$;
2. if $t = \sigma(t'_1, \ldots, t'_k)$ for some $\sigma \in \Sigma_k$, $k \geq 0$, and trees $t'_1, \ldots, t'_k \in T_\Sigma(X_n)$, then $t(\underline{t}) = \sigma(t'_1(\underline{t}), \ldots, t'_k(\underline{t}))$.

If $t \in T_\Sigma(X_1)$, thus $\underline{t} = (t_1)$ is a one-tuple, we may omit the parentheses and write tt_1 for $t(t_1)$. This will be done mostly when $t = \zeta \in CT_\Sigma$ is a Σ-context and t_1 is either a variable-free tree or a context, i.e. $t_1 \in T_\Sigma \cup CT_\Sigma$.

Sometimes it will be convenient to think about a Σ-tree as a mapping from a tree domain to $\Sigma \cup X$. In this setting, the *domain* $\mathrm{dom}(t)$ of a tree t is a finite subset of $[\max R]^*$ and is defined as follows:

1. when $t \in X \cup \Sigma_0$, $\mathrm{dom}(t) = \{\epsilon\}$;
2. when $t = \sigma(t_1, \ldots, t_n)$ for some arity $n > 0$, symbol $\sigma \in \Sigma_n$ and trees t_i, $i \in [n]$, then $\mathrm{dom}(t) = \{\epsilon\} \cup \{i \cdot v : i \in [n], v \in \mathrm{dom}(t_i)\}$.

When $t \in T_\Sigma(X)$ is a tree and $v \in \mathrm{dom}(t)$ is a *node* (or *vertex*) of t, $t(v)$ denotes the *label* of v in t, that is,

1. $t(\epsilon) = t$ for any $t \in X \cup \Sigma_0$;
2. $t(\epsilon) = \sigma$, when $t = \sigma(t_1, \ldots, t_n)$ for some $\sigma \in \Sigma_n$, $t_1 \ldots, t_n \in T_\Sigma(X)$;
3. $t(i \cdot v) = t_i(v)$, when $t = \sigma(t_1, \ldots, t_n)$ for some $\sigma \in \Sigma_n$, $t_1, \ldots, t_n \in T_\Sigma(X)$.

$\mathrm{Root}(t)$ is a shorthand for $t(\epsilon)$.

Another notion regarding trees is that of the *subtree* $t|_v$ of a tree t rooted at $v \in \mathrm{dom}(t)$. The domain of this tree is $\mathrm{dom}(t|_v) = \{w : v \cdot w \in \mathrm{dom}(t)\}$, and the labeling is given by $t|_v(w) = t(v \cdot w)$ for each $w \in \mathrm{dom}(t|_v)$. It is clear that these notions are well-defined. A subtree $t|_v$ of t is called a *proper subtree* of t if $v \neq \epsilon$ and is called an *immediate subtree* of t if $v \in [\max R]$. (Note that a proper subtree of a tree is not always a proper tree.)

The above notions are extended to *tuples* of trees as follows. When $\underline{t} = (t_1, \ldots, t_n)$ is an n-tuple of trees, let $\mathrm{dom}(\underline{t})$ denote the set $\{i \cdot v : i \in [n], v \in \mathrm{dom}(t_i)\}$. Furthermore, for any node $i \cdot v \in \mathrm{dom}(\underline{t})$ of $\underline{t}$ above, let $\underline{t}(i \cdot v) = t_i(v)$ and $\underline{t}|_{i \cdot v} = t_i|_v$. These notions are also well-defined.

We consider a (Σ-)*tree language* any set of variable-free Σ-trees, i.e. any $L \subseteq T_\Sigma$. 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 $\zeta^{-1}(L) = \{t : \zeta t \in L\}$.

Suppose Σ and Δ are signatures having the same rank type R. A tree $t \in T_\Delta$ is a *relabeling* of a tree $s \in T_\Sigma$ if $\mathrm{dom}(t) = \mathrm{dom}(s)$, and for any vertex $v \in \mathrm{dom}(t)$ and variable $x \in X$, $t(v) = x$ if and only if $s(v) = x$, i.e. we get the tree s from t by relabeling each node which are labeled by some symbol from Σ, but preserving the variables.

When Σ and Δ are signatures having the same rank type R and $h : \Sigma \to \Delta$ is a rank-preserving map (i.e. for any $n \in R$ and $\sigma \in \Sigma_n$, the symbol $h(\sigma)$ is contained in Δ_n), then h induces a *literal tree homomorphism* from $T_\Sigma(X)$ to $T_\Delta(X)$, also denoted h as follows: for any tree $s \in T_\Sigma(X)$ let $\mathrm{dom}(h(s)) = \mathrm{dom}(s)$ and for any vertex $v \in \mathrm{dom}(s)$ define

$$(h(s))(v) \quad = \quad \begin{cases} s(v) & \text{if } s(v) \in X; \\ h(s(v)) & \text{if } s(v) \in \Sigma. \end{cases}$$

Given a signature Σ of rank type R and an arbitrary set A, which is disjoint from both Σ and X, we also define the set $T_{\Sigma,A}(X_n)$ of ΣAX_n*-polynomial symbols* for each n as the least set satisfying the following conditions:

1. any $x \in X_n$ is a ΣAX_n-polynomial symbol;
2. any $a \in A$ is a ΣAX_n-polynomial symbol;
3. when $\sigma \in \Sigma_k$ is a symbol and $p_1, \ldots, p_k$ are ΣAX_n-polynomial symbols, then $\sigma(p_1, \ldots, p_k)$ is also a ΣAX_n-polynomial symbol.

The ΣAX_n-polynomial symbols are also called n-ary ΣA-polynomial symbols. The set $CT_{\Sigma,A}$ of ΣA*-polynomial contexts* is the subset of $T_{\Sigma,A}(X_1)$ consisting of the polynomial symbols in which x_1 occurs exactly once. The set $T_{\Sigma,A}(X)$ of ΣAX-polynomial symbols

is the union $\bigcup_{n\geq 0} T_{\Sigma,A}(X_n)$. A polynomial symbol which is not a variable is called *proper*. Elements of the set $T_{\Sigma,A} = T_{\Sigma,A}(X_0)$ (that is, the variable-free polynomial symbols) are called ΣA-polynomial symbols.

We will also use occasionally the notation $\mathrm{Root}(p)$ to denote the root symbol of the polynomial symbol $p \in T_{\Sigma,A}(X)$, that is,

$$\mathrm{Root}(p) \quad = \quad \begin{cases} p & \text{if } p \in X \cup A; \\ \sigma & \text{if } p = \sigma(p_1, \ldots, p_k) \text{ for some } \sigma \in \Sigma_k, p_i \in T_{\Sigma,A}(X). \end{cases}$$

The *domain* $\mathrm{dom}(p)$ of a polynomial symbol p and *relabelings* of polynomial symbols are defined analogously to the respective notions for trees. When Σ and Δ are signatures, $h_1 : \Sigma \to \Delta$ is a rank-preserving map, A and B are sets disjoint from Σ and Δ, respectively, and $h_2 : A \to B$ is a function, then the pair (h_1, h_2) *induces* a relabeling $T_{\Sigma,A}(X) \to T_{\Delta,B}(X)$ that replaces the label of each node v of a polynomial symbol $p \in T_{\Sigma,A}(X)$ by $h_1(v)$ if $p(v) \in \Sigma$ and by $h_2(v)$ if $p(v) \in A$. When $p(v) \in X$, then the label remains unchanged. When either h_1 or h_2 is the identity function (so that $\Sigma \subseteq \Delta$ or $A \subseteq B$), we say that the relabeling is induced by h_2 or h_1, respectively.

1.3 Σ-tree automata

Given a signature Σ of some rank type R, a *(Σ-)tree automaton* $\mathbb{A} = (A, \Sigma)$ consists of a nonempty *state* (or *carrier*) *set* A and for each symbol $\sigma \in \Sigma_n$, there is an associated *elementary operation* $\sigma^{\mathbb{A}} : A^n \to A$. Elements of the state set A are usually called *states*. When $\sigma \in \Sigma_n$ is a symbol, $\mathrm{Img}_{\mathbb{A}}(\sigma)$ denotes the *image set* $\{\sigma^{\mathbb{A}}(a_1, \ldots, a_n) : a_1, \ldots, a_n \in A\}$ of σ in $\mathbb{A}$. When the set A of states is finite, we call $\mathbb{A}$ a *finite tree automaton*; when even $|A| = 1$, $\mathbb{A}$ is called a *trivial tree automaton*.

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 operations $\sigma^{\mathbb{A}}$, then we say that $\mathbb{B}$ is a *subautomaton* of $\mathbb{A}$. When also $B \subset A$ holds, $\mathbb{B}$ is called a *proper* 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 $A \times B$ and the elementary operations $\sigma^{\mathbb{A}\times\mathbb{B}}\big((a_1, b_1), \ldots, (a_n, b_n)\big) = \big(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \sigma^{\mathbb{B}}(b_1, \ldots, b_n)\big)$ for all $\sigma \in \Sigma_n$ and states $a_1, \ldots, a_n \in A$, $b_1, \ldots, b_n \in B$.

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 $\mathbb{A}$, i.e. for each $\delta \in \Delta_n$ there exists a symbol

$\sigma \in \Sigma_n$ having $\delta^{\mathbb{B}} = \sigma^{\mathbb{A}}$. Observe that this relation is *not* symmetric. When Σ and Δ are signatures over the same rank type R, $\mathbb{A} = (A, \Sigma)$ is a tree automaton and $h : \Delta \to \Sigma$ is a rank-preserving map, then h *determines* the renaming $\mathbb{B} = (A, \Delta)$ of $\mathbb{A}$ where $\delta^{\mathbb{B}} = (h(\delta))^{\mathbb{A}}$ for each $\delta \in \Delta$.

Given a Σ-tree automaton $\mathbb{A} = (A, \Sigma)$, we associate a *term function* $t^{\mathbb{A}} : A^m \to A$ to any tree $t \in T_\Sigma(X_m)$ defined as:

1. any x_i, $i \in [m]$ induces the *projection* to the i-th coordinate, i.e. $x_i^{\mathbb{A}}(a_1, \ldots, a_m) = a_i$;
2. any constant symbol $\sigma \in \Sigma_0$ induces the constant function $A^m \to A$ having the value $\sigma^{\mathbb{A}}$;
3. any tree $t = \sigma(t_1, \ldots, t_n)$ with $\sigma \in \Sigma_n$ for some $n > 0$ induces the composition of the elementary operation $\sigma^{\mathbb{A}}$ with the tupling of the term functions $t_i^{\mathbb{A}}$, $i \in [n]$, i.e. for any $\underline{a} = (a_1, \ldots, a_m)$, $t^{\mathbb{A}}(\underline{a}) = \sigma^{\mathbb{A}}(t_1^{\mathbb{A}}(\underline{a}), \ldots, t_n^{\mathbb{A}}(\underline{a}))$.

Term functions that are induced by proper trees are called *proper term functions*.

Thus, any tree $t \in T_\Sigma$ has a constant function as associated term function. When $t^{\mathbb{A}}$ is the constant function having the value $a \in A$, we say that t *evaluates* to a in $\mathbb{A}$ and denote this by $t^{\mathbb{A}} = a$, identifying the constant-valued function with its value. Then, if a Σ-tree automaton $\mathbb{A} = (A, \Sigma)$ is given along with a set $A' \subseteq A$ of *final states*, we define the tree language $L_{\mathbb{A},A'}$ *recognized by* $\mathbb{A}$ *(with the set* A' *of final states)* as the set $\{t \in T_\Sigma : t^{\mathbb{A}} \in A'\}$. A tree language L is *recognizable* by the tree automaton $\mathbb{A}$ if $L = L_{\mathbb{A},A'}$ for some set $A' \subseteq A$ of final states.

When the rank type R of a signature Σ contains 0, the *connected part* of a tree automaton $\mathbb{A} = (A, \Sigma)$ is the subautomaton (A', Σ) of $\mathbb{A}$ where $A' = \{t^{\mathbb{A}} : t \in T_\Sigma\}$ consists of the *accessible* states of A. Note that since 0 is contained in R, T_Σ is not empty, thus A' cannot be empty either. It is easy to see that (A', Σ) is the least subautomaton of $\mathbb{A}$. Also when Σ is a signature of rank type R which contains 0, we call a Σ-tree automaton $\mathbb{A} = (A, \Sigma)$ *connected* if it is generated by the constants (or equivalently, if it has no proper subautomata).

When $\mathbb{A} = (A, \Sigma)$ is a tree automaton and $p \in T_{\Sigma,A}(X_n)$, then p induces an *(n-ary) polynomial function* $p^{\mathbb{A}} : A^n \to A$ as follows:

1. $x_i^{\mathbb{A}}$ is the projection π_i to the i-th coordinate for any $x_i \in X_n$;
2. each symbol $\sigma \in \Sigma_0$ induces the constant function $A^n \to A$ with value $\sigma^{\mathbb{A}}$;
3. each $a \in A$ induces the constant function $A^n \to A$ with value a;

4. when $p = \sigma(p_1, \ldots, p_k)$, then $p^{\mathbb{A}}$ is defined as the composition of $\sigma^{\mathbb{A}}$ with the target tupling of the functions $p_i^{\mathbb{A}}$, $i \in [k]$, i.e. for any $\underline{a} = (a_1, \ldots, a_n)$ let $p^{\mathbb{A}}(\underline{a}) = \sigma^{\mathbb{A}}(p_1^{\mathbb{A}}(\underline{a}), \ldots, p_k^{\mathbb{A}}(\underline{a}))$.

Polynomial functions of $\mathbb{A} = (A, \Sigma)$ induced by (proper) ΣA-polynomial contexts are called *(proper) translations of* $\mathbb{A}$.

We note that ΣX_n-trees are $\Sigma\emptyset X_n$-polynomial symbols and any ΣX_n-tree t induces the same term function (when t is viewed as a tree) and polynomial function (when t is viewed as a $\Sigma A X_n$-polynomial symbol) in any tree automaton (A, Σ). Thus, we could define trees as special polynomial symbols and term functions as polynomial functions induced by trees. The reason why we define these notions this way is to emphasize that trees play a central role in this thesis.

We call a tree language $L \subseteq T_\Sigma$ *regular* if it is recognizable by some finite tree automaton. It is well-known (see e.g. [25]) that if L is a regular tree language, then it has only finitely many quotients and each of these quotients is regular.

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, i.e. $h(\sigma^{\mathbb{A}}(a_1, \ldots, a_n)) = \sigma^{\mathbb{B}}(h(a_1), \ldots, h(a_n))$ holds for all arities $n \in R$, symbol $\sigma \in \Sigma_n$ and states $a_1, \ldots, a_n \in A$. We call $\mathbb{B}$ a *homomorphic image* or a *quotient* of $\mathbb{A}$ if there exists a surjective homomorphism from $\mathbb{A}$ to $\mathbb{B}$; if this homomorphism is even injective (in which case it is called an *isomorphism*), then $\mathbb{A}$ and $\mathbb{B}$ are said to be *isomorphic* (it is easy to see that in this case the inverse of the given homomorphism is also a bijective homomorphism, thus the definition is symmetric). We call a quotient of a subautomaton of a tree automaton $\mathbb{A}$ a *divisor* of $\mathbb{A}$.

Homomorphic images are strongly related to congruences. Suppose $\mathbb{A} = (A, \Sigma)$ is a Σ-tree automaton and $\Theta \subseteq A \times A$ is an equivalence relation on A which is compatible with all the elementary operations of $\mathbb{A}$, i.e. whenever $\sigma \in \Sigma_n$ is an n-ary symbol and $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ are states with $a_i \Theta b_i$ for each $i \in [n]$, then also $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) \Theta \sigma^{\mathbb{A}}(b_1, \ldots, b_n)$ holds. Then Θ is called a *congruence* of $\mathbb{A}$. Clearly, the identity relation $\Delta_A = \{(a, a) : a \in A\}$ and the complete relation $\nabla_A = A \times A$ over A are congruences of $\mathbb{A}$; these are called the *trivial* congruences, whereas a congruence Θ which is different from both Δ_A and ∇_A is a *nontrivial* congruence of $\mathbb{A}$.

It is clear that the intersection $\bigcap_{i \in I} \Theta_i$ of any family $(\Theta_i)_{i \in I}$ of congruences of $\mathbb{A}$ is also a congruence. Thus, when $\mathbb{A}$ is a given tree automaton, the set $\mathrm{Con}(\mathbb{A})$ of congruences of $\mathbb{A}$ forms a complete lattice. In this lattice, the meet and join operators are defined as $\bigwedge_{i \in I} \Theta_i = \bigcap_{i \in I} \Theta_i$ and $\bigvee_{i \in I} \Theta_i = \bigcap(\Theta \in \mathrm{Con}(\mathbb{A}) : \bigcup_{i \in I} \Theta_i \subseteq \Theta)$; the least (resp. greatest) element of $\mathrm{Con}(\mathbb{A})$ is $\Delta_{\mathbb{A}}$ (resp. $\nabla_{\mathbb{A}}$). When Θ is in $\mathrm{Con}(\mathbb{A})$ and $a \in A$, then the congruence class

of A/Θ containing a will be denoted a/Θ, or sometimes just $\overline{a}$ when Θ is understood.

When Θ is a congruence of $\mathbb{A} = (A, \Sigma)$, then the *factor automaton* $\mathbb{A}/\Theta = (B, \Sigma)$ is defined as follows:

1. the state set of $\mathbb{A}/\Theta$ is $B = \{a/\Theta : a \in A\}$;
2. for any symbol $\sigma \in \Sigma_n$, the elementary operation $\sigma^{\mathbb{A}/\Theta}$ maps the n-tuple $(a_1/\Theta, \ldots, a_n/\Theta)$ to the congruence class $\sigma^{\mathbb{A}}(a_1, \ldots, a_n)/\Theta$.

From the congruence properties it is clear that the above interpretation of the symbols is well-defined. Obviously, $\mathbb{A}/\Theta$ is a quotient of $\mathbb{A}$.

The Homomorphism Theorem states that for any Σ-tree automaton $\mathbb{A} = (A, \Sigma)$ and homomorphic image $\mathbb{B} = (B, \Sigma)$ of $\mathbb{A}$ there exists a congruence Θ of $\mathbb{A}$ such that the factor automaton $\mathbb{A}/\Theta$ is isomorphic to $\mathbb{B}$. When h is a homomorphism from $\mathbb{A}$ onto $\mathbb{B}$, we may define Θ by $a_1 \Theta a_2 \Leftrightarrow h(a_1) = h(a_2)$, which is called the *kernel* of h.

A nontrivial Σ-tree automaton $\mathbb{A} = (A, \Sigma)$ is said to be *subdirectly irreducible* if whenever $\mathbb{A}$ is isomorphic to a subautomaton of some direct product $\prod_{i \in I} \mathbb{A}/\Theta_i$ for some family $(\Theta_i)_{i \in I}$ of congruences of $\mathbb{A}$, then $\Theta_j = \Delta_A$ holds for some $j \in I$. Any nontrivial Σ-tree automaton $\mathbb{A}$ is either subdirectly irreducible or is isomorphic to a subautomaton of the direct product of a family $(\mathbb{A}/\Theta_i)_{i \in I}$, where each Θ_i is a nontrivial congruence of $\mathbb{A}$. In this case $\mathbb{A}$ is *subdirectly reducible*. Clearly when $\mathbb{A}$ is finite, the above I can be also assumed finite.

It is well-known that for any tree language $L \subseteq T_\Sigma$ there exists a *minimal automaton* $\mathbb{A}_L$, unique up to isomorphism, having the following properties:

1. $\mathbb{A}_L$ is connected.
2. L is recognized by $\mathbb{A}_L$.
3. $\mathbb{A}_L$ is a divisor of any Σ-tree automaton $\mathbb{B}$ by which L is recognizable.

Thus, a tree language L is regular if and only if its minimal automaton $\mathbb{A}_L$ is finite. Given a finite tree automaton $\mathbb{A} = (A, \Sigma)$ with a set $A' \subseteq A$ of final states, the minimal automaton $\mathbb{A}_L$ of the recognized language $L = L_{\mathbb{A},A'}$ can be effectively constructed.

We call a nonempty class $\mathbf{V}$ of finite tree automata a *pseudovariety of finite tree automata* if it is closed under taking direct products, homomorphic images, subautomata and renamings. It is clear that the intersection of any family of pseudovarieties is also a pseudovariety (observe that the intersection cannot be empty since every pseudovariety contains all the trivial tree automata), thus for any class $\mathbf{K}$ of finite tree automata there

exists a least pseudovariety $\langle \mathbf{K} \rangle$ containing $\mathbf{K}$. We also say that $\langle \mathbf{K} \rangle$ is the pseudovariety *generated by* $\mathbf{K}$. When $\mathbf{K} = \{\mathbb{A}_1, \ldots, \mathbb{A}_n\}$ is a finite class, we write $\langle \mathbb{A}_1, \ldots, \mathbb{A}_n \rangle$ for $\langle \mathbf{K} \rangle$.

When $\mathbf{K}$ is a class of finite tree automata, we let $\mathbf{K}^c$ denote the class consisting of the connected parts of the members of $\mathbf{K}$.

When $\mathbb{A}$ and $\mathbb{B}$ are connected Σ-tree automata, their *connected direct product*, also denoted $\mathbb{A} \times \mathbb{B}$ is the connected part of their direct product. Similarly, we say that the tree automaton $\mathbb{A} = (A, \Sigma)$ is a *connected renaming* of the tree automaton $\mathbb{B} = (B, \Delta)$ if $\mathbb{A}$ is the connected part of a renaming of $\mathbb{B}$. (Note that any homomorphic image of a connected tree automaton is connected, thus there is no need to define a notion such as "connected homomorphism".) When we write $\mathbb{A} \times \mathbb{B}$, it will be always clear whether this notation stands for a direct product or a connected direct product. Observe that no connected tree automaton $\mathbb{A}$ has a proper subautomaton. Moreover, when $\mathbb{A}$ and $\mathbb{B}$ are connected Σ-tree automata, then there is at most one homomorphism from $\mathbb{A}$ to $\mathbb{B}$.

A nonempty class $\mathbf{V}^c$ of finite connected tree automata is called a *pseudovariety of finite connected tree automata* if it is closed under connected renamings, connected direct products and homomorphic images. When $\mathbf{K}^c$ is a class of finite connected tree automata, there exists a least pseudovariety of finite connected tree automata, also denoted $\langle \mathbf{K}^c \rangle$, containing $\mathbf{K}^c$. We also say that $\langle \mathbf{K}^c \rangle$ is *generated* by $\mathbf{K}^c$.

Since our logics will be defined over variable-free trees, in connection with logical definability it will suffice to deal with pseudovarieties of finite connected tree automata. Also the following hold:

1. When $\mathbf{V}$ is a pseudovariety of finite tree automata, $\mathbf{V}^c$ is a pseudovariety of finite connected tree automata.

2. Let $\mathbf{K}$ be an arbitrary class of finite tree automata. Then $\langle \mathbf{K} \rangle^c$, the class of all connected tree automata contained in the pseudovariety of finite tree automata generated by $\mathbf{K}$, coincides with $\langle \mathbf{K}^c \rangle$, the pseudovariety of finite connected tree automata generated by the connected parts of the members of $\mathbf{K}$.

The class of pseudovarieties of finite (connected) tree automata is a complete lattice. In this lattice, when $(\mathbf{V}_i)_{i \in I}$ is a family of pseudovarieties of finite (connected) tree automata, $\bigwedge_{i \in I} \mathbf{V}_i = \bigcap_{i \in I} \mathbf{V}_i$ is their meet and $\bigvee_{i \in I} \mathbf{V}_i = \langle \bigcup_{i \in I} \mathbf{V}_i \rangle$ is their join; the least element contains only the trivial tree automata and the greatest element contains every finite (connected) tree automaton.

A closely related notion is that of literal varieties of tree languages. 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, i.e. when Σ and

Δ are signatures having the same rank type, $\zeta \in T_\Delta$ is a Δ-context, $K, L \in T_\Delta$ are tree languages contained in $\mathcal{L}$ and $h : T_\Sigma \to T_\Delta$ is a literal tree homomorphism, then each of the languages $\zeta^{-1}L$, $K \cap L$, $\overline{L} = T_\Delta - L$ over Δ and the language $h^{-1}(L)$ over Σ is a member of $\mathcal{L}$.

The class of literal varieties of tree languages also forms a complete lattice: when $(\mathcal{L}_i)_{i\in I}$ is a family of literal varieties of tree languages, $\bigwedge_{i\in I} \mathcal{L}_i = \bigcap_{i\in I} \mathcal{L}_i$ is their meet and their join $\bigvee_{i\in I} \mathcal{L}_i$ is the smallest literal variety of tree languages containing $\bigcup_{i\in I} \mathcal{L}_i$. (From the definition of literal varieties it is clear that when $(\mathcal{L}_i)_{i\in I}$ is a family of literal varieties of tree languages, then their intersection $\bigcap_{i\in I} \mathcal{L}_i$ is also a literal variety of tree languages, thus the concept of "smallest literal variety containing $\bigcup_{i\in I} \mathcal{L}_i$" is well-defined.)

There exists an *Eilenberg correspondence* [9] between two of the lattices defined above:

1. The lattice of all pseudovarieties of finite connected tree automata is isomorphic to the lattice of all literal varieties of tree languages.

2. The following mapping establishes an order isomorphism from the lattice of all pseudovarieties of finite connected tree automata to the lattice of all literal varieties of tree languages:

$$\mathbf{V}^c \quad \mapsto \quad \mathcal{L}_{\mathbf{V}^c} = \{L : L \text{ is recognizable in some } \mathbb{A} \in \mathbf{V}^c\}.$$

3. The following mapping establishes the order isomorphism that is the inverse of the above isomorphism:

$$\mathcal{L} \quad \mapsto \quad \mathbf{V}^c_{\mathcal{L}} = \langle\{\mathbb{A}_L : L \in \mathcal{L}\}\rangle.$$

   Recall that $\mathbb{A}_L$ stands for the minimal automaton of the tree language L.

For more information on tree automata we refer the reader to [26]; several different notions of (pseudo)varieties and Eilenberg correspondence between classes of automata and languages were investigated in [1, 13, 37, 44, 47, 48].

1.4 Logics on trees

In this section we briefly recall the syntax and semantics of two classical logics on trees, first-order logic and the temporal logic CTL.

Let Σ be a signature over the rank type R where $0 \in R$.

The set of first-order formulas over Σ (equipped with the descendant relation and the successor relations), denoted $\mathrm{FO}(<, S_i)$, is the least set satisfying the following conditions:

1. for any symbol $\sigma \in \Sigma$ and variable $x \in X$, $P_\sigma(x)$ is a formula;
2. for any variables $x, y \in X$, $(x = y)$ and $(x < y)$ are formulas;
3. for any variables $x, y \in X$ and integer $i \in [\max R]$, $S_i(x, y)$ is a formula;
4. if φ is a formula, then $(\neg\varphi)$ is also a formula;
5. if φ_1 and φ_2 are formulas, then $(\varphi_1 \vee \varphi_2)$ is a formula;
6. if $x \in X$ is a variable and φ is a formula, then $(\exists x\varphi)$ is a formula.

Note that we use the same set X of variables as in the case of ΣX-trees. This will not cause any confusion.

Let $t \in T_\Sigma$ be a tree, φ an $\mathrm{FO}(<, S_i)$-formula and $\mathbf{v} : X \to \mathrm{dom}(t)$ a mapping from the set of variables to the domain of t. We say that the pair $(t, \mathbf{v})$ *satisfies* φ, denoted $(t, \mathbf{v}) \models \varphi$, if one of the following holds:

1. $\varphi = P_\sigma(x)$ for some σ and x with $t(\mathbf{v}(x)) = \sigma$;
2. $\varphi = (x = y)$ for some $x, y \in X$ and $\mathbf{v}(x) = \mathbf{v}(y)$;
3. $\varphi = (x < y)$ for some $x, y \in X$ and $\mathbf{v}(x) \prec \mathbf{v}(y)$ (i.e. $\mathbf{v}(x)$ is a proper prefix of $\mathbf{v}(y)$, or y is a proper descendant of x);
4. $\varphi = S_i(x, y)$ for some $x, y \in X$ and $i \in [\max R]$, and $\mathbf{v}(y) = \mathbf{v}(x) \cdot i$;
5. $\varphi = (\neg\varphi_1)$ and it is not the case that $(t, \mathbf{v}) \models \varphi_1$;
6. $\varphi = (\varphi_1 \vee \varphi_2)$ and $(t, \mathbf{v}) \models \varphi_1$ or $(t, \mathbf{v}) \models \varphi_2$;
7. $\varphi = (\exists x\varphi_1)$ and there exists a node $u \in \mathrm{dom}(t)$ such that $(t, \mathbf{v}[x \mapsto u]) \models \varphi_1$. Here $\mathbf{v}[x \mapsto u]$ is the assignment from X to $\mathrm{dom}(t)$ where $\mathbf{v}[x \mapsto u](x) = u$ and for any $y \neq x$, $\mathbf{v}[x \mapsto u](y) = \mathbf{v}(y)$.

In order to define the semantics of $\mathrm{FO}(<, S_i)$ over trees without a given mapping $\mathbf{v}$ of the variables, we first have to define the set $\mathrm{FreeVar}(\varphi)$ of free variables in the formula φ as follows:

1. for any $\sigma \in \Sigma$ and $x \in X$, $\mathrm{FreeVar}(P_\sigma(x)) = \{x\}$;
2. for any $x, y \in X$, $\mathrm{FreeVar}((x = y)) = \{x, y\}$;
3. for any $x, y \in X$, $\mathrm{FreeVar}((x < y)) = \{x, y\}$;
4. for any $x, y \in X$ and $i \in [\max R]$, $\mathrm{FreeVar}(S_i(x, y)) = \{x, y\}$;
5. for any formula φ, $\mathrm{FreeVar}((\neg\varphi)) = \mathrm{FreeVar}(\varphi)$;
6. for any formulas φ_1, φ_2, $\mathrm{FreeVar}((\varphi_1 \vee \varphi_2)) = \mathrm{FreeVar}(\varphi_1) \cup \mathrm{FreeVar}(\varphi_2)$;
7. for any formula φ and variable $x \in X$, $\mathrm{FreeVar}((\exists x\varphi)) = \mathrm{FreeVar}(\varphi) - \{x\}$.

A formula φ is called a *sentence* if $\mathrm{FreeVar}(\varphi) = \emptyset$. It is easy to see that if φ is a formula, t is a Σ-tree and $\mathbf{v_1}, \mathbf{v_2} : X \to \mathrm{dom}(t)$ are two assignments of the variables that agree on $\mathrm{FreeVar}(\varphi)$, then $(t, \mathbf{v_1}) \models \varphi$ if and only if $(t, \mathbf{v_2}) \models \varphi$. Thus, for any sentence φ and Σ-tree t, the following are equivalent:

1. There exists an assignment $\mathbf{v} : X \to \mathrm{dom}(t)$ with $(t, \mathbf{v}) \models \varphi$.
2. For any assignment $\mathbf{v} : X \to \mathrm{dom}(t)$, $(t, \mathbf{v}) \models \varphi$.

Thus, one can define $t \models \varphi$ for the tree t and the sentence φ if $(t, \mathbf{v})$ satisfies φ for some assignment, or for all assignments $\mathbf{v} : X \to \mathrm{dom}(t)$.

A sentence φ over Σ *defines* the language $L_\varphi = \{t \in T_\Sigma : t \models \varphi\}$. The class $\mathbf{FO}(<, \mathbf{S_i})$ of $\mathrm{FO}(<, S_i)$-definable tree languages is $\{L_\varphi : \varphi \text{ is a } \mathrm{FO}(<, S_i)\text{-formula}\}$.

The fragment $\mathrm{FO}(<)$ of $\mathrm{FO}(<, S_i)$ is obtained by excluding the successor relations S_i from the syntax. It is clear that $\mathrm{FO}(<)$ is weaker than $\mathrm{FO}(<, S_i)$, since e.g. it cannot distinguish the trees $\sigma_2(\sigma_0, \nu_0)$ and $\sigma_2(\nu_0, \sigma_0)$.

Now we turn to the temporal logic CTL.

The set of CTL-*formulas* over Σ is the least set satisfying the following conditions:

1. for any $\sigma \in \Sigma$, p_σ is a formula;
2. if φ is a formula, then $(\neg\varphi)$ and $(\mathrm{EX}\varphi)$ are also formulas;
3. if φ_1 and φ_2 are formulas, then $(\varphi_1 \vee \varphi_2)$ and $(\mathrm{EU}(\varphi_1, \varphi_2))$ are also formulas.

The semantics of CTL is defined as follows: for a tree $t \in T_\Sigma$ and a CTL-formula φ over Σ we say that t *satisfies* φ, in notation $t \models \varphi$ if and only if one of the following holds:

1. $\varphi = p_\sigma$ and $\text{Root}(t) = \sigma$;
2. $\varphi = (\neg\varphi_1)$ and it is not the case that $t \models \varphi_1$;
3. $\varphi = (\varphi_1 \vee \varphi_2)$ and $t \models \varphi_1$ or $t \models \varphi_2$;
4. $\varphi = (\text{EX}\varphi_1)$ and $t = \sigma(t_1, \ldots, t_n)$ for some $\sigma \in \Sigma_n$ and trees $t_1, \ldots, t_n \in T_\Sigma$, and there exists an index $i \in [n]$ with $t_i \models \varphi_1$;
5. $\varphi = (\text{EU}(\varphi_1, \varphi_2))$ and there exists a node $v \in \text{dom}(t)$ such that both of the following two conditions hold:
   (a) $t|_v \models \varphi_2$;
   (b) for any $u \prec v$, $t|_u \models \varphi_1$.

A CTL-formula φ *defines the tree language* $L_\varphi = \{t \in T_\Sigma : t \models \varphi\}$. **CTL** denotes the class of all CTL-definable tree languages[1].

Two formulas, φ and ψ, of any logic are said to be *equivalent*, denoted $\varphi \equiv \psi$ if they define the same language, i.e. $L_\varphi = L_\psi$.

[1]The reader familiar with the logic CTL may observe the fact the unary modality EG is left out from the definition of the logic. This is due to the fact that for finite trees, $\text{EG}(\varphi)$ is equivalent to $\text{EU}(\varphi, \varphi \wedge \neg\text{EX}\top)$, where $\top$ is a formula satisfied by any tree.

Chapter 2

Future temporal logics and the definability problem

In this chapter we recall from [15] the operator FTL, which associates to each class $\mathcal{L}$ of tree languages a temporal logic FTL($\mathcal{L}$). This class of logics covers a wide range of branching time future temporal logics; for instance, the logic CTL is equivalent to a logic FTL($\mathcal{L}$) for a specific class $\mathcal{L}$ of simple tree languages.

After introducing the logic FTL($\mathcal{L}$), we state its definability problem, and present an algebraic characterization of its expressive power, at least when the parameter class $\mathcal{L}$ consists of regular tree languages and possesses a natural property. This characterization is not effective, but as an application we show decidability of the definability problem of FTL($\mathcal{L}$) for some special classes $\mathcal{L}$ related to fragments of CTL.

In the last section of this chapter we present a game-based, Ehrenfeucht-Fraïssé type approach and provide another characterization of the logic FTL($\mathcal{L}$), this time for an arbitrary finite class $\mathcal{L}$ of tree languages. This last section contains results that are still unpublished.

This chapter deals with variable-free trees only. For the whole section we now fix a rank type R; to avoid trivial situations we assume that $0 \in R$ to ensure that T_Σ is nonempty for any signature Σ (of rank type R), moreover, we also assume that R contains at least one positive integer.

In the definition of the semantics of FTL($\mathcal{L}$) we use a lexicographic ordering of signatures; we assume that each signature Σ (all of them are of rank type R) comes with a fixed lexicographic ordering, denoted $<_\Sigma$ or just $<$ when Σ is understood.

2.1 The logics FTL($\mathcal{L}$)

This section is devoted to the definition of the logics FTL($\mathcal{L}$). For further use, we also state some properties of these logics.

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. If φ is a formula (of depth d), then $(\neg\varphi)$ is also a formula (of depth d).
3. If φ_1 and φ_2 are formulas (having maximal depth d), then $(\varphi_1 \vee \varphi_2)$ is also a formula (of depth d).
4. If Δ is a signature, $L \subseteq T_\Delta$ is a tree language and for each $\delta \in \Delta$, φ_δ is a formula (still over Σ and having maximal depth d), then

$$L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta} \tag{2.1}$$

   is also a formula (of depth $d+1$).

The notion of *subformulas* of a formula φ is defined as usual.

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. $\varphi = (\neg\varphi_1)$ and it is not the case that $t \models \varphi_1$;
3. $\varphi = (\varphi_1 \vee \varphi_2)$ and $t \models \varphi_1$ or $t \models \varphi_2$;
4. $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ 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 (such a relabeling always exists since the signatures Σ and Δ have the common rank type R); a vertex v with $t(v) \in \Sigma_n$ is relabeled to $\delta \in \Delta_n$ if one of the following holds:
   - either $t|_v \models \varphi_\delta$ and δ is the first (least wrt. $<_\Delta$) such element of Δ_n;
   - or $t|_v \not\models \varphi_{\delta'}$ for any $\delta' \in \Delta_n$, and δ is the last (greatest wrt. $<_\Delta$) element of Δ_n.

For any formula φ over Σ, we let L_φ denote the *language defined by* φ: $L_\varphi = \{t \in T_\Sigma : t \models \varphi\}$.

We will consider subsets of formulas associated with 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 (2.1) 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 FTL($\mathcal{L}$).

We say the formulas φ and ψ are *equivalent*, denoted $\varphi \equiv \psi$ if $L_\varphi = L_\psi$. We also 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. The relation $s \equiv_{\mathcal{L}} t$ denotes that s and t satisfy the same set of FTL($\mathcal{L}$)-formulas, regardless of depth.

We will use the Boolean connectives $\wedge$ (conjunction) and $\rightarrow$ (implication) as abbreviations. Moreover, for any signature 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, $t \models \mathtt{tt}_n$ if and only if $\mathrm{Root}(t) \in \Sigma_n$.

It was shown in [15] that the operator **FTL** preserves regularity and is a closure operator on classes of regular languages, i.e. the following holds:

THEOREM 2.1.1 *For any class $\mathcal{L}$ of regular tree languages the following all hold:*

1. $\mathbf{FTL}(\mathcal{L})$ *is a class of regular tree languages;*
2. $\mathcal{L} \subseteq \mathbf{FTL}(\mathcal{L})$*;*
3. $\mathbf{FTL}(\mathbf{FTL}(\mathcal{L})) \subseteq \mathbf{FTL}(\mathcal{L})$*.*

Moreover, if $\mathcal{L}_1$ and $\mathcal{L}_2$ are classes of regular tree languages, then $\mathcal{L}_1 \subseteq \mathcal{L}_2$ implies $\mathbf{FTL}(\mathcal{L}_1) \subseteq \mathbf{FTL}(\mathcal{L}_2)$.

Given a tree $t \in T_\Sigma$ and a family $(\varphi_\delta)_{\delta \in \Delta}$ of formulas over Σ, we say that the family $(\varphi_\delta)_{\delta \in \Delta}$ is *deterministic* over t if for any node $v \in \mathrm{dom}(t)$ there exists at most one $\delta \in \Delta$ with $t|_v \models \varphi_\delta$, moreover, this δ, if exists, is contained in Δ_n, where n is the arity of $t(v)$. The notion is extended to tuples of Σ-trees: a family of formulas is deterministic over a tuple $\underline{t} = (t_1, \ldots, t_n)$ if it is deterministic over each t_i, $i \in [n]$. The family is called deterministic if it is deterministic over any tree $t \in T_\Sigma$.

Given a tree $t \in T_\Sigma$ and a family $(\varphi_\delta)_{\delta \in \Delta}$ of formulas over Σ, we say that the family $(\varphi_\delta)_{\delta \in \Delta}$ is *complete* over t if for any node $v \in \mathrm{dom}(t)$ there exists at least one $\delta \in \Delta_n$ with $t|_v \models \varphi_\delta$, where n is the arity of $t(v)$. A family is complete over a tuple $(t_1, \ldots, t_n)$ of Σ-trees if it is complete over each t_i, $i \in [n]$; and it is complete if it is complete over every $t \in T_\Sigma$.

A formula φ is called *deterministic* if for each subformula of φ of the form (2.1) the family $(\varphi_\delta)_{\delta\in\Delta}$ is deterministic; *completeness* of a formula is defined similarly.

It is clear that when φ is a complete and deterministic formula, then the particular ordering of the signatures does not affect the language L_φ. Also observe that for any class $\mathcal{L}$ of tree languages and FTL($\mathcal{L}$)-formula φ, there exists an equivalent, complete and deterministic FTL($\mathcal{L}$)-formula φ'. This can be shown by induction on the depth of the formula: when the depth is 0, φ has no subformulas of the form (2.1); when $\varphi = (\neg\varphi_1)$ or $\varphi = (\varphi_1 \vee \varphi_2)$, we can define $\varphi' = (\neg\varphi'_1)$ or $\varphi' = (\varphi'_1 \vee \varphi'_2)$, respectively. Finally, when $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, we can define φ' as

$$L(\delta \mapsto \varphi^*_\delta)_{\delta\in\Delta},$$

where for each $\delta \in \Delta_n$,

$$\varphi^*_\delta \quad = \quad \begin{cases} \mathtt{tt}_n \wedge \bigwedge\limits_{\delta'<\delta} (\neg\varphi'_{\delta'}) & \text{if } \delta \text{ is the last element of } \Delta_n; \\ \mathtt{tt}_n \wedge \varphi'_\delta \wedge \bigwedge\limits_{\delta'<\delta} (\neg\varphi'_{\delta'}) & \text{otherwise.} \end{cases}$$

From the definition of the semantics and the formulas $\mathtt{tt}_n$, and by the induction hypothesis, it is clear that the resulting formula is indeed equivalent to φ, and is complete and deterministic.

Thus, for any formula φ of FTL($\mathcal{L}$) there exists an equivalent, complete and deterministic formula φ' of FTL($\mathcal{L}$), having at most the same depth as φ.

The following has been also shown in [15]:

THEOREM 2.1.2 *For any class $\mathcal{L}$ of tree languages, $\mathbf{FTL}(\mathcal{L})$ is closed under the Boolean operations and inverse literal homomorphisms.*

Moreover, when $\mathcal{L}$ is a class of regular tree languages, then $\mathbf{FTL}(\mathcal{L})$ is a literal variety of tree languages if and only if for each quotient L of a language in $\mathcal{L}$ it holds that $L \in \mathbf{FTL}(\mathcal{L})$.

2.2 Products of tree automata

In this section we define the cascade [43], Moore and strict Moore [18] products of tree automata, as well as the connected variants of these notions, giving rise to (connected) cascade, Moore and strict Moore pseudovarieties of finite (connected) tree automata. These pseudovarieties (especially the connected Moore pseudovarieties) will play a fundamental role in the algebraic characterization of the logics FTL($\mathcal{L}$) developed in the next section.

2.2.1 The Moore product and Moore pseudovarieties

In this subsection we define the three product operations mentioned above, and provide several examples for further use.

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

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

for all $\sigma \in \Sigma_n$, $n \in R$ and $(a_1, b_1), \ldots, (a_n, b_n) \in A \times B$. (Note that when $n = 0$, δ is a function of σ).

We state the following lemma:

LEMMA 2.2.1 *Suppose $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are tree automata and $\mathbb{C} = (C, \Sigma) = \mathbb{A} \times_\gamma \mathbb{B}$ is a cascade product. Then for any $\Sigma C X_n$-polynomial symbol p and states $a_1, \ldots, a_n \in A$ of $\mathbb{A}$, there exists a relabeling $p' \in \Sigma A X_n$ and a relabeling $p'' \in \Delta B X_n$ of p such that*

$$p^{\mathbb{C}}\big((a_1, b_1), \ldots, (a_n, b_n)\big) \quad = \quad \big(p'^{\mathbb{A}}(a_1, \ldots, a_n), p''^{\mathbb{B}}(b_1, \ldots, b_n)\big),$$

for any $b_1, \ldots, b_n \in B$. Moreover, p' can be assumed to be the relabeling of p determined by the projection $A \times B \to A$, $(a, b) \mapsto a$.

Proof. Let $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, $a_1, \ldots, a_n$ as given above. We define the relabelings p' and p'' of a polynomial symbol $p \in T_{\Sigma,C}(X_n)$ as follows:

1. If $p = x_i \in X_n$, let $p' = p'' = x_i$.
2. If $p = (a, b) \in A \times B$, let $p' = a$ and $p'' = b$.
3. If $p = \sigma(p_1, \ldots, p_k)$, let $p' = \sigma(p'_1, \ldots, p'_k)$ and $p'' = \delta(p''_1, \ldots, p''_k)$, where $\delta = \gamma_k\big(p_1'^{\mathbb{A}}(a_1, \ldots, a_n), \ldots, p_k'^{\mathbb{A}}(a_1, \ldots, a_n), \sigma\big)$.

It is easy to check that p' and p'' satisfy the required properties for any states $b_1, \ldots, b_n \in B$. ∎

We call a 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 $a_1, \ldots, a_n$. Moore products of the above form will be written as $\mathbb{A} \times_\alpha \mathbb{B}$, i.e. by specifying the function $\alpha : A \times \Sigma \to \Delta$.

We call a Moore product $\mathbb{A} \times_\alpha \mathbb{B}$, $\alpha : A \times \Sigma \to \Delta$ 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 $\sigma \in \Sigma_n$. Accordingly, we will sometimes specify a strict Moore product by the function β.

Note that the direct product is a special case of the Moore product but it is not a special case of the strict Moore product. Any class of finite tree automata which is closed under the Moore product is also closed under the direct product, but the same fact is not true for the strict Moore product: the class of all finite tree automata $\mathbb{A} = (A, \Sigma)$ such that for each $n \in R$ and $a_1, \ldots, a_n$ the cardinality of the set $\{\sigma^{\mathbb{A}}(a_1, \ldots, a_n) : \sigma \in \Sigma_n\}$ is at most 2 is closed under the strict Moore product but not under the direct product.

REMARK 2.2.2 In the classical case, i.e. when $R = \{0, 1\}$, the function β in a strict Moore product $\mathbb{A} \times_\beta \mathbb{B}$ assigns an output symbol in Δ_1 to each pair $(a, 1)$, so that forgetting about the symbols of rank 0, $(\mathbb{A}, \beta)$ may be viewed as a Moore automaton with input alphabet Σ_1 and output alphabet Δ_1. This explains the terminology.

When $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are connected tree automata and γ is a family $(\gamma_n)_{n \in R}$ of functions, $\gamma_n : A^n \times \Sigma \to \Delta_n$, the *connected cascade product* of $\mathbb{A}$ and $\mathbb{B}$ determined by γ, also written as $\mathbb{A} \times_\gamma \mathbb{B}$, is the connected part of the cascade product of $\mathbb{A}$ and $\mathbb{B}$ determined by γ. The notions of *connected Moore product* and *connected strict Moore product* are defined analogously.

Recall that when $\mathbf{V}$ is a class of finite tree automata, then $\mathbf{V}^c$ denotes the class of finite connected tree automata consisting of the connected parts of the members of $\mathbf{V}$. Thus, when $\mathbf{V}$ is a pseudovariety of finite tree automata, $\mathbf{V}^c$ consists of the connected tree automata contained in $\mathbf{V}$ and is a pseudovariety of finite connected tree automata. We will use the superscript ^{c} to indicate that a class is a pseudovariety of finite connected tree automata.

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 $\mathbb{B} \in \mathbf{W}$. Moreover, let $\mathbf{V} \times_M \mathbf{W}$ denote the pseudovariety of finite tree automata generated by all Moore products $\mathbb{A} \times_\alpha \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}$ and $\mathbb{B} \in \mathbf{W}$. Similarly, let $\mathbf{V} \times_s \mathbf{W}$ and $\mathbf{V} \times_c \mathbf{W}$ respectively denote the pseudovarieties of finite tree automata generated by all strict Moore products and cascade products $\mathbb{A} \times_\alpha \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}$ and $\mathbb{B} \in \mathbf{W}$. Moreover,

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

The above notions are extended also to finite connected tree automata as follows. 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 $\mathbb{B} \in \mathbf{W}^c$. Also, let $\mathbf{V}^c \times_M \mathbf{W}^c$ denote the pseudovariety of finite connected tree automata generated by all connected Moore products $\mathbb{A} \times_\alpha \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}^c$ and $\mathbb{B} \in \mathbf{W}^c$. Similarly, let $\mathbf{V}^c \times_s \mathbf{W}^c$ and $\mathbf{V}^c \times_c \mathbf{W}^c$ respectively denote the pseudovarieties of finite connected tree automata generated by all connected strict Moore products and connected cascade products $\mathbb{A} \times_\alpha \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}^c$ and $\mathbb{B} \in \mathbf{W}^c$. Finally, when $\mathbf{K}^c$ is an arbitrary class of finite connected tree automata, let $\langle\mathbf{K}^c\rangle_M$, $\langle\mathbf{K}^c\rangle_s$, $\langle\mathbf{K}^c\rangle_c$ respectively denote the least pseudovariety of finite connected tree automata containing $\mathbf{K}^c$ which is closed under the connected Moore product, the connected strict Moore product, and the connected cascade product. We call pseudovarieties of the form $\langle\mathbf{K}^c\rangle_s$ and $\langle\mathbf{K}^c\rangle_M$ *connected strict Moore pseudovarieties* and *connected Moore pseudovarieties*, respectively, and pseudovarieties of the form $\langle\mathbf{K}^c\rangle_c$ *connected cascade pseudovarieties*.

When $\mathbf{K} = \{\mathbb{A}_1, \ldots, \mathbb{A}_n\}$, for some finite tree automata $\mathbb{A}_1, \ldots, \mathbb{A}_n$, we will write $\langle\mathbb{A}_1, \ldots, \mathbb{A}_n\rangle_M$ for $\langle\mathbf{K}\rangle_M$. The notations $\langle\mathbb{A}_1, \ldots, \mathbb{A}_n\rangle_s$ and $\langle\mathbb{A}_1, \ldots, \mathbb{A}_n\rangle_c$ are used in the same way. Each of these notations is extended to connected tree automata as well. It will always be clear whether $\langle\mathbb{A}_1, \ldots, \mathbb{A}_n\rangle_M$ ($\langle\mathbb{A}_1, \ldots, \mathbb{A}_n\rangle_s$,$\langle\mathbb{A}_1, \ldots, \mathbb{A}_n\rangle_c$, resp.) stands for the generated Moore (strict Moore, cascade, resp.) pseudovariety or for the generated connected Moore (strict Moore, cascade, resp.) pseudovariety.

PROPOSITION 2.2.3 *For any pseudovarieties* $\mathbf{V}$ *and* $\mathbf{W}$ *of finite tree automata,* $\mathbf{V} \times_M \mathbf{W}$ *is the class of all divisors of Moore products* $\mathbb{A} \times_\alpha \mathbb{B}$ *with* $\mathbb{A} \in \mathbf{V}$ *and* $\mathbb{B} \in \mathbf{W}$. *A similar fact holds for* $\mathbf{V} \times_c \mathbf{W}$ *and* $\mathbf{V} \times_s \mathbf{W}$.

Proof. Let $\mathbb{A}, \mathbb{B}$ be finite Σ-tree automata contained in $\mathbf{V}$, $\mathbb{A}' = (A', \Delta^1)$ and $\mathbb{B}' = (B', \Delta^2)$ finite tree automata contained in $\mathbf{W}$, and let $\mathbb{A} \times_\alpha \mathbb{A}'$, $\mathbb{B} \times_\beta \mathbb{B}'$ be Moore products. We show that $(\mathbb{A} \times_\alpha \mathbb{A}') \times (\mathbb{B} \times_\beta \mathbb{B}')$ is isomorphic to a Moore product $\mathbb{C} \times_\gamma \mathbb{C}'$, where $\mathbb{C} \in \mathbf{V}$ and $\mathbb{C}' \in \mathbf{W}$.

Indeed, consider the direct product $\mathbb{C} = \mathbb{A} \times \mathbb{B}$ and define $\mathbb{C}'$ as the tree automaton $(A' \times B', \Delta)$, where for each $n \in R$, $\Delta_n = \Delta^1_n \times \Delta^2_n$ and the function symbols are interpreted componentwise, i.e.

$$(\delta_1, \delta_2)^{\mathbb{C}'}((a'_1, b'_1), \ldots, (a'_n, b'_n)) \quad = \quad (\delta_1^{\mathbb{A}'}(a'_1, \ldots, a'_n), \delta_2^{\mathbb{B}'}(b'_1, \ldots, b'_n))$$

for each $\delta_1 \in \Delta_n^1$, $\delta_2 \in \Delta_n^2$, $a'_1, \ldots, a'_n \in A'$ and $b'_1, \ldots, b'_n \in B'$. It is easy to see that $\mathbb{C}'$ is a direct product of Δ-renamings of $\mathbb{A}'$ and $\mathbb{B}'$.

Now $(\mathbb{A} \times_\alpha \mathbb{A}') \times (\mathbb{B} \times_\beta \mathbb{B}')$ is isomorphic to the Moore product $\mathbb{C} \times_\gamma \mathbb{C}'$ where γ is defined by

$$\gamma\big((a,b),\sigma\big) = \big(\alpha(a,\sigma), \beta(b,\sigma)\big)$$

for each $(a,b) \in C$, $\sigma \in \Sigma$. An isomorphism $\mathbb{C} \times_\gamma \mathbb{C}' \to (\mathbb{A} \times_\alpha \mathbb{A}') \times (\mathbb{B} \times_\beta \mathbb{B}')$ is given by the map $((a,b),(a',b')) \mapsto ((a,a'),(b,b'))$ for all states $((a,b),(a',b'))$ of $\mathbb{C} \times_\gamma \mathbb{C}'$.

It is easy to see that any renaming of a Moore product $\mathbb{A} \times_\alpha \mathbb{B}$ is isomorphic to a Moore product $\mathbb{A}' \times_{\alpha'} \mathbb{B}$ where $\mathbb{A}'$ is a renaming of $\mathbb{A}$. Since any divisor of a divisor of $\mathbb{A}$ is also a divisor of $\mathbb{A}$, the claim is proved for Moore pseudovarieties.

The case of the cascade and the strict Moore product is analogous. ■

It is easy to check that for any pseudovarieties $\mathbf{V}$ and $\mathbf{W}$ of finite tree automata we have $(\mathbf{V} \times_M \mathbf{W})^c = \mathbf{V}^c \times_M \mathbf{W}^c$, and similarly for the strict Moore and the cascade products. Thus Proposition 2.2.3 implies the following:

COROLLARY 2.2.4 *For any pseudovarieties $\mathbf{V}^c$ and $\mathbf{W}^c$ of finite connected tree automata, $\mathbf{V}^c \times_M \mathbf{W}^c$ is the class of all quotients of connected Moore products $\mathbb{A} \times_\alpha \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}^c$ and $\mathbb{B} \in \mathbf{W}^c$. A similar fact holds for $\mathbf{V}^c \times_c \mathbf{W}^c$ and $\mathbf{V}^c \times_s \mathbf{W}^c$.*

REMARK 2.2.5 The Moore product is associative on pseudovarieties: For any pseudovarieties of finite tree automata $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$,

$$(\mathbf{V}_1 \times_M \mathbf{V}_2) \times_M \mathbf{V}_3 \quad = \quad \mathbf{V}_1 \times_M (\mathbf{V}_2 \times_M \mathbf{V}_3).$$

A similar fact holds for the cascade product.

We now give some examples of Moore pseudovarieties. 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 $[n]$. Moreover, we say that a Σ-tree automaton $\mathbb{A}$ is *idempotent* if it satisfies the equations

$$\sigma^{\mathbb{A}}(x_1, \ldots, x_n) \quad = \quad \sigma^{\mathbb{A}}(y_1, \ldots, y_n)$$

for all $\sigma \in \Sigma_n$, $n \in R$, $n > 0$ where $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}$. Note that $\mathbb{A}$ is idempotent if and only if for each $\sigma \in \Sigma_n$ and $a_1, \ldots, a_n \in A$, $\sigma^{\mathbb{A}}(a_1, \ldots, a_n)$ depends only on the set $\{a_1, \ldots, a_n\}$. In particular, any idempotent tree automaton is commutative.

PROPOSITION 2.2.6 *Suppose that $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ is a Moore product. If $\mathbb{A}$ and $\mathbb{B}$ are commutative, or idempotent, then so is $\mathbb{C}$.*

Proof. Assume that $\mathbb{A}$ is a Σ-tree automaton and $\mathbb{B}$ is a Δ-tree automaton, so that $\alpha : A \times \Sigma \to \Delta$. Suppose that $\mathbb{A}$ and $\mathbb{B}$ are commutative. Then for all $\sigma \in \Sigma_n$, $n \in R$, $n > 0$, $a_1, \ldots, a_n \in A$ and permutations π,

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

Thus, when $b_1, \ldots, b_n \in B$ and $\delta = \alpha(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \sigma)$,

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

The argument is similar in the idempotent case. ■

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

PROPOSITION 2.2.7 *Suppose that $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ is a Moore product. If $\mathbb{A}$ and $\mathbb{B}$ are stutter invariant, then so is $\mathbb{C}$.*

Proof. Assume that $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are tree automata, so that $\alpha : A \times \Sigma \to \Delta$. Suppose that $\mathbb{A}$ and $\mathbb{B}$ are stutter invariant. Let $\sigma \in \Sigma_n$, $n > 0$, $n \in R$, $a_1, \ldots, a_n \in A$, $b_1, \ldots, b_n \in B$. Since $\mathbb{A}$ is stutter invariant, we have

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

Thus,

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

Let δ denote this symbol. Then, using the assumption that $\mathbb{B}$ is also stutter invariant,

$$\begin{aligned} & \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) \\ = \; & (\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \delta^{\mathbb{B}}(b_1, \ldots, b_n)) \\ = \; & (\sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, \sigma^{\mathbb{A}}(a_1, \ldots, a_n)), \delta^{\mathbb{B}}(b_1, \ldots, b_{n-1}, \delta^{\mathbb{B}}(b_1, \ldots, b_n))) \\ = \; & \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_{n-1}, b_{n-1}), \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)))). \end{aligned}$$

■

It is clear that any divisor of a commutative tree automaton is commutative; similarly, idempotence and stutter invariance are also preserved under taking divisors. This implies the following:

COROLLARY 2.2.8 *The classes of commutative, idempotent, and stutter invariant finite tree automata, denoted* **Com**, **Idem** *and* **Stu**, *respectively, are all Moore pseudovarieties, as is any intersection of these classes.*

It is easy to check that when a class $\mathbf{V}$ of finite tree automata forms a Moore (cascade, resp.) pseudovariety of finite tree automata, then the class $\mathbf{V}^c$ forms a connected Moore (connected cascade, resp.) pseudovariety of finite connected tree automata. Thus the following also holds:

COROLLARY 2.2.9 *The classes of commutative, idempotent, and stutter invariant finite connected tree automata, denoted* $\mathbf{Com}^c$, $\mathbf{Idem}^c$ *and* $\mathbf{Stu}^c$, *respectively, are all connected Moore pseudovarieties, as is any intersection of these classes.*

2.2.2 Definite tree automata

Let Σ be a signature, $k \geq 0$ an integer and $s, t \in T_\Sigma(X)$ trees. We say that *s and t agree up to depth k* if one of the following holds:

1. $k = 0$ and $\mathrm{Root}(s) = \mathrm{Root}(t)$;
2. $s = t \in X$;
3. $k > 0$, $s = \sigma(s_1, \ldots, s_n)$, $t = \sigma(t_1, \ldots, t_n)$ holds for some $\sigma \in \Sigma_n$ and ΣX-trees $s_1, \ldots, s_n, t_1, \ldots, t_n$, moreover, the trees s_i and t_i agree up to depth $k-1$ for each i.

Agreement of polynomial symbols $p, q \in T_{\Sigma,A}(X)$ up to some depth k is defined analogously.

We say that a Σ-tree automaton $\mathbb{A}$ is *k-definite*, for some $k \geq 1$, if for all polynomial symbols $p, q \in T_{\Sigma,A}$, if p and q agree up to depth $k-1$, then $p^{\mathbb{A}} = q^{\mathbb{A}}$. By extension, 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. Thus, $\mathbf{D}_k^c$ and $\mathbf{D}^c$ denote the class of all finite k-definite connected tree automata and the class of all finite definite connected tree automata, respectively.

It is clear that each $\mathbf{D}_k$ is a pseudovariety of finite tree automata as is the class $\mathbf{D}$.

We call a tree language $L \subseteq T_\Sigma$ *k-definite* for some $k \geq 1$ if for any two trees $s, t \in T_\Sigma$, if s and t agree up to depth $k-1$, then $s \in L \Leftrightarrow t \in L$. Also, we say that a tree language is *definite* if it is k-definite for some k. For each k, let $\mathcal{D}_k$ denote the class of all k-definite tree languages and let $\mathcal{D}$ be the union of the classes $\mathcal{D}_k$, so that $\mathcal{D}$ is the class of all definite tree languages. Then each $\mathcal{D}_k$ is the literal variety of tree languages corresponding to $\mathbf{D}_k^c$ by the Eilenberg correspondence, moreover, $\mathcal{D}$ is the literal variety corresponding to $\mathbf{D}^c$. Definite tree languages were studied e.g. in [33].

The following fact was shown in [12]:

PROPOSITION 2.2.10 *For all $n, m \geq 1$, it holds that $\mathbf{D}_n \times_c \mathbf{D}_m = \mathbf{D}_{n+m}$ and thus $\mathbf{D}_n^c \times_c \mathbf{D}_m^c = \mathbf{D}_{n+m}^c$. Hence, $\mathbf{D} = \langle \mathbf{D}_1 \rangle_c$ and $\mathbf{D}^c = \langle \mathbf{D}_1^c \rangle_c$.*

Observe that a Σ-tree automaton $\mathbb{A}$ is 1-definite if and only if $\mathrm{Img}_{\mathbb{A}}(\sigma)$ is a singleton set for each $\sigma \in \Sigma$.

PROPOSITION 2.2.11 *$\mathbf{D}_1$ is a Moore pseudovariety of finite tree automata.*

Proof. We only prove that if $\mathbb{A}$ and $\mathbb{B}$ are in $\mathbf{D}_1$, and $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ is any Moore product of $\mathbb{A}$ and $\mathbb{B}$, then $\mathbb{C}$ is also in $\mathbf{D}_1$. Let us write in more detail $\mathbb{A} = (A, \Sigma)$, $\mathbb{B} = (B, \Delta)$, $\mathbb{C} = (C, \Sigma)$ and $\alpha : A \times \Sigma \to \Delta$. Assume that $\sigma \in \Sigma_n$, $n \in R$ and $(a_1, b_1), \ldots, (a_n, b_n)$ and $(c_1, d_1), \ldots, (c_n, d_n)$ are all in C. Then

$$\begin{aligned} \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) &= (\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \delta^{\mathbb{B}}(b_1, \ldots, b_n)) \\ \sigma^{\mathbb{C}}((c_1, d_1), \ldots, (c_n, d_n)) &= (\sigma^{\mathbb{A}}(c_1, \ldots, c_n), \delta'^{\mathbb{B}}(d_1, \ldots, d_n)) \end{aligned}$$

where

$$\begin{aligned} \delta &= \alpha(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \sigma) \\ \delta' &= \alpha(\sigma^{\mathbb{A}}(c_1, \ldots, c_n), \sigma). \end{aligned}$$

However, $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) = \sigma^{\mathbb{A}}(c_1, \ldots, c_n)$, since $\mathbb{A} \in \mathbf{D}_1$, and thus $\delta = \delta'$. But then $\delta^{\mathbb{B}}(b_1, \ldots, b_n) = \delta'^{\mathbb{B}}(d_1, \ldots, d_n)$, since $\mathbb{B} \in \mathbf{D}_1$. Thus

$$\sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) = \sigma^{\mathbb{C}}((c_1, d_1), \ldots, (c_n, d_n)).$$

■

The *Boolean signature* Bool has exactly two symbols of rank n, for each $n \in R$, the symbols $\uparrow_n$ and $\downarrow_n$.

PROPOSITION 2.2.12 *The Moore pseudovariety* $\mathbf{D}_1$ *of finite tree automata is generated by the two-element* Bool*-tree automaton* $\mathbb{D}_0 = \big(\{0,1\}, \mathrm{Bool}\big)$*, where for each* $n \in R$ *and* $a_1, \ldots, a_n \in \{0,1\}$ *we have*

$$\begin{aligned} \uparrow_n^{\mathbb{D}_0}(a_1, \ldots, a_n) &= 1 \text{ and} \\ \downarrow_n^{\mathbb{D}_0}(a_1, \ldots, a_n) &= 0; \end{aligned}$$

that is, $\langle \mathbb{D}_0 \rangle_M = \mathbf{D}_1$.

For any Σ, let $\mathbb{A}_\Sigma$ denote the Σ-tree automaton whose state set is Σ such that $\sigma^{\mathbb{A}_\Sigma}(\sigma_1, \ldots, \sigma_n) = \sigma$ for all $\sigma_1, \ldots, \sigma_n \in \Sigma$ and $\sigma \in \Sigma_n$. It is clear that $\mathbb{A}_\Sigma$ is in $\mathbf{D}_1^c$. Moreover, any finite connected Σ-tree automaton $\mathbb{B} \in \mathbf{D}_1^c$ is a quotient of $\mathbb{A}_\Sigma$. As an extension, when A is some set, let $\mathbb{A}_\Sigma^A$ denote the tree automaton whose state set is $\Sigma + A$ (that is, the disjoint union of Σ and A) and where each function symbol $\sigma \in \Sigma$ is interpreted as a constant function with value σ. Clearly, $\mathbb{A}_\Sigma^A$ is contained in $\mathbf{D}_1$ for any signature Σ and finite set A. Also observe that any 1-definite Σ-tree automaton $\mathbb{A} = (A, \Sigma)$ is a quotient of $\mathbb{A}_\Sigma^A$.

PROPOSITION 2.2.13 *For any pseudovariety* $\mathbf{V}$ *of finite tree automata we have*

$$\mathbf{V} \times_M \mathbf{D}_1 = \mathbf{D}_1 \times \mathbf{V}.$$

Consequently, $\mathbf{V}^c \times_M \mathbf{D}_1^c = \mathbf{D}_1^c \times \mathbf{V}^c$.

Proof. Suppose that $\mathbb{A} = (A, \Sigma)$ is a Σ-tree automaton in $\mathbf{V}$ and $\mathbb{B} = (B, \Delta)$ is a Δ-tree automaton in $\mathbf{D}_1$. Consider a Moore product $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$. Since $\mathbb{B}$ is in $\mathbf{D}_1$, for each $\delta \in \Delta_n$ there exists some $\overline{\delta} \in B$ such that $\mathrm{Img}_{\mathbb{B}}(\delta) = \{\overline{\delta}\}$. Thus, $\mathbb{C}$ is a quotient of the direct product $\mathbb{A}_\Sigma^B \times \mathbb{A}$: a homomorphism is given by $(b, a) \mapsto (a, b)$ and $(\sigma, a) \mapsto (a, \overline{\alpha(a, \sigma)})$ for each $a \in A$, $b \in B$ and $\sigma \in \Sigma$. ■

PROPOSITION 2.2.14 *For any pseudovariety* $\mathbf{V}$ *of finite tree automata,*

$$\mathbf{D}_1 \times_M \mathbf{V} = \mathbf{D}_1 \times \mathbf{V}.$$

Thus, $\mathbf{D}_1^c \times_M \mathbf{V}^c = \mathbf{D}_1^c \times \mathbf{V}^c$ *also holds for any pseudovariety* $\mathbf{V}^c$ *of finite connected tree automata.*

Proof. We only have to prove that $\mathbf{D}_1 \times_M \mathbf{V} \subseteq \mathbf{D}_1 \times \mathbf{V}$. Let $\mathbb{A} = (A, \Sigma)$ be a finite 1-definite Σ-tree automaton, $\mathbb{B} = (B, \Delta) \in \mathbf{V}$ a finite Δ-tree automaton and $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ a Moore product.

Let $\mathbb{B}'$ be the renaming of $\mathbb{B}$ determined by the rank preserving map $\sigma \mapsto \alpha(\overline{\sigma}, \sigma)$, where $\overline{\sigma}$ is the unique element of $\mathrm{Img}_{\mathbb{A}}(\sigma)$. Then $\mathbb{C}$ is a homomorphic image of the direct product $\mathbb{A}_\Sigma^A \times \mathbb{B}'$: a homomorphism is given by $(a, b) \mapsto (a, b)$ and $(\sigma, b) \mapsto (\overline{\sigma}, b)$ for each $a \in A$, $b \in B$ and $\sigma \in \Sigma$. ■

COROLLARY 2.2.15 *For any Moore pseudovariety* $\mathbf{V}$ *of finite tree automata,* $\mathbf{D}_1 \times \mathbf{V}$ *is a Moore pseudovariety. Thus,* $\langle \mathbf{D}_1 \cup \mathbf{V} \rangle_M = \mathbf{D}_1 \times \mathbf{V}$. *Analogous facts hold for connected Moore pseudovarieties.*

Proof.

$$\begin{aligned}
(\mathbf{D}_1 \times \mathbf{V}) \times_M (\mathbf{D}_1 \times \mathbf{V}) &\subseteq (\mathbf{D}_1 \times_M \mathbf{V}) \times_M (\mathbf{D}_1 \times_M \mathbf{V}) \\
&= \mathbf{D}_1 \times_M (\mathbf{V} \times_M \mathbf{D}_1) \times_M \mathbf{V} \\
&= \mathbf{D}_1 \times_M (\mathbf{D}_1 \times_M \mathbf{V}) \times_M \mathbf{V} \\
&= (\mathbf{D}_1 \times_M \mathbf{D}_1) \times_M (\mathbf{V} \times_M \mathbf{V}) \\
&= \mathbf{D}_1 \times_M \mathbf{V} \\
&= \mathbf{D}_1 \times \mathbf{V}.
\end{aligned}$$

■

2.2.3 Relating the three products

In this subsection we relate the Moore product to the strict Moore product and to the cascade product. As a first observation we note that the Moore product is strictly weaker than the cascade product, since $\langle \mathbb{D}_0 \rangle_M = \mathbf{D}_1 \subset \mathbf{D} = \langle \mathbb{D}_0 \rangle_c$.

PROPOSITION 2.2.16 *For any pseudovarieties* $\mathbf{V}$ *and* $\mathbf{W}$ *of finite tree automata,*

$$\mathbf{V} \times_M \mathbf{W} \quad \subseteq \quad (\mathbf{V} \times \mathbf{D}_1) \times_s \mathbf{W}.$$

Thus, if $\mathbf{D}_1 \subseteq \mathbf{V}$, *then* $\mathbf{V} \times_s \mathbf{W} = \mathbf{V} \times_M \mathbf{W}$.

Proof. Let $\mathbb{A}$ be a Σ-tree automaton in $\mathbf{V}$, $\mathbb{B}$ a Δ-tree automaton in $\mathbf{W}$, and let $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ be a Moore product. It suffices to show that there exists a strict Moore product $\mathbb{C}' = \mathbb{A}' \times_\beta \mathbb{B}$ with $\mathbb{A}' \in \mathbf{V} \times \mathbf{D}_1$ which can be mapped homomorphically onto $\mathbb{C}$.

Let $\mathbb{A}' = \mathbb{A} \times \mathbb{A}_\Sigma$ and let β be the function $\beta((a, \sigma), n) = \alpha(a, \sigma)$, for all $a \in A$ and $\sigma \in \Sigma_n$. When $\sigma \notin \Sigma_n$, define $\beta((a, \sigma), n)$ arbitrarily. A homomorphism $\mathbb{C}' \to \mathbb{C}$ is defined by $((a, \sigma), b) \mapsto (a, b)$. ■

COROLLARY 2.2.17 *For any pseudovariety* $\mathbf{V}$ *of finite tree automata, if* $\mathbf{D}_1 \subseteq \mathbf{V}$ *then* $\langle \mathbf{V} \rangle_M = \langle \mathbf{V} \rangle_s$.

Now we turn our attention to the cascade product.

Suppose that $\mathbb{A}$ and $\mathbb{B}$ are finite Σ-tree and Δ-tree automata, respectively, and consider a cascade product $\mathbb{A} \times_\gamma \mathbb{B}$. We will give two constructions of a Moore product $\mathbb{A}_0 \times_\alpha \mathbb{B}$ such that $\mathbb{A} \times_\gamma \mathbb{B}$ is a quotient of $\mathbb{A}_0 \times_\alpha \mathbb{B}$. In the first construction, $\mathbb{A}_0 = \mathbb{A}_\gamma$ will be a cascade product of $\mathbb{A}$ and a finite 1-definite tree automaton. In the second construction, $\mathbb{A}_0 = \mathrm{Del}(\mathbb{A})$ will be a divisor of a Moore product of $\mathbb{A}$ and a finite 2-definite tree automaton.

Define $\mathbb{A}_\gamma = (A_\gamma, \Sigma)$ as the cascade product $\mathbb{A} \times_\gamma \mathbb{A}_\Delta$. Then, the function symbols are interpreted as

$$\sigma^{\mathbb{A}_\gamma}\big((a_1, \delta_1), \ldots, (a_n, \delta_n)\big) \quad = \quad \big(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \gamma_n(a_1, \ldots, a_n, \sigma)\big),$$

for all $\sigma \in \Sigma_n$, $(a_i, \delta_i) \in A \times \Delta$, $i \in [n]$.

LEMMA 2.2.18 *Any cascade product* $\mathbb{A} \times_\gamma \mathbb{B}$ *of a* Σ*-tree automaton* $\mathbb{A}$ *and a* Δ*-tree automaton* $\mathbb{B}$ *is a quotient of a strict Moore product* $\mathbb{A}_\gamma \times_\beta \mathbb{B}$.

Proof. We define β so that $\beta((a, \delta), n) = \delta$, for all $(a, \delta) \in A_\gamma$ with $\delta \in \Delta_n$. A homomorphism $\mathbb{A}_\gamma \times_\beta \mathbb{B} \to \mathbb{A} \times_\gamma \mathbb{B}$ maps a state $((a, \delta), b)$ of $\mathbb{A}_\gamma \times_\beta \mathbb{B}$ to (a, b). ■

COROLLARY 2.2.19 *For any pseudovarieties* $\mathbf{V}$ *and* $\mathbf{W}$ *of finite tree automata,*

$$\mathbf{V} \times_c \mathbf{W} \quad \subseteq \quad (\mathbf{V} \times_c \mathbf{D}_1) \times_s \mathbf{W}.$$

Thus, if $\mathbf{V} \times_c \mathbf{D}_1 \subseteq \mathbf{V}$, *then* $\mathbf{V} \times_c \mathbf{W} = \mathbf{V} \times_M \mathbf{W} = \mathbf{V} \times_s \mathbf{W}$.

Proof. We have to show that each cascade product $\mathbb{A} \times_\gamma \mathbb{B}$ with $\mathbb{A} \in \mathbf{V}$ and $\mathbb{B} \in \mathbf{W}$ is contained in $(\mathbf{V} \times_c \mathbf{D}_1) \times_s \mathbf{W}$. But by Lemma 2.2.18, $\mathbb{A} \times_\gamma \mathbb{B}$ is a quotient of a strict Moore product $\mathbb{A}_\gamma \times_\beta \mathbb{B}$, and clearly, $\mathbb{A}_\gamma \in \mathbf{V} \times_c \mathbf{D}_1$. ■

COROLLARY 2.2.20 *Suppose that* $\mathbf{V}$ *is a Moore pseudovariety with* $\mathbf{D}_1 \subseteq \mathbf{V}$. *Then* $\mathbf{V}$ *is a cascade pseudovariety if and only if* $\mathbf{V} \times_c \mathbf{D}_1 \subseteq \mathbf{V}$.

Proof. Suppose that $\mathbf{V} \times_c \mathbf{D}_1 \subseteq \mathbf{V}$. Then $\mathbf{V} \times_c \mathbf{V} = \mathbf{V} \times_M \mathbf{V}$, so that $\mathbf{V}$ is a Moore pseudovariety if and only if it is a cascade pseudovariety. ■

We now turn to the second construction. Let $\mathbb{A} = (A, \Sigma)$ be a finite Σ-tree automaton. We denote the set $A \cup \bigcup_{n \in R}(\Sigma_n \times A^n)$ by $\hat{A}$.

Let $\hat{\Sigma}$ denote the signature with

$$\hat{\Sigma}_n \quad = \quad \{\sigma_a : \sigma \in \Sigma_n, a \in \mathrm{Img}_{\mathbb{A}}(\sigma)\},$$

for each $n \in R$. Note that since $\mathbb{A}$ is finite, $\hat{\Sigma}$ is also finite, and thus a signature (of rank type R).

Now we construct a finite $\hat{\Sigma}$-tree automaton $\hat{\mathbb{A}} = (\hat{A}, \hat{\Sigma})$ as follows. For any $\sigma_a \in \hat{\Sigma}_n$ and $b_1, \ldots, b_n \in \hat{A}$, let

$$\sigma_a^{\hat{\mathbb{A}}}(b_1, \ldots, b_n) \quad = \quad \begin{cases} (\sigma, a_1, \ldots, a_n) & \text{if } \sigma^{\mathbb{A}}(a_1, \ldots, a_n) = a; \\ (\sigma, c_1, \ldots, c_n) & \text{otherwise,} \end{cases}$$

where for each $i \in [n]$,

$$a_i \quad = \quad \begin{cases} b_i & \text{if } b_i \in A; \\ \sigma_i^{\mathbb{A}}(a_{i1}, \ldots, a_{ik_i}) & \text{if } b_i = (\sigma_i, a_{i1}, \ldots, a_{ik_i}) \text{ for some } \sigma_i \in \Sigma_{k_i}, a_{i1}, \ldots, a_{ik_i} \in A, \end{cases}$$

and $c_1, \ldots, c_n \in A$ are arbitrary fixed elements of A, depending only on σ and a, with $\sigma^{\mathbb{A}}(c_1, \ldots, c_n) = a$.

The following fact is straightforward from the above definition.

LEMMA 2.2.21 *For any (variable-free) $\hat{\Sigma}\hat{A}$-polynomial symbol p with $\mathrm{Root}(p) = \sigma_a$ for some $\sigma \in \Sigma_n$ and $a \in A$ it holds that $p^{\hat{\mathbb{A}}} = (\sigma, a_1, \ldots, a_n)$ for some $a_1, \ldots, a_n$ with $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) = a$.*

With this lemma, we can prove the following:

LEMMA 2.2.22 *For any finite tree automaton $\mathbb{A}$, the tree automaton $\hat{\mathbb{A}}$ is contained in $\mathbf{D}_2$.*

Proof. Let p, q be $\hat{\Sigma}\hat{A}$-polynomial symbols that agree up to depth one. We show that $p^{\hat{\mathbb{A}}} = q^{\hat{\mathbb{A}}}$.

When $\mathrm{Root}(p) \in \hat{A}$, then from $\mathrm{Root}(p) = \mathrm{Root}(q)$ this clearly holds. Otherwise, write $p = \sigma_a(p_1, \ldots, p_n)$ and $q = \sigma_a(q_1, \ldots, q_n)$. Then for each $i \in [n]$, $\mathrm{Root}(p_i) = \mathrm{Root}(q_i)$. We define the states $a_1, \ldots, a_n \in A$ as follows: let

$$a_i \quad = \quad \begin{cases} \sigma_i^{\mathbb{A}}(a_{i1}, \ldots, a_{ik_i}) & \text{if } \mathrm{Root}(p_i) = (\sigma_i, a_{i1}, \ldots, a_{ik_i}) \in \Sigma_{k_i} \times A^{k_i}; \\ b_i & \text{if } \mathrm{Root}(p_i) = \sigma_{ib_i} \text{ for some } \sigma_i \in \Sigma, b_i \in \mathrm{Img}_{\mathbb{A}}(\sigma_i); \\ p_i & \text{otherwise (i.e. when } p_i \in A). \end{cases}$$

Then,

$$p^{\hat{\mathbb{A}}} = q^{\hat{\mathbb{A}}} \quad = \quad \begin{cases} (\sigma, a_1, \ldots, a_n) & \text{if } \sigma^{\mathbb{A}}(a_1, \ldots, a_n) = a; \\ (\sigma, c_1, \ldots, c_n) & \text{otherwise,} \end{cases}$$

where the states c_i are fixed elements depending only on σ and a. ■

Suppose that $\mathbb{A}$ is a finite Σ-tree automaton. Then the *delay automaton of* $\mathbb{A}$ is the following Σ-tree automaton $\mathrm{Del}(\mathbb{A})$. The set of states of $\mathrm{Del}(\mathbb{A})$ is $\hat{A}$ (thus, $\mathrm{Del}(\mathbb{A})$ is also finite). The operations are defined as follows. Given states z_i, $i \in [n]$ and $\sigma \in \Sigma_n$, let

$$\sigma^{\mathrm{Del}(\mathbb{A})}(z_1, \ldots, z_n) \quad = \quad (\sigma, a_1, \ldots, a_n),$$

where

$$a_i \quad = \quad \begin{cases} z_i & \text{if } z_i \in A; \\ \sigma_i^{\mathbb{A}}(a_{i1}, \ldots, a_{ik_i}) & \text{if } z_i = (\sigma_i, a_{i1}, \ldots, a_{ik_i}) \in \Sigma_{k_i} \times A^{k_i}. \end{cases}$$

Note that $\mathbb{A}$ is a quotient of $\mathrm{Del}(\mathbb{A})$, a homomorphism $\mathrm{Del}(\mathbb{A}) \to \mathbb{A}$ is the map $(\sigma, a_1, \ldots, a_n) \mapsto \sigma^{\mathbb{A}}(a_1, \ldots, a_n)$ for each $\sigma \in \Sigma_n$, $a_1, \ldots, a_n \in A$ and $a \mapsto a$ for each $a \in A$. More generally, we show that the delay automaton and the strict Moore product together can be used to simulate a cascade product:

LEMMA 2.2.23 *Any cascade product $\mathbb{A} \times_\gamma \mathbb{B}$ of a finite Σ-tree automaton $\mathbb{A}$ and a finite Δ-tree automaton $\mathbb{B}$ is a quotient of a strict Moore product $\mathrm{Del}(\mathbb{A}) \times_\beta \mathbb{B}$.*

Proof. We define β as

$$\beta((\sigma, a_1, \ldots, a_n), n) \quad = \quad \gamma(a_1, \ldots, a_n, \sigma),$$

for all $a_1, \ldots, a_n \in A$ and $\sigma \in \Sigma_n$. A homomorphism $\mathrm{Del}(\mathbb{A}) \times_\beta \mathbb{B} \to \mathbb{A} \times_\gamma \mathbb{B}$ maps $((\sigma, a_1, \ldots, a_n), b)$ to $(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), b)$, and (a, b) to (a, b) for all $\sigma \in \Sigma_n$, $a, a_1, \ldots, a_n \in A$ and $b \in B$. ■

REMARK 2.2.24 Actually $\mathbb{A}_\gamma$ is a quotient of $\mathrm{Del}(\mathbb{A})$, so Lemma 2.2.23 follows from Lemma 2.2.18.

LEMMA 2.2.25 *For any finite Σ-tree automaton $\mathbb{A}$, the tree automaton $\mathrm{Del}(\mathbb{A})$ divides the Moore product $\mathbb{B} = \mathbb{A} \times_\alpha \hat{\mathbb{A}}$ determined by the map $\alpha(a, \sigma) = \sigma_a$, for all $a \in A$ and $\sigma \in \Sigma$.*

Proof. Let $\mathbb{B}'$ be the subautomaton of $\mathbb{B}$ consisting the states of the form (a, a) and of the form $(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), (\sigma, a_1, \ldots, a_n))$. A homomorphism $\mathbb{B}' \to \mathrm{Del}(\mathbb{A})$ is given by the map $(a, a) \mapsto a$ for each $a \in A$ and

$$(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), (\sigma, a_1, \ldots, a_n)) \quad \mapsto \quad (\sigma, a_1, \ldots, a_n)$$

for all $\sigma \in \Sigma_n$, $n \in R$, $a_1, \ldots, a_n \in A$. ■

COROLLARY 2.2.26 *For any pseudovariety* $\mathbf{V}$ *of finite tree automata and tree automaton* $\mathbb{A} \in \mathbf{V}$, $\mathrm{Del}(\mathbb{A})$ *is contained in* $\mathbf{V} \times_M \mathbf{D}_2$.

COROLLARY 2.2.27 *For any pseudovarieties* $\mathbf{V}$ *and* $\mathbf{W}$ *of finite tree automata,*

$$\mathbf{V} \times_c \mathbf{W} \quad \subseteq \quad (\mathbf{V} \times_M \mathbf{D}_2) \times_s \mathbf{W}.$$

Thus, if $\mathbf{V} \times_M \mathbf{D}_2 \subseteq \mathbf{V}$, *then* $\mathbf{V} \times_c \mathbf{W} = \mathbf{V} \times_M \mathbf{W} = \mathbf{V} \times_s \mathbf{W}$.

So finally, we have the following:

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

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

In particular, from $\mathbf{D} = \langle \mathbf{D} \rangle_c$ *we have* $\mathbf{D} = \langle \mathbf{D}_2 \rangle_M$.

Analogous facts hold for classes of finite connected tree automata.

Summarizing the results of this subsection, we conclude the section with the following corollary:

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* $\mathbf{D}$.

2.3 Definability and membership

In this section we state a characterization theorem that establishes a correspondence between the definability problem of logics of the form FTL($\mathcal{L}$) and the membership problem of connected Moore pseudovarieties of finite connected tree automata, at least in the case when $\mathcal{L}$ is a class of regular tree languages satisfying a natural property.

When $\mathcal{L}$ is a class of tree languages, the *definability problem* of FTL($\mathcal{L}$) is stated as follows: given a tree language L, is L contained in $\mathbf{FTL}(\mathcal{L})$?

Also, when $\mathbf{V}^c$ is a pseudovariety of finite connected tree automata, the *membership* problem of $\mathbf{V}^c$ is stated as follows: given a finite connected tree automaton $\mathbb{A}$, is it contained in $\mathbf{V}^c$?

Following [15], we also associate a logic 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 FTL($\mathbf{K}^c$) be the logic FTL($\mathcal{L}_{\mathbf{K}^c}$) and define $\mathbf{FTL}(\mathbf{K}^c) = \mathbf{FTL}(\mathcal{L}_{\mathbf{K}^c})$. In [15] it was shown that for any class $\mathcal{L}$ of regular tree languages, $\mathbf{FTL}(\mathcal{L})$ is a literal variety of tree languages if and only if $\mathbf{FTL}(\mathcal{L}) = \mathbf{FTL}(\mathbf{K}^c)$ for some class $\mathbf{K}^c$ of finite connected tree automata.

Our aim is to show that for every class $\mathbf{K}^c$ of finite connected tree automata, a language L is definable in the logic FTL($\mathbf{K}^c$) if and only if its minimal tree automaton $\mathbb{A}_L$ belongs to $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$, the least connected Moore pseudovariety of finite connected tree automata containing $\mathbf{K}^c$ and the 1-definite connected tree automata. As a consequence of this fact we obtain that the Eilenberg correspondence establishes a bijection between connected Moore pseudovarieties of finite connected tree automata containing $\mathbf{D}_1^c$ and literal varieties of tree languages of the form $\mathbf{FTL}(\mathbf{K}^c)$. Moreover, we obtain that for any class $\mathcal{L}$ of regular tree languages such that any quotient of each language in $\mathcal{L}$ belongs to $\mathbf{FTL}(\mathcal{L})$, a language L is in $\mathbf{FTL}(\mathcal{L})$ if and only if $\mathbb{A}_L$ is in the least connected Moore pseudovariety of finite connected tree automata containing $\mathbf{D}_1^c$ and the minimal automata of the languages in $\mathcal{L}$.

We note that the following are equivalent for any regular tree language L and class $\mathbf{V}$ of finite tree automata:

1. $\mathbb{A}_L$ is contained in the least connected Moore pseudovariety of finite connected tree automata containing $\mathbf{V}^c$;

2. $\mathbb{A}_L$ is contained in the least Moore pseudovariety of finite tree automata containing $\mathbf{V}$.

(This is due to the fact that the two above classes contain the same connected tree automata and for any regular tree language L, the minimal automaton $\mathbb{A}_L$ is connected).

PROPOSITION 2.3.1 *Suppose that $\mathcal{L}$ is a class of regular tree languages, $\mathbb{A}$ is a finite connected Σ-tree automaton, $\mathbb{B}$ is a finite connected Δ-tree automaton and $\mathbb{C} = (C, \Sigma) = \mathbb{A} \times_\alpha \mathbb{B}$ is a connected Moore product of $\mathbb{A}$ and $\mathbb{B}$. If every language recognizable by $\mathbb{A}$ or $\mathbb{B}$ belongs to $\mathbf{FTL}(\mathcal{L})$, then every language recognizable by $\mathbb{C}$ also belongs to $\mathbf{FTL}(\mathcal{L})$.*

Proof. It suffices to show that for each $(a, b) \in C$, the language $h^{-1}((a, b))$ belongs to $\mathbf{FTL}(\mathcal{L})$, where h denotes the unique homomorphism $T_\Sigma \to \mathbb{C}$. Recall that every connected Moore product can be viewed as a connected cascade product. Let $\overline{\alpha}$ denote the family of functions $A^n \times \Sigma_n \to \Delta_n$, $n \in R$ determined by α such that $\mathbb{A} \times_{\overline{\alpha}} \mathbb{B}$ is the connected cascade product corresponding to the connected Moore product $\mathbb{A} \times_\alpha \mathbb{B}$. By assumption, for each $a \in A$ there exists a formula τ_a over Σ in FTL($\mathcal{L}$) defining the tree language $h^{-1}(\pi^{-1}(a))$, where π denotes the projection $C \to A$, $(a, b) \mapsto a$. For each $b \in B$, let $T_b = \{s \in T_\Delta : s^{\mathbb{B}} = b\}$.

It is shown in [15] that $h^{-1}((a, b))$ is definable by the formula

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

where

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

Here for each i, $X_i\tau_{a_i}$ holds for a tree if and only if its ith immediate subtree exists and satisfies the formula τ_{a_i}. But since our connected cascade product is derived from a connected Moore product, we have $\overline{\alpha_n}(a_1, \ldots, a_n, \sigma) = \delta$ if and only if $\alpha(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \sigma) = \delta$. Thus we can redefine φ_δ as

$$\varphi_\delta \quad = \quad \bigvee_{\alpha(a',\sigma)=\delta} p_\sigma \wedge \tau_{a'}. \tag{2.2}$$

Since by assumption each T_b is definable in FTL($\mathcal{L}$), and **FTL** is a closure operator, it follows that there is a formula in FTL($\mathcal{L}$) which is equivalent to φ. ■

PROPOSITION 2.3.2 *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 complete and 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) connected tree automata. Then $L_\varphi \subseteq T_\Sigma$ is recognizable by a connected strict Moore product of $\mathbb{A}$ and $\mathbb{B}$.*

Proof. The proof of the corresponding fact in [15], Proposition 28 actually constructs a connected strict Moore product. ■

From these results, we can derive:

THEOREM 2.3.3 *For any class $\mathbf{K}^c$ of finite connected tree automata, every language in $\mathbf{FTL}(\mathbf{K}^c)$ is recognizable by some tree automaton in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$.*

Proof. Let φ denote a complete and deterministic formula over Σ in $\mathrm{FTL}(\mathbf{K}^c)$. We show that L_φ is recognizable by some tree automaton in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$. When φ is p_σ, for some $\sigma \in \Sigma$, then L_φ is 1-definite and thus recognizable by some tree automaton in $\mathbf{D}_1^c$. We proceed 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 $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$, $i = 1, 2$. Then L_φ is recognizable by the connected direct product $\mathbb{A}_1 \times \mathbb{A}_2$ which is also in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$. 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 $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$, then it follows by Proposition 2.3.2 that L_φ is recognizable by some tree automaton in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$. (Note that since $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$ is closed with respect to the connected direct product, we may assume without loss of generality that each L_{φ_δ} is recognizable by the same tree automaton $\mathbb{A}$ in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$). ■

Note that $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M = \langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_s$ is just $\mathbf{D}_1^c \times \langle \mathbf{K}^c \rangle_M$.

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 $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$.*

Proof. It is clear that $\mathbf{FTL}(\mathbf{K}^c)$ contains the 1-definite tree languages and thus $\mathbf{FTL}(\mathbf{K}^c) = \mathbf{FTL}(\mathbf{D}_1^c \cup \mathbf{K}^c)$, by Theorem 2.1.1. Let us define the *rank* of $\mathbb{A} \in \langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$ to be the least number of Moore product and quotient operations needed to generate $\mathbb{A}$ from $\mathbf{D}_1^c \cup \mathbf{K}^c$. We prove by induction on the rank of $\mathbb{A}$ that every language recognizable by $\mathbb{A}$ is in $\mathbf{FTL}(\mathbf{D}_1^c \cup \mathbf{K}^c)$. When the rank is 0 we have $\mathbb{A} \in \mathbf{D}_1^c \cup \mathbf{K}^c$ 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 $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$ of smaller rank, or $\mathbb{A}$ is a connected Moore product of some tree automata in $\langle \mathbf{D}_1^c \cup \mathbf{K}^c \rangle_M$ 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 2.3.1. Conversely, by Theorem 2.3.3, every language in $\mathbf{FTL}(\mathbf{K}^c)$ is recognizable by some tree 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 in the least connected (strict) Moore pseudovariety of finite connected tree automata containing $\mathbf{D}_1^c$ and the minimal automata of the languages in $\mathcal{L}$.*

COROLLARY 2.3.6 *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 $\mathbf{FTL}(\mathcal{V}) = \mathcal{V}$.*

Proof. By Theorem 2.3.4, $\mathbf{FTL}(\mathbf{V}^c) = \mathcal{L}_{\langle \mathbf{D}_1^c \cup \mathbf{V}^c \rangle_M}$. When $\mathbf{V}^c$ is a connected Moore pseudovariety containing $\mathbf{D}_1^c$, then $\langle \mathbf{D}_1^c \cup \mathbf{V}^c \rangle_M = \mathbf{V}^c$ and thus $\mathbf{FTL}(\mathbf{V}^c) = \mathcal{L}_{\mathbf{V}^c}$. Let $\mathbf{V}_1^c$ and $\mathbf{V}_2^c$ be connected Moore pseudovarieties of finite connected tree automata containing $\mathbf{D}_1^c$. By the Eilenberg correspondence, $\mathbf{V}_1^c \subseteq \mathbf{V}_2^c$ if and only if $\mathcal{L}_{\mathbf{V}_1^c} \subseteq \mathcal{L}_{\mathbf{V}_2^c}$, i.e., when $\mathbf{FTL}(\mathbf{V}_1^c) \subseteq \mathbf{FTL}(\mathbf{V}_2^c)$. Assume finally that $\mathcal{V}$ is a literal variety of regular tree languages containing $\mathcal{D}_1$ with $\mathcal{V} = \mathbf{FTL}(\mathcal{V})$. By the Eilenberg correspondence, there exists a pseudovariety $\mathbf{V}^c$ of finite connected tree automata with $\mathcal{V} = \mathcal{L}_{\mathbf{V}^c}$. Since $\mathcal{D}_1 \subseteq \mathcal{V}$, we have $\mathbf{D}_1^c \subseteq \mathbf{V}^c$. Thus, $\mathcal{V} = \mathcal{L}_{\mathbf{V}^c} = \mathbf{FTL}(\mathbf{V}^c)$. ■

In [15] it has been shown that the map $\mathbf{V}^c \mapsto \mathbf{FTL}(\mathbf{V}^c)$, restricted to connected cascade pseudovarieties containing $\mathbf{D}^c$, is an order isomorphism between such connected cascade pseudovarieties of finite connected tree automata and literal varieties $\mathcal{V}$ of tree languages containing the definite tree languages and satisfying $\mathbf{FTL}(\mathcal{V}) = \mathcal{V}$. Combining this result with Corollary 2.3.6, we obtain a new proof of the fact, proved in Corollary 2.2.29, that any connected Moore pseudovariety containing $\mathbf{D}^c$ is a connected cascade pseudovariety.

2.4 Application

In this section, we present some applications of Theorem 2.3.3. The characterization provided by the Theorem is not effective by itself; in the major part of this section we show that some fragments of the temporal logic CTL have a decidable definability problem.

2.4.1 Fragments of CTL

Below we assume that the lexicographic order on Bool (see page 29) satisfies $\uparrow_n < \downarrow_n$, for each $n \in R$ (as we have seen before, the particular ordering is not important).

Let $L_{\mathrm{EF}^+} \subseteq T_{\mathrm{Bool}}$ denote the regular 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. [45].

In the same way, the strict and non-strict EG modalities are closely related to the modal operators associated to the following languages, where we use the letter p to range over the maximal paths of a tree, while u and v range over non-root vertices and all vertices, respectively:

$$\begin{aligned} L_{\mathrm{EG}^+} &= \{t \in T_{\mathrm{Bool}} : \exists p \forall u \in p\ t(u) \in \{\uparrow_n : n \in R\}\} \\ L_{\mathrm{EG}^*} &= \{t \in T_{\mathrm{Bool}} : \exists p \forall v \in p\ t(v) \in \{\uparrow_n : n \in R\}\}. \end{aligned}$$

It is easy to check that the minimal tree automaton of L_{EF^+} is the automaton $\mathbb{E}^+_{\mathrm{EF}} = (\{0,1,2\}, \mathrm{Bool})$, where the interpretation of the function symbols is the following:

$$\uparrow_n^{\mathbb{E}^+_{\mathrm{EF}}}(x_1,\ldots,x_n) = \begin{cases} 1 & \text{if } x_i = 0 \text{ for all } i \in [n]; \\ 2 & \text{otherwise} \end{cases}$$

and

$$\downarrow_n^{\mathbb{E}^+_{\mathrm{EF}}}(x_1,\ldots,x_n) = \begin{cases} 0 & \text{if } x_i = 0 \text{ for all } i \in [n]; \\ 2 & \text{otherwise.} \end{cases}$$

Moreover, the minimal tree automaton of L_{EF^*} is $\mathbb{E}^*_{\mathrm{EF}} = (\{0,1\}, \mathrm{Bool})$, where the interpretation of the function symbols is the following:

$$\uparrow_n^{\mathbb{E}^*_{\mathrm{EF}}}(x_1,\ldots,x_n) = 1$$

and

$$\downarrow_n^{\mathbb{E}^*_{\mathrm{EF}}}(x_1,\ldots,x_n) = \begin{cases} 0 & \text{if } x_i = 0 \text{ for all } i \in [n]; \\ 1 & \text{otherwise.} \end{cases}$$

As for the languages L_{EG^+} and L_{EG^*}, their minimal automata are $\mathbb{E}^+_{\mathrm{EG}} = (\{0,1,2\}, \mathrm{Bool})$ and $\mathbb{E}^*_{\mathrm{EG}} = (\{0,1\}, \mathrm{Bool})$ with

$$\begin{aligned} \uparrow_0^{\mathbb{E}^+_{\mathrm{EG}}} &= 2; \\ \downarrow_0^{\mathbb{E}^+_{\mathrm{EG}}} &= 1; \\ \uparrow_0^{\mathbb{E}^*_{\mathrm{EG}}} &= 1; \\ \downarrow_0^{\mathbb{E}^*_{\mathrm{EG}}} &= 0, \end{aligned}$$

and when $n > 0$,

$$\uparrow_n^{\mathbb{E}^+_{\mathrm{EG}}}(x_1,\ldots,x_n) = \begin{cases} 0 & \text{if } x_i \in \{0,1\} \text{ for all } i \in [n]; \\ 2 & \text{otherwise,} \end{cases}$$

$$\downarrow_n^{\mathbb{E}^+_{\mathrm{EG}}}(x_1,\ldots,x_n) = \begin{cases} 0 & \text{if } x_i \in \{0,1\} \text{ for all } i \in [n]; \\ 1 & \text{otherwise,} \end{cases}$$

and

$$\uparrow_n^{\mathbb{E}^*_{\mathrm{EG}}}(x_1,\ldots,x_n) = \begin{cases} 0 & \text{if } x_i = 0 \text{ for all } i \in [n]; \\ 1 & \text{otherwise,} \end{cases}$$

$$\downarrow_n^{\mathbb{E}^*_{\mathrm{EG}}}(x_1,\ldots,x_n) = 0.$$

It is easy to show that any quotient of the language L_{EF^+} is in $\mathbf{FTL}(\{L_{\mathrm{EF}^+}\})$: any language of the form $\zeta^{-1}(L_{\mathrm{EF}^+})$ where ζ is a proper Bool-context, is either T_{Bool} (when ζ has a non-root vertex labeled $\uparrow_n$ for some n), or $L_{\mathrm{EF}^+} \cup \{t \in T_{\mathrm{Bool}} : \mathrm{Root}(t) = \uparrow_n \text{ for some } n\}$ (when all non-root vertices but x_1 of ζ are labeled by $\downarrow_n$ for some n); these languages are clearly definable in $\mathrm{FTL}(\{L_{\mathrm{EF}^+}\})$. Analogous facts hold also for the other languages.

Thus we have that the logics corresponding to the languages L_{EF^+}, L_{EF^*}, L_{EG^+} and L_{EG^*} are exactly the logics corresponding to the automata $\mathbb{E}^+_{\mathrm{EF}}$, $\mathbb{E}^*_{\mathrm{EF}}$, $\mathbb{E}^+_{\mathrm{EG}}$ and $\mathbb{E}^*_{\mathrm{EG}}$. Let us denote these logics by $\mathrm{CTL}(\mathrm{EF}^+)$, $\mathrm{CTL}(\mathrm{EF}^*)$, $\mathrm{CTL}(\mathrm{EG}^+)$ and $\mathrm{CTL}(\mathrm{EG}^*)$, respectively.

PROPOSITION 2.4.1 *Let* Y *denote either* F *or* G. *Then a tree language is definable in* $\mathrm{CTL}(\mathrm{EY}^+)$ *if and only if its minimal automaton is in the connected Moore pseudovariety* $\langle \mathbf{D}_1^c \cup \{\mathbb{E}^+_{\mathrm{EY}}\}\rangle_M$ *of finite connected tree automata. Similarly, a tree language is definable in* $\mathrm{CTL}(\mathrm{EY}^*)$ *if and only if its minimal automaton is in the connected Moore pseudovariety* $\langle \mathbf{D}_1^c \cup \{\mathbb{E}^*_{\mathrm{EY}}\}\rangle_M$ *of finite connected tree automata.*

In the following subsections, we give decidable characterizations of the connected Moore pseudovarieties $\langle \mathbf{D}_1^c \cup \{\mathbb{E}^+_{\mathrm{EF}}\}\rangle_M$ and $\langle \mathbf{D}_1^c \cup \{\mathbb{E}^*_{\mathrm{EF}}\}\rangle_M$, and hence of the logics $\mathrm{CTL}(\mathrm{EF}^+)$ and $\mathrm{CTL}(\mathrm{EF}^*)$.

2.4.2 Some Moore properties of tree automata

When $\mathbb{A}$ is a Σ-tree automaton, let $\preceq_{\mathbb{A}}$ denote the *accessibility relation* of $\mathbb{A}$, that is, $a \preceq_{\mathbb{A}} b$ holds for $a, b \in A$ if and only if for some ΣA-polynomial context p we have $p^{\mathbb{A}}(a) = b$. Clearly, $\preceq_{\mathbb{A}}$ is a reflexive and transitive relation (a *preorder*). The relation $\sim_{\mathbb{A}}$ over A is defined by $a \sim_{\mathbb{A}} b$ if and only if $a \preceq_{\mathbb{A}} b$ and $b \preceq_{\mathbb{A}} a$ hold; since $\preceq_{\mathbb{A}}$ is a preorder, $\sim_{\mathbb{A}}$ is an equivalence relation. $a \prec_{\mathbb{A}} b$ denotes $a \preceq_{\mathbb{A}} b \ \wedge \ b \not\preceq_{\mathbb{A}} a$.

Before we continue with a deeper investigation of these relations, we need the following auxiliary results:

LEMMA 2.4.2 *Let* $\mathbb{A} = (A, \Sigma)$ *be a finite tree automaton and* Θ *a congruence of* $\mathbb{A}$. *Let* $\overline{a}$ *denote the congruence class* a/Θ, *and let* $\mathbb{B}$ *denote the factor automaton* $\mathbb{A}/\Theta$. *Let* $a, b \in A$ *with* $\overline{a} \sim_{\mathbb{B}} \overline{b}$. *Then there exist* $a', b' \in A$ *such that* $a'\Theta a$, $b'\Theta b$ *and* $a' \sim_{\mathbb{A}} b'$.

Similarly, if $\overline{a} \preceq_{\mathbb{B}} \overline{b}$ *then there exists* $b' \in A$ *with* $b'\Theta b$ *and* $a \preceq_{\mathbb{A}} b'$.

Proof. We only prove the first claim.

Suppose $\mathbb{A}$, Θ, a and b satisfy the assumptions above. From $\overline{a} \preceq_{\mathbb{B}} \overline{b}$ there exists a ΣB-polynomial context p_1 with $p_1^{\mathbb{B}}(\overline{a}) = \overline{b}$. Similarly, from $\overline{b} \preceq_{\mathbb{B}} \overline{a}$ there exists a ΣB-polynomial context p_2 with $p_2^{\mathbb{B}}(\overline{b}) = \overline{a}$. Let q_1 be a ΣA-polynomial context such that $h(q_1) = p_1$ for the relabeling h induced by the map $A \to B$, $a \mapsto \overline{a}$. Such a relabeling always exists, since $\mathbb{B}$ is a homomorphic image of $\mathbb{A}$. Let q_2 be a ΣA-polynomial context with $h(q_2) = p_2$. Consider the following sequence $a_0, b_0, a_1, b_1, \ldots$ of states of $\mathbb{A}$ as follows: let $a_0 = a$, $b_i = q_1^{\mathbb{A}}(a_i)$ and $a_{i+1} = q_2^{\mathbb{A}}(b_i)$ for each $i \geq 0$. Then for all i we have $a_i \Theta a$ and $b_i \Theta b$.

From the definition of the sequence it is clear that $a_j \preceq_{\mathbb{A}} a_k$ for all $j < k$. Since A is finite, there exist integers $j < k$ such that $a_j = a_k$. For the states $a' = a_j$ and $b' = b_j$, all the required properties hold. ■

We state a corollary of Lemma 2.2.1:

COROLLARY 2.4.3 *Suppose that* $\mathbb{C} = (C, \Sigma) = \mathbb{A} \times_\gamma \mathbb{B}$ *is a cascade product of the tree automata* $\mathbb{A} = (A, \Sigma)$ *and* $\mathbb{B} = (B, \Delta)$. *Then for any* (a, b) *and* $(a', b') \in C$, *if* $(a, b) \preceq_{\mathbb{C}} (a', b')$ *then* $a \preceq_{\mathbb{A}} a'$ *and* $b \preceq_{\mathbb{B}} b'$. *Thus, if* $(a, b) \sim_{\mathbb{C}} (a', b')$ *then* $a \sim_{\mathbb{A}} a'$ *and* $b \sim_{\mathbb{B}} b'$.

Proof. Suppose $(a, b) \preceq_{\mathbb{C}} (a', b')$ holds. Then there exists a ΣC-polynomial context p with $p^{\mathbb{C}}((a, b)) = (a', b')$. By Lemma 2.2.1, there exists a ΣA-polynomial context p' and a ΔB-polynomial context p'' (depending on p and a) such that $p'^{\mathbb{A}}(a) = a'$ and $p''^{\mathbb{B}}(b) = b'$. Thus, $a \preceq_{\mathbb{A}} a'$ and $b \preceq_{\mathbb{B}} b'$ indeed hold. ■

A tree automaton $\mathbb{A}$ is called *monotone* if $\preceq_{\mathbb{A}}$ is a partial order (or equivalently, when $\sim_{\mathbb{A}} = \Delta_A$).

It is known, c.f., e.g. [15], that any homomorphic image of a finite monotone tree automaton is monotone. Moreover, as noted in [15], any cascade product of monotone tree automata is monotone, so that finite monotone tree automata form a cascade pseudovariety of finite tree automata.

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

COROLLARY 2.4.4 *Monotonicity is a cascade property.*

Let **Mon** denote the class of all finite monotone tree automata. Then, $\mathbf{Mon}^c$ denotes

the class of all finite monotone connected tree automata, which is a connected cascade pseudovariety.

REMARK 2.4.5 Finite monotone word automata (corresponding to the case $R = \{0, 1\}$) have been studied in detail in [9, 37], for example. They correspond to finite $\mathcal{R}$-trivial monoids. In [37], finite monotone automata are called *extensive*. The equivalence of finite extensive automata and finite automata having an $\mathcal{R}$-trivial monoid was extended to tree automata in [39]. Monotone tree automata were also investigated in [23].

The second property we will study is that of maximal dependency.

DEFINITION 2.4.6 *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* $i, j \in [n-1]$, *we have* $\sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, a_n) = \sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, a'_n)$.

Note that for unary symbols $\sigma \in \Sigma_1$ this property is trivial, so that when $R = \{0, 1\}$, all tree automata are maximal dependent.

PROPOSITION 2.4.7 *Any homomorphic image of a finite maximal dependent tree automaton is also maximal dependent.*

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite maximal dependent tree automaton and Θ a congruence of $\mathbb{A}$. The congruence class a/Θ of a state a will be denoted by $\overline{a}$. We have to prove that the factor automaton $\mathbb{B} = \mathbb{A}/\Theta$ is maximal dependent.

Now let $\sigma \in \Sigma_n$ be a function symbol and $\overline{a_1}, \ldots, \overline{a_{n-1}}, \overline{b}, \overline{c}$ states of $\mathbb{B}$ such that for some indices $i, j \in [n-1]$ both $\overline{b} \preceq_{\mathbb{B}} \overline{a_i}$ and $\overline{c} \preceq_{\mathbb{B}} \overline{a_j}$ hold. By Lemma 2.4.2, there exist $a'_i, a'_j \in A$ such that $\overline{a'_i} = \overline{a_i}$, $\overline{a'_j} = \overline{a_j}$, $b \preceq_{\mathbb{A}} a'_i$ and $c \preceq_{\mathbb{A}} a'_j$. So from the congruence properties and the maximal dependency of $\mathbb{A}$ we get:

$$\begin{aligned}
\sigma^{\mathbb{B}}(\overline{a_1}, \overline{a_2}, \ldots, \overline{a_{n-1}}, \overline{b}) &= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{i-1}}, \overline{a'_i}, \overline{a_{i+1}}, \ldots, \overline{a_{n-1}}, \overline{b}) \\
&= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{i-1}}, \overline{a'_i}, \overline{a_{i+1}}, \ldots, \overline{a_{n-1}}, \overline{a'_i}) \\
&= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{i-1}}, \overline{a_i}, \overline{a_{i+1}}, \ldots, \overline{a_{n-1}}, \overline{a_i}) \\
&= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{i-1}}, \overline{a_i}, \overline{a_{i+1}}, \ldots, \overline{a_{n-1}}, \overline{a_j}) \\
&= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{j-1}}, \overline{a'_j}, \overline{a_{j+1}}, \ldots, \overline{a_{n-1}}, \overline{a'_j}) \\
&= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{j-1}}, \overline{a'_j}, \overline{a_{j+1}}, \ldots, \overline{a_{n-1}}, \overline{c}) \\
&= \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_{j-1}}, \overline{a_j}, \overline{a_{j+1}}, \ldots, \overline{a_{n-1}}, \overline{c}).
\end{aligned}$$

(Note: we do not change $\overline{a_i}$ and $\overline{a_j}$ in one single step in order to handle the case $i = j$ properly.) So $\mathbb{B} = \mathbb{A}/\Theta$ is indeed maximal dependent. ■

PROPOSITION 2.4.8 *If $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are maximal dependent tree automata, then any Moore product $\mathbb{C} = (C, \Sigma) = \mathbb{A} \times_\alpha \mathbb{B}$ is also maximal dependent.*

Proof. Let $\sigma \in \Sigma_n$, $(a_1, b_1), \ldots, (a_{n-1}, b_{n-1}), (y_1, y_2), (z_1, z_2) \in C$ and let $i, j \in [n-1]$ be indices such that both $(y_1, y_2) \preceq_{\mathbb{C}} (a_i, b_i)$ and $(z_1, z_2) \preceq_{\mathbb{C}} (a_j, b_j)$ hold.

By Corollary 2.4.3, we have $y_1 \preceq_{\mathbb{A}} a_i$, $z_1 \preceq_{\mathbb{A}} a_j$, $y_2 \preceq_{\mathbb{B}} b_i$ and $z_2 \preceq_{\mathbb{B}} b_j$. Let $a = \sigma^{\mathbb{A}}(a_1, a_2, \ldots, a_{n-1}, y_1)$. Since $\mathbb{A}$ is maximal dependent, we also have $a = \sigma^{\mathbb{A}}(a_1, a_2, \ldots, a_{n-1}, z_1)$.

Thus,

$$\begin{aligned} & \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_{n-1}, b_{n-1}), (y_1, y_2)) \\ = \; & \big(a, \alpha(a, \sigma)^{\mathbb{B}}(b_1, b_2, \ldots, b_{n-1}, y_2)\big) \\ = \; & \big(a, \alpha(a, \sigma)^{\mathbb{B}}(b_1, b_2, \ldots, b_{n-1}, z_2)\big) \\ = \; & \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_{n-1}, b_{n-1}), (z_1, z_2)), \end{aligned}$$

since $\mathbb{B}$ is maximal dependent.

So $\mathbb{C}$ is indeed maximal dependent. ■

COROLLARY 2.4.9 *Maximal dependency is a Moore property.*

Let **MaxDep** denote the Moore pseudovariety of finite tree automata containing exactly the finite maximal dependent tree automata. Then, $\mathbf{MaxDep}^c$ denotes the connected Moore pseudovariety of finite maximal dependent connected tree automata.

DEFINITION 2.4.10 *We call a tree automaton $\mathbb{A} = (A, \Sigma)$* component dependent *if for any function symbol $\sigma \in \Sigma_n$ and $a_1 \sim_{\mathbb{A}} a'_1, \ldots, a_n \sim_{\mathbb{A}} a'_n \in A$ it holds that $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) = \sigma^{\mathbb{A}}(a'_1, \ldots, a'_n)$.*

Note that in any component dependent automaton $\mathbb{A}$ the relation $\sim_{\mathbb{A}}$ is a congruence relation. However, this latter property is not a Moore property (not preserved under taking Moore products); this is the reason why we have to define the stronger concept of component dependency. Also note that any monotone tree automaton is trivially component dependent, as well as any 1-definite tree automaton.

PROPOSITION 2.4.11 *Any homomorphic image of a finite component dependent tree automaton is component dependent.*

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite component dependent tree automaton, Θ a congruence of $\mathbb{A}$ and let $\mathbb{B}$ denote the factor automaton $\mathbb{A}/\Theta$. For each $a \in A$, let $\overline{a}$ denote the congruence class a/Θ. Let $\sigma \in \Sigma_n$ be a function symbol and $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ such that $\overline{a_i} \sim_{\mathbb{B}} \overline{b_i}$ for all $i \in [n]$.

Then, by Lemma 2.4.2 there exist $a'_1, \ldots, a'_n, b'_1, \ldots, b'_n \in A$ such that $a'_i \Theta a_i$, $b_i \Theta b'_i$ and $a'_i \sim_{\mathbb{A}} b'_i$ for all $i \in [n]$. Then we have

$$\begin{aligned} \sigma^{\mathbb{B}}(\overline{a_1}, \ldots, \overline{a_n}) &= \sigma^{\mathbb{B}}(\overline{a'_1}, \ldots, \overline{a'_n}) \\ &= \sigma^{\mathbb{B}}(\overline{b'_1}, \ldots, \overline{b'_n}) \\ &= \sigma^{\mathbb{B}}(\overline{b_1}, \ldots, \overline{b_n}). \end{aligned}$$

■

PROPOSITION 2.4.12 *Let $\mathbb{A}$ and $\mathbb{B}$ be component dependent tree automata and $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ a Moore product. Then $\mathbb{C}$ is also component dependent.*

Proof. Let $\sigma \in \Sigma_n$ be a function symbol and $(a_1, b_1), \ldots, (a_n, b_n)$, $(a'_1, b'_1), \ldots, (a'_n, b'_n)$ be states of $\mathbb{C}$ such that $(a_i, b_i) \sim_{\mathbb{C}} (a'_i, b'_i)$ for all $i \in [n]$.

By Corollary 2.4.3 we know that $a_i \sim_{\mathbb{A}} a'_i$ and $b_i \sim_{\mathbb{B}} b'_i$ for all $i \in [n]$. Let δ denote the symbol $\alpha(a, \sigma)$ where $a = \sigma^{\mathbb{A}}(a_1, \ldots, a_n)$. From the component dependency of $\mathbb{A}$ we know that $a = \sigma^{\mathbb{A}}(a'_1, \ldots, a'_n)$ also holds. Now we have the following:

$$\begin{aligned} \sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) &= (a, \delta^{\mathbb{B}}(b_1, \ldots, b_n)) \\ &= (a, \delta^{\mathbb{B}}(b'_1, \ldots, b'_n)) \\ &= \sigma^{\mathbb{C}}((a'_1, b'_1), \ldots, (a'_n, b'_n)). \end{aligned}$$

■

COROLLARY 2.4.13 *Component dependency is a Moore property.*

Let **CompDep** denote the Moore pseudovariety containing exactly the finite component dependent tree automata. Then, the class $\mathbf{CompDep}^c$ of all finite component dependent connected tree automata is a connected Moore pseudovariety.

The last property considered in this section is the following:

DEFINITION 2.4.14 *Call a tree automaton $\mathbb{A} = (A, \Sigma)$* componentwise unique *if for any $a, b \in A$ and proper ΣA-polynomial contexts p_1, p_2 such that $\mathrm{Root}(p_1) = \mathrm{Root}(p_2)$, $p_1^{\mathbb{A}}(a) = b$ and $p_2^{\mathbb{A}}(b) = a$, it holds that $a = b$.*

PROPOSITION 2.4.15 *Any homomorphic image of a finite componentwise unique tree automaton is componentwise unique.*

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite componentwise unique tree automaton and Θ a congruence of $\mathbb{A}$. We write $\overline{a}$ for a/Θ as before. Let $\mathbb{B} = (B, \Sigma)$ be the factor automaton $\mathbb{A}/\Theta$. Let $\overline{a}, \overline{b} \in B$ and p_1, p_2 be proper ΣB-polynomial contexts such that $\mathrm{Root}(p_1) = \mathrm{Root}(p_2)$, $p_1^{\mathbb{B}}(\overline{a}) = \overline{b}$ and $p_2^{\mathbb{B}}(\overline{b}) = \overline{a}$.

Let q_1 and q_2 respectively be ΣA-polynomial contexts such that $h(q_1) = p_1$ and $h(q_2) = p_2$ for the relabeling h induced by the mapping $A \to B$, $a \mapsto \overline{a}$. Such relabelings always exist, moreover, $\mathrm{Root}(q_1) = \mathrm{Root}(p_1) = \mathrm{Root}(p_2) = \mathrm{Root}(q_2)$ holds.

Consider the sequence $a_0, b_0, a_1, b_1, \ldots \in A$ of states such that $a_0 = a$, $b_i = q_1^{\mathbb{A}}(a_i)$ and $a_{i+1} = q_2^{\mathbb{A}}(b_i)$ for each $i \geq 0$. Clearly, $\overline{a_i} = \overline{a}$ and $\overline{b_i} = \overline{b}$ for all $i \geq 0$. Since A is finite, there exist integers $0 \leq j < k$ with $a_j = a_k$. Thus, we have $q_1^{\mathbb{A}}(a_j) = b_j$ and $(q_2(q_1q_2)^{k-j-1})^{\mathbb{A}}(b_j) = a_k = a_j$.

Moreover $\mathrm{Root}(p_1) = \mathrm{Root}(p_2(p_1p_2)^{k-j-1})$, since $\mathrm{Root}(p_1) = \mathrm{Root}(p_2)$. From the componentwise uniqueness of $\mathbb{A}$ we have $a_j = b_j$, so $\overline{a} = \overline{a_j} = \overline{b_j} = \overline{b}$. ■

PROPOSITION 2.4.16 *Any Moore product of two componentwise unique tree automata is also componentwise unique.*

Proof. Let $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ be componentwise unique tree automata and $\mathbb{C} = \mathbb{A} \times_\alpha \mathbb{B}$ a Moore product. Let $(a_1, b_1), (a_2, b_2) \in C$ be two states of $\mathbb{C}$ and p_1, p_2 proper ΣC-polynomial contexts such that $\mathrm{Root}(p_1) = \mathrm{Root}(p_2)$, $p_1^{\mathbb{C}}((a_1, b_1)) = (a_2, b_2)$ and $p_2^{\mathbb{C}}((a_2, b_2)) = (a_1, b_1)$.

Then there exist proper ΣA-polynomial contexts p_1' and p_2' with $p_1'^{\mathbb{A}}(a_1) = a_2$ and $p_2'^{\mathbb{A}}(a_2) = a_1$, moreover $\mathrm{Root}(p_1') = \mathrm{Root}(p_1) = \mathrm{Root}(p_2) = \mathrm{Root}(p_2')$. Since $\mathbb{A}$ is componentwise unique we have that $a_1 = a_2$. Let a denote a_1 from now on.

Now consider the automaton $\mathbb{B}$. From $p_1^{\mathbb{C}}(a, b_1) = (a, b_2)$ we have that there exists a proper ΔB-polynomial context p_1'' such that both $p_1''^{\mathbb{B}}(b_1) = b_2$ and $\mathrm{Root}(p_1'') = \alpha(a, \mathrm{Root}(p_1))$ hold. Also, since $p_2^{\mathbb{C}}(a, b_2) = (a, b_1)$ we have that there exists a proper ΔB-polynomial context p_2'' with $p_2''^{\mathbb{B}}(b_2) = b_1$ and $\mathrm{Root}(p_2'') = \alpha(a, \mathrm{Root}(p_2))$. Since $\mathrm{Root}(p_1) = \mathrm{Root}(p_2)$ we have $\mathrm{Root}(p_1'') = \mathrm{Root}(p_2'')$, and from the componentwise uniqueness of $\mathbb{B}$ it follows that $b_1 = b_2$, so $\mathbb{C}$ is indeed componentwise unique. ■

COROLLARY 2.4.17 *Componentwise uniqueness is a Moore property.*

Let **CompUnique** denote the Moore pseudovariety containing exactly the finite componentwise unique tree automata. Also, $\mathbf{CompUnique}^c$ denotes the corresponding connected Moore pseudovariety.

The following relation holds between the properties we defined so far:

PROPOSITION 2.4.18

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

Consequently,

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

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite tree automaton that is component dependent, maximal dependent and commutative. Suppose $a, b \in A$ are states and p, q are proper ΣA-polynomial contexts such that $p^{\mathbb{A}}(a) = b$, $q^{\mathbb{A}}(b) = a$ and $\mathrm{Root}(p) = \mathrm{Root}(q) = \sigma \in \Sigma_n$. We prove that $a = b$ also holds in this case.

Write $p = \sigma(p_1, \ldots, p_n)$. There exists exactly one index $i_0 \in [n]$ such that p_{i_0} is a ΣA-polynomial context. For each $i \in [n]$, let a_i denote the state $p_i^{\mathbb{A}}$ if $i \neq i_0$, and let a_{i_0} be the state $p_{i_0}^{\mathbb{A}}(a)$. We clearly have $a \preceq_{\mathbb{A}} a_{i_0} \preceq_{\mathbb{A}} b$, moreover, $a_i \preceq_{\mathbb{A}} b$ for each $i \in [n]$. Thus, since $a \sim_{\mathbb{A}} b$, we have $a_{i_0} \sim_{\mathbb{A}} a$ and $a_i \preceq_{\mathbb{A}} a_{i_0}$ for each $i \in [n]$.

It follows now from commutativity and maximal dependency that

$$b = \sigma^{\mathbb{A}}(a_1, \ldots, a_n) \quad = \quad \sigma^{\mathbb{A}}(a_{i_0}, \ldots, a_{i_0}).$$

We have thus proved that there exists $a' \in A$ with $a' \sim_{\mathbb{A}} a$ and $b = \sigma^{\mathbb{A}}(a', \ldots, a')$. Analogously, there exists a state $b' \sim_{\mathbb{A}} b$ with $a = \sigma^{\mathbb{A}}(b', \ldots, b')$.

But since $a' \sim_{\mathbb{A}} a \sim_{\mathbb{A}} b \sim_{\mathbb{A}} b'$ and $\mathbb{A}$ is component dependent, we have $a = b$. ■

2.4.3 Characterizing $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$

Recall the automaton $\mathbb{E}_{\mathrm{EF}}^+$ defined on page 39. In this subsection, we characterize the connected Moore pseudovariety $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$ by three easily verifiable conditions.

It can be checked easily that $\mathbb{E}_{\mathrm{EF}}^+$ is a finite, monotone, commutative and maximal dependent connected tree automaton, so by the results of Subsection 2.4.2,

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

We will show that the reverse inclusion is also true.

First we claim the following facts about commutative and maximal dependent tree automata:

PROPOSITION 2.4.19 *Let $\mathbb{A} = (A, \Sigma)$ be a commutative and maximal dependent tree automaton. Let $\sigma \in \Sigma_{j+k}$ be a function symbol, where $j \geq 1$ and $k > 0$, and assume that $a_1, \ldots, a_j, b_1, \ldots, b_k, c_1, \ldots, c_k \in A$ such that for all $i \in [k]$ there exist $i', i'' \in [j]$ such that $b_i \preceq_{\mathbb{A}} a_{i'}$ and $c_i \preceq_{\mathbb{A}} a_{i''}$. Then $\sigma^{\mathbb{A}}(a_1, \ldots, a_j, b_1, \ldots, b_k) = \sigma^{\mathbb{A}}(a_1, \ldots, a_j, c_1, \ldots, c_k)$.*

This statement can be proved easily by a straightforward induction on k.

PROPOSITION 2.4.20 *Any commutative maximal dependent tree automaton is idempotent.*

The proof of this statement is analogous to that of Proposition 2.4.22 below.

For a tree automaton $\mathbb{A} = (A, \Sigma)$, we define the function $\mathrm{Values}^+_{\mathbb{A}}$ mapping Σ-trees to finite subsets of A as follows. Given a Σ-tree t we define $\mathrm{Values}^+_{\mathbb{A}}(t)$ to be the (finite) set $\{s^{\mathbb{A}} : s \text{ is a proper subtree of } t\}$.

We prove a couple of facts concerning the function $\mathrm{Values}^+_{\mathbb{A}}$.

PROPOSITION 2.4.21 *Let $\mathbb{A} = (A, \Sigma)$ be a monotone tree automaton and $t = \sigma(t_1, \ldots, t_n)$ a Σ-tree. Then for any maximal element a of $\mathrm{Values}^+_{\mathbb{A}}(t)$ (with respect to $\preceq_{\mathbb{A}}$) there exists an index $i_a \in [n]$ such that $t^{\mathbb{A}}_{i_a} = a$.*

Proof. Let a be a maximal element of $\mathrm{Values}^+_{\mathbb{A}}(t)$. From the definition of $\mathrm{Values}^+_{\mathbb{A}}(t)$ we have that there exists a proper subtree s of t with $s^{\mathbb{A}} = a$. Since s is a proper subtree of t, there exists an integer $i_a \in [n]$ such that s is a subtree of t_{i_a}. For this t_{i_a} we clearly have $a = s^{\mathbb{A}} \preceq_{\mathbb{A}} t^{\mathbb{A}}_{i_a} \in \mathrm{Values}^+_{\mathbb{A}}(t)$.

Since a is a maximal element of $\mathrm{Values}^+_{\mathbb{A}}(t)$, it follows that $a = t^{\mathbb{A}}_{i_a}$ completing the proof. ■

PROPOSITION 2.4.22 *Let $\mathbb{A} = (A, \Sigma)$ be a monotone, commutative and maximal dependent tree automaton. Assume that $s, t \in T_\Sigma$ are Σ-trees such that $\mathrm{Values}^+_{\mathbb{A}}(s) = \mathrm{Values}^+_{\mathbb{A}}(t)$ and $\mathrm{Root}(s) = \mathrm{Root}(t)$. Then $s^{\mathbb{A}} = t^{\mathbb{A}}$.*

Proof. Let $t = \sigma(t_1, \ldots, t_n)$ and $s = \sigma(s_1, \ldots, s_n)$ be two Σ-trees satisfying the assumptions. Let $\{a_1, \ldots, a_k\}$ be the set of maximal elements of $\mathrm{Values}^+_{\mathbb{A}}(t) = \mathrm{Values}^+_{\mathbb{A}}(s)$ (this set, as well as $\mathrm{Values}^+_{\mathbb{A}}(t)$ itself, is finite). From Proposition 2.4.21 we have that for all $i \in [k]$ there exist $i', i'' \in [n]$ such that $t^{\mathbb{A}}_{i'} = s^{\mathbb{A}}_{i''} = a_i$. Of course if $i \neq j$ then $i' \neq j'$ and $i'' \neq j''$. Let π and ν be permutations of $[n]$ with $t^{\mathbb{A}}_{\pi(i)} = s^{\mathbb{A}}_{\nu(i)} = a_i$ for all $i \in [k]$. Then, for

any $k < i \leq n$ there exist indices $i', i'' \in [k]$ with $t^{\mathbb{A}}_{\pi(i)} \preceq_{\mathbb{A}} a_{i'}$ and $s^{\mathbb{A}}_{\nu(i)} \preceq_{\mathbb{A}} a_{i''}$. Applying Proposition 2.4.19 we have

$$\begin{aligned} t^{\mathbb{A}} &= \sigma^{\mathbb{A}}(t^{\mathbb{A}}_1, \ldots, t^{\mathbb{A}}_n) \\ &= \sigma^{\mathbb{A}}(t^{\mathbb{A}}_{\pi(1)}, t^{\mathbb{A}}_{\pi(2)}, \ldots, t^{\mathbb{A}}_{\pi(n)}) \\ &= \sigma^{\mathbb{A}}(t^{\mathbb{A}}_{\pi(1)}, \ldots, t^{\mathbb{A}}_{\pi(k)}, s^{\mathbb{A}}_{\nu(k+1)}, \ldots, s^{\mathbb{A}}_{\nu(n)}) \\ &= \sigma^{\mathbb{A}}(s^{\mathbb{A}}_{\nu(1)}, \ldots, s^{\mathbb{A}}_{\nu(n)}) \\ &= \sigma^{\mathbb{A}}(s^{\mathbb{A}}_1, \ldots, s^{\mathbb{A}}_n) \\ &= s^{\mathbb{A}}. \end{aligned}$$

■

Remark 2.4.23 The implication

$$\mathrm{Values}^+_{\mathbb{A}}(s) = \mathrm{Values}^+_{\mathbb{A}}(t) \ \wedge \ \mathrm{Root}(s) = \mathrm{Root}(t) \quad \Rightarrow \quad s^{\mathbb{A}} = t^{\mathbb{A}}$$

appearing in Proposition 2.4.22 corresponds to the notion of "typeset dependency" of [5].

In the next couple of statements we connect the tree automaton $\mathbb{E}^+_{\mathrm{EF}}$ and the functions $\mathrm{Values}^+_{\mathbb{A}}(t)$.

Definition 2.4.24 *Let $\mathbb{A} = (A, \Sigma)$ be a tree automaton. We say that a tree automaton $\mathbb{B} = (B, \Sigma)$ is a* Values$^+$-automaton of $\mathbb{A}$ *if there exists a mapping $h : B \to P(A)$ to the power set $P(A)$ of A such that for any tree $t \in T_\Sigma$ it holds that $h(t^{\mathbb{B}}) = \mathrm{Values}^+_{\mathbb{A}}(t)$.*

Observe that h is uniquely determined on the connected part of $\mathbb{B}$ and is thus unique whenever $\mathbb{B}$ is connected. Thus, in this case we call h the canonical Values$^+$-mapping *from $\mathbb{B}$ to $\mathbb{A}$.*

We have the following:

Proposition 2.4.25 *Let $\mathbb{A}$ be a finite connected tree automaton. There exists a finite* Values$^+$*-automaton of $\mathbb{A}$ in the connected Moore pseudovariety $\langle \mathbb{A}, \mathbb{E}^+_{\mathrm{EF}} \rangle_M$.*

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite connected tree automaton with $A = \{a_1, \ldots, a_k\}$. For each state $a \in A$ define the connected Moore product $\mathbb{B}[a] = \mathbb{A} \times_{\alpha[a]} \mathbb{E}^+_{\mathrm{EF}}$, where

$$\alpha[a](a', \sigma) = \begin{cases} \uparrow_n & \text{if } a = a'; \\ \downarrow_n & \text{otherwise} \end{cases}$$

for all $a' \in A$ and $\sigma \in \Sigma_n$, $n \geq 0$. Then for any Σ-tree t we have $t^{\mathbb{B}[a]} = (t^{\mathbb{A}}, e)$, where $e = 2$ if and only if $a \in \mathrm{Values}^+_{\mathbb{A}}(t)$.

Now the connected direct product $\prod_{a \in A} \mathbb{B}[a]$ is isomorphic to a tree automaton $\mathbb{C} = (C, \Sigma)$ all of whose states are of the form $(a, e_1, \ldots, e_k)$ with $a \in A$ and $e_1, \ldots, e_k \in \{0, 1, 2\}$, moreover, for any Σ-tree t, $t^{\mathbb{C}} = (t^{\mathbb{A}}, e_1, \ldots, e_k)$ for the integers $e_1, \ldots, e_k$ with $e_i = 2$ if and only if $a_i \in \mathrm{Values}^+_{\mathbb{A}}(t)$. Now considering the function $h : C \to P(A)$ that is defined by $h\big((a, e_1, \ldots, e_k)\big) = \{a_i : i \in [k], e_i = 2\}$, we get that $\mathbb{C}$ is a Values$^+$-automaton of $\mathbb{A}$. ■

When R contains an integer > 1, then any finite monotone tree automaton $\mathbb{A} = (A, \Sigma)$ has a unique maximal element with respect to the partial order $\preceq_{\mathbb{A}}$. Indeed, from finiteness we get that at least one maximal element exists; and if a, b are both maximal elements and if the arity of a symbol σ is at least two, then by $a, b \preceq_{\mathbb{A}} \sigma^{\mathbb{A}}(a, b, a, \ldots, a)$ we have $a = b$. However, when $R = \{0, 1\}$, there may be several maximal elements. In order to be able to handle the two cases in a unified manner, we now prove:

PROPOSITION 2.4.26 *Assume that $\mathbb{A} = (A, \Sigma)$ is a finite monotone tree automaton. Then $\mathbb{A}$ is a renaming of a monotone tree automaton $\mathbb{A}'$ which has a unique maximal element $a_{\max}$ with respect to $\preceq_{\mathbb{A}'}$.*

Moreover, if $\mathbb{A}$ is commutative, maximal dependent, stutter invariant or subdirectly irreducible, then so is $\mathbb{A}'$.

Proof. Suppose that $\mathbb{A} = (A, \Sigma)$ is a finite monotone tree automaton. If it has more than one maximal element, then we can add a new symbol σ to Σ of rank > 0, and define the corresponding operation $\sigma^{\mathbb{A}}$ as a constant function with value a fixed maximal element of A with respect to $\preceq_{\mathbb{A}}$. The congruences of the new automaton $\mathbb{A}'$ are exactly the same as the congruences of $\mathbb{A}$, so that $\mathbb{A}$ is subdirectly irreducible if and only if $\mathbb{A}'$ is. It is also straightforward to check that commutativity, maximal dependency and stutter invariance are also preserved. ■

PROPOSITION 2.4.27 *Suppose $\mathbb{A}$ is a subdirectly irreducible finite monotone tree automaton with a unique maximal element $a_{\max}$. Then the set $A^- = A - \{a_{\max}\}$ also has a unique maximal element with respect to $\preceq_{\mathbb{A}}$.*

Proof. Assume that $\mathbb{A}$ has a unique maximal element $a_{\max}$. It is clear that for each maximal element $a \in A^-$, the relation that collapses a and $a_{\max}$ and keeps all other states separated is a congruence relation. Moreover, for different maximal elements $a, a' \in A^-$, the intersection of the two congruences is Δ_A, showing that $\mathbb{A}$ is subdirectly reducible.

Thus, if $\mathbb{A}$ is subdirectly irreducible, then there is a unique maximal element in A^- with respect to $\preceq_{\mathbb{A}}$. ■

Suppose now that $\mathbb{A} = (A, \Sigma)$ is a subdirectly irreducible finite monotone tree automaton which has a unique maximal element $a_{\max}$. Then let a_0 denote the unique maximal element of $A^- = A - \{a_{\max}\}$.

Define $\mathbb{A}^-$ to be the Σ-tree automaton (A^-, Σ) with

$$\sigma^{\mathbb{A}^-}(a_1, \ldots, a_n) \quad = \quad \begin{cases} \sigma^{\mathbb{A}}(a_1, \ldots, a_n) & \text{if } \sigma^{\mathbb{A}}(a_1, \ldots, a_n) \in A^-; \\ a_0 & \text{otherwise.} \end{cases}$$

PROPOSITION 2.4.28 *$\mathbb{A}^-$ is a quotient of $\mathbb{A}$.*

Proof. Map each $a \in A^-$ to a and $a_{\max}$ to a_0. ■

PROPOSITION 2.4.29 *Under the above assumptions on the subdirectly irreducible finite monotone tree automaton $\mathbb{A} = (A, \Sigma)$, if $\mathbb{A}$ is commutative, maximal dependent and connected then $\mathbb{A}$ is contained in the connected Moore pseudovariety $\langle \mathbb{A}^-, \mathbb{E}^+_{\mathrm{EF}} \rangle_M$.*

Proof. From Proposition 2.4.25 we know that there exists a tree automaton $\mathbb{B} = (B, \Sigma)$ in $\langle \mathbb{A}^-, \mathbb{E}^+_{\mathrm{EF}} \rangle_M$ that is a Values$^+$-automaton of $\mathbb{A}^-$. Let $h : B \to P(A^-)$ be the associated canonical Values$^+$-mapping. Consider the connected Moore product $\mathbb{C} = \mathbb{B} \times_\alpha \mathbb{E}^+_{\mathrm{EF}}$, where

$$\alpha(b, \sigma) = \begin{cases} \uparrow_n & \text{if there exists a } \Sigma\text{-tree } t \text{ such that} \\ & \mathrm{Root}(t) = \sigma, t^{\mathbb{A}} = a_{\max} \text{ and } \mathrm{Values}^+_{\mathbb{A}}(t) = h(b); \\ \downarrow_n & \text{otherwise.} \end{cases}$$

It is clear that for any Σ-tree t we have $t^{\mathbb{C}} = (t^{\mathbb{B}}, e)$ for some $e \in \{0, 1, 2\}$.

We claim that $e > 0$ if and only if $t^{\mathbb{A}} = a_{\max}$. We prove this claim by induction on the height of t. Let t be the tree $\sigma(t_1, \ldots, t_n)$ (if t is a constant symbol, then $n = 0$).

Assume that $t^{\mathbb{C}} = (t^{\mathbb{B}}, 0)$. From the definition of $\mathbb{E}^+_{\mathrm{EF}}$ it has to be the case $t_i^{\mathbb{C}} = (t_i^{\mathbb{B}}, 0)$ for all $i \in [n]$; from the induction hypothesis it follows that $t_i^{\mathbb{A}} \neq a_{\max}$ for any $i \in [n]$. In this case we have that $h(b) = \mathrm{Values}^+_{\mathbb{A}^-}(t) = \mathrm{Values}^+_{\mathbb{A}}(t)$, since if $a_{\max}$ occurred in $\mathrm{Values}^+_{\mathbb{A}}(t)$, then for some index $i \in [n]$ we would have $t_i^{\mathbb{A}} = a_{\max}$. Since $t^{\mathbb{C}} = (t^{\mathbb{B}}, 0)$, we also have that $\alpha(t^{\mathbb{B}}, \sigma) = \downarrow_n$. From the definition of α we get that there is no tree t' such that each of the conditions $\mathrm{Root}(t') = \sigma$, $t'^{\mathbb{A}} = a_{\max}$ and $\mathrm{Values}^+_{\mathbb{A}}(t') = h(b)$ hold. Since the first and the third condition both hold for t, it has to be the case $t^{\mathbb{A}} \neq a_{\max}$.

Now assume that $t^{\mathbb{C}} = (t^{\mathbb{B}}, e)$ for some $e \in \{1, 2\}$. We have two cases.

i) If there exists an index $i \in [n]$ such that $t_i^{\mathbb{C}} = (t_i^{\mathbb{B}}, e')$ for some $e' \in \{1, 2\}$, then by the induction hypothesis we have $t_i^{\mathbb{A}} = a_{\max}$. Since $a_{\max}$ is the maximal element of A, also $t^{\mathbb{A}} = a_{\max}$.

ii) If $t_i^{\mathbb{C}} = (t_i^{\mathbb{B}}, 0)$ for all $i \in [n]$, then it has to be the case $\alpha(t^{\mathbb{B}}, \sigma) = \uparrow_n$. Moreover, by the induction hypothesis we have $t_i^{\mathbb{A}} \neq a_{\max}$ for any $i \in [n]$; from this it follows that $\mathrm{Values}^+_{\mathbb{A}}(t) = \mathrm{Values}^+_{\mathbb{A}^-}(t)$ (since $a_{\max}$ cannot occur in $\mathrm{Values}^+_{\mathbb{A}}(t)$). From the definition of α we have that there exists a tree t' with $\mathrm{Root}(t') = \sigma$, $t'^{\mathbb{A}} = a_{\max}$ and $\mathrm{Values}^+_{\mathbb{A}}(t') = h(t^{\mathbb{B}})$. But since $\mathrm{Root}(t) = \sigma$ and $\mathrm{Values}^+_{\mathbb{A}} = h(t^{\mathbb{B}})$, from Proposition 2.4.22 we get that $t^{\mathbb{A}} = t'^{\mathbb{A}} = a_{\max}$.

Finally, form the connected direct product $\mathbb{A}^- \times \mathbb{C}$. It is clear that whenever $(a, (b, e))$ is a state of this automaton then there is a tree t with $t^{\mathbb{A}^-} = a$ and $h(b) = \mathrm{Values}^+_{\mathbb{A}^-}(t)$. Mapping each $(a, (b, 0))$ to a and $(a_0, (b, 1))$ and $(a_0, (b, 2))$ to $a_{\max}$, we obtain a homomorphism $\mathbb{A}^- \times \mathbb{C} \to \mathbb{A}$. ■

As a corollary we get the main result of this subsection:

THEOREM 2.4.30 *The connected Moore pseudovariety $\langle \mathbb{E}^+_{\mathrm{EF}} \rangle_M$ contains exactly the finite connected tree automata that are monotone, commutative and maximal dependent:*

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

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite, monotone, commutative and maximal dependent connected tree automaton. We prove by induction on $n = |A|$ that $\mathbb{A}$ is contained in $\langle \mathbb{E}^+_{\mathrm{EF}} \rangle_M$.

When $n = 1$, $\mathbb{A}$ is a trivial tree automaton contained in any pseudovariety of finite connected tree automata.

Suppose that $n > 0$ and we have proved the claim for all $i < n$. We have two cases: either $\mathbb{A}$ is subdirectly reducible or not.

i) If $\mathbb{A}$ is subdirectly reducible, then $\mathbb{A}$ is isomorphic to a connected direct product $\prod_{i \in [k]} \mathbb{A}/\Theta_i$ for some finite family $(\Theta_i)_{i \in [k]}$ of nontrivial congruences of $\mathbb{A}$. Since each $\mathbb{A}/\Theta_i$ has less than n states, we get from the induction hypothesis that each $\mathbb{A}/\Theta_i$ is contained in $\langle \mathbb{E}^+_{\mathrm{EF}} \rangle_M$, thus $\mathbb{A}$ is also contained in $\langle \mathbb{E}^+_{\mathrm{EF}} \rangle_M$.

ii) If $\mathbb{A}$ is subdirectly irreducible, then by Proposition 2.4.26 $\mathbb{A}$ is a connected renaming of some subdirectly irreducible, finite, monotone, commutative and maximal dependent connected tree automaton $\mathbb{A}_0$ having a unique maximal element $a_{\max}$.

From Proposition 2.4.29 we get that $\mathbb{A}_0$ is contained in $\langle \mathbb{A}_0^-, \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$. From the induction hypothesis, $\mathbb{A}_0^-$ is contained in $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$, thus $\mathbb{A}_0$ (hence also $\mathbb{A}$) is contained in $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$, proving the statement.

■

2.4.4 Characterizing $\langle \mathbb{E}_{\mathrm{EF}}^* \rangle_M$

Since $\mathbb{E}_{\mathrm{EF}}^*$ (see page 39) is a quotient of $\mathbb{E}_{\mathrm{EF}}^+$, it holds that $\langle \mathbb{E}_{\mathrm{EF}}^* \rangle_M \subseteq \langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$. We will prove that $\langle \mathbb{E}_{\mathrm{EF}}^* \rangle_M$ is a proper subclass of $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$ characterized by one extra condition: stutter invariance. Indeed, $\mathbb{E}_{\mathrm{EF}}^*$ is a monotone, commutative, maximal dependent and stutter invariant connected tree automaton, so that

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

In this subsection we will show that the reverse inclusion also holds. Our argument requires somewhat more work than in the case of $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M$.

Let $\mathbb{A} = (A, \Sigma)$ be a tree automaton. We define the mapping $\mathrm{Values}_{\mathbb{A}}^* : T_\Sigma \to P(A)$ mapping Σ-trees to finite subsets of A as follows: $\mathrm{Values}_{\mathbb{A}}^*(t) = \{s^{\mathbb{A}} : s \text{ is a subtree of } t\}$. The notion of a Values*-*automaton* is defined accordingly: a tree automaton $\mathbb{B} = (B, \Sigma)$ is a Values*-automaton of $\mathbb{A}$ if there exists a mapping $h : B \to P(A)$ such that for any tree $t \in T_\Sigma$ it holds that $h(t^{\mathbb{B}}) = \mathrm{Values}_{\mathbb{A}}^*(t)$. This mapping h is uniquely determined on the connected part of $\mathbb{B}$, thus when $\mathbb{B}$ is connected, we call h *the canonical* Values*-*mapping* from $\mathbb{B}$ to $\mathbb{A}$.

We have a result which is similar to Proposition 2.4.25:

Proposition 2.4.31 *Let $\mathbb{A}$ be a finite connected tree automaton. Then there exists a finite* Values*-*automaton of $\mathbb{A}$ in the connected Moore pseudovariety $\langle \mathbb{A}, \mathbb{E}_{\mathrm{EF}}^* \rangle_M$.*

Proof. Let $\mathbb{A} = (A, \Sigma)$ with $A = \{a_1, \ldots, a_k\}$. For every $a \in A$ we define the connected Moore product $\mathbb{B}[a] = \mathbb{A} \times_{\alpha[a]} \mathbb{E}_{\mathrm{EF}}^*$ as follows: for each $a' \in A$ and $\sigma \in \Sigma_n$ let

$$\alpha[a](a', \sigma) = \begin{cases} \uparrow_n & \text{if } a' = a; \\ \downarrow_n & \text{otherwise.} \end{cases}$$

It is easy to see that for any tree t we have $t^{\mathbb{B}[a]} = (t^{\mathbb{A}}, e)$, where $e = 1$ if and only if $a \in \mathrm{Values}_{\mathbb{A}}^*(t)$.

From this fact we can conclude that the connected direct product $\prod_{a \in A} \mathbb{B}[a]$ is isomorphic to a tree automaton $\mathbb{C}$ all of whose states are of the form $(a, e_1, \ldots, e_k)$ with $a \in A$,

$e_1, \ldots, e_k \in \{0, 1\}$, moreover, for any tree t we have $t^{\mathbb{C}} = (t^{\mathbb{A}}, e_1, \ldots, e_k)$ where $e_i = 1$ if and only if $a_i \in \mathrm{Values}^*_{\mathbb{A}}(t)$.

Thus, $\mathbb{C}$ is a Values*-automaton of $\mathbb{A}$ with the canonical Values*-mapping $(a, e_1, \ldots, e_k) \mapsto \{a_i : i \in [k], e_i = 1\}$. ■

We have the following property:

PROPOSITION 2.4.32 *Let $\mathbb{A}$ be a commutative, maximal dependent and stutter invariant tree automaton. Let $\sigma \in \Sigma_n$ be a symbol and $a_1, \ldots, a_n \in A$ be states such that $a_n \in \mathrm{Img}_{\mathbb{A}}(\sigma)$ and $a_1, \ldots, a_{n-1} \preceq_{\mathbb{A}} a_n$. Then $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) = a_n$.*

Proof. Let $b_1, \ldots, b_n \in A$ be states with $\sigma^{\mathbb{A}}(b_1, \ldots, b_n) = a_n$. Note that in this case $a_i, b_i \preceq_{\mathbb{A}} a_n$ holds for all $i \in [n]$. From the definition of stutter invariance and applying the commutativity of $\mathbb{A}$ and Proposition 2.4.19 we get:

$$\begin{aligned} a_n &= \sigma^{\mathbb{A}}(b_1, \ldots, b_n) \\ &= \sigma^{\mathbb{A}}(b_1, \ldots, b_{n-1}, a_n) \\ &= \sigma^{\mathbb{A}}(a_1, \ldots, a_{n-1}, a_n). \end{aligned}$$

■

We will introduce a decomposition method similar to the one in the previous subsection, but with more workaround. We define the following auxiliary automaton:

DEFINITION 2.4.33 *Let $\mathbb{A}_{\mathrm{Aux}}$ be the (finite, connected) tree automaton $(\{0, 1, 2\}, \Gamma)$, where $\Gamma_n = \{l_n, s_n, e_n, o_n\}$ for all $n \in R$ and the interpretation of the function symbols is the following: for all $x_1, \ldots, x_n \in \{0, 1, 2\}$,*

$$\begin{aligned} l_n^{\mathbb{A}_{\mathrm{Aux}}}(x_1, \ldots, x_n) &= \max_{i \in [n]} x_i, \\ s_n^{\mathbb{A}_{\mathrm{Aux}}}(x_1, \ldots, x_n) &= \begin{cases} 1 & \text{if } x_i < 2 \text{ for all } i \in [n]; \\ 2 & \text{otherwise,} \end{cases} \\ e_n^{\mathbb{A}_{\mathrm{Aux}}}(x_1, \ldots, x_n) &= \begin{cases} 2 & \text{if either } x_i = 2 \text{ for some } i \in [n] \\ & \text{or } x_i = 0 \text{ for all } i \in [n]; \\ 1 & \text{otherwise,} \end{cases} \\ o_n^{\mathbb{A}_{\mathrm{Aux}}}(x_1, \ldots, x_n) &= 2. \end{aligned}$$

The function symbols l, s, e and o are the first letters of the words "less", "set", "equal" and "over", respectively. Informally, the application of the operations corresponding to these function symbols will have the meaning that the value of the currently evaluated

tree is less than a specific value, set to this value, equal to this value and over this value, respectively. The difference between "set" and "equal" is that in the latter case we require that at least one of the immediate subtrees also has to evaluate to this specific value.

We first note that $\mathbb{A}_{\text{Aux}}$ is in the pseudovariety considered in this subsection:

PROPOSITION 2.4.34 *$\mathbb{A}_{\text{Aux}}$ is contained in the connected Moore pseudovariety $\langle \mathbb{E}^*_{\text{EF}} \rangle_M$.*

Proof. Let $\mathbb{A} = (\{0,1\}, \Gamma)$ be the renaming of $\mathbb{E}^*_{\text{EF}}$ given by the assignment $l^{\mathbb{A}}_n = e^{\mathbb{A}}_n = \downarrow^{\mathbb{E}^*_{\text{EF}}}_n$ and $o^{\mathbb{A}}_n = s^{\mathbb{A}}_n = \uparrow^{\mathbb{E}^*_{\text{EF}}}_n$ for each $n \in R$. Let us define the connected Moore product $\mathbb{B} = \mathbb{A} \times_\alpha \mathbb{E}^*_{\text{EF}}$, where

$$\alpha(a,\sigma) = \begin{cases} \uparrow_n & \text{if either } \sigma = o_n \text{ or both } \sigma = e_n \text{ and } a = 0; \\ \downarrow_n & \text{otherwise} \end{cases}$$

for all $a \in \{0,1\}$ and $\sigma \in \{l_n, s_n, e_n, o_n\}$, $n \in R$.

Then the following hold for all $n \in R$ and $x_1, \ldots, x_n \in \{0,1\} \times \{0,1\}$:

$$l^{\mathbb{B}}_n(x_1,\ldots,x_n) = \begin{cases} (0,0) & \text{if } x_i = (0,0) \text{ for all } i \in [n]; \\ (1,0) & \text{if } x_i \in \{(0,0),(1,0)\} \text{ for all } i \in [n] \\ & \text{and } x_i = (1,0) \text{ for some } i \in [n]; \\ (0,1) \text{ or } (1,1) & \text{otherwise.} \end{cases}$$

$$s^{\mathbb{B}}_n(x_1,\ldots,x_n) = \begin{cases} (1,0) & \text{if } x_i \in \{(0,0),(1,0)\} \text{ for all } i \in [n]; \\ (1,1) & \text{otherwise.} \end{cases}$$

$$e^{\mathbb{B}}_n(x_1,\ldots,x_n) = \begin{cases} (1,0) & \text{if } x_i \in \{(0,0),(1,0)\} \text{ for all } i \in [n] \\ & \text{and } x_i = (1,0) \text{ for some } i \in [n]; \\ (0,1) \text{ or } (1,1) & \text{otherwise.} \end{cases}$$

$$o^{\mathbb{B}}_n(x_1,\ldots,x_n) = (1,1).$$

Since the automaton $\mathbb{A}_{\text{Aux}}$ is a homomorphic image of $\mathbb{B}$ under the map $(0,0) \mapsto 0$, $(1,0) \mapsto 1$ and $(0,1),(1,1) \mapsto 2$ our claim is proven. ■

We now use $\mathbb{A}_{\text{Aux}}$ for decomposing finite, monotone, commutative, maximal dependent and stutter invariant connected tree automata.

LEMMA 2.4.35 *Let $\mathbb{A} = (A, \Sigma)$ be a subdirectly irreducible, finite, monotone, commutative, maximal dependent and stutter invariant connected tree automaton with the unique maximal element $a_{\max}$. Then $\mathbb{A}$ is contained in the connected Moore pseudovariety $\langle \mathbb{A}^-, \mathbb{E}^*_{\text{EF}} \rangle_M$.*

Proof. From Proposition 2.4.27 we get that $A^- = A - \{a_{\max}\}$ also has a unique maximal element a_0.

From Proposition 2.4.31 we have that there exists a Values*-automaton $\mathbb{B} = (B, \Sigma)$ of $\mathbb{A}^-$ in the pseudovariety $\langle \mathbb{A}^-, \mathbb{E}^*_{\mathrm{EF}} \rangle_M$. Let $h : B \to P(A^-)$ be the canonical Values*-mapping from $\mathbb{B}$ to $\mathbb{A}^-$.

Since $\mathbb{A}^-$ is monotone, for any tree t it holds that $t^{\mathbb{A}^-}$ is the unique maximal element of $\mathrm{Values}^*_{\mathbb{A}^-}(t)$. As a shorthand, when $b \in B$, we let $\overline{b}$ denote the unique maximal element of the set $h(b)$. Note that since $\mathbb{B}$ is connected, $\overline{b}$ is defined for each $b \in B$.

Let us define the connected Moore product $\mathbb{C} = \mathbb{B} \times_\alpha \mathbb{A}_{\mathrm{Aux}}$, where

$$\alpha(b,\sigma) = \begin{cases} l_n & \text{if } \overline{b} \neq a_0; \\ s_n & \text{if } \overline{b} = a_0, \text{ and there exists a tree } t \in T_\Sigma \text{ with } t^{\mathbb{A}} = a_0 \text{ such} \\ & \text{that both } \mathrm{Values}^+_{\mathbb{A}}(t) = h(b) - \{a_0\} \text{ and } \mathrm{Root}(t) = \sigma \text{ hold}; \\ o_n & \text{if } \overline{b} = a_0 \text{ and no tree } t \in T_\Sigma \text{ exists with both } t^{\mathbb{A}} = a_0 \\ & \text{and } \mathrm{Root}(t) = \sigma; \\ e_n & \text{otherwise.} \end{cases}$$

Note that the last case applies when $\overline{b} = a_0$ and there is a tree $t \in T_\Sigma$ with $t^{\mathbb{A}} = a_0$ and $\mathrm{Root}(t) = \sigma$, but $\mathrm{Values}^+_{\mathbb{A}}(t) \neq h(b) - \{a_0\}$ for any such tree t.

It is clear that for any tree $t \in T_\Sigma$ we have $t^{\mathbb{C}} = (t^{\mathbb{B}}, e)$ for some $e \in \{0,1,2\}$. We claim that $e = 1$ if and only if $t^{\mathbb{A}} = a_0$, moreover, $e = 2$ if and only if $t^{\mathbb{A}} = a_{\max}$. We prove this by induction on the height of $t = \sigma(t_1, \ldots, t_n)$. Let $t_i^{\mathbb{C}} = (t_i^{\mathbb{B}}, e_i)$ for all $i \in [n]$.

Note also that if $t^{\mathbb{A}} \neq a_{\max}$, then $\mathrm{Values}^*_{\mathbb{A}}(t) = \mathrm{Values}^*_{\mathbb{A}^-}(t) = h(t^{\mathbb{B}})$.

Assume $t^{\mathbb{A}} = a_0$. It is clear that $t_i^{\mathbb{A}} \preceq a_0$ for all $i \in [n]$. We have two cases:

i) If $t_i^{\mathbb{A}} \prec a_0$ for all $i \in [n]$, then by the induction hypothesis $e_i = 0$ for all i. Moreover, in this case $\mathrm{Values}^+_{\mathbb{A}}(t) = \mathrm{Values}^*_{\mathbb{A}}(t) - \{a_0\} = h(t^{\mathbb{B}}) - \{a_0\}$, $\mathrm{Root}(t) = \sigma$ and $\overline{t^{\mathbb{B}}} = a_0$, so $e = s_n^{\mathbb{A}_{\mathrm{Aux}}}(e_1, \ldots, e_n) = s_n^{\mathbb{A}_{\mathrm{Aux}}}(0, \ldots, 0) = 1$.

ii) If there exists an integer $i \in [n]$ such that $t_i^{\mathbb{A}} = a_0$, then $\alpha(t^{\mathbb{B}}, \sigma)$ is either s_n or e_n (depending on $h(t^{\mathbb{B}})$). From the induction hypothesis we have that $\{e_1, \ldots, e_n\} = \{0,1\}$ or $\{e_1, \ldots, e_n\} = \{1\}$, thus, from the definition of $\mathbb{A}_{\mathrm{Aux}}$ we get that $e = s_n^{\mathbb{A}_{\mathrm{Aux}}}(e_1, \ldots, e_n) = e_n^{\mathbb{A}_{\mathrm{Aux}}}(e_1, \ldots, e_n) = 1$ indeed holds.

Now assume $t^{\mathbb{A}} = a_{\max}$. We have three cases.

i) If $t_i^{\mathbb{A}} = a_{\max}$ for some $i \in [n]$, then by the induction hypothesis $e_i = 2$. Since $\mathbb{A}_{\mathrm{Aux}}$ is monotone, $e = 2$ indeed holds in this case.

ii) If $t_i^{\mathbb{A}} \prec a_{\max}$ for all $i \in [n]$ and there exists an integer $i \in [n]$ such that $t_i^{\mathbb{A}} = a_0$, then (since $t_i^{\mathbb{A}} \prec a_{\max}$ is equivalent to $t_i^{\mathbb{A}} \preceq a_0$), by Proposition 2.4.32 we have that $a_0 \notin \mathrm{Img}_{\mathbb{A}}(\sigma)$ (since otherwise we would have $t^{\mathbb{A}} = a_0$). From this it follows that $e = o_n^{\mathbb{A}_{\mathrm{Aux}}}(e_1, \ldots, e_n) = 2$.

iii) If $t_i^{\mathbb{A}} \prec a_0$ for all $i \in [n]$ then we have $\mathrm{Values}_{\mathbb{A}}^{+}(t) = \mathrm{Values}_{\mathbb{A}}^{*}(t) - \{a_{\max}\} = \mathrm{Values}_{\mathbb{A}^-}^{*}(t) - \{a_0\}$. From Proposition 2.4.22 we get that there is no tree t' with $\mathrm{Root}(t') = \sigma$, $t'^{\mathbb{A}} = a_0 \neq a_{\max} = t^{\mathbb{A}}$ and $\mathrm{Values}_{\mathbb{A}}^{+}(t') = \mathrm{Values}_{\mathbb{A}}^{+}(t)$. It follows that $\alpha(t^{\mathbb{B}}, \sigma) = e_n$ and by the induction hypothesis we have $e_i = 0$ for all $i \in [n]$. Finally we have $e = e_n^{\mathbb{A}_{\mathrm{Aux}}}(0, \ldots, 0) = 2$.

The last fact we have to show is that if $t^{\mathbb{A}} \notin \{a_0, a_{\max}\}$ then $e = 0$. Assume that $t^{\mathbb{A}} = a \prec a_0$. Then $t_i^{\mathbb{A}} \prec a_0$ for all $i \in [n]$. By the induction hypothesis we have $e_i = 0$ for all $i \in [n]$. Now from the definition of α we get $e = l_n^{\mathbb{A}_{\mathrm{Aux}}}(0, 0, \ldots, 0) = 0$ (since from $t^{\mathbb{A}} \notin \{a_0, a_{\max}\}$ we have $\overline{t^{\mathbb{B}}} = t^{\mathbb{A}^-} \neq a_0$).

From the above facts it follows that $\mathbb{A}$ is a homomorphic image of $\mathbb{C}$ under the function that maps each element (b, e) of $\mathbb{C}$ to $\overline{b}$ if $e = 0$, to a_0 if $e = 1$, and to $a_{\max}$ if $e = 2$. Since $\mathbb{C}$ is in $\langle \mathbb{A}^-, \mathbb{E}_{\mathrm{EF}}^{*} \rangle_M$, the result follows. ■

From the above lemma we get the main result of this subsection as a consequence (which can be proved analogously to Theorem 2.4.30):

THEOREM 2.4.36 *The connected Moore pseudovariety $\langle \mathbb{E}_{\mathrm{EF}}^{*} \rangle_M$ contains exactly the finite connected tree automata that are monotone, commutative, maximal dependent and stutter invariant:*

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

2.4.5 Adding $\mathbb{D}_0$

Our aim in this subsection is to provide characterizations of the connected Moore pseudovarieties $\langle \mathbb{E}_{\mathrm{EF}}^{+}, \mathbb{D}_0 \rangle_M = \langle \mathbb{E}_{\mathrm{EF}}^{+} \rangle_M \times \mathbf{D}_1^c$ and $\langle \mathbb{E}_{\mathrm{EF}}^{*}, \mathbb{D}_0 \rangle_M = \langle \mathbb{E}_{\mathrm{EF}}^{*} \rangle_M \times \mathbf{D}_1^c$. Since both $\mathbb{E}_{\mathrm{EF}}^{+}$ and $\mathbb{E}_{\mathrm{EF}}^{*}$ are contained in $\mathbf{Mon}^c$ but $\mathbb{D}_0$ is not, first we study the pseudovariety $\mathbf{Mon}^c \times \mathbf{D}_1^c$ of finite connected tree automata.

We make the following observation:

PROPOSITION 2.4.37

$$\mathbf{Mon} \times \mathbf{D}_1 \;\subseteq\; \mathbf{CompDep} \cap \mathbf{CompUnique}.$$

Proof. We know that monotone automata are both component dependent and componentwise unique. It is clear that the same holds for 1-definite automata. ■

Our aim now is to show that we in fact have equality. In order to prove this, we introduce another notion: we say that a tree automaton $\mathbb{A} = (A, \Sigma)$ has the *disjoint images property*, if for any pair of symbols $\sigma_1 \neq \sigma_2 \in \Sigma$, the sets $\mathrm{Img}_{\mathbb{A}}(\sigma_1)$ and $\mathrm{Img}_{\mathbb{A}}(\sigma_2)$ are disjoint.

Recall the definition of the 1-definite automata of the form $\mathbb{A}^A_\Sigma$ (see page 30).

LEMMA 2.4.38 *Let $\mathbb{A} = (A, \Sigma)$ be a finite, component dependent and componentwise unique connected tree automaton having the disjoint images property. Then there exists a congruence Θ of $\mathbb{A}$ such that the following both hold:*

1. *$\mathbb{A}/\Theta$ is monotone;*
2. *$\mathbb{A}$ divides the direct product $(\mathbb{A}/\Theta) \times \mathbb{A}^A_\Sigma$.*

Proof. Assume $\mathbb{A} = (A, \Sigma)$ is such a tree automaton. We claim that the relation $\sim_{\mathbb{A}}$ satisfies the required properties. Since $\mathbb{A}$ is component dependent, $\sim_{\mathbb{A}}$ is a congruence. Since $\sim_{\mathbb{A}}$ is a congruence, we clearly have by Lemma 2.4.2 that $\mathbb{B} = (B, \Sigma) = \mathbb{A}/\sim_{\mathbb{A}}$ is monotone. So it remains to show that $\mathbb{A}$ is a divisor of the direct product $\mathbb{C} = \mathbb{B} \times \mathbb{A}^A_\Sigma$. Let $\overline{a}$ denote the class $a/\sim_{\mathbb{A}}$.

Let $(\overline{a}, \sigma)$ be a state of $\mathbb{C}$ with $a \in A$ and $\sigma \in \Sigma$. Then there exists at most one $a' \in A$, $a' \sim_{\mathbb{A}} a$ such that $a' \in \mathrm{Img}_{\mathbb{A}}(\sigma)$. Indeed, suppose $\sigma^{\mathbb{A}}(a_1, \ldots, a_n) = a' \sim_{\mathbb{A}} a$ and $\sigma^{\mathbb{A}}(b_1, \ldots, b_n) = a'' \sim_{\mathbb{A}} a$. Then $a' \sim_{\mathbb{A}} a''$, thus there exists ΣA-polynomial contexts p and q with $p^{\mathbb{A}}(a') = a''$ and $q^{\mathbb{A}}(a'') = a'$. If either p or q is x_1, then we clearly have $a' = a''$. Suppose both p and q are proper. Then, from the disjoint images property and that $a', a'' \in \mathrm{Img}_{\mathbb{A}}(\sigma)$, $\mathrm{Root}(p) = \mathrm{Root}(q) = \sigma$ holds, which implies $a' = a''$, since $\mathbb{A}$ is componentwise unique.

Let $\mathbb{C}'$ be the subautomaton of $\mathbb{C}$ consisting of the states of the form $(\overline{a}, a)$, $a \in A$ and of the form $(\overline{a}, \sigma)$ with $a \in \mathrm{Img}_{\mathbb{A}}(\sigma)$ (it is easy to see that this set indeed determines a subautomaton of $\mathbb{C}$). Then, a homomorphism $\mathbb{C}' \to \mathbb{A}$ is given by the function which maps each state $(\overline{a}, a)$ to a and each state $(\overline{a}, \sigma)$ to the unique state a' with $\overline{a} = \overline{a'}$ and $a' \in \mathrm{Img}_{\mathbb{A}}(\sigma)$. ■

From this we get the desired result:

COROLLARY 2.4.39

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

Proof. Let $\mathbb{A} = (A, \Sigma)$ be a finite, component dependent and componentwise unique tree automaton. Consider the direct product $\mathbb{A} \times \mathbb{A}_\Sigma$. It is still a finite, component dependent and componentwise unique tree automaton since $\mathbb{A}_\Sigma$ also has these properties; moreover, it has the disjoint images property. So by Lemma 2.4.38 we have that $\mathbb{A} \times \mathbb{A}_\Sigma$ is contained in the pseudovariety $\mathbf{Mon} \times \mathbf{D}_1$ of finite tree automata. Since $\mathbb{A}$ is a homomorphic image of this automaton, the same holds for $\mathbb{A}$. ■

Both of the aimed characterizations are consequences of the following theorem:

THEOREM 2.4.40 *Let* $\mathbf{V}$ *be a pseudovariety of finite tree automata such that* $\mathbf{D}_1 \subseteq \mathbf{V} \subseteq \mathbf{Mon} \times \mathbf{D}_1$. *Then* $(\mathbf{Mon} \cap \mathbf{V}) \times \mathbf{D}_1 = \mathbf{V}$.

Proof. We have that

$$\begin{aligned}(\mathbf{Mon} \cap \mathbf{V}) \times \mathbf{D}_1 &\subseteq (\mathbf{Mon} \times \mathbf{D}_1) \cap (\mathbf{V} \times \mathbf{D}_1) \\ &\subseteq (\mathbf{Mon} \times \mathbf{D}_1) \cap \mathbf{V} \text{ (since } \mathbf{D}_1 \subseteq \mathbf{V}) \\ &\subseteq \mathbf{V}.\end{aligned}$$

Let $\mathbb{A} = (A, \Sigma) \in \mathbf{V}$. From Lemma 2.4.38 we know that $\mathbb{A}$ divides some direct product $\mathbb{B} \times \mathbb{A}_\Sigma^A$, where $\mathbb{B}$ is a monotone quotient of $\mathbb{A}$. From this it follows that $\mathbb{B} \in \mathbf{Mon} \cap \mathbf{V}$ and thus $\mathbb{A} \in (\mathbf{Mon} \cap \mathbf{V}) \times \mathbf{D}_1$; so $\mathbf{V} \subseteq (\mathbf{Mon} \cap \mathbf{V}) \times \mathbf{D}_1$. ■

As consequences we get:

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

Proof. By Proposition 2.4.18, $\mathbf{CompDep}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c = \mathbf{CompUnique}^c \cap \mathbf{CompDep}^c \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c$. Applying Corollary 2.4.39 this further equals $(\mathbf{Mon}^c \times \mathbf{D}_1^c) \cap \mathbf{Com}^c \cap \mathbf{MaxDep}^c$. Let $\mathbf{V}^c$ denote this pseudovariety of finite connected tree automata. By Theorem 2.4.30 we have $\langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M = \mathbf{Mon}^c \cap \mathbf{V}^c$. Since we also have $\mathbf{D}_1^c \subseteq \mathbf{V}^c \subseteq \mathbf{Mon}^c \times \mathbf{D}_1^c$, thus by Theorem 2.4.40

$$\begin{aligned}\langle \mathbb{E}_{\mathrm{EF}}^+, \mathbb{D}_0 \rangle_M &= \langle \mathbb{E}_{\mathrm{EF}}^+ \rangle_M \times \mathbf{D}_1^c \\ &= (\mathbf{Mon}^c \cap \mathbf{V}^c) \times \mathbf{D}_1^c \\ &= \mathbf{V}^c,\end{aligned}$$

proving i). The proof of ii) is analogous. ■

From Theorem 2.3.3 we get the following alternative formulation:

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 finite tree automaton $\mathbb{A} = (A, \Sigma)$ and a set $F \subseteq A$ whether the regular language $L(\mathbb{A}, F)$ is in $\mathbf{CTL}(\mathrm{EF}^+)$, or in $\mathbf{CTL}(\mathrm{EF}^*)$. In [19] we have shown polynomial-time decision procedures for the membership problems of each of the connected 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 $\langle \mathbb{E}^*_{\mathrm{EF}}, \mathbb{D}_0 \rangle_M$, which proves that the definability problem of each of the fragments $\mathrm{CTL}(\mathrm{EF}^+)$ and $\mathrm{CTL}(\mathrm{EF}^*)$ is decidable in polynomial time, when the language L is given by its minimal tree automaton. For the fragment $\mathbf{CTL}(\mathrm{EF}^+)$ this was already shown in [5] using different methods.

In [56] (see also [34]), it was proved that in the case of word automata (which corresponds to the case when $R = \{0, 1\}$) it holds that a regular language L is definable in the logic $\mathrm{CTL}(\mathrm{EF}^+)$ if and only if it is definable in $\mathrm{CTL}(\mathrm{EF}^*)$ and its minimal automaton is stutter invariant. We note that for word languages, the logic $\mathrm{CTL}(\mathrm{EF}^*)$ corresponds to TL[F], the fragment of linear temporal logic determined by the strict F modality, since in that case there is no branching and the path quantifiers E and A coincide. In the same way, $\mathrm{CTL}(\mathrm{EF}^*)$ corresponds to $\mathrm{TL}[\mathrm{F}_{\mathrm{sf}}]$, linear temporal logic equipped with only the non-strict (also called stutter-free) F-modality.

Corollary 2.4.42 extends this result to the case of regular languages over finite trees. Let us inspect the word case in more detail. For word languages the properties of commutativity and maximal dependency are trivial. Moreover, component dependency implies componentwise uniqueness. So in the word case the above definability theorem can be reformulated as follows: a word language L is definable in $\mathrm{CTL}(\mathrm{EF}^+)$ if and only if L is regular and its minimal automaton $\mathbb{A}_L$ is component dependent; moreover, L is definable in $\mathrm{CTL}(\mathrm{EF}^*)$ if and only if $\mathbb{A}_L$ is component dependent and stutter invariant. This yields exactly the same characterization and decision procedure as the one described in [56], where component dependency is defined in terms of forbidden patterns in the labeled graph of the minimal automaton.

We get yet another interesting fact from Theorem 2.4.30 for the classical case. Since both maximal dependency and commutativity are trivial properties, we have that $\langle \mathbb{E}^+_{\mathrm{EF}} \rangle_M = \mathbf{Mon}^c$ in the word case (and similarly, from Theorem 2.4.36 we get that $\langle \mathbb{E}^*_{\mathrm{EF}} \rangle_M$ contains exactly the finite connected automata that are both monotone and stutter invariant).

2.5 Ehrenfeucht-Fraïssé type games

In this section we define the so-called n-round $\mathcal{L}$-game for any class $\mathcal{L}$ of tree languages and number $n \geq 0$ of rounds and show 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 $t_0, t_1 \in T_\Sigma$. 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 with $(\widehat{t_0}, \widehat{t_1})(x) \neq (\widehat{t_0}, \widehat{t_1})(y)$. If he cannot do so, Spoiler wins. Otherwise, an $(n-1)$-round $\mathcal{L}$-game is played on the pair $((t_0, t_1)|_x, (t_0, t_1)|_y)$ of trees. Whoever wins the subgame, also wins the game.

Clearly, for any class $\mathcal{L}$ of languages, number $n \geq 0$ 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 will show that the relations $\equiv^n_{\mathcal{L}}$ and $\sim^n_{\mathcal{L}}$ coincide. Recall that $s \equiv^n_{\mathcal{L}} t$ if and only if the trees s and t satisfy the same set of FTL($\mathcal{L}$)-formulas of depth at most n.

Proposition 2.5.1 *For any class $\mathcal{L}$ of languages, integer n and trees $s, t \in T_\Sigma$, if $s \sim^n_{\mathcal{L}} t$ then $s \equiv^n_{\mathcal{L}} t$.*

Proof. We argue by induction on n and contraposition. Let s, t be Σ-trees with $s \not\equiv^n_{\mathcal{L}} t$.

If $n = 0$, then $\mathrm{Root}(s) \neq \mathrm{Root}(t)$, thus Spoiler indeed wins the $\mathcal{L}$-game on (s, t) in 0 rounds.

Suppose that $n > 0$ and we have proved the claim for $n - 1$. Then, $s \not\equiv^n_{\mathcal{L}} t$ implies either $s \not\equiv^{n-1}_{\mathcal{L}} t$, or s and t can be separated by some complete and deterministic FTL($\mathcal{L}$)-formula $\varphi = L(\delta \mapsto \varphi_\delta)$ of depth n.

If $s \not\equiv^{n-1}_{\mathcal{L}} t$, we can apply the induction hypothesis and get that Spoiler wins the $(n-1)$-round $\mathcal{L}$-game on (s, t), thus also wins the n-round $\mathcal{L}$-game on these trees.

Suppose that s and t can be separated by the complete and deterministic FTL($\mathcal{L}$)-formula $\varphi = L(\delta \mapsto \varphi_\delta)$ of depth n, say $s \models \varphi$ and $t \not\models \varphi$. Then let Spoiler choose the language L and the relabelings $\widehat{s}$ and $\widehat{t}$ that are the characteristic trees of s and t, respectively, determined by the family $(\varphi_\delta)_{\delta\in\Delta}$. From the semantics we get that $\widehat{s} \in L$ and $\widehat{t} \notin L$ indeed hold, thus this is a valid move. Now suppose Duplicator responds by choosing the nodes x and y of (s,t) such that $\delta = (\widehat{s},\widehat{t})(x) \neq (\widehat{s},\widehat{t})(y)$. Since the family $(\varphi_\delta)_{\delta\in\Delta}$ is complete and deterministic, we have that φ_δ separates the trees $(s,t)|_x$ and $(s,t)|_y$. Since the depth of this formula is at most $n-1$, applying the induction hypothesis we get that Spoiler wins the $(n-1)$-round subgame, thus also the whole game. ■

PROPOSITION 2.5.2 *For any class $\mathcal{L}$ of languages, integer n and trees $s,t \in T_\Sigma$, if $s \equiv^n_{\mathcal{L}} t$, then $s \sim^n_{\mathcal{L}} t$.*

Proof. We again argue by induction on n and contraposition. Let s,t be trees with $s \not\sim^n_{\mathcal{L}} t$.

If $n = 0$, then $\mathrm{Root}(s) \neq \mathrm{Root}(t)$, thus $s \not\equiv^0_{\mathcal{L}} t$.

Suppose that $n > 0$ and we have proved the claim for $n-1$. We have to consider two cases: either Spoiler already wins the $(n-1)$-round game, or he chooses an $L \in \mathcal{L}$ and two relabelings of the trees in the first step when playing a winning strategy.

If Spoiler already wins the $(n-1)$-round $\mathcal{L}$-game, then by the induction hypothesis $s \not\equiv^{n-1}_{\mathcal{L}} t$ holds, which clearly implies $s \not\equiv^n_{\mathcal{L}} t$.

Now without loss of generality, assume Spoiler chooses the tree language $L \in \mathcal{L}$ over Δ and the relabelings $\widehat{s} \in L$, $\widehat{t} \notin L$ of s and t, when playing his winning strategy. Then for any pair x,y of nodes of (s,t) with $(\widehat{s},\widehat{t})(x) \neq (\widehat{s},\widehat{t})(y)$, Spoiler wins the $(n-1)$-round $\mathcal{L}$-game on $((s,t)|_x,(s,t)|_y)$. Applying the induction hypothesis, we get that for any such pair (x,y) there exists an FTL($\mathcal{L}$)-formula $\varphi_{x,y}$ of depth at most $n-1$ with $(s,t)|_x \models \varphi_{x,y}$ and $(s,t)|_y \not\models \varphi_{x,y}$.

For each $\delta \in \Delta$, let us define the formula

$$\varphi_\delta \quad = \quad \bigvee_{(\widehat{s},\widehat{t})(x)=\delta} \bigwedge_{(\widehat{s},\widehat{t})(y)\neq\delta} \varphi_{x,y},$$

where x and y range over the nodes of (s,t).

First, observe that,

$$(\widehat{s},\widehat{t})(z) = \delta \quad \Rightarrow \quad (s,t)|_z \models \varphi_\delta$$

for any node z of (s,t) and symbol $\delta \in \Delta$. Thus, the family $(\varphi_\delta)_{\delta\in\Delta}$ is complete over (s,t).

Also,

$$(s,t)|_z \models \varphi_\delta \quad \Rightarrow \quad (\widehat{s},\widehat{t})(z) = \delta.$$

Indeed, suppose $(\widehat{s},\widehat{t})(z) \neq \delta$ and $(s,t)|_z \models \varphi_\delta$. Then there exists a node x with $(\widehat{s},\widehat{t})(x) = \delta$ such that $(s,t)|_z \models \bigwedge_{(\widehat{s},\widehat{t})(y)\neq\delta} \varphi_{x,y}$. Then $(s,t)|_z \models \varphi_{x,z}$, which contradicts the definition of the formula $\varphi_{x,z}$.

Thus the family $(\varphi_\delta)_{\delta\in\Delta}$ is complete and deterministic over (s,t), moreover, $\widehat{s}$ and $\widehat{t}$ are the characteristic trees of s and t, respectively, determined by the family $(\varphi_\delta)_{\delta\in\Delta}$. Now since $\widehat{s} \in L$ and $\widehat{t} \notin L$, we conclude that the FTL($\mathcal{L}$)-formula $L(\delta \mapsto \varphi_\delta)$ of depth n separates s and t, thus the statement is proven. ■

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 $t \notin L$.

Proof. Suppose $\mathcal{L}$ is a finite class of tree languages, L is a tree language and $n \geq 0$ is an integer such that Spoiler wins the n-round $\mathcal{L}$-game on any pair (s,t) of trees with $s \in L$ and $t \notin L$.

Then for any such pair (s,t) of trees there exists an FTL($\mathcal{L}$)-formula $\varphi_{s,t}$ such that $s \models \varphi_{s,t}$ and $t \not\models \varphi_{s,t}$. Each of these formulas is of depth at most n.

Since $\mathcal{L}$ is finite, by standard arguments from finite model theory, it follows that, up to equivalence, there exist only a finite number of formulas of depth at most n.

Thus, for any tree $s \in L$, the "infinitary conjunction" $\bigwedge_{t\notin L} \varphi_{s,t}$ is equivalent to some FTL($\mathcal{L}$)-formula φ_s of depth at most n. Also the "infinitary disjunction" $\bigvee_{s\in L} \varphi_s$ is equivalent to some formula φ; it is straightforward to see that $L_\varphi = L$ indeed holds, proving ii) → i). The other direction is a direct consequence of Corollary 2.5.3. ■

Chapter 3

Aperiodicity

In this chapter we define a hierarchy of aperiodicity notions for finite tree automata by considering n-tuples of proper trees in n variables and the semigroup of vector-valued term functions induced by them. We say that a finite tree automaton is n-aperiodic for some integer $n > 0$ if the semigroup of all such term functions is aperiodic. Moreover, we say that a finite tree automaton is strongly aperiodic if it is n-aperiodic for each n. We show that n-aperiodic finite tree automata form a proper hierarchy, and the class of 1-aperiodic finite tree automata is properly contained in the class of context aperiodic finite tree automata, when the rank type R contains an integer greater than 1. We also show that in this case the class of strongly aperiodic finite tree automata properly contains the class of all finite definite tree automata. Moreover, we establish that each of these classes is a cascade pseudovariety of finite tree automata. We also study the complexity of the membership problem of these classes. In particular, we establish a P-time algorithm for testing strong aperiodicity. We also provide an extension of the above aperiodicity notions which is motivated by the Krohn-Rhodes decomposition theory [9] and study a modified version of aperiodicity obtained by taking polynomial functions instead of term functions. Finally, we relate aperiodicity to logic.

3.1 The notion of n-aperiodicity

For the balance of this section, we fix an arbitrary rank type R, containing at least one positive integer. Let $\mathbb{A} = (A, \Sigma)$ be a 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 $t_i^{\mathbb{A}} : A^n \to A$, $i \in [m]$. 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. We let $C(\mathbb{A})$ denote the subsemigroup of $S_1(\mathbb{A})$ consisting of the term functions induced by the proper contexts if this set is not empty. Otherwise we let $C(\mathbb{A})$ be a trivial semigroup.

PROPOSITION 3.1.1 *When $n \leq m$, $S_n(\mathbb{A})$ divides $S_m(\mathbb{A})$.*

Proof. First note that since $n \leq m$, we have $T_\Sigma(X_n) \subseteq T_\Sigma(X_m)$. Consider the functions $f = \langle t_1^{\mathbb{A}}, \ldots, t_m^{\mathbb{A}} \rangle \in S_m(\mathbb{A})$ such that for each $i \in [n]$, t_i is a proper tree in $T_\Sigma(X_m)$ not containing any occurrence of a variable x_j with $j > n$. These functions form a subsemigroup T of $S_m(\mathbb{A})$. Moreover, $S_n(\mathbb{A})$ is a homomorphic image of T, one homomorphism being the map that takes any f of the above form to $\langle t_1^{\mathbb{A}}, \ldots, t_n^{\mathbb{A}} \rangle$, where now each t_i is considered as a tree in $T_\Sigma(X_n)$ inducing a function $A^n \to A$. ■

The following fact is clear:

PROPOSITION 3.1.2 *$C(\mathbb{A})$ is a subsemigroup of $S_1(\mathbb{A})$. When R is either $\{1\}$ or $\{0,1\}$, $C(\mathbb{A}) = S_1(\mathbb{A})$.*

Recall from [9] that a finite semigroup S is called *aperiodic* if it contains no nontrivial group, or equivalently, when there exists an integer $k \geq 1$ such that $s^k = s^{k+1}$, for all $s \in S$.

DEFINITION 3.1.3 *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 $n \geq 1$.*

DEFINITION 3.1.4 *We call a finite Σ-tree automaton $\mathbb{A}$* context aperiodic *if $C(\mathbb{A})$ is aperiodic.*

The notion of context aperiodicity was introduced by Thomas in [54] under the name aperiodicity. In the classical case when $R = \{1\}$ or $R = \{0,1\}$, context aperiodicity and 1-aperiodicity coincide with aperiodicity, or counter-freeness, see below.

EXAMPLE 3.1.5 In case of semigroups, the signature contains a single binary symbol. Since there are no proper contexts, every finite semigroup is context aperiodic. We show that a finite semigroup S is 1-aperiodic if and only if its exponent is 1 or 2. (Recall that the exponent of S is the least positive integer d such that $s^k = s^{k+d}$ for all $s \in S$.)

Suppose first that S is a finite semigroup which is 1-aperiodic. For each integer $k > 1$, consider the proper tree $x_1^k = x_1 \cdot (x_1 \cdot \ldots \cdot (x_1 \cdot x_1) \ldots)$ in the variable x_1. Since S is

1-aperiodic, there exists some integer m such that the equation $x_1^{k^{m+1}} = x_1^{k^m}$ holds in S. This implies that its exponent d is a divisor of $k^{m+1} - k^m = k^m(k-1)$. Thus, each prime appearing in the prime decomposition of d must divide $k(k-1)$. Since this holds for all $k > 1$, we conclude that $d = 1$ or d is a power of 2. But when $k = 3$, then d must divide $3^m \cdot 2$ for some m, so that the exponent is 1 or 2.

When the exponent is 1, S is aperiodic. It is then clear that S is 1-aperiodic. Assume that the exponent is 2. Then there is an integer n such that $s^{n+2} = s^n$ for all $s \in S$. Consider any tree x_1^k, where $k \geq 2$. When $k^m \geq n$, we have that $s^{k^{m+1}} = s^{k^m}$ for all $s \in S$. This means that $x_1^{k^m}$ and $x_1^{k^{m+1}}$ induce the same term functions. It follows that S is 1-aperiodic.

Example 3.1.6 Recall that a semigroup S is *nilpotent* if there exist an element $0 \in S$ and an integer $n > 0$ such that $S^n = 0$. It is clear that any finite nilpotent semigroup is strongly aperiodic. We show that every finite 2-aperiodic semigroup S is nilpotent. To this end, first note that the exponent of S is 1 or 2 by the above argument. But if the exponent is 2, then there exist some $s \in S$ and $n > 0$ such that $s^n = s^{n+2}$ but $s^n \neq s^{n+1}$. Now both s^n and s^{n+1} are fixed points of the proper term function induced by x_1^3. Thus, by Proposition 3.1.9 below, S is not 2-aperiodic. We conclude that the exponent of S is 1, i.e., S is aperiodic, so that there exists $n > 0$ such that $x^{n+1} = x^n$ holds in S. Consider now the pair (x_2^2, x_1^2) of proper trees. Since S is 2-aperiodic, there exists some k such that $2^k \geq n$ and $x_1^{2^k} = x_2^{2^{k+1}}$ holds in S. But since $2^k \geq n$, $x_1^{2^k} = x_1^n$ and $x_2^{2^{k+1}} = x_2^n$, so that $x_1^n = x_2^n$ also holds. Now this implies that S has a single idempotent e and $e = s^n$ for each $s \in S$. But then since $xy^n = xx^n = x^n$ and $y^n x = x^n$ also hold, $Se = eS = e$. Since all long enough products over S have a factorization which contains an idempotent, it follows that S is nilpotent.

Since any divisor of an aperiodic semigroup is aperiodic, from Propositions 3.1.1 and 3.1.2 we have:

Proposition 3.1.7 *Every 1-aperiodic finite tree automaton is context aperiodic, and when $n \leq m$, every m-aperiodic tree automaton is n-aperiodic.*

When A is a finite set and S is a semigroup of functions $A \to A$ (product in S is function composition), then the pair (A, S) is called a *transformation semigroup*, cf. [9]. The *(left) action* of S on A is defined by $sa := s(a)$, for each $a \in A$ and $s \in S$. A transformation semigroup (A, S) is called *counter-free* when there exists no $s \in S$ which induces a nontrivial (cyclic) permutation of a subset of A. It is well-known that S is aperiodic if and only if (A, S) is counter-free. Note that for each n, $(A^n, S_n(\mathbb{A}))$ is a transformation semigroup as is $(A, C(\mathbb{A}))$.

PROPOSITION 3.1.8 *A finite Σ-tree automaton $\mathbb{A}$ is n-aperiodic if and only if $(A^n, S_n(\mathbb{A}))$ is counter-free. Moreover, it is context aperiodic if and only if $(A, C(\mathbb{A}))$ is counter-free.*

For later use we prove:

PROPOSITION 3.1.9 *For any integer $n > 0$, $2n$-aperiodic finite tree automaton $\mathbb{A}$ and proper term function $f : A^n \to A^n$ it holds that f has at most one fixed point.*

Proof. Let $n > 0$ be an integer, $\mathbb{A}$ a $2n$-aperiodic finite Σ-tree automaton, $t_1, \ldots, t_n \in T_\Sigma(X_n)$ proper trees and $(a_1, \ldots, a_n)$, $(b_1, \ldots, b_n) \in A^n$ two fixed points of $f = \langle t_1^{\mathbb{A}}, \ldots, t_n^{\mathbb{A}} \rangle$. We have to show that $a_i = b_i$ for all $i \in [n]$.

For any $i \in [n]$ let s_i denote the tree that results from t_i by replacing each occurrence of x_j by x_{n+j}, for each $j \in [n]$ (so we add n to the index of each variable). Now the proper term function

$$f' \quad = \quad \langle s_1^{\mathbb{A}}, \ldots, s_n^{\mathbb{A}}, t_1^{\mathbb{A}}, \ldots, t_n^{\mathbb{A}} \rangle : A^{2n} \to A^{2n}$$

satisfies both

$$f'(a_1, \ldots, a_n, b_1, \ldots, b_n) \quad = \quad (b_1, \ldots, b_n, a_1, \ldots, a_n)$$

and

$$f'(b_1, \ldots, b_n, a_1, \ldots, a_n) \quad = \quad (a_1, \ldots, a_n, b_1, \ldots, b_n).$$

Since $\mathbb{A}$ is $2n$-aperiodic, $(a_1, \ldots, a_n, b_1, \ldots, b_n) = (b_1, \ldots, b_n, a_1, \ldots, a_n)$, hence $a_i = b_i$ for all $i \in [n]$. ■

PROPOSITION 3.1.10 *Suppose $\mathbb{A}$ is a finite tree automaton which is* not *n-aperiodic for the integer $n > 0$. Then there exists a proper term function $f : A^n \to A^n$ having at least two different fixed points.*

Proof. If $\mathbb{A}$ is not n-aperiodic, then there exist a proper term function $f : A^n \to A^n$ and a subset $B \subseteq A^n$ with $k > 1$ elements such that the restriction of f to B is a cyclic permutation of B. But then each element of B is a fixed point of the proper term function f^k. ■

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.*

REMARK 3.1.12 We can show that a finite tree automaton $\mathbb{A}$ is strongly aperiodic if and only if no proper term function $A^n \to A^n$ has two or more different fixed points where $n = |A|^2$.

3.2 The 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 $A \times B$ of states 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 $t = \gamma_n(a_1, \ldots, a_n, \sigma)$.

REMARK 3.2.1 When the range of each γ_n only contains trees of the form $\delta(x_1, \ldots, x_n)$, where $\delta \in \Delta_n$, we may view each γ_n as a function $A^n \times \Sigma_n \to \Delta_n$ and thus $\mathbb{A} \times_\gamma \mathbb{B}$ can be viewed as a cascade product.

REMARK 3.2.2 The generalized cascade product can be defined in terms of the cascade product. We say that a tree automaton $\mathbb{B}'$ is a *derived automaton* of a tree automaton $\mathbb{B}$ if $\mathbb{B}$ and $\mathbb{B}'$ have the same set of states and each basic operation of $\mathbb{B}'$ is a proper term function of $\mathbb{B}$. Any generalized cascade product $\mathbb{A} \times_\gamma \mathbb{B}$ of the finite tree automata $\mathbb{A}$ and $\mathbb{B}$ is clearly a cascade product $\mathbb{A} \times_\alpha \mathbb{B}'$ for some derived automaton $\mathbb{B}'$ of $\mathbb{B}$ and for some α.

DEFINITION 3.2.3 *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.*

REMARK 3.2.4 A cascade pseudovariety of finite tree automata is a generalized cascade pseudovariety if and only if it is closed under derived automata.

In the rest of this section we show that for each $n > 0$, the class of all n-aperiodic finite tree automata is a generalized cascade pseudovariety. Hence, the class of all strongly aperiodic finite tree automata is also a generalized cascade pseudovariety.

We begin with the following lemma.

LEMMA 3.2.5 *Let $n > 0$ be an integer and $\mathbb{A}$ an n-aperiodic finite tree automaton. Then any divisor of $\mathbb{A}$ is also n-aperiodic.*

Proof. It is clear that taking subautomata preserves n-aperiodicity.

Let $\mathbb{B} = (B, \Sigma)$ be a homomorphic image of $\mathbb{A}$ under the mapping $a \mapsto \overline{a}$. Let $\underline{t}$ be an n-tuple of proper ΣX_n-trees. Since $\mathbb{A}$ is n-aperiodic, there exists an integer $N > 0$ such

that $(\underline{t}^{\mathbb{A}})^N = (\underline{t}^{\mathbb{A}})^{N+1}$. Hence, for any n-tuple of states $\overline{a_1}, \ldots, \overline{a_n} \in B$ it holds that

$$\begin{aligned}(\underline{t}^{\mathbb{B}})^N(\overline{a_1}, \ldots, \overline{a_n}) &= \overline{(\underline{t}^{\mathbb{A}})^N(a_1, \ldots, a_n)} \\ &= \overline{(\underline{t}^{\mathbb{A}})^{N+1}(a_1, \ldots, a_n)} \\ &= (\underline{t}^{\mathbb{B}})^{N+1}(\overline{a_1}, \ldots, \overline{a_n}),\end{aligned}$$

showing that $\mathbb{B}$ is indeed n-aperiodic. ■

Before proving that the generalized cascade product also preserves n-aperiodicity, we make the following observation, that is similar to Lemma 2.2.1:

LEMMA 3.2.6 *Let $\mathbb{A}$ be a Σ-tree automaton, $\mathbb{B}$ a Δ-tree automaton and $\mathbb{C} = \mathbb{A} \times_\gamma \mathbb{B}$ a generalized cascade product of $\mathbb{A}$ and $\mathbb{B}$. Then for any integer $n \geq 0$, proper tree $t \in T_\Sigma(X_n)$ and states $a_1, \ldots, a_n \in A$ there exists a proper tree $s \in T_\Delta(X_n)$ such that*

$$t^{\mathbb{C}}\big((a_1, b_1), \ldots, (a_n, b_n)\big) \quad = \quad \big(t^{\mathbb{A}}(a_1, \ldots, a_n), s^{\mathbb{B}}(b_1, \ldots, b_n)\big)$$

for any $(a_1, b_1), \ldots, (a_n, b_n) \in C$.

LEMMA 3.2.7 *Let $n > 0$ be an integer, $\mathbb{A}$ an n-aperiodic finite Σ-tree automaton, $\mathbb{B}$ an n-aperiodic finite Δ-tree automaton, and $\mathbb{C} = \mathbb{A} \times_\gamma \mathbb{B}$ a generalized cascade product of $\mathbb{A}$ and $\mathbb{B}$. Then $\mathbb{C}$ is also n-aperiodic.*

Proof. We know that there exist integers N_A and N_B such that $f^{N_A} = f^{N_A+1}$ and $g^{N_B} = g^{N_B+1}$ for all proper term functions $f : A^n \to A^n$ and $g : B^n \to B^n$ of $\mathbb{A}$ and $\mathbb{B}$, respectively. We show that this implies that $h^{N_A+N_B} = h^{N_A+N_B+1}$ for all proper term functions $h : (A \times B)^n \to (A \times B)^n$ of $\mathbb{C}$.

Let $\underline{t} = (t_1, \ldots, t_n)$ be an n-tuple of proper ΣX_n-trees. Let us denote by $((a'_1, b'_1), \ldots, (a'_n, b'_n))$ the n-tuple $(\underline{t}^{\mathbb{C}})^{N_A}((a_1, b_1), \ldots, (a_n, b_n))$. Applying Lemma 3.2.6 to the trees $(t_1, \ldots, t_n)$ and the states $a'_1, \ldots, a'_n \in A$ we get that there exist proper trees $s_1, \ldots, s_n \in T_\Delta(X_n)$ such that for any $(a'_1, y_1), \ldots, (a'_n, y_n) \in C$ the value of $\underline{t}^{\mathbb{C}}((a'_1, y_1), \ldots, (a'_n, y_n))$ can be written as

$$\big((t_1^{\mathbb{A}}(a'_1, \ldots, a'_n), s_1^{\mathbb{B}}(y_1, \ldots, y_n)), \ldots, (t_n^{\mathbb{A}}(a'_1, \ldots, a'_n), s_n^{\mathbb{B}}(y_1, \ldots, y_n))\big).$$

From the definition of $a'_1, \ldots, a'_n$ we get that $t_i^{\mathbb{A}}(a'_1, \ldots, a'_n) = a'_i$ for all $i \in [n]$, so

$$\underline{t}^{\mathbb{C}}((a'_1, y_1), \ldots, (a'_n, y_n)) \quad = \quad \big((a'_1, s_1^{\mathbb{B}}(y_1, \ldots, y_n)), \ldots, (a'_n, s_n^{\mathbb{B}}(y_1, \ldots, y_n))\big).$$

Iterating this from $(a'_1, b'_1), \ldots, (a'_n, b'_n)$ it follows that for any $k \geq 0$ we can write $(\underline{t}^{\mathbb{C}})^k\big((a'_1, b'_1), \ldots, (a'_n, b'_n)\big)$ as

$$\big((a'_1, (s_1^{\mathbb{B}})^k(b'_1, \ldots, b'_n)), \ldots, (a'_n, (s_n^{\mathbb{B}})^k(b'_1, \ldots, b'_n))\big).$$

But we know that $(s_i^{\mathbb{B}})^{N_B} = (s_i^{\mathbb{B}})^{N_B+1}$ holds for all $i \in [n]$. Summing up we get the following equality:

$$\begin{aligned}
& (\underline{t}^{\mathbb{C}})^{N_A+N_B}((a_1,b_1),\ldots,(a_n,b_n)) \\
= \; & (\underline{t}^{\mathbb{C}})^{N_B}((a'_1,b'_1),\ldots,(a'_n,b'_n)) \\
= \; & \big((a'_1,(s_1^{\mathbb{B}})^{N_B}(b'_1,\ldots,b'_n)),\ldots,(a'_n,(s_n^{\mathbb{B}})^{N_B}(b'_1,\ldots,b'_n))\big) \\
= \; & \big((a'_1,(s_1^{\mathbb{B}})^{N_B+1}(b'_1,\ldots,b'_n)),\ldots,(a'_n,(s_n^{\mathbb{B}})^{N_B+1}(b'_1,\ldots,b'_n))\big) \\
= \; & (\underline{t}^{\mathbb{C}})^{N_B+1}((a'_1,b'_1),\ldots,(a'_n,b'_n)) \\
= \; & (\underline{t}^{\mathbb{C}})^{N_A+N_B+1}((a_1,b_1),\ldots,(a_n,b_n)),
\end{aligned}$$

so $\mathbb{C}$ is indeed n-aperiodic. ■

For each n, let $\mathbf{SAper}_n$ denote the class of all n-aperiodic finite tree automata. Thus, $\mathbf{SAper} = \bigcap_{n\geq 1} \mathbf{SAper}_n$ is the class of all strongly aperiodic finite tree automata. Moreover, let $\mathbf{CAper}$ denote the class of all context aperiodic finite tree automata. From the above facts we get the following:

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

In a similar way, we have:

THEOREM 3.2.9 $\mathbf{CAper}$ *is a cascade pseudovariety of finite tree automata.*

Recall that $\mathbf{D}$ denotes the class of all finite definite tree automata.

COROLLARY 3.2.10

$$\mathbf{D} \quad \subseteq \quad \mathbf{SAper}.$$

Proof. It was shown in [12] that $\mathbf{D}$ is the least cascade pseudovariety of finite tree automata containing $\mathbb{D}_0$. It is easy to check that $\mathbb{D}_0$ is strongly aperiodic. ■

3.3 Strict containments

By Propositions 3.1.1, 3.1.2 and Corollary 3.2.10,

$$\mathbf{CAper} \supseteq \mathbf{SAper}_1 \supseteq \mathbf{SAper}_2 \supseteq \ldots \supseteq \mathbf{SAper} \supseteq \mathbf{D} \tag{3.1}$$

is a decreasing chain. In this section we prove that when R contains an integer > 1, then each of the containments in (3.1) is strict. But first we treat the case when $R = \{1\}$ or $R = \{0, 1\}$. (Recall that in order to avoid trivial situations, we assume that the rank type contains at least one positive integer.)

PROPOSITION 3.3.1 *When $R = \{1\}$ or $R = \{0, 1\}$, it holds that*

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

Proof. So let $R = \{1\}$ or $R = \{0, 1\}$. The first equality is clear. We also know that $\mathbf{SAper} \supseteq \mathbf{D}$. If we can show that $\mathbf{SAper}_2 \subseteq \mathbf{D}$, then it follows that $\mathbf{SAper}_2 = \mathbf{SAper} = \mathbf{D}$. Moreover, $\mathbf{SAper}_1 \supset \mathbf{SAper}_2$, since there exist counter-free finite automata which are not definite. For instance, the tree automaton $\mathbb{E}^+_{\mathrm{EF}}$ (see page 39) is finite, counter-free and not definite for any rank type.

To prove that $\mathbf{SAper}_2 \subseteq \mathbf{D}$, suppose that $\mathbb{A}$ is *not* definite. When $R = \{1\}$, it is known that $\mathbb{A}$ has a proper term function having two or more different fixed points, cf. [9]. Thus, by Proposition 3.1.9, $\mathbb{A}$ is not 2-aperiodic. If $0 \in R$ the same reasoning applies, since if $\mathbb{A}$ is not definite, then the automaton $\mathbb{A}'$ obtained from $\mathbb{A}$ by removing the constants (so that the rank type of $\mathbb{A}'$ is $\{1\}$) is also not definite. ■

Thus, when R does not contain any integer > 1, then the hierarchy (3.1) collapses. In the remaining part of this section we will show that when R contains an integer > 1, then the hierarchy is proper.

PROPOSITION 3.3.2 *If R contains an integer $n > 1$ then $\mathbf{SAper}_1 \subset \mathbf{CAper}$.*

Proof. Let Σ contain the symbol σ of rank $n > 1$. Define the Σ-tree automaton $\mathbb{A}$ on the set $A = \{0, 1, 2\}$ of states as follows:

$$\sigma^{\mathbb{A}}(x_1, \ldots, x_n) \quad = \quad \begin{cases} 0 & \text{if } x_1 = \ldots = x_n = 1; \\ 1 & \text{if } x_1 = \ldots = x_n = 0; \\ 2 & \text{otherwise.} \end{cases}$$

Let the interpretation of any other symbol be a constant function. (We may add all constant functions if Σ is large enough.) This tree automaton is not 1-aperiodic, since the proper term function $A \to A$ induced by the tree $\sigma(x_1, \ldots, x_1)$ maps 0 to 1 and 1 to 0.

However, $\mathbb{A}$ is context aperiodic: for any proper term function $f : A \to A$ induced by a proper context it holds that f^2 is a constant function, so that $f^2 = f^3$. ■

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

Proof. Given n, we will first prove the claim assuming that the rank type contains an integer $m \geq n$.

We are going to construct a tree automaton $\mathbb{A}$ on the set

$$\{0, 1, \ldots, n, 1', \ldots, n'\}$$

having at least the operations $\sigma_i^{\mathbb{A}} : A^m \to A$, $i \in [n]$. For each $i \in [n]$, we define

$$\begin{aligned} \sigma_i^{\mathbb{A}}(1, \ldots, n, \ldots, n) &= i' \\ \sigma_i^{\mathbb{A}}(1', \ldots, n', \ldots, n') &= i. \end{aligned}$$

In all remaining cases, the operations $\sigma_i^{\mathbb{A}}$ return 0. For each integer $k \in R$, $k \neq m$, we take a single operation symbol in Σ_k and interpret it as the constant function $A^k \to A$ with value 0. This defines the tree automaton $\mathbb{A}$.

For each $i \in [n]$, let $s_i = \sigma_i(x_1, \ldots, x_n, \ldots, x_n) \in T_\Sigma(X_n)$. If $f = \langle s_1^{\mathbb{A}}, \ldots, s_n^{\mathbb{A}} \rangle$, then $f(1, \ldots, n) = (1', \ldots, n')$ and $f(1', \ldots, n') = (1, \ldots, n)$, showing that $\mathbb{A}$ is not n-aperiodic.

To prove that $\mathbb{A}$ is $(n-1)$-aperiodic, consider any proper tree $t \in T_\Sigma(X_{n-1})$. We show that $t^{\mathbb{A}} : A^{n-1} \to A$ is constant with value 0. This is clearly true when an operation symbol different from one of the σ_i appears in t. Suppose now that all operations symbols appearing in t are in the set $\{\sigma_1, \ldots, \sigma_n\}$. Then t has a subtree of the form $s = \sigma_i(x_{j_1}, \ldots, x_{j_m})$. Since the first n variables $x_{j_1}, \ldots, x_{j_n}$ cannot be all distinct, $s^{\mathbb{A}}$ is the constant function $A^{n-1} \to A$ with value 0. It follows now that $t^{\mathbb{A}}$ is also this function.

We show how to modify the above construction when each integer in R is less than n. Let m denote the maximal integer in R, so that $m > 1$. Let k denote the least integer with $1 + k(m-1) \geq n$, so that $k \geq 2$. We take symbols $\sigma_{i,1}, \ldots, \sigma_{i,k}$ of rank m, for all $i \in [n]$. Consider the trees in $T_\Sigma(X_n)$,

$$\begin{aligned} s_{i,1} &= \sigma_{i,1}(x_1, \ldots, x_m) \\ s_{i,2} &= \sigma_{i,2}(s_{i,1}, x_{m+1}, \ldots, x_{m+m-1}) \\ &\vdots \\ s_{i,k-1} &= \sigma_{i,k-1}(s_{i,k-2}, x_{m+(k-2)(m-1)+1}, \ldots, x_{m+(k-1)(m-1)}) \\ s_{i,k} &= \sigma_{i,k}(s_{i,k-1}, x_{m+(k-1)m+1}, \ldots, x_n, \ldots, x_n) \end{aligned}$$

Let $s_i = s_{i,k}$, $i \in [n]$. Now define the interpretation of the $\sigma_{i,j}$ so that

$$\begin{aligned} s_i^{\mathbb{A}}(1, \ldots, n) &= i' \\ s_i^{\mathbb{A}}(1', \ldots, n') &= i, \end{aligned}$$

for all i, moreover, for each i and $j < k$,

$$s_{i,j}^{\mathbb{A}}(1, \ldots, n) \quad \text{and} \quad s_{i,j}^{\mathbb{A}}(1', \ldots, n')$$

are *new* elements. The set A is the union of the set $\{0, 1, \ldots, n, 1', \ldots, n'\}$ with these new elements. In all other cases, the operations $\sigma_{i,j}$ return 0. As before, for all $p \neq m$, we take a single operation symbol in Σ_p whose interpretation is a constant function with value 0. We omit the formal verification of the correctness of this construction. ■

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

Proof. Let Σ be a signature of rank type R and $\sigma \in \Sigma_k$ a function symbol with $k > 1$. Define the Σ-tree automaton $\mathbb{A}$ as follows. Let $A = \{0, 1, 2\}$ and

$$\sigma^{\mathbb{A}}(a_1, \ldots, a_k) \quad = \quad \begin{cases} 1 & \text{if } 0 \in \{a_1, \ldots, a_k\} \subseteq \{0, 1\}; \\ 2 & \text{otherwise,} \end{cases}$$

for all $a_1, \ldots, a_k \in A$. All other symbols in Σ are interpreted as a constant function with value 2.

It is clear that for any proper tree $t \in T_\Sigma(X_n)$, $t^{\mathbb{A}}(a_1, \ldots, a_n) \in \{1, 2\}$ holds; moreover, if $a_1, \ldots, a_n \in \{1, 2\}$, then $t^{\mathbb{A}}(a_1, \ldots, a_n) = 2$. It follows that for any proper term function $f : A^n \to A^n$, each component of f^2 is a constant function with value 2.

We show that $\mathbb{A}$ is not definite. Consider the following sequences p_i, q_i of ΣA-polynomial symbols: $p_0 = \sigma(0, \ldots, 0)$, $p_{n+1} = \sigma(0, \ldots, 0, p_n)$ and $q_0 = \sigma(1, \ldots, 1)$, $q_{n+1} = \sigma(0, \ldots, 0, q_n)$. It is easy to check that $p_n^{\mathbb{A}} = 1$ and $q_n^{\mathbb{A}} = 2$ holds for any integer $n \geq 0$, hence—since the polynomial symbols p_n and q_n agree up to depth n—$\mathbb{A}$ is not definite. ■

3.4 Decidability and complexity

Since for any finite Σ-tree automaton $\mathbb{A}$ and for any $n \geq 1$, $S_n(\mathbb{A})$ is finite and effectively constructible, it is clear that there exists an algorithm to decide, given a finite Σ-tree automaton $\mathbb{A}$ and an integer n, whether $\mathbb{A}$ is n-aperiodic. Also, context aperiodicity is decidable. Strong aperiodicity of a finite tree automaton is not immediately decidable, since the definition of strong aperiodicity involves a condition for each n. (However, see Remark 3.1.12.)

In this section, we show that strong aperiodicity is decidable in polynomial time. On the other hand, it is known that deciding aperiodicity of classical automata (the case when $R = \{1\}$) is PSPACE-complete, cf. [4, 7]. We use this fact to show that for any fixed n, the membership problem of the class $\mathbf{SAper}_n$ is PSPACE-hard.

Let $\mathbb{A}$ be a finite Σ-tree automaton and consider the direct product $\mathbb{A} \times \mathbb{A}$ of $\mathbb{A}$ with itself. Let $B \subseteq A \times A$ a set of state pairs. We denote by $[B]$ the following set containing state

pairs:

$$\{t^{\mathbb{A}\times\mathbb{A}}\big((a_1,b_1),\ldots,(a_n,b_n)\big) : n \geq 0,\ t \in T_\Sigma(X_n) \text{ proper}, (a_i,b_i) \in B\}.$$

Note that for any $\mathbb{A}$ and $B \subseteq A \times A$, the set $[B]$ is computable in time polynomial in $|\mathbb{A}|$, the *size* of $\mathbb{A}$ defined as $n + \sum_{i\in R} s_i n^i$, where $n = |A|$ and $s_i = |\Sigma_i|$, for all $i \in R$. We make the following observation:

LEMMA 3.4.1 *The following are equivalent for any finite Σ-tree automaton $\mathbb{A}$:*

i) There exist an integer n and a proper term function $f : A^n \to A^n$ of $\mathbb{A}$ with at least two different fixed points.

ii) There exist an integer n and a proper term function $f : (A \times A)^n \to (A \times A)^n$ of the direct product $\mathbb{A} \times \mathbb{A}$ such that f has a fixed point of the form $\big((a_1,b_1),\ldots,(a_n,b_n)\big)$ with $a_i \neq b_i$ for some $i \in [n]$.

PROPOSITION 3.4.2 *The following are equivalent for any finite Σ-tree automaton $\mathbb{A}$:*

i) $\mathbb{A}$ is not strongly aperiodic;

ii) there exist an integer $n > 0$ and proper trees $t_1,\ldots,t_n \in T_\Sigma(X_n)$ such that $(t_1,\ldots,t_n)^{\mathbb{A}}$ has at least two different fixed points;

iii) there exist a set $S \subseteq A \times A$ and states $a \neq b \in A$ such that $(a,b) \in S \subseteq [S]$;

iv) the mapping $P(A \times A) \to P(A \times A)$ defined by $B \mapsto [B] \cap B$ has a fixed point S which contains a pair (a,b) with $a \neq b$;

v) the greatest fixed point of the mapping $B \mapsto [B] \cap B$ contains some pair (a,b) with $a \neq b$.

Proof. i) $\to$ ii). We have already proved this (Corollary 3.1.11).

ii) $\to$ iii). Let $n > 0$ be an integer, $t_1,\ldots,t_n \in T_\Sigma(X_n)$ proper trees and $(a_1,\ldots,a_n) \neq (b_1,\ldots,b_n)$ different state tuples such that both $(a_1,\ldots,a_n)$ and $(b_1,\ldots,b_n)$ are fixed points of $(t_1,\ldots,t_n)^{\mathbb{A}}$. Then $\big((a_1,b_1),\ldots,(a_n,b_n)\big)$ is a fixed point of $(t_1,\ldots,t_n)^{\mathbb{A}\times\mathbb{A}}$. Hence, choosing $S = \{(a_i,b_i) : i \in [n]\}$ we get that $S \subseteq [S]$, and since $(a_1,\ldots,a_n) \neq (b_1,\ldots,b_n)$, there exists a state pair $(a_i,b_i) \in S$ with $a_i \neq b_i$.

iii) $\to$ iv). This is a simple reformulation.

iv) → v). The mapping $B \mapsto [B] \cap B$ is monotone, so it has a greatest fixed point. Clearly, if some fixed point contains an element (a, b) with $a \neq b$, then the greatest fixed point also contains this element.

v) → i). Suppose $S = \{(a_1, b_1), \ldots, (a_n, b_n)\}$ is the greatest fixed point. Since S is a fixed point, $S = [S] \cap S$. Moreover, by assumption, $a_i \neq b_i$ holds for some $i \in [n]$. It follows (from the definition of $[S]$) that there exist proper trees $t_1, \ldots, t_n \in T_\Sigma(X_n)$ such that $\big((a_1, b_1), \ldots, (a_n, b_n)\big)$ is a fixed point of $(t_1, \ldots, t_n)^{\mathbb{A}\times\mathbb{A}}$. Now we can apply Lemma 3.4.1 and Corollary 3.1.11 and get the result. ■

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

Proof. The condition v) of Proposition 3.4.2 is clearly decidable in polynomial time: we have to compute $F^{n^2}(A \times A)$, where $F : P(A \times A) \to P(A \times A)$ is defined by $F(B) = [B] \cap B$ and n is $|A|$, and check whether the resulting set contains an ordered pair whose components are different.

Since F is computable by a polynomial time procedure, its iteration for n^2 steps runs in polynomial time. ■

It is clear that for any fixed n, testing whether a tree automaton is in $\mathbf{SAper}_n$ can be done in exponential time, since there are exponentially many proper term functions $A^n \to A^n$ (in terms of $|A|$), which can be even enumerated in exponential time; also checking whether $f^k = f^{k+1}$ holds for some $k \geq 1$ can also be done in exponential time. The following proposition establishes a PSPACE lower bound.

PROPOSITION 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.*

Proof. Consider an ordinary automaton, i.e., a Σ-tree automaton $\mathbb{A}$, where Σ is of rank type $\{1\}$. We construct a Δ-tree automaton $\mathbb{B}$, for some Δ, which is n-aperiodic if and only if $\mathbb{A}$ is aperiodic. We define $B = \{0\} \cup \{(a, j) : a \in A, j \in [n]\}$ so that B has $|A| \times n + 1$ states. Then let $\Delta = \Delta_n = \{(\sigma, i) : \sigma \in \Sigma,\ i \in [n]\}$. The operations $(\sigma, i)^{\mathbb{A}}$ are defined as follows. Suppose that $\sigma^{\mathbb{A}}(a) = b$, for some $a, b \in A$ and $\sigma \in \Sigma$. Then we let $(\sigma, i)^{\mathbb{A}}((a, 1), \ldots, (a, n)) = (b, i)$. In all remaining cases, the operation returns 0. For each ΣX_n-tree t, define $\underline{t} = (t_1, \ldots, t_n) \in T_\Delta(X_n)^n$ as follows. If $t = x_1$ then $t_i = x_i$, for each i. If $t = \sigma(s)$, then $t_i = (\sigma, i)(\underline{s})$, for each i. Now if $t^{\mathbb{A}}(a) = b$, then $\underline{t}^{\mathbb{B}}(\underline{a}) = \underline{b}$, where $\underline{a} = ((a, 1), \ldots, (a, n))$ and $\underline{b}$ is defined in the same way. It follows that if $\mathbb{A}$ is not aperiodic, then $\mathbb{B}$ is not n-aperiodic.

We still have to show that if $\mathbb{B}$ is not n-aperiodic then $\mathbb{A}$ is not aperiodic either. For this reason, we define the paths in a tree $t \in T_\Delta(X_n)$ as follows. If $t = x_i$ then $\mathrm{paths}(t) = \{x_1\}$. If $t = (\sigma, i)(t_1, \ldots, t_n)$ then $\mathrm{paths}(t) = \{\sigma(s) : s \in \bigcup_{j=1}^n \mathrm{paths}(t_j)\}$. Note that $\mathrm{paths}(t) \subseteq T_\Sigma(X_1)$.

Now assume that $\mathbb{B}$ is not n-aperiodic, so that there exist some $\underline{q_1}, \ldots, \underline{q_k} \in B^n$, $k > 1$, a vector $\underline{t} = (t_1, \ldots, t_n) \in T_\Delta(X_n)^n$ of proper trees such that $\underline{t}^{\mathbb{B}}(\underline{q_i}) = \underline{q_{i+1}}$, for all $i \in [k-1]$ and $\underline{t}^{\mathbb{B}}(\underline{q_k}) = \underline{q_1}$. Now each t_i contains a subtree of the form $(\sigma, j)(x_{k_1}, \ldots, x_{k_n})$. However, the variables $x_{k_1}, \ldots, x_{k_n}$ must be all distinct, since otherwise $t_i^{\mathbb{B}}$ would be constant with value 0. Also, the components of each $\underline{q_i}$ must also be all distinct, moreover, the components of each $\underline{q_j}$ must give a permutation of the states $(a, 1), \ldots, (a, n)$ for some $a \in A$, since otherwise we would have $t_i^{\mathbb{B}}(\underline{q_j}) = 0$. Also, if $(\sigma, j)(s_1, \ldots, s_n)$ is any subtree of some component of $\underline{t}$, then for any $\underline{q_i}$ there must exist a state $a \in A$ such that $s_1^{\mathbb{B}}(\underline{q_i}), \ldots, s_n^{\mathbb{B}}(\underline{q_i})$ is the sequence $(a, 1), \ldots, (a, n)$. It then follows that there is a permutation π such that for each $\underline{q_i}$ there is some state $a_i \in A$ with $\underline{q_i} = ((a_i, \pi(1)), \ldots, (a_i, \pi(n))$. Of course, the states a_i are all distinct. Thus, if we take any tree $s \in \mathrm{paths}(t_j)$, for any j, then we have $s^{\mathbb{A}}(a_i) = a_{i+1}$ for all $i \in [k-1]$ and $s^{\mathbb{A}}(a_k) = a_1$. This shows that $\mathbb{A}$ is not aperiodic. ■

3.5 Aperiodicity and logic

In this section we relate the above aperiodicity notions to formal logic. We establish the following facts for tree languages over signatures of rank type $R = \{0, 2\}$.

1. If a tree language is in **CTL**, then its minimal tree automaton is in $\mathbf{SAper}_1$.
2. There exists a tree language in **CTL** whose minimal automaton is not in $\mathbf{SAper}_2$.
3. There exists a tree language definable in FO($<$) which is not 1-aperiodic.
4. There exists a 1-aperiodic tree language which is not definable in FO($<, S_i$).

The above results can all be extended to rank types containing 0 and at least one integer > 1.

In the rest of this section we fix $R = \{0, 2\}$.

PROPOSITION 3.5.1 *If L is a tree language in* **CTL**, *then the minimal automaton of L is in* $\mathbf{SAper}_1$.

Proof. It was shown in [14, 15] that a tree language is in **CTL** if and only if its minimal tree automaton is in the least cascade pseudovariety of finite tree automata containing the tree automaton over the two-element set $\{0, 1\}$ equipped with the binary `or` function and the two constants $0, 1$. Our claim thus follows from the facts that this tree automaton is 1-aperiodic and that 1-aperiodic finite tree automata form a generalized cascade pseudovariety of finite tree automata, cf. Theorem 3.2.8. ■

PROPOSITION 3.5.2 *There exists a tree language in* $\mathbf{CTL}(\mathrm{EF}^*)$ *whose minimal tree automaton is not in* $\mathbf{SAper}_2$.

Proof. Recall the tree automaton $\mathbb{E}^*_{\mathrm{EF}}$. In this automaton, the identity function is a proper term function (induced by the proper tree $\downarrow_2 (x_1, x_1)$) having two fixed points. Thus $\mathbb{E}^*_{\mathrm{EF}} \notin \mathbf{SAper}_2$. ■

PROPOSITION 3.5.3 *There exists a (regular) tree language in* $\mathbf{FO}(<)$ *whose minimal tree automaton is not* 1*-aperiodic.*

Proof. Let us define the signature Σ with $\Sigma_2 = \{\sigma, \nu_1, \nu_2\}$ and $\Sigma_0 = \{c\}$. We construct a regular tree language $L \subseteq T_\Sigma$ in $\mathbf{FO}(<)$ whose minimal automaton is not 1-aperiodic. Let L be the following language:

$$L = \{t \in T_\Sigma : \mathrm{AllPaths}(t) \cap (\sigma\nu_1\sigma\nu_2)^* c \neq \emptyset\}.$$

Here, $\mathrm{AllPaths}(t)$ is the set of all words which are label sequences of maximal paths in t, i.e.

$$\mathrm{AllPaths}(t) \quad = \quad \begin{cases} t & \text{if } t \in \Sigma_0; \\ \bigcup\limits_{i=1,2} \{\sigma' \cdot v : v \in \mathrm{AllPaths}(t_i)\} & \text{if } t = \sigma'(t_1, t_2). \end{cases}$$

Now L is definable in $\mathrm{FO}(<)$: a tree $t \in T_\Sigma$ belongs to L if and only if there exists a vertex v such that the following hold:

1. the label of v is c (hence v is a leaf);
2. if y is a successor of x and $y < v$ then x is labeled σ if and only if y is labeled in $\{\nu_1, \nu_2\}$, and x is labeled in $\{\nu_1, \nu_2\}$ if and only if y is labeled σ;
3. if y is a second successor ("grandchild") of x and $y < v$ then x is labeled ν_1 if and only if y is labeled ν_2, and x is labeled ν_2 if and only if y is labeled ν_1;

4. the root is labeled in $\{\sigma, c\}$;

5. if v is a successor of x then x is labeled ν_2.

It is clear that all these conditions are expressible in FO($<$). Now consider the following infinite sequence of trees t_i: $t_0 = c$, and for all $i \geq 0$, $t_{i+1} = \sigma(\nu_1(t_i, t_i), \nu_2(t_i, t_i))$. It is clear that $t_i \in L$ if and only if i is even. Thus, the minimal automaton of L is not 1-aperiodic, since for the term function $f : A \to A$ induced by the proper tree $\sigma(\nu_1(x_1, x_1), \nu_2(x_1, x_1))$ we have $f^i(c^{\mathbb{A}}) = c^{\mathbb{A}}$ if and only if i is even. ■

PROPOSITION 3.5.4 *There exists a regular tree language not contained in* $\mathbf{FO}(<, S_1, S_2)$ *whose minimal automaton is* 1*-aperiodic.*

Proof. Let $\Sigma_0 = \{\mathbf{0}, \mathbf{1}\}$, $\Sigma_2 = \{\wedge, \vee\}$, and let us define the following tree automaton $\mathbb{A}$ on $A = \{0, 1, \perp\}$ that was introduced in [40]: $\mathbf{0}^{\mathbb{A}} = 0$, $\mathbf{1}^{\mathbb{A}} = 1$ and

$$\wedge^{\mathbb{A}}(x, y) = \begin{cases} 1 & \text{if } x = y = 1; \\ 0 & \text{if } \{x, y\} = \{0, 1\}; \\ \perp & \text{otherwise,} \end{cases} \qquad \vee^{\mathbb{A}}(x, y) = \begin{cases} 0 & \text{if } x = y = 0; \\ 1 & \text{if } \{x, y\} = \{0, 1\}; \\ \perp & \text{otherwise.} \end{cases}$$

Let $L = \{t \in T_\Sigma : t^{\mathbb{A}} = 1\}$. It has been shown in [40] that L is not contained in $\mathbf{FO}(<, S_1, S_2)$. It is easy to check that $\mathbb{A}$ is 1-aperiodic. We note that $\mathbb{A}$ is not 2-aperiodic, since both 1 and $\perp$ are fixed points of the term function $A \to A$ induced by $\wedge(x_1, x_1)$. ■

3.6 A generalization

In this section, we provide a generalization of the aperiodicity notions studied above. The following definition is motivated by the Krohn-Rhodes decomposition theorem [9].

DEFINITION 3.6.1 *Suppose that* $\mathcal{G}$ *is a class of finite simple groups closed under division. We say that a finite tree automaton* $\mathbb{A}$ *belongs to the class* $\mathbf{Aut}(\mathcal{G}, n)$ *for some positive integer* n *exactly when each simple group dividing* $S_n(\mathbb{A})$ *is in* $\mathcal{G}$*. The intersection of all classes* $\mathbf{Aut}(\mathcal{G}, n)$ *is* $\mathbf{Aut}(\mathcal{G})$*. Moreover, we let* $\mathbf{Aut}_c(\mathcal{G})$ *denote the class of all finite tree automata* $\mathbb{A}$ *such that every simple group divisor of* $C(\mathbb{A})$ *is in* $\mathcal{G}$*.*

When $\mathcal{G}$ contains only the trivial groups, then the above classes are just $\mathbf{SAper}_n$, $\mathbf{SAper}$ and $\mathbf{CAper}$, respectively. The following generalization of Theorems 3.2.8 and 3.2.9 holds.

THEOREM 3.6.2 *For each n, $\mathbf{Aut}(\mathcal{G}, n)$ is a generalized cascade pseudovariety of finite tree automata, as is $\mathbf{Aut}(\mathcal{G})$. The class $\mathbf{Aut}_c(\mathcal{G})$ is a cascade pseudovariety of finite tree automata.*

We omit the proof which is based on the following facts. For the definition of the wreath product of transformation semigroups we refer to [9].

LEMMA 3.6.3 *Suppose that $\mathbb{A}$ and $\mathbb{B}$ are Σ-tree automata such that $\mathbb{A}$ is a subautomaton or a homomorphic image of $\mathbb{B}$. Then for each n, $(A^n, S_n(\mathbb{A}))$ divides $(B^n, S_n(\mathbb{B}))$. Similarly, $(A, C(\mathbb{A}))$ divides $(B, C(\mathbb{B}))$.*

LEMMA 3.6.4 *Suppose that $\mathbb{A}$ is a Σ-tree automaton, $\mathbb{B}$ is a Δ-tree automaton and consider a generalized cascade product $\mathbb{A} \times_\alpha \mathbb{B}$. Then for each n, $((A \times B)^n, S_n(\mathbb{A} \times_\alpha \mathbb{B}))$ divides a wreath product of $(A^n, S_n(\mathbb{A}))$ and $(B^n, S_n(\mathbb{B}))$.*

3.7 Aperiodicity for polynomials

In this section we study a variant of the aperiodicity notions resulting by considering polynomial functions instead of term functions. We will show that in this case the hierarchy collapses at $n = 2$.

Let $\mathbb{A}$ be a Σ-tree automaton of rank type R over the set A. For any $n \geq 0$ and $m > 0$, we let each m-tuple $\underline{p} = (p_1, \ldots, p_m)$ of ΣAX_n-polynomial symbols induce 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 m functions $p_i^{\mathbb{A}} : A^n \to A$, $i \in [m]$. When each p_i is proper, $\underline{p}^{\mathbb{A}}$ is also called proper. For each $n \geq 1$, let $S_n^{(p)}(\mathbb{A})$ denote the semigroup of proper polynomial functions $A^n \to A^n$ and let $C^{(p)}(\mathbb{A})$ denote the semigroup of proper translations of $\mathbb{A}$.

For each n, we let $\mathbf{SAper}_n^{(p)}$ denote the class of all finite tree automata $\mathbb{A}$ such that $S_n^{(p)}(\mathbb{A})$ is an aperiodic semigroup. We define the classes $\mathbf{SAper}^{(p)}$ and $\mathbf{CAper}^{(p)}$ in the same way.

The following was proved in [12], Corollary 3.12.

LEMMA 3.7.1 *The following are equivalent for any finite Σ-tree automaton $\mathbb{A}$:*

i) $\mathbb{A}$ is definite;

ii) any proper translation of $\mathbb{A}$ has at most one fixed point.

THEOREM 3.7.2 *The following are equivalent for any finite Σ-tree automaton $\mathbb{A}$:*

i) $\mathbb{A}$ *is definite;*

ii) $\mathbb{A} \in \mathbf{SAper}^{(p)}$*;*

iii) $\mathbb{A} \in \mathbf{SAper}_2^{(p)}$*;*

iv) Each proper translation of $\mathbb{A}$ *has at most one fixed point.*

Proof. The first and last conditions are equivalent by Lemma 3.7.1, and the first condition implies the second by Corollary 3.2.10. It is clear that the second condition implies the third. Finally, the third implies the fourth by Proposition 3.1.9. ■

COROLLARY 3.7.3 *It is decidable in polynomial time whether a finite tree automaton is in* $\mathbf{SAper}^{(p)}$.

Proof. A polynomial time algorithm is implicit in [12]. ■

PROPOSITION 3.7.4

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

Proof. The inclusions are clear. The second inclusion is strict by the proof of Proposition 3.3.2. The first is strict since the tree automaton on $\{0, 1\}$ with the binary `or` function as the single operation is in $\mathbf{SAper}_1^{(p)} \setminus \mathbf{D}$. ■

PROPOSITION 3.7.5 *Suppose the rank type R contains an integer > 1. Then for each n there exist finite tree automata $\mathbb{A}$ and $\mathbb{B}$ such that both $\mathbb{A}$ and $\mathbb{B}$ are $(n-1)$-aperiodic, neither of them is n-aperiodic, moreover, $\mathbb{A}$ is contained in $\mathbf{SAper}_1^{(p)}$ and $\mathbb{B}$ is not.*

Proof. First we deal with the case when R contains an integer $k \geq n$.

Consider the tree automaton $\mathbb{A}$ constructed in the proof of Proposition 3.3.3. We have proved that $\mathbb{A}$ is $(n-1)$-aperiodic and not n-aperiodic. It is easy to see that for every non-constant term function $f : A \to A$ of $\mathbb{A}^{(p)}$ there exists *at most one* state a with $f(a) \neq 0$, moreover, this state a is different from 0. From this we can conclude that $f^2 = f^3$ holds for any such function. Thus, $\mathbb{A}$ is contained in $\mathbf{SAper}_1^{(p)}$.

Now we construct the tree automaton $\mathbb{B}$ by modifying the construction of $\mathbb{A}$ as follows. The state set of $\mathbb{B}$ is $B = \{0, \ldots, n\} \cup \{1'\}$. For each $i \in [n]$, we define $\sigma_i^{\mathbb{B}}$ as follows:

$$\sigma_i^{\mathbb{B}}(1, \ldots, n, \ldots, n) = \sigma_i^{\mathbb{B}}(1', 2, \ldots, n, \ldots, n) = i \text{ if } i > 1;$$

$$\sigma_1^{\mathbb{B}}(1, \ldots, n, \ldots, n) = 1' \text{ and } \sigma_1^{\mathbb{B}}(1', 2, \ldots, n, \ldots, n) = 1.$$

In all remaining cases the functions $\sigma_i^{\mathbb{B}}$ return 0, and all other function symbols are interpreted as a constant function with value 0.

Now by the same argument as in the proof or Proposition 3.3.3 we get that $\mathbb{B}$ is $(n-1)$-aperiodic but not n-aperiodic. Also, since for the ΣBX_1-polynomial symbol $p = \sigma_1(x_1, 2, \ldots, n)$ we have $p^{\mathbb{B}}(1) = 1'$ and $p^{\mathbb{B}}(1') = 1$, $\mathbb{B}$ is not contained in $\mathbf{SAper}_1^{(p)}$.

If R contains only integers less than n, then we can modify $\mathbb{B}$ as in the proof of Proposition 3.3.3. ■

PROPOSITION 3.7.6 *Suppose R contains some integer $k > 1$. Then there exists an automaton $\mathbb{A}$ which is strongly aperiodic, and is not contained in $\mathbf{SAper}_1^{(p)}$.*

Proof. We define the following Σ-tree automaton $\mathbb{A}$, where $\Sigma_i = \{\sigma_i\}$ for each $i \in R$ and $A = \{0, 1, 2\}$. For each $i \in R$ with $i \neq k$, let $\sigma_i^{\mathbb{A}}$ be the constant function with value 2 and let us define $\sigma_k^{\mathbb{A}}$ as follows:

$$\sigma_k^{\mathbb{A}}(a_1, \ldots, a_k) = \begin{cases} 1 & \text{if } \{a_1, \ldots, a_k\} = \{0, 2\}; \\ 2 & \text{otherwise.} \end{cases}$$

Now it is easy to see that for any $n \geq 1$ and tree $t \in T_\Sigma(X_n)$ it holds that $t^{\mathbb{A}}(a_1, \ldots, a_n) \in \{1, 2\}$ for any $a_1, \ldots, a_n \in A$. Moreover, if each a_i is 1 or 2, then $t^{\mathbb{A}}(a_1, \ldots, a_n) = 2$. It follows that for any tuple of trees $\underline{t} = (t_1, \ldots, t_n) \in \big(T_\Sigma(X_n)\big)^n$, $(\underline{t}^{\mathbb{A}})^2$ is a constant function, hence $\mathbb{A}$ is strongly aperiodic.

However, since the function induced by $\sigma_k(0, \ldots, 0, x_1)$ maps 1 to 2 and 2 to 1, $\mathbb{A}$ is not contained in $\mathbf{SAper}_1^{(p)}$. ■

COROLLARY 3.7.7 *As shown in Figure 3.1, the class $\mathbf{SAper}_1^{(p)}$ nontrivially intersects $\mathbf{SAper}$ and each member of the hierarchy $\mathbf{SAper}_n$ for $n \geq 2$.*

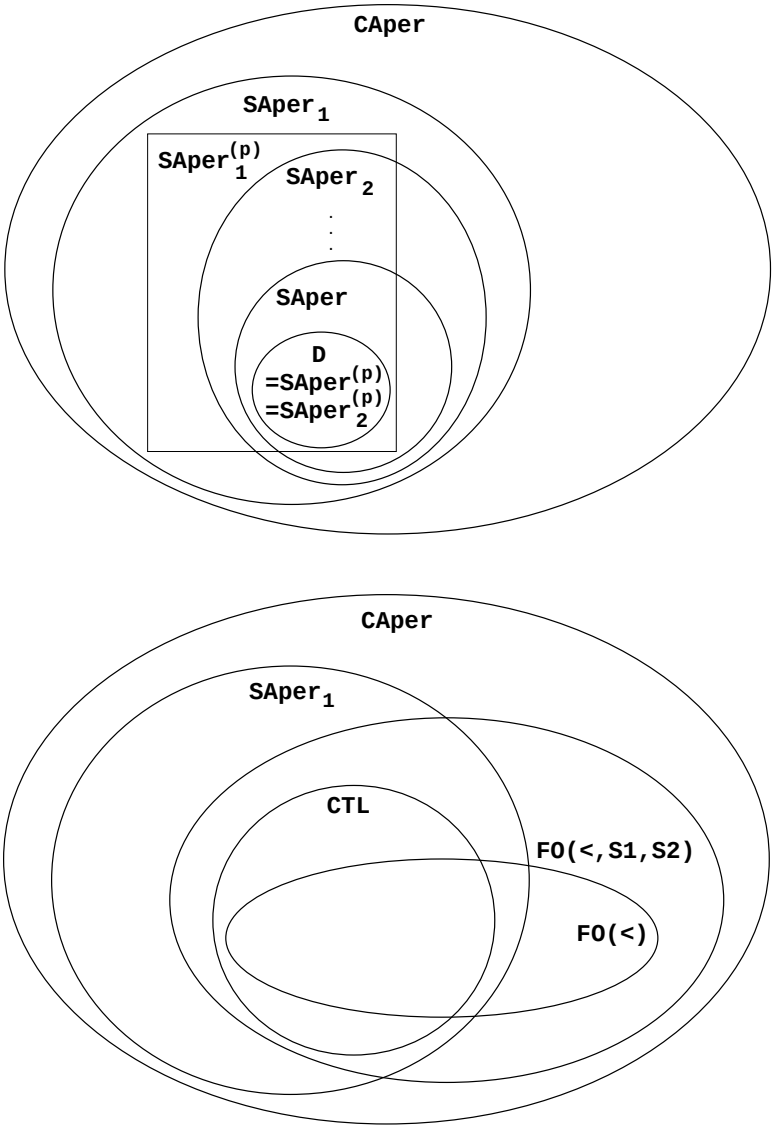

Figure 3.1: The aperiodicity hierarchies.

Summary

This thesis is concerned with the definability problem of several classes of logics on finite trees. Two main areas of this topic can be identified here. The first area is the algebraic characterization of the logic FTL($\mathcal{L}$), when $\mathcal{L}$ is a class of regular tree languages and all the quotients $\zeta^{-1}L$ for $\zeta \in CT_\Sigma$, $L \subseteq T_\Sigma$, $L \in \mathcal{L}$ are expressible in FTL($\mathcal{L}$). The second area addresses different notions of aperiodicity in relation to logics on trees.

Chapter 1 is a preliminary one. It contains a brief summary of earlier results related to the thesis, basic notions and well-known theorems from the fields of formal language theory, tree automata and logic.

Chapter 2 is devoted to the family of the temporal logics FTL($\mathcal{L}$), and provides an algebraic characterization of its definability problem. To each class $\mathcal{L}$ of tree languages, a branching time future temporal logic FTL($\mathcal{L}$) is associated, according to [14, 15]. The class $\mathbf{FTL}(\mathcal{L})$ of tree languages is defined as the class of FTL($\mathcal{L}$)-definable tree languages. In [15], the following has been proved:

Suppose $\mathcal{L}$ is a class of regular tree languages such that quotients and the next modalities are expressible in FTL($\mathcal{L}$). *Then a tree language L is contained in* $\mathbf{FTL}(\mathcal{L})$ *if and only if L is regular, and its minimal automaton $\mathbb{A}_L$ is contained in the least pseudovariety of finite tree automata containing the minimal automata of the members of $\mathcal{L}$ and a specific definite tree automaton $\mathbb{D}_0$, which is closed under the cascade product.*

Here, the condition "quotients are expressible in FTL($\mathcal{L}$)" holds if whenever $L \in \mathcal{L}$ is a tree language over Σ, and $\zeta \in CT_\Sigma$ is a Σ-context, then the language $\zeta^{-1}L$ is contained in $\mathbf{FTL}(\mathcal{L})$.

Also, the condition "the next modalities are expressible in FTL($\mathcal{L}$)" is equivalent to stating that for any FTL($\mathcal{L}$)-formula φ over a signature Σ of rank type R and integer $i \in [\max R]$ there exists an FTL($\mathcal{L}$)-formula $X_i\varphi$ with $t \models X_i\varphi$ for a tree $t \in T_\Sigma$ if and only if $t = \sigma(t_1, \ldots, t_n)$ for some $n \geq i$, $\sigma \in \Sigma_n$ and $t_1, \ldots, t_n$ such that $t_i \models \varphi$.

Theorem 2.3.4 and Corollary 2.3.5 strengthen this result. In Section 2.2 we identified a restricted form of the cascade product, called the Moore product, and related it to the

cascade product. Also an even more restrictive variant, the strict Moore product has been introduced. Using the Moore product, the following stronger result was proved (published in [18]):

Suppose $\mathcal{L}$ is a class of regular tree languages such that quotients are expressible in $\mathrm{FTL}(\mathcal{L})$. *Then a tree language L is contained in* $\mathbf{FTL}(\mathcal{L})$ *if and only if L is regular, and its minimal automaton $\mathbb{A}_L$ is contained in the least pseudovariety of finite tree automata containing the minimal automata of the members of $\mathcal{L}$ and a specific definite tree automaton $\mathbb{D}_0$, which is closed under the Moore product.*

This characterization establishes a correspondence between definability in the logic $\mathrm{FTL}(\mathcal{L})$ and membership in a pseudovariety of finite tree automata (actually, the correspondence is an order isomorphism between the lattice of literal varieties of tree languages closed under the operator **FTL** and the lattice of connected Moore pseudovarieties of finite connected tree automata containing $\mathbb{D}_0$). However, the characterization is not effective. The characterization of [15] was applicable to the CTL fragments CTL(EX) and CTL(EF + EX), but not to the fragments $\mathrm{CTL}(\mathrm{EF}^*)$ and $\mathrm{CTL}(\mathrm{EF}^+)$, the fragment in which only the non-strict, resp. strict variant of the EF modality ("there exists a subtree") is allowed, since in these logics, the next modalities are not expressible.

In Section 2.3 we gave decidable characterizations of the fragments $\mathrm{CTL}(\mathrm{EF}^*)$ and $\mathrm{CTL}(\mathrm{EF}^+)$: Corollary 2.4.42, published in [19], states the following:

A tree language $L \subseteq T_\Sigma$ is definable in the temporal logic $\mathrm{CTL}(\mathrm{EF}^+)$ *if and only if it is regular and its minimal tree automaton $\mathbb{A}_L$ is commutative, maximal dependent and component dependent, i.e. satisfies the equations*

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

for any $\sigma \in \Sigma_n$ and permutation π, the implications

$$y \preceq x_i, z \preceq x_j \quad \Rightarrow \quad \sigma(x_1, \ldots, x_{n-1}, y) = \sigma(x_1, \ldots, x_{n-1}, z)$$

for each $\sigma \in \Sigma_n$, where $\preceq$ is the accessibility relation of $\mathbb{A}_L$, and the implications

$$x_1 \sim y_1, \ldots, x_n \sim y_n \quad \Rightarrow \quad \sigma(x_1, \ldots, x_n) = \sigma(y_1, \ldots, y_n)$$

for any $\sigma \in \Sigma_n$, where $x \sim y$ holds if and only if $x \preceq y$ and $y \preceq x$.

A tree language $L \subseteq T_\Sigma$ is definable in $\mathrm{CTL}(\mathrm{EF}^*)$ *if and only if it is definable in* $\mathrm{CTL}(\mathrm{EF}^+)$ *and its minimal automaton is stutter invariant, i.e. satisfies the equations*

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

for each $\sigma \in \Sigma_n$.

This characterization also shows that the following questions are both decidable in polynomial time: given a tree language L by its minimal tree automaton $\mathbb{A}_L$, is L definable in CTL(EF$^+$) or in CTL(EF*)? This was already known for the fragment CTL(EF$^+$), see [5], whereas the characterization of CTL(EF*) solves a problem mentioned open in [5] (which has also been solved independently in [58]).

In Section 2.5, we provided another characterization in terms of two-player games. We defined for each class $\mathcal{L}$ of (not necessarily regular) tree languages and integer $n \geq 0$ the n-round $\mathcal{L}$-game that is played on a pair of trees between two competing players, Spoiler and Duplicator. Corollary 2.5.3 relates FTL($\mathcal{L}$)-definability and the existence of a winning strategy as follows:

For any finite class $\mathcal{L}$ of tree languages, a language L is contained in $\mathbf{FTL}(\mathcal{L})$ if and only if there exists a number $n \geq 0$ such that Spoiler wins the n-round $\mathcal{L}$-game on L, that is, whenever $s \in L$ and $t \notin L$ are two trees, Spoiler has a winning strategy in the n-round $\mathcal{L}$-game played on (s, t).

Chapter 3 is devoted to several aperiodicity notions. When Σ is a signature and $\mathbb{A} = (A, \Sigma)$ is a finite Σ-tree automaton, any n-tuple $(t_1, \ldots, t_n)$ of proper ΣX_n-trees induces a mapping $(t_1, \ldots, t_n)^{\mathbb{A}} : A^n \to A^n$. For any $n > 0$, these functions form a semigroup $S_n(\mathbb{A})$, with the composition operation as product. We say that the finite tree automaton $\mathbb{A}$ is n-aperiodic if the semigroup $S_n(\mathbb{A})$ is aperiodic, i.e. for some K the equation $s^K = s^{K+1}$ holds for any element s of $S_n(\mathbb{A})$. Moreover, $\mathbb{A}$ is said to be strongly aperiodic if it is n-aperiodic for each n. Following [54], we say that $\mathbb{A}$ is context aperiodic if the semigroup $C(\mathbb{A})$ of term functions induced by proper Σ-contexts is aperiodic. We denote the class of all n-aperiodic finite tree automata by $\mathbf{SAper}_n$, the class of all strongly aperiodic finite tree automata by **SAper** and the class of all context aperiodic finite tree automata by **CAper**. Each of these classes but **CAper** forms a generalized cascade pseudovariety, whereas **CAper** forms a cascade pseudovariety.

The inclusions

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

hold for any rank type R. In Proposition 3.3.1 we showed that when $R = \{0, 1\}$ or $R = \{1\}$, that is, in the case of words, the hierarchy collapses:

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

However, in any non-classical case (when the rank type contains an integer greater than 1) the hierarchy is strict:

$$\mathbf{CAper} \supset \mathbf{SAper}_1 \supset \mathbf{SAper}_2 \supset \cdots \supset \mathbf{SAper} \supset \mathbf{D},$$

see Propositions 3.3.2, 3.3.3 and 3.3.4.

The complexity of the membership problems was also studied: we showed that when R is a non-classical rank type, then for each $n > 0$ it is PSPACE-hard to decide whether a tree automaton is n-aperiodic (Proposition 3.4.4). In contrast, we gave a polynomial time decision procedure for testing strong aperiodicity (Theorem 3.4.3).

We related the above aperiodicity notions to logic. In Section 3.5 we showed that the following hold for binary trees (when $R = \{0, 2\}$):

1. If a tree language is in **CTL**, then its minimal tree automaton is in $\mathbf{SAper}_1$.
2. There exists a tree language in **CTL** whose minimal automaton is not in $\mathbf{SAper}_2$.
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($<, S_i$).

The arguments can be extended to any non-classical rank type.

We also introduced a slight modification of the notion of n-aperiodicity, involving the semigroup of proper polynomial functions instead of proper term functions. It turned out (Proposition 3.7.4) that in this case the hierarchy also collapses:

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

As a last result, we showed that $\mathbf{SAper}_1^{(p)}$ is incomparable with $\mathbf{SAper}_n$, for each $n > 1$, as well as with **SAper** (Corollary 3.7.7).

The results of Chapter 3 were published in [17].

Összefoglalás

Jelen dolgozat bizonyos, véges fákon értelmezett logikák definiálhatósági problémájával foglalkozik. Ezen a témán belül két nagyobb érintett terület különül el: az első az FTL($\mathcal{L}$) alakú logikák algebrai karakterizációja, ahol $\mathcal{L}$ reguláris fanyelvek osztálya és ahol minden $\zeta^{-1}L$ alakú hányados (ζ itt Σ-context, L pedig $\mathcal{L}$-beli Σ-fanyelv) kifejezhető FTL($\mathcal{L}$)-ben. Ez a tulajdonság a továbbiakban a következőképp szerepel: ,,a hányadosok kifejezhetők FTL($\mathcal{L}$)-ben". A második részben az aperiodicitás különböző változatait vizsgáltuk.

Az első fejezet bevezetés: itt definiáltuk a dolgozatban használt jelöléseket a formális nyelvek, faautomaták és logika területeiről, egyszersmind a felhasznált korábbi eredményeket is itt ismertettük.

A második fejezetben vezettük be az FTL elágazó, jövő idejű temporális logikát, illetve ennek FTL($\mathcal{L}$) alakú töredékeit, ahol $\mathcal{L}$ fanyelvek egy osztálya. Az FTL($\mathcal{L}$) logikában definiálható nyelvek osztályát a [14, 15] munkákat követve $\mathbf{FTL}(\mathcal{L})$ jelöli. A fejezetben továbbá algebrai karakterizációt adtunk az $\mathbf{FTL}(\mathcal{L})$ nyelvosztályra, ha $\mathcal{L}$ reguláris fanyelvek egy olyan osztálya, melyre a hányadosok kifejezhetők FTL($\mathcal{L}$)-ben. [15]-ben az alábbi karakterizációs tétel került igazolásra:

Legyen $\mathcal{L}$ reguláris nyelvek egy olyan osztálya, melyre a hányadosok és a rákövetkezési modalitások kifejezhetők FTL($\mathcal{L}$)*-ben. Akkor egy L fanyelv pontosan akkor esik az* $\mathbf{FTL}(\mathcal{L})$ *osztályba, ha reguláris és minimális automatája benne van a véges faautomatáknak abban a legszűkebb pszeudovarietásában, mely tartalmazza $\mathcal{L}$ elemeinek minimális automatáit, egy bizonyos $\mathbb{D}_0$ 1-definit faautomatát és amely zárt faautomaták kaszkád szorzataira.*

Itt ,,a rákövetkezési modalitások kifejezhetők FTL($\mathcal{L}$)-ben" azt jelenti, hogy tetszőleges φ, az R rangtípusú Σ ábécé fölötti FTL($\mathcal{L}$)-formulára és $i \in [\max R]$ egészre létezik egy Σ fölötti $X_i\varphi$ formula FTL($\mathcal{L}$)-ben, melyre $t \models X_i\varphi$ pontosan akkor teljesül egy $t \in T_\Sigma$ fára, ha t-nek létezik az i-edik közvetlen részfája, és az kielégíti a φ formulát.

A 2.3.4. tétel és a 2.3.5. következmény ezt az eredményt erősítik. Azonosítottuk faautomaták kaszkád szorzatának egy speciális esetét, amit Moore szorzatnak neveztünk (és melyet a 2.2-es szakaszban definiáltunk). Ezt a szorzatot és egy további speciális esetét, a szigorú Moore szorzatot összehasonlítottuk a kaszkád szorzattal. A Moore szorzatot

felhasználva az alábbi, [18]-ban publikált összefüggést igazoltuk:

Legyen $\mathcal{L}$ reguláris fanyelvek egy olyan osztálya, melyre a hányadosok kifejezhetők FTL($\mathcal{L}$)*-ben. Akkor egy L fanyelv pontosan akkor esik* **FTL**($\mathcal{L}$)*-be, ha reguláris és minimális automatája beletartozik a véges faautomaták azon legszűkebb pszeudovarietásába, mely tartalmazza $\mathcal{L}$ elemeinek minimális automatáit, a $\mathbb{D}_0$ automatát és amely zárt faautomaták Moore szorzataira.*

Ez a karakterizáció az FTL($\mathcal{L}$) logikák definiálhatósági kérdése és faautomaták pszeudovarietásainak eldöntési kérdése között létesít kapcsolatot. Valójában ez a kapcsolat egy hálóizomorfizmus, az **FTL** operátorra zárt literális fanyelv-varietások és véges, összefüggő faautomatáknak a $\mathbb{D}_0$-t tartalmazó és a Moore szorzatra zárt pszeudovarietásai által alkotott teljes hálók között. Ugyanakkor, a karakterizáció nem effektív. A [15]-ben megadott jellemzés alkalmazható volt annak igazolására, hogy a CTL logika két szintaktikus töredéke, CTL(EX) és CTL(EX + EF) definiálhatósági problémája eldönthető. Azonban ha tekintjük az CTL(EF*) és CTL(EF$^+$) töredékeket, melyekben egyetlen modalitás, az EF („existence in the future") megengedő ill. szigorú változata szerepel, ezekre nem alkalmazható a [15]-beli jellemzési tétel, mivel azokban a rákövetkezési modalitások nem kifejezhetők. A 2.3-as szakaszban a Moore szorzat segítségével megadott jellemzést alkalmazva igazoltuk, hogy ennek a két logikának is eldönthető a definiálhatósági problémája. A kapott eredmény (2.4.42. következmény), mely [19]-ban került publikálásra, így szól:

Az $L \subseteq T_\Sigma$ fanyelv pontosan akkor definiálható a CTL(EF$^+$) *logikában, ha reguláris és minimális faautomatája kommutatív, maximális elemfüggő és komponensfüggő, azaz teljesíti a*

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

azonosságot minden $\sigma \in \Sigma_n$ szimbólumra és π permutációra, valamint az

$$y \preceq x_i, z \preceq x_j \quad \Rightarrow \quad \sigma(x_1, \ldots, x_{n-1}, y) = \sigma(x_1, \ldots, x_{n-1}, z)$$

és az

$$x_1 \sim y_1, \ldots, x_n \sim y_n \quad \Rightarrow \quad \sigma(x_1, \ldots, x_n) = \sigma(y_1, \ldots, y_n)$$

implikációkat minden $\sigma \in \Sigma_n$-re. Itt $\preceq$ jelöli a minimális automata elérhetőségi relációját, $\sim$ pedig az $a \sim b \Leftrightarrow a \preceq b \wedge b \preceq a$ ekvivalenciarelációt.

Továbbá, az $L \subseteq T_\Sigma$ fanyelv pontosan akkor definiálható a CTL(EF*) *logikában, ha definiálható a* CTL(EF$^+$)*-ban és minimális automatája „dadogás-invariáns", vagyis teljesül benne a*

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

egyenlőség minden $\sigma \in \Sigma_n$-re.

A fenti karakterizáció szerint mindkét kérdés polinom időben eldönthető; ez korábban is ismert volt a CTL(EF$^+$) esetére, ld. [5], míg a CTL(EF*) jellemzése egy [5]-ben nyitottként felvetett probléma volt. Ez utóbbit [58] eltérő módszerrel, függetlenül megoldotta.

A 2.5. szakaszban egy újabb jellemzést adtunk, kétszemélyes játékok segítségével. Fanyelvek egy (tetszőleges) $\mathcal{L}$ osztályához és az $n \geq 0$ egészhez definiáltuk az n-fordulós $\mathcal{L}$-játékot, mely egy kétszemélyes, fák egy (s,t) párján játszott játék. A 2.5.3. következmény kapcsolja össze a fenti játékot az FTL($\mathcal{L}$)-definiálhatósággal:

Fanyelvek tetszőleges $\mathcal{L}$ véges osztályára és L fanyelvre fennáll, hogy az L nyelv pontosan akkor esik bele az $\mathbf{FTL}(\mathcal{L})$ *osztályba, ha van egy olyan $n \geq 0$ szám, melyre az n-fordulós $\mathcal{L}$-játékban a kezdő játékosnak van nyerő stratégiája, valahányszor a játékot fák olyan (s,t) párján játsszák, ahol $s \in L$ és $t \notin L$.*

A harmadik fejezetben az aperiodicitás különböző fogalmait vizsgáltuk. Ha Σ egy rangolt ábécé és $\mathbb{A} = (A, \Sigma)$ egy véges Σ-faautomata, akkor nemtriviális ΣX_n-fák bármely $(t_1, \ldots, t_n)$ vektora meghatároz egy $A^n \to A^n$ transzformációt. Ezek a transzformációk egy félcsoportot alkotnak, melyet $S_n(\mathbb{A})$ jelöl (és melyben a szorzás a transzformációk kompozíciója). Az $\mathbb{A}$ véges faautomatát n-aperiodikusnak mondjuk, ha az $S_n(\mathbb{A})$ félcsoport aperiodikus, vagyis ha benne az $x^K = x^{K+1}$ azonosság fennáll valamely K-ra. Továbbá, az $\mathbb{A}$ véges faautomata erősen aperiodikus, ha minden $n \geq 1$-re n-aperiodikus. Követve [54]-et, $\mathbb{A}$-t környezet-aperiodikusnak (context aperiodic) nevezzük, ha a nemtriviális környezetek által indukált leképezések félcsoportja aperiodikus. Az n-aperiodikus véges faautomaták osztályát $\mathbf{SAper}_n$, az erősen aperiodikusokét **SAper**, a környezet-aperiodikusokét pedig **CAper** jelöli. A legutóbbi osztály véges faautomatáknak egy kaszkád szorzatra zárt pszeudovarietása, míg a többi az általánosított kaszkád szorzatára is zárt pszeudovarietás.

A

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

tartalmazások minden R rangtípusra fennállnak. A 3.3.1. állításban megmutattuk, hogy ha $R = \{0,1\}$ vagy $R = \{1\}$, azaz mikor fáink valójában szavak, a hierarchia összeomlik, ekkor

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

Ugyanakkor ha R tartalmaz egy 1-nél nagyobb egészet (másképp fogalmazva, ha R nemklasszikus), a hierarchia a 3.3.2, 3.3.3 és 3.3.4 állítások szerint valódi:

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

Vizsgáltuk továbbá a fenti osztályok eldöntési problémájának bonyolultságát is: megmutattuk, hogy ha R nem-klasszikus rangtípus, akkor bármely rögzített n-re PSPACE-nehéz kérdés annak eldöntése, hogy egy adott faautomata n-aperiodikus vagy sem (3.4.4. állítás). Ezzel ellentétben, az erős aperiodicitás eldöntésére megadtunk egy polinom idejű algoritmust (3.4.3. tétel).

Az aperiodicás fogalmainak és a logikai definiálhatóságnak a kapcsolatát is vizsgáltuk: a 3.5-ös szakasz eredményei szerint bináris fákra ($R = \{0, 2\}$) a következők állnak fenn:

1. Ha L definiálható CTL-ben, akkor minimális faautomatája 1-aperiodikus.
2. Létezik CTL-ben definiálható nyelv, melynek minimális faautomatája nem 2-aperiodikus.
3. Létezik FO($<$)-ben definiálható fanyelv, melynek minimális automatája nem 1-aperiodikus.
4. Létezik reguláris fanyelv, mely nem definiálható FO($<, S_i$)-ben, de melynek minimális automatája 1-aperiodikus.

Az eredmények bármely nem-klasszikus rangtípusra kiterjeszthetők.

Bevezettük továbbá az n-aperiodicitásnak egy kissé módosított fogalmát, melynek definíciójában a nemtriviális polinomvektorok által indukált leképezések szerepelnek (favektorok helyett). A 3.7.4. állítás kimondja, hogy az így definiált hierarchia minden rangtípusra összeomlik:

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

A fejezet zárásaként megmutattuk, hogy a polinom értelemben 1-aperiodikus véges faautomaták osztálya a tartalmazás reláció szerint összehasonlíthatatlan bármely $\mathbf{SAper}_n$ osztállyal csakúgy, mint **SAper**-rel (3.7.7. következmény).

A 3. fejezet eredményeit [17]-ben publikáltuk.

Bibliography

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

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

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

[4] L. Bernátsky. Regular expression star-freeness is PSPACE-complete. *Acta Cybernetica*, 13:1–21, 1997.

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

[6] J. R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundl. Math.*, 6:66–92, 1960.

[7] S. Cho and Dung T. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88:99–116, 1991.

[8] J. Cohen, D. Perrin and J.-E. Pin. On the expressive power of temporal logic. *J. Comput. System Sci.*, 46:271–294, 1983.

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

[10] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the AMS*, 98:21–52, 1961.

[11] 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.

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

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

[14] 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.

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

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

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

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

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

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

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

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

[23] F. Gécseg and B. Imreh. On α_i-product of automata. *Acta Cybernetica* 8(2):135–141, 1987.

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

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

[26] G. Grätzer. *Universal Algebra*, Springer, 1979.

[27] 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.

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

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

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

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

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

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

[34] D. Peled and T. Wilke. Stutter-invariant temporal properties are expressible without the next-time operator, *Inf. Proc. Lett.*, 63:243–246, 1997.

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

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

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

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

[39] V. Piirainen. Monotone algebras, R-trivial monoids and a variety of tree languages. Bulletin of the EATCS 84:189-194, 2004.

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

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

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

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

[44] S. Salehi. *Varieties of Tree Languages*. TUCS Dissertations, No. 64, 2006.

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

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

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

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

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

[50] 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.

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

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

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

[54] 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.

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

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

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

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

Index