Publ. Math. Debrecen
77/3-4 (2010), 277–297

A family of temporal logics on finite trees

By ZOLTÁN ÉSIK (Szeged) and SZABOLCS IVÁN (Szeged)

This paper is dedicated to Professor P. Dömösi

Abstract. We associate a temporal logic XTL($\mathcal{L}$) with each class $\mathcal{L}$ of (regular) tree languages and provide both an algebraic and a game-theoretic characterization of the expressive power of the logic XTL($\mathcal{L}$).

1. Introduction

A characterization of a logic on trees or words is called effective if it gives rise to an effective procedure to decide whether a property of trees or words is expressible in the logic. The property is usually modeled by a tree or word language and is given by a finite automaton. For example, it is known that a word language is definable in the first-order logic FO($<$) or in Linear Temporal Logic (LTL) if and only if its minimal automaton is finite and counter-free, or alternatively, if and only if its syntactic monoid is finite and aperiodic [16], [18]. Since it is decidable (*PSPACE*-complete) whether a finite automaton is counter-free, this characterization of FO($<$) (or LTL) is effective.

An algebraic characterization of first-order logic on finite trees using "preclones of finite algebras" has been given in [11]. However, this result does not provide any effective algorithm. In fact, finding an effective characterization of

Mathematics Subject Classification: 03B44,68Q70,91A40.
Key words and phrases: branching time future temporal logics, Ehrenfeucht–Fraïssé games, strict Moore-product.
This research was supported by OTKA grant no. K75249 and the program TÁMOP-4.2.2/08/1/2008-0008 of the Hungarian National Development Agency.

the expressive power of first-order logic on trees (with both the successor relations and the partial order relation derived from the successor relations) has been a long standing open problem, cf. [14], [17], [23].[1] With a few exceptions, there is no effective characterization known for temporal logics on (finite and/or infinite) trees. Most notably, no effective characterization of the logic CTL [5] is known.

In this paper we consider only finite trees. In [6], a logic FTL($\mathcal{L}$) was associated with each class $\mathcal{L}$ of regular tree languages. Under the assumption that the next modalities are expressible (and an additional technical condition), a characterization of the languages definable in FTL($\mathcal{L}$) was obtained using pseudovarieties of finite tree automata and cascade products. It was argued that by selecting particular (finite) language classes $\mathcal{L}$, most of the familiar temporal logics can be covered. In [8], we removed the extra condition on the next modalities by making use of a modified version of the cascade product, called the Moore-product. The logics FTL($\mathcal{L}$) contain "built in" atomic formulas describing the label of the root of a tree. This has the disadvantage that some classes of tree languages do not possess a characterization in terms of the logics FTL($\mathcal{L}$). For example, considering only unary trees, which correspond to words, no nontrivial variety of group languages can be derived from these logics.

In this paper, we introduce a generalization of the logics FTL($\mathcal{L}$). We associate yet another logic, called XTL($\mathcal{L}$), with each class $\mathcal{L}$ of tree languages. In the first part of the paper we show that, when $\mathcal{L}$ ranges over subclasses of regular tree languages (and satisfies a technical condition), then the classes of languages definable in XTL($\mathcal{L}$) are in a one-to-one correspondence with those pseudovarieties of finite tree automata which are closed under a variant of the Moore-product.

In the second part of the paper we provide a game-theoretic characterization of the logics XTL($\mathcal{L}$). With each class $\mathcal{L}$ of tree languages, we associate an Ehrenfeucht–Fraïssé-type game, called the XTL($\mathcal{L}$)-game, between "Spoiler" and "Duplicator". We obtain that two trees s, t can be separated by an XTL($\mathcal{L}$)-formula of "depth n" if and only if Spoiler has a winning strategy in the n-round XTL($\mathcal{L}$)-game on (s, t). We also discuss a modification of the game that characterizes the logics FTL($\mathcal{L}$).

The paper is ended by a few examples derived from the main theorems providing game-theoretic characterizations of some familiar logics, including a version of CTL for finite trees, and some of its fragments. This paper is an expanded and improved version of the extended abstract [10].

[1]The case when one has only the successor relations has been studied in [3] where an effective characterization has been found.

2. Preliminaries

A *rank type* is a nonempty finite set R of nonnegative integers containing 0. A *ranked alphabet* Σ (of rank type R) is a union $\bigcup_{n \in R} \Sigma_n$ of pairwise disjoint, finite nonempty sets of symbols. Elements of Σ_0 are also called *constant symbols*. We assume that each ranked alphabet Σ comes with a fixed lexicographic ordering denoted $<_\Sigma$, or just $<$ when Σ is understood.

For the whole paper we now fix an arbitrary rank type R.

Given a ranked alphabet Σ, the set T_Σ of Σ-*trees* is the least set such that whenever $\sigma \in \Sigma_k$, $k \in R$ is a symbol and $t_1, \ldots, t_k$ are Σ-trees, then $\sigma(t_1, \ldots, t_k)$ is also a Σ-tree. When σ is a constant symbol, we often write σ for the tree $\sigma()$. A (Σ)-*tree language* L is any subset of T_Σ.

We can also view a Σ-tree as a map from a tree domain to Σ. In this setting, the *domain* $\mathrm{dom}(t)$ of a tree t is defined inductively as follows. When $t = \sigma \in \Sigma_0$, $\mathrm{dom}(t) = \{\epsilon\}$, the singleton set whose unique element is the empty word. Suppose that $t = \sigma(t_1, \ldots, t_n)$, where $n > 0$. Then $\mathrm{dom}(t) = \{\epsilon\} \cup \bigcup_{i=1}^{n} \{i \cdot v : v \in \mathrm{dom}(t_i)\}$. Elements of $\mathrm{dom}(t)$ are also called *nodes* of t. Then, a Σ-tree $t = \sigma(t_1, \ldots, t_n)$ can be viewed as a mapping $t : \mathrm{dom}(t) \to \Sigma$ defined inductively as follows: $t(\epsilon) = \sigma$, and for any node $i \cdot v \in \mathrm{dom}(t)$, $t(i \cdot v) = t_i(v)$. We define $\mathrm{Root}(t) = t(\epsilon)$. When $t(v) \in \Sigma_n$, we also say that v is a node of *rank* n. When t is a Σ-tree and s is a Δ-tree such that $\mathrm{dom}(t) = \mathrm{dom}(s)$, s is called a Δ-*relabeling* of t.

When t is a Σ-tree and $v \in \mathrm{dom}(t)$ is a node of t, the *subtree* of t rooted at v is defined as the tree $t|_v$ with $\mathrm{dom}(t|_v) = \{w : v \cdot w \in \mathrm{dom}(t)\}$ and $t|_v(w) = t(v \cdot w)$. We extend the above notions to tuples of trees as follows: when $\underline{t} = (t_1, \ldots, t_n)$ is an n-tuple of trees, let $\mathrm{dom}(\underline{t}) = \bigcup_{i=1}^{n} \{i \cdot v : v \in \mathrm{dom}(t_i)\}$, and for any node $i \cdot v \in \mathrm{dom}(\underline{t})$, let $\underline{t}(i \cdot v) = t_i(v)$ and $\underline{t}|_{i \cdot v} = t_i|_v$.

Suppose Σ and Δ are ranked alphabets and h is a rank-preserving mapping $\Sigma \to \Delta$, i.e., for any $n \in R$ and $\sigma \in \Sigma_n$, $h(\sigma)$ is contained in Δ_n. Then h determines a *literal tree homomorphism* $T_\Sigma \to T_\Delta$, also denoted h, defined as follows: for any tree $t \in T_\Sigma$, let $\mathrm{dom}(h(t)) = \mathrm{dom}(t)$, and for any node $v \in \mathrm{dom}(t)$, let $h(t)(v) = h(t(v))$. Thus, $h(t)$ is a Δ-relabeling of t.

When Σ is a ranked alphabet, let $\Sigma(\bullet)$ denote its enrichment by a new constant symbol $\bullet$. A Σ-*context* is a tree $\zeta \in T_{\Sigma(\bullet)}$ in which $\bullet$ occurs exactly once. When ζ is a Σ-context and t is a Σ-tree, $\zeta(t)$ denotes the Σ-tree resulting from ζ by substituting t in place of the "hole" $\bullet$. When $L \subseteq T_\Sigma$ is a tree language and ζ is a Σ-context, the *quotient of L with respect to ζ* is the tree language $\zeta^{-1}(L) = \{t : \zeta(t) \in L\}$.

Suppose Σ is a ranked alphabet. A Σ-*algebra* $\mathbb{A} = (A, \Sigma)$ consists of a nonempty *set A of states* and for each symbol $\sigma \in \Sigma_n$ an associated *elementary operation* $\sigma^{\mathbb{A}} : A^n \to A$. *Subalgebras, homomorphisms, quotients* etc. are defined as usual, cf. [13]. A Σ-*tree automaton* is a Σ-algebra which contains no proper subalgebra. A tree automaton $\mathbb{A} = (A, \Sigma)$ is called *finite* if A is finite; if $|A| = 1$, $\mathbb{A}$ is called *trivial*.

In any Σ-algebra $\mathbb{A}$, any tree $t \in T_\Sigma$ *evaluates* to a state $t^{\mathbb{A}} \in A$ defined as usual. Thus, a Σ-algebra $\mathbb{A} = (A, \Sigma)$ is a tree automaton if and only if all of its states are accessible, i.e. for each $a \in A$ there exists some tree $t \in T_\Sigma$ with $t^{\mathbb{A}} = a$. The *connected part* of a Σ-algebra $\mathbb{A}$ is the tree automaton which is the subalgebra of $\mathbb{A}$ determined by the states $t^{\mathbb{A}}$, where t ranges over T_Σ.

Suppose that $\mathbb{A}$ is a Σ-tree automaton. When also a set $A' \subseteq A$ is given, $\mathbb{A}$ *recognizes* the tree language $L_{\mathbb{A},A'} = \{t : t^{\mathbb{A}} \in A'\}$ with the set A' of *final states*. When $A' = \{a\}$ is a singleton set, we write just $L_{\mathbb{A},a}$. 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. A tree language is called *regular* if it is recognizable by some finite tree automaton.

We say that the tree automaton $\mathbb{B} = (B, \Delta)$ is a *renaming* of the tree automaton $\mathbb{A} = (A, \Sigma)$ if $B \subseteq A$ and each elementary operation of $\mathbb{B}$ is a restriction of an elementary operation of $\mathbb{A}$. When $\mathbb{A} = (A, \Sigma)$ is a tree automaton, Δ is a ranked alphabet and $h : \Delta \to \Sigma$ is a rank-preserving mapping, then h *determines* the renaming $\mathbb{B}$ which is the connected part of the algebra $\mathbb{A}' = (A, \Delta)$ where for each $\delta \in \Delta$, $\delta^{\mathbb{A}'} = (h(\delta))^{\mathbb{A}}$.

When $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Sigma)$ are tree automata, their *direct product* $\mathbb{A} \times \mathbb{B}$ is the connected part of the Σ-algebra $\mathbb{C} = (A \times B, \Sigma)$, where for each $\sigma \in \Sigma_n$ and states $a_1, \ldots, a_n \in A$, $b_1, \ldots, b_n \in B$,

$$\sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) = (\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \sigma^{\mathbb{B}}(b_1, \ldots, b_n)).$$

We call a nonempty class $\mathbf{V}$ of finite tree automata a *pseudovariety of finite tree automata* if it is closed under renamings, direct products and quotients. A closely related notion is that of literal varieties of tree languages: a nonempty class $\mathcal{V}$ of regular tree languages is a *literal variety of tree languages* if it is closed under the Boolean operations, quotients and inverse literal homomorphisms.

There exists an *Eilenberg correspondence* between the lattice of pseudovarieties of finite tree automata and the lattice of literal varieties of tree languages: the mapping

$$\mathbf{K} \mapsto \mathcal{V}_{\mathbf{K}} = \{L : L \text{ is recognizable by some member of } \mathbf{K}\},$$

restricted to pseudovarieties, establishes an order isomorphism between the two lattices. For more information on (literal) varieties of tree languages the reader is referred to [19], [20], [21], [6].

3. The logic XTL($\mathcal{L}$)

In this section we introduce an extension of the logics FTL($\mathcal{L}$) defined in [6] and further investigated in [8], [9].

Each modal operator of the logic CTL corresponds to a regular tree language in a canonical way, cf. [6]. For example, consider the ranked alphabet Bool which contains exactly two symbols, $\uparrow_n$ and $\downarrow_n$ for each $n \in R$. As a shorthand, let $\mathrm{UP} = \{\uparrow_n \colon n \in R\}$ and $\mathrm{DOWN} = \{\downarrow_n \colon n \in R\}$. (For technical reasons, we fix an arbitrary ordering $<_{\mathrm{Bool}}$ satisfying $\uparrow_n <_{\mathrm{Bool}} \downarrow_n$ for each $n \in R$.) Then the EF^* (nonstrict existential future) modality corresponds to the regular tree language in T_{Bool} consisting of those trees having at least one node labeled in UP. Further examples are given in Examples 1 and 2. Conversely, as argued in [6], each regular tree language can in turn be seen as a modal operator. This allows us to treat various temporal logics on trees in a unified manner. We make these ideas more precise in the following definitions.

Let $\mathcal{L}$ be a class of tree languages and let Σ be a ranked alphabet. The set of XTL($\mathcal{L}$)-*formulas* over Σ is the least set satisfying the following conditions:

(1) The symbol $\downarrow$ is an XTL($\mathcal{L}$)-formula (of depth 0).

(2) For any ranked alphabet Δ, rank-preserving mapping $\pi : \Sigma \to \Delta$ and Δ-tree language $L \in \mathcal{L}$, (L, π) is an (atomic) XTL($\mathcal{L}$)-formula (of depth 0).

(3) When φ is an XTL($\mathcal{L}$)-formula (of depth d), then $(\neg\varphi)$ is also an XTL($\mathcal{L}$)-formula (of depth d).

(4) When φ and ψ are XTL($\mathcal{L}$)-formulas (of maximal depth d), then $(\varphi \vee \psi)$ is also an XTL($\mathcal{L}$)-formula (of depth d).

(5) When Δ is a ranked alphabet, $L \in \mathcal{L}$ is a Δ-tree language and for each $\delta \in \Delta$, φ_δ is an XTL($\mathcal{L}$)-formula over Σ (of maximal depth d), then $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ is an XTL($\mathcal{L}$)-formula (of depth $d+1$).

We now turn to the definition of the semantics. We need to define what it means that a Σ-tree t *satisfies* an XTL($\mathcal{L}$)-formula φ over Σ, in notation $t \models \varphi$. Since Boolean connectives and the falsity symbol $\downarrow$ are handled as usual, we only concentrate on two types of formulas.

(1) If $\varphi = (L, \pi)$ for some rank-preserving mapping $\pi : \Sigma \to \Delta$ and Δ-tree language $L \in \mathcal{L}$, then $t \models \varphi$ if and only if $\pi(t)$ is contained in L;

(2) If $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ then $t \models \varphi$ if and only if the *characteristic tree* $\hat{t}$ of t determined by the family $(\varphi_\delta)_{\delta\in\Delta}$ is contained in L.

Here $\hat{t}$ is a Δ-relabeling of t, defined as follows: for every node $v \in \mathrm{dom}(t)$ with $t(v) \in \Sigma_n$, $\hat{t}(v) = \delta$, where δ is either the first symbol in Δ_n with $t|_v \models \varphi_\delta$; or there is no such symbol and δ is the last element of Δ_n.

We use the usual shorthands $\uparrow$ for $(\neg\downarrow)$ and $(\varphi \wedge \psi)$ for $\neg((\neg\varphi) \vee (\neg\psi))$.

An XTL($\mathcal{L}$)-formula over the ranked alphabet Σ *defines* the tree language $L_\varphi = \{t \in T_\Sigma : t \models \varphi\}$. **XTL**($\mathcal{L}$) denotes the class of tree languages definable by some XTL($\mathcal{L}$)-formula. We say that two formulas, φ and ψ are *equivalent* if $L_\varphi = L_\psi$.

The logic FTL($\mathcal{L}$) [6] differs from the logic XTL($\mathcal{L}$) in that the atomic formulas over Σ are $\downarrow$ and the formulas p_σ, where $\sigma \in \Sigma$, defining the language of all Σ-trees whose root is labeled σ. We let **FTL**($\mathcal{L}$) denote the class of tree languages definable by the formulas of the logic FTL($\mathcal{L}$).

Example 1. Let $R = \{0, 2\}$, $\Sigma_2 = \{f\}$, $\Sigma_0 = \{a, b\}$. Consider the rank-preserving mapping $\pi : \Sigma \to \mathrm{Bool}$ given by $\pi(f) =\downarrow_2$, $\pi(a) =\uparrow_0$ and $\pi(b) =\downarrow_0$. Let L_{even} be the set of all trees in T_{Bool} with an even number of nodes labeled in UP. Then the formula $\psi = \neg(L_{\mathrm{even}}, \pi)$ defines the set of all Σ-trees having an odd number of leaves labeled a. Let $\varphi_{\uparrow_2}$ be the formula ψ defined above, and let $\varphi_\delta =\downarrow$ for all $\delta \in \mathrm{Bool}$, $\delta \neq\uparrow_2$. Then the formula $L_{\mathrm{even}}(\delta \mapsto \varphi_\delta)_{\delta\in\mathrm{Bool}}$ defines the set of all Σ-trees with an even number of non-leaf subtrees having an odd number of leaves labeled a.

Example 2. In this example let $R = \{0, 1\}$. When Σ is a ranked alphabet (of rank type R), then any Σ-tree determines a word over Σ_1 which is the sequence of node labels from the root to the leaf of the tree not including the leaf label. By extension, each tree language over Σ determines a word language over Σ_1. Let L'_{even} be the set of all trees in T_{Bool} with an even number of nodes labeled $\uparrow_1$, and let $\mathcal{L} = \{L'_{\mathrm{even}}\}$. Then a tree language $K \subseteq T_\Sigma$ is definable in XTL($\mathcal{L}$) if and only if the word language determined by K is a (regular) group language whose syntactic group is a p-group for $p = 2$, see [22]. There is no class $\mathcal{L}'$ such that FTL($\mathcal{L}'$) would define the same language class.

The operators **FTL** and **XTL** are related by Proposition 1 below. Let us define the Bool-tree language

$$L_\uparrow = \{t \in T_{\mathrm{Bool}} : \mathrm{Root}(t) \in \mathrm{UP}\}.$$

Proposition 1. *For any class $\mathcal{L}$ of tree languages,*

$$\mathbf{FTL}(\mathcal{L}) = \mathbf{XTL}(\mathcal{L} \cup \{L_\uparrow\}).$$

Proof. Let Σ be a ranked alphabet. It is clear that for each $\sigma \in \Sigma_n$, the formulas p_σ and $(L_\uparrow, \pi)$ define the same language, where $\pi : \Sigma \to \text{Bool}$ maps σ to $\uparrow_n$ and all other symbols to a symbol in DOWN. It follows by a straightforward induction argument that $\mathbf{FTL}(\mathcal{L}) \subseteq \mathbf{XTL}(\mathcal{L} \cup \{L_\uparrow\})$.

Now let ψ be an XTL$(\mathcal{L} \cup \{L_\uparrow\})$-formula over the ranked alphabet Σ. By induction on the structure of ψ, we construct an FTL$(\mathcal{L})$-formula ψ' defining the language L_ψ.

(1) When $\psi =\downarrow$, then $\psi' =\downarrow$.

(2) Suppose $\psi = (L_\uparrow, \pi)$ for some rank-preserving mapping $\pi : \Sigma \to \text{Bool}$. Then we define ψ' as $\bigvee_{\pi(\sigma)\in \text{UP}} p_\sigma$.

(3) Suppose $\psi = (L, \pi)$ for some Δ-tree language $L \in \mathcal{L}$ and rank-preserving mapping $\pi : \Sigma \to \Delta$. Then we define ψ' as $L(\delta \mapsto \psi_\delta)$, where $\psi_\delta = \bigvee_{\pi(\sigma)=\delta} p_\sigma$ for each δ.

(4) When $\psi = (\neg\psi_1)$ or $\psi = (\psi_1 \vee \psi_2)$, we define ψ' as $(\neg\psi_1')$ and $(\psi_1' \vee \psi_2')$, respectively.

(5) When $\psi = L(\delta \mapsto \psi_\delta)_{\delta\in\Delta}$ for some Δ-tree language $L \in \mathcal{L}$, we define $\psi' = L(\delta \mapsto \psi'_\delta)_{\delta\in\Delta}$.

(6) Finally, when $\psi = L_\uparrow(\delta \mapsto \psi_\delta)_{\delta\in\text{Bool}}$, we define ψ' as $\bigvee_{n\in R} \psi_{\uparrow_n}$. □

In [6], it has been shown that **FTL** is a closure operator preserving regularity. Thus, when $\mathcal{L}$ is a class of regular tree languages then $\mathbf{FTL}(\mathcal{L})$ only contains regular tree languages. Moreover, $\mathbf{FTL}(\mathcal{L})$ is closed under the Boolean operations and inverse literal homomorphisms, and is closed under quotients if and only if each quotient of any language in $\mathcal{L}$ belongs to $\mathbf{FTL}(\mathcal{L})$. The same facts hold for the operator **XTL**, with almost the same proofs.

Theorem 1. (1) *The operator* **XTL** *is a closure operator: for any classes* $\mathcal{L}$, $\mathcal{L}'$ *of tree languages,*

(a) $\mathcal{L} \subseteq \mathbf{XTL}(\mathcal{L})$;

(b) $\mathbf{XTL}(\mathbf{XTL}(\mathcal{L})) \subseteq \mathbf{XTL}(\mathcal{L})$,

(c) *if* $\mathcal{L} \subseteq \mathcal{L}'$, *then* $\mathbf{XTL}(\mathcal{L}) \subseteq \mathbf{XTL}(\mathcal{L}')$.

(2) *When* $\mathcal{L}$ *is a class of regular tree languages, then so is* $\mathbf{XTL}(\mathcal{L})$.

(3) *For any class* $\mathcal{L}$ *of tree languages,* $\mathbf{XTL}(\mathcal{L})$ *is closed under the Boolean operations and inverse literal homomorphisms, and is closed under quotients if and only if each quotient of any language in* $\mathcal{L}$ *is in* $\mathbf{XTL}(\mathcal{L})$.

4. Definability and membership

In this section we recall from [8] the notion of the strict Moore-product of tree automata and that of strict Moore pseudovarieties, and relate the operator **XTL** to strict Moore pseudovarieties.

Suppose $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are tree automata and $\alpha : A \times R \to \Delta$ is a rank-preserving mapping, i.e., for any $n \in R$ and $a \in A$, $\alpha(a, n)$ is contained in Δ_n. Then the *strict Moore-product of* $\mathbb{A}$ *and* $\mathbb{B}$ *determined by* α is the tree automaton $\mathbb{A} \times_\alpha \mathbb{B}$ which is the connected part of the algebra $\mathbb{C} = (A \times B, \Sigma)$, where for each $\sigma \in \Sigma_n$ and $a_1, \ldots, a_n \in A$, $b_1, \ldots, b_n \in B$,

$$\sigma^{\mathbb{C}}((a_1, b_1), \ldots, (a_n, b_n)) = (\sigma^{\mathbb{A}}(a_1, \ldots, a_n), \delta^{\mathbb{B}}(b_1, \ldots, b_n))$$

with $\delta = \alpha(\sigma^{\mathbb{A}}(a_1, \ldots, a_n), n)$.

A pseudovariety $\mathbf{V}$ of finite tree automata is called a *strict Moore pseudovariety* if it is also closed under the strict Moore-product. It is clear that for any class $\mathbf{K}$ of finite tree automata there exists a least strict Moore pseudovariety $\langle \mathbf{K} \rangle_s$ containing $\mathbf{K}$.

Proposition 2. *Suppose $\mathbb{A} = (A, \Sigma)$ is a tree automaton and $\mathcal{L}$ is a class of tree languages such that each tree language recognizable by $\mathbb{A}$ is in $\mathbf{XTL}(\mathcal{L})$. Then any tree language recognizable by a renaming or quotient of $\mathbb{A}$ is also in $\mathbf{XTL}(\mathcal{L})$.*

Proof. When $\mathbb{B} = (A, \Delta)$ is the renaming of $\mathbb{A} = (A, \Sigma)$ determined by the rank-preserving mapping $\pi : \Delta \to \Sigma$, then each language L recognizable by $\mathbb{B}$ is of the form $\pi^{-1}(K)$, for some Σ-tree language K recognizable by $\mathbb{A}$. Since $\mathbf{XTL}(\mathcal{L})$ is closed under inverse literal homomorphisms, the claim is proved for renamings.

When $\mathbb{B}$ is a quotient of $\mathbb{A}$, each language recognizable by $\mathbb{B}$ is also recognizable by $\mathbb{A}$, which proves the claim for quotients. □

Proposition 3. *Suppose $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Sigma)$ are finite tree automata and $\mathcal{L}$ is a class of tree languages such that each tree language recognizable by either $\mathbb{A}$ or $\mathbb{B}$ is in $\mathbf{XTL}(\mathcal{L})$. Then each tree language recognizable by the direct product $\mathbb{A} \times \mathbb{B}$ is also in $\mathbf{XTL}(\mathcal{L})$.*

Proof. It suffices to show that whenever $a \in A$ and $b \in B$ are states, then the tree language $L_{\mathbb{A}\times\mathbb{B},(a,b)}$ is definable in $\mathrm{XTL}(\mathcal{L})$. But when φ_a defines the tree language $L_{\mathbb{A},a}$ and φ_b defines $L_{\mathbb{B},b}$, then $\varphi_a \wedge \varphi_b$ defines $L_{\mathbb{A}\times\mathbb{B},(a,b)}$. □

Proposition 4. *Suppose $\mathbb{A} = (A, \Sigma)$ and $\mathbb{B} = (B, \Delta)$ are finite tree automata and $\mathcal{L}$ is a class of tree languages such that each tree language recognizable by either $\mathbb{A}$ or $\mathbb{B}$ is in $\mathbf{XTL}(\mathcal{L})$. Then each tree language recognizable by any strict Moore-product $\mathbb{A} \times_\alpha \mathbb{B}$ is also in $\mathbf{XTL}(\mathcal{L})$.*

Proof. It suffices to show that whenever $a \in A$ and $b \in B$, then the tree language $L_{\mathbb{A}\times_\alpha\mathbb{B},(a,b)}$ is definable in XTL($\mathcal{L}$). By assumption, $L_{\mathbb{B},b}$ is definable in XTL($\mathcal{L}$), and for each $a' \in A$, $L_{\mathbb{A},a'}$ is definable by some XTL($\mathcal{L}$)-formula $\tau_{a'}$. Then $L_{\mathbb{A}\times_\alpha\mathbb{B},(a,b)}$ is definable by the XTL(**XTL**($\mathcal{L}$))-formula $\tau_a \wedge L_{\mathbb{B},b}(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$, where for each $\delta \in \Delta_n$,

$$\varphi_\delta = \bigvee_{\alpha(a',n)=\delta} \tau_{a'}.$$

Since by Theorem 1, **XTL** is a closure operator, the above formula is equivalent to some XTL($\mathcal{L}$)-formula. □

Using Propositions 2, 3 and 4 we get:

Proposition 5. *Suppose* **K** *is a class of finite tree automata and* $\mathcal{L}$ *is a class of tree languages such that each tree language recognizable by some member of* **K** *is definable in* XTL($\mathcal{L}$)*. Then each tree language recognizable by some automaton in* $\langle\mathbf{K}\rangle_s$ *is also definable in* XTL($\mathcal{L}$)*.*

The converse also holds:

Proposition 6. *Suppose* $\mathcal{L}$ *is a class of (regular) tree languages and* **K** *is a class of finite tree automata such that each member of* $\mathcal{L}$ *is recognizable by some automaton in* **K***. Then every tree language definable in* XTL($\mathcal{L}$) *is recognizable by some automaton in* $\langle\mathbf{K}\rangle_s$*.*

Proof. We argue by induction on the structure of the XTL($\mathcal{L}$)-formula φ over Σ.

(1) If $\varphi =\downarrow$, L_φ is the empty set which is recognizable by any tree automaton in $\langle\mathbf{K}\rangle_s$.

(2) Suppose $\varphi = (L, \pi)$ for some Δ-tree language $L \in \mathcal{L}$ and rank-preserving mapping $\pi : \Sigma \to \Delta$. By assumption, L is recognizable by some tree automaton $\mathbb{B} = (B, \Delta)$ contained in **K**. Then L_φ is recognizable by the renaming of $\mathbb{B}$ determined by π.

(3) Suppose $\varphi = (\neg\varphi_1)$. By the induction hypothesis, L_{φ_1} is recognizable by some member $\mathbb{A}$ of $\langle\mathbf{K}\rangle_s$. Then L_φ is also recognizable by $\mathbb{A}$.

(4) Suppose $\varphi = (\varphi_1 \vee \varphi_2)$. By the induction hypothesis, L_{φ_i} is recognizable by some member $\mathbb{A}_i$ of $\langle\mathbf{K}\rangle_s$, $i = 1, 2$. Then L_φ is recognizable by the direct product $\mathbb{A}_1 \times \mathbb{A}_2$.

(5) Suppose $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ for some Δ-tree language $L \in \mathcal{L}$ and family $(\varphi_\delta)_{\delta\in\Delta}$ of XTL($\mathcal{L}$)-formulas. By the induction hypothesis, each L_{φ_δ} is recognizable by some member $\mathbb{A}_\delta$ of $\langle\mathbf{K}\rangle_s$ with some set $A'_\delta \subseteq A_\delta$ of final

states. Moreover, by assumption L is recognizable by some $\mathbb{B} = (B, \Delta) \in \mathbf{K}$ with some set B' of final states. Let us define the strict Moore-product $\mathbb{C} = (\prod_{\delta\in\Delta} \mathbb{A}_\delta) \times_\alpha \mathbb{B}$, where for each state $(a_\delta)_{\delta\in\Delta}$ of the direct product $(\prod_{\delta\in\Delta} \mathbb{A}_\delta)$ and integer $n \in R$, $\alpha((a_\delta)_{\delta\in\Delta}, n) = \overline{\delta} \in \Delta_n$ if one of the following holds:

(a) either $a_{\overline{\delta}} \in A'_{\overline{\delta}}$ and $\overline{\delta}$ is the first such element of Δ_n;

(b) or $a_{\delta'} \notin A'_{\delta'}$ for each $\delta' \in \Delta_n$ and $\overline{\delta}$ is the last element of Δ_n.

Then L_φ is recognized by $\mathbb{C}$ with the set $\{((a_\delta)_{\delta\in\Delta}, b) : a_\delta \in A_\delta,\ b \in B'\}$ of final states. □

Propositions 5 and 6 imply the following characterization:

Theorem 2. *For any class* $\mathbf{K}$ *of finite tree automata,*

$$\mathcal{V}_{\langle\mathbf{K}\rangle_s} = \mathbf{XTL}(\mathcal{V}_\mathbf{K}).$$

Corollary 1. *The mapping* $\mathbf{K} \mapsto \mathcal{V}_\mathbf{K}$ *establishes an order isomorphism between the lattice of strict Moore pseudovarieties of finite tree automata and the lattice of literal varieties of tree languages* $\mathcal{V}$ *satisfying* $\mathbf{XTL}(\mathcal{V}) = \mathcal{V}$.

Observe that Proposition 6 implies also that the operator **XTL** preserves regularity, i.e., when $\mathcal{L}$ is a class of regular tree languages, $\mathbf{XTL}(\mathcal{L})$ is also a class of regular tree languages.

5. Ehrenfeucht–Fraïssé-type games

In this section we give a game-theoretic characterization of the logics $\mathrm{XTL}(\mathcal{L})$.

Let $\mathcal{L}$ be a class of tree languages, $n \geq 0$ an integer, and let s, t be Σ-trees for some ranked alphabet Σ. The *n-round* $\mathrm{XTL}(\mathcal{L})$*-game on the pair* (s,t) *of trees* is played between two competing players, Spoiler and Duplicator, according to the following rules:

(1) If there exists an atomic formula (L, π) which is satisfied by exactly one of the trees s and t, then Spoiler wins. Otherwise, Step 2 follows.

(2) If $n = 0$, Duplicator wins. Otherwise, Step 3 follows.

(3) Spoiler chooses a tree language $L \in \mathcal{L}$, over some ranked alphabet Δ, and Δ-relabelings $\hat{s}$ and $\hat{t}$ of s and t, respectively, such that exactly one of $\hat{s}$ and $\hat{t}$ is contained in L. If he cannot do so, Duplicator wins; otherwise, Step 4 follows.

(4) Duplicator chooses two nodes of the pair (s,t), x and y, of the same rank, such that $(\hat{s},\hat{t})(x) \neq (\hat{s},\hat{t})(y)$. (For the notation see the 5th paragraph of Section 2.) If he cannot do so, Spoiler wins. Otherwise, an $(n-1)$-round XTL($\mathcal{L}$)-game is played on the pair $((s,t)|_x,(s,t)|_y)$. The player winning the subgame also wins the whole game.

Clearly, for any class $\mathcal{L}$ of tree languages, integer $n \geq 0$ and pair (s,t) of Σ-trees, one of the players has a winning strategy in the n-round XTL($\mathcal{L}$)-game played on (s,t). Let $s \sim_{\mathcal{L}}^n t$ denote that Duplicator has a winning strategy in the n-round XTL($\mathcal{L}$)-game on the pair (s,t). Also, when s and t are Σ-trees for some ranked alphabet Σ, $\mathcal{L}$ is a class of tree languages and $n \geq 0$ is an integer, let $s \equiv_{\mathcal{L}}^n t$ denote that s and t satisfy the same set of XTL($\mathcal{L}$)-formulas (over Σ) having depth at most n.

Proposition 7. *For any class $\mathcal{L}$ of tree languages, integer $n \geq 0$, ranked alphabet Σ, and pair s, t of Σ-trees, if $s \sim_{\mathcal{L}}^n t$ then $s \equiv_{\mathcal{L}}^n t$.*

PROOF. We argue by induction on n, and by contraposition. Suppose $s \not\equiv_{\mathcal{L}}^n t$.

When $n=0$, there exists an XTL($\mathcal{L}$)-formula (L,π) for some Δ-tree language $L \in \mathcal{L}$ and rank-preserving mapping $\pi : \Sigma \to \Delta$ separating s and t. Then exactly one of the Δ-trees $\pi(s)$ and $\pi(t)$ is contained in L, thus Spoiler indeed wins the 0-round XTL($\mathcal{L}$)-game on (s,t).

Let $n>0$ and suppose that we have proved the claim for $n-1$. From $s \not\equiv_{\mathcal{L}}^n t$ we get that either $s \not\equiv_{\mathcal{L}}^{n-1} t$, or there exists an XTL($\mathcal{L}$)-formula $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ of depth n separating s and t.

When $s \not\equiv_{\mathcal{L}}^{n-1} t$ then by the induction hypothesis $s \not\sim_{\mathcal{L}}^{n-1} t$, and thus $s \not\sim_{\mathcal{L}}^n t$.

Assume now that s and t are separated by the XTL($\mathcal{L}$)-formula $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ of depth n, say $s \models \varphi$ and $t \not\models \varphi$. Without loss of generality we may assume that the family $(\varphi_\delta)_{\delta\in\Delta}$ is *deterministic*, i.e. for any tree $t \in T_\Sigma$ there exists exactly one $\delta \in \Delta_k$ with $t \models \varphi_\delta$, where k is the arity of the root symbol of t. To see this, consider any family $(\psi_\delta)_{\delta\in\Delta}$ of XTL($\mathcal{L}$)-formulas. Then the family $(\psi'_\delta)_{\delta\in\Delta}$ defined as

$$\psi'_\delta = \begin{cases} \psi_\delta \ \wedge \ \neg\bigvee_{\delta'\in\Delta_k,\delta'<\delta} \psi_{\delta'} & \text{if } \delta\in\Delta_k \text{ is not the maximal element of } \Delta_k; \\ \neg\bigvee_{\delta'\in\Delta_k,\delta'<\delta} \psi_{\delta'} & \text{otherwise,} \end{cases}$$

is a deterministic family of formulas equivalent to $(\psi_\delta)_{\delta\in\Delta}$, i.e., for any tree t, the respective characteristic trees coincide.

A winning strategy for Spoiler is given as follows: let Spoiler choose the Δ-tree language $L \in \mathcal{L}$ and the characteristic trees $\hat{s}$ and $\hat{t}$ of s and t, respectively, determined by the family $(\varphi_\delta)_{\delta\in\Delta}$. From the semantics of XTL($\mathcal{L}$) we get that $\hat{s} \in L$ and $\hat{t} \notin L$, thus this is a valid move. Now assume Duplicator responds by choosing some nodes x, y of (s,t) of the same rank such that $(\hat{s},\hat{t})(x) \neq (\hat{s},\hat{t})(y)$. Let $\overline{\delta} = (\hat{s},\hat{t})(x)$. Since the family $(\varphi_\delta)_{\delta\in\Delta}$ is deterministic, $\varphi_{\overline{\delta}}$ separates $(s,t)|_x$ and $(s,t)|_y$. Since $\varphi_{\overline{\delta}}$ is of depth at most $n-1$, applying the induction hypothesis we get that Spoiler wins the $(n-1)$-round XTL($\mathcal{L}$)-game on $((s,t)|_x, (s,t)|_y)$, and thus wins the whole game. □

Proposition 8. *For any class $\mathcal{L}$ of tree languages, integer $n \geq 0$, ranked alphabet Σ 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 by contraposition. Let s, t be Σ-trees with $s \not\sim^n_{\mathcal{L}} t$.

If $n = 0$, then for some ranked alphabet Δ, rank-preserving mapping $\pi : \Sigma \to \Delta$ and Δ-tree language $L \in \mathcal{L}$, exactly one of the trees $\pi(s)$ and $\pi(t)$ is contained in L. Thus, the XTL($\mathcal{L}$)-formula (L, π) of depth 0 separates s and t.

Suppose that $n > 0$ and we have proved the claim for $n-1$. We consider two cases. If Spoiler has a winning strategy in the $(n-1)$-round XTL($\mathcal{L}$)-game, then by the induction hypothesis we have $s \not\equiv^{n-1}_{\mathcal{L}} t$, which clearly implies $s \not\equiv^n_{\mathcal{L}} t$. Otherwise, suppose that Spoiler chooses a Δ-tree language $L \in \mathcal{L}$ and two relabelings of the trees s and t in the first step following his winning strategy in the n-round game. Let the two relabelings be $\hat{s} \in L$ and $\hat{t} \notin L$. Then for any pair x, y of nodes of (s,t) of the same rank with $(\hat{s},\hat{t})(x) \neq (\hat{s},\hat{t})(y)$, Spoiler has a winning strategy in the $(n-1)$-round XTL($\mathcal{L}$)-game on $((s,t)|_x, (s,t)|_y)$. Applying the induction hypothesis, we get that for any such pair (x, y) there exists an XTL($\mathcal{L}$)-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 $\delta \in \Delta_k$, let us define the formula

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

where x and y range over the nodes of (s,t) of rank k. Observe that

$$(\hat{s},\hat{t})(z) = \delta \Rightarrow (s,t)|_z \models \varphi_\delta \tag{1}$$

for any node z of (s,t) and symbol $\delta \in \Delta$. Also, if z is a k-ary node of (s,t), then

$$(s,t)|_z \models \varphi_\delta \Rightarrow (\hat{s},\hat{t})(z) = \delta. \tag{2}$$

Indeed, suppose that z is a k-ary node, $(\hat{s},\hat{t})(z) \neq \delta$ and $(s,t)|_z \models \varphi_\delta$. Then there exists a node x with $(\hat{s},\hat{t})(x) = \delta$ such that $(s,t)|_z \models \bigwedge_{(\hat{s},\hat{t})(y)\neq\delta} \varphi_{x,y}$, where y

ranges over all nodes of (s,t) of rank k. Then $(s,t)|_z \models \varphi_{x,z}$, which contradicts the definition of the formula $\varphi_{x,z}$.

From (1) and (2) we get that $\hat{s}$ and $\hat{t}$ are the characteristic trees of s and t, respectively, determined by the family $(\varphi_\delta)_{\delta\in\Delta}$. Now since $\hat{s} \in L$ and $\hat{t} \notin L$, we conclude that the XTL($\mathcal{L}$)-formula $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$ of depth n separates s and t, completing the proof. □

Theorem 3. *For any class $\mathcal{L}$ of tree languages and any $n \geq 0$, the relations $\sim^n_{\mathcal{L}}$ and $\equiv^n_{\mathcal{L}}$ coincide.*

Corollary 2. *The following are equivalent for any* finite *class $\mathcal{L}$ of tree languages and any tree language L:*

i) $L \in \mathbf{XTL}(\mathcal{L})$;

ii) *there exists an integer $n \geq 0$ such that for all $s \in L$ and $t \notin L$, Spoiler has a winning strategy in the n-round* XTL($\mathcal{L}$)*-game on (s,t).*

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 XTL($\mathcal{L}$)-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 XTL($\mathcal{L}$)-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 XTL($\mathcal{L}$)-formula φ_s of depth at most n. Also the "infinitary disjunction" $\bigvee_{s\in L} \varphi_s$ is equivalent to some XTL($\mathcal{L}$)-formula φ; it is straightforward to see that $L_\varphi = L$ indeed holds, proving ii) $\to$ i). The other direction is a direct consequence of Theorem 3. □

6. Modified games

We have argued that the logics FTL($\mathcal{L}$) may be seen as special cases of the logics XTL($\mathcal{L}$). We may thus modify the game introduced in the previous section to obtain a game-theoretic characterization of the logics FTL($\mathcal{L}$). In this section, we introduce for each $n \geq 0$ and class $\mathcal{L}$ of tree languages the *n-round* FTL($\mathcal{L}$)*-game* characterizing the expressive power of FTL($\mathcal{L}$). Second, we introduce a *modified n-round* XTL($\mathcal{L}$)*-game*, applicable to certain classes $\mathcal{L}$ of tree languages. This game resembles the original Ehrenfeucht–Fraïssé game more than the n-round XTL($\mathcal{L}$)-game of the previous section. A combination of the two modifications is also introduced. By selecting special language classes $\mathcal{L}$, in the last section we derive games for some familiar temporal logics on finite trees related to CTL, cf. [1], [24].

Let $\mathcal{L}$ be a class of tree languages, $n \geq 0$, and let s, t be Σ-trees. The *n-round* FTL($\mathcal{L}$)*-game on the pair* (s, t) is played between Spoiler and Duplicator according to the same rules as the n-round XTL($\mathcal{L}$)-game, except for the first step which gets replaced by:

1'. If $\mathrm{Root}(s) \neq \mathrm{Root}(t)$, Spoiler wins. Otherwise, Step 2 follows.

(We may also modify step 4 by dropping the requirement that x and y have the same rank.) The following characterization theorem holds:

Theorem 4. *For any class $\mathcal{L}$ of tree languages, integer $n \geq 0$ and trees $s, t \in T_\Sigma$, Duplicator has a winning strategy in the n-round* FTL($\mathcal{L}$)*-game if and only if s and t satisfy the same set of* FTL($\mathcal{L}$)*-formulas of depth at most n. Consequently, if $\mathcal{L}$ is finite, then for any tree language L, $L \in$* **FTL**($\mathcal{L}$) *if and only if there exists an $n \geq 0$ such that Spoiler has a winning strategy in the n-round* FTL($\mathcal{L}$)*-game on any pair (s, t) of trees with $s \in L$ and $t \notin L$.*

Now we turn to the modified n-round XTL($\mathcal{L}$)-game. Recall that each ranked alphabet Σ comes with a fixed lexicographic ordering $<_\Sigma$. We define the following partial order $\preceq_\Sigma$ on Σ-trees: when $s, t \in T_\Sigma$, let $s \preceq_\Sigma t$ if and only if $\mathrm{dom}(s) = \mathrm{dom}(t)$ and for any node $v \in \mathrm{dom}(s)$, either $s(v) = t(v)$ or $t(v)$ is the last element of the corresponding Σ_n with respect to $<_\Sigma$. If in addition $s \neq t$ holds, then we write $s \prec_\Sigma t$.

Let $\mathcal{L}$ be a class of tree languages, let $n \geq 0$, and let s, t be Σ-trees. The *modified n-round* XTL($\mathcal{L}$)*-game on the pair* (s, t) is played between Spoiler and Duplicator according to the following rules:

(1–2) These steps are the same as in the n-round XTL($\mathcal{L}$)-game.

(3) Spoiler chooses one of the two trees, say s, some Δ-tree language $L \in \mathcal{L}$ and a relabeling $\hat{s}$ of s such that $\hat{s} \in L$ and for any $s' \in T_\Delta$, if $\hat{s} \prec_\Delta s'$ then $s' \notin L$. (That is, $\hat{s}$ is a *maximal* relabeling of s in L). If he cannot do so, Duplicator wins, otherwise Step 4 follows.

(4) Duplicator chooses a maximal relabeling $\hat{t}$ of t in the language L. If he cannot do so (i.e., t has no relabeling in L), then Spoiler wins, otherwise Step 5 follows.

(5) Spoiler chooses a node y of t such that $\delta = \hat{t}(y)$ is *not* the last element of

the respective Δ_k. If he cannot do so, Duplicator wins, otherwise Step 6 follows.

(6) Duplicator chooses a node x of s with $\hat{s}(x) = \delta$. If he cannot do so, Spoiler wins. Otherwise, a modified $(n-1)$-round XTL($\mathcal{L}$)-game is played on the pair $(s|_x, t|_y)$. The player winning the subgame also wins the whole game.

It is clear that for any class $\mathcal{L}$ of tree languages, $n \geq 0$, ranked alphabet Σ and Σ-trees s, t, one of the players has a winning strategy in the modified n-round XTL($\mathcal{L}$)-game on (s, t). Let $s \approx^n_{\mathcal{L}} t$ denote that Duplicator possesses such a strategy.

When Σ is a ranked alphabet, we also define the following partial ordering $\leq_\Sigma$ on Σ-trees: let $s \leq_\Sigma t$ if and only if $\mathrm{dom}(s) = \mathrm{dom}(t)$, and for any node $x \in \mathrm{dom}(s)$, $s(x) \leq_\Sigma t(x)$. We omit the subscript when it is clear from the context.

We say that a tree language $L \subseteq T_\Sigma$ is *downwards closed* if whenever s and t are Σ-trees with $s \leq_\Sigma t$ and $t \in L$, then also $s \in L$.

Proposition 9. *For any class $\mathcal{L}$ of downwards closed tree languages, integer $n \geq 0$, ranked alphabet Σ and trees $s, t \in T_\Sigma$, if $s \approx^n_{\mathcal{L}} t$ then $s \equiv^n_{\mathcal{L}} t$.*

Proof. The proof of this statement is similar to that of Proposition 7: only the case when s and t are separated by some XTL($\mathcal{L}$)-formula $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$ of depth $n > 0$ needs to be elaborated. Again, we can assume that $s \models \varphi$, $t \not\models \varphi$ and that the family $(\varphi_\delta)_{\delta \in \Delta}$ is deterministic.

We give a winning strategy for Spoiler as follows. Let s' be the characteristic tree of s determined by the family $(\varphi_\delta)_{\delta \in \Delta}$. Let Spoiler choose the tree s, the tree language L over the alphabet Δ and an arbitrary maximal relabeling $\hat{s}$ of s in L satisfying $s' \preceq_\Delta \hat{s}$. Since $s' \in L$, such a tree is guaranteed to exist. Note that for any node x of s, if $\hat{s}(x) = \delta$ is not the maximal element of the respective Δ_k, then $s|_x \models \varphi_\delta$.

Suppose Duplicator responds by choosing a maximal relabeling $\hat{t}$ of t in L. We claim that there exists a node y of t such that $\overline{\delta} = \hat{t}(y)$ is not the last element of the respective Δ_k, moreover, $t|_y \not\models \varphi_{\overline{\delta}}$. Indeed, suppose this is not the case. Then (since the family $(\varphi_\delta)_{\delta \in \Delta}$ is deterministic) the characteristic tree t' of t determined by $(\varphi_\delta)_{\delta \in \Delta}$ satisfies $t' \preceq_\Delta \hat{t}$, and hence also $t' \leq_\Delta \hat{t}$, so that $t' \in L$ since L is downwards closed. This contradicts the assumption that $t \not\models \varphi$. Let Spoiler choose such a node y of t in Step 5 and let $\overline{\delta} \in \Delta_k$ be the label $\hat{t}(y)$.

Assume Duplicator responds by choosing a node x of s with $\hat{s}(x)=\overline{\delta}$. Since $\overline{\delta}$ is not the maximal element of Δ_k, $s|_x \models \varphi_{\overline{\delta}}$. Hence, the formula $\varphi_{\overline{\delta}}$ of depth at most $n-1$ separates $s|_x$ and $t|_y$. Applying the induction hypothesis we get that $s|_x \not\approx_{\mathcal{L}}^{n-1} t|_y$, thus $s \not\approx_{\mathcal{L}}^n t$, proving the statement. □

Proposition 10. *For any class $\mathcal{L}$ of downwards closed tree languages, integer $n \geq 0$, ranked alphabet Σ and trees $s, t \in T_\Sigma$, if $s \equiv_{\mathcal{L}}^n t$ then $s \approx_{\mathcal{L}}^n t$.*

Proof. The proof of this statement is similar to that of Proposition 8. We only elaborate the case when $s \not\approx_{\mathcal{L}}^n t$ and $s \approx_{\mathcal{L}}^{n-1} t$ hold for $n > 0$.

Suppose that Spoiler chooses the maximal relabeling $\hat{s}$ of s in L for some Δ-tree language $L \in \mathcal{L}$ according to his winning strategy. Then, for any maximal relabeling $\hat{t}$ of t in L, Spoiler can pick a node $y_{\hat{t}}$ of t (and of $\hat{t}$) such that $\hat{t}(y_{\hat{t}}) = \delta_{\hat{t}}$ is not the maximal element of the respective Δ_k, moreover, Spoiler wins the $(n-1)$-round game on any pair $(s|_x, t|_{y_{\hat{t}}})$ with $\hat{s}(x) = \delta_{\hat{t}}$. Applying the induction hypothesis we get that for any maximal relabeling $\hat{t}$ and node x of s with $\hat{s}(x) = \delta_{\hat{t}}$, there exists an XTL($\mathcal{L}$)-formula $\varphi_{\hat{t},x}$ of depth at most $n-1$ separating $s|_x$ and $t|_{y_{\hat{t}}}$, say $s|_x \models \varphi_{\hat{t},x}$ and $t|_{y_{\hat{t}}} \not\models \varphi_{\hat{t},x}$.

Now let us define the formula

$$\varphi_\delta = \bigwedge_{\delta_{\hat{t}}=\delta} \bigvee_{\hat{s}(x)=\delta} \varphi_{\hat{t},x}$$

for each $\delta \in \Delta$ that is not the maximal element of the respective Δ_k, where $\hat{t}$ ranges over the maximal relabelings of t in L and x ranges over the nodes of s. Moreover, whenever δ is the maximal element of the respective Δ_k, let φ_δ be the formula $\uparrow$. Finally, let φ stand for the XTL($\mathcal{L}$)-formula $L(\delta \mapsto \varphi_\delta)_{\delta\in\Delta}$.

We claim that $t \not\models \varphi$. Indeed,

$t \models \varphi \Leftrightarrow$ the characteristic tree t' of t determined by $(\varphi_\delta)_{\delta\in\Delta}$ is in L
$\Leftrightarrow$ for some relabeling $\hat{t} \in L$ of t we have $t|_y \models \varphi_{\hat{t}(y)}$ for all nodes y of t such that $\hat{t}(y)$ is not the maximal element of the respective Δ_k (since L is downwards closed)
$\Leftrightarrow$ for some maximal relabeling $\hat{t} \in L$ of t we have $t|_y \models \varphi_{\hat{t}(y)}$ for all nodes y of t such that $\hat{t}(y)$ is not the maximal element of the respective Δ_k.

However, the latter is clearly not possible. Indeed, suppose that $t|_{y_{\hat{t}}} \models \varphi_{\delta_{\hat{t}}}$. Then, $t|_{y_{\hat{t}}} \models \bigvee_{\hat{s}(x)=\delta_{\hat{t}}} \varphi_{\hat{t},x}$, and thus $t|_{y_{\hat{t}}} \models \varphi_{\hat{t},x}$ for some node x of s with $\hat{s}(x) = \delta_{\hat{t}}$, contradicting the definition of the formulas $\varphi_{\hat{t},x}$.

We also claim that $s \models \varphi$. Since L is downwards closed and $\hat{s}$ is in L, it suffices to show that $s' \leq_\Delta \hat{s}$, where s' is the characteristic tree of s determined by the family $(\varphi_\delta)_{\delta\in\Delta}$. Thus, it suffices to show that for any node z of s for which $\hat{s}(z) = \delta$ is not the maximal element of the respective Δ_k, we have $s|_z \models \varphi_\delta$ (implying $s'(z) \leq_\Delta \hat{s}(z)$). This is clear, since for any relabeling $\hat{t}$ with $\delta_{\hat{t}} = \delta$ we have $s|_z \models \varphi_{\hat{t},z}$ by the definition of the formulas $\varphi_{\hat{t},z}$.

Hence the XTL($\mathcal{L}$)-formula φ of depth at most n separates s and t, thus $s \not\equiv^n_{\mathcal{L}} t$ and the statement is proved. □

Propositions 9 and 10 imply the following characterization:

Theorem 5. *Suppose $\mathcal{L}$ is a class of downwards closed tree languages. Then for any $n \geq 0$ and trees $s, t \in T_\Sigma$, Duplicator has a winning strategy on (s, t) in the modified n-round* XTL($\mathcal{L}$)*-game if and only if s and t satisfy the same set of* XTL($\mathcal{L}$)*-formulas of depth at most n. Consequently, if $\mathcal{L}$ is finite, then for any tree language L, $L \in$* **XTL**($\mathcal{L}$) *if and only if there exists some $n \geq 0$ such that Spoiler has a winning strategy in the modified n-round* XTL($\mathcal{L}$)*-game on any pair (s, t) of trees with $s \in L$ and $t \notin L$.*

It is possible to combine the FTL($\mathcal{L}$)-game and the modified XTL($\mathcal{L}$)-game. We call the resulting game the *modified n-round* FTL($\mathcal{L}$)*-game*. A characterization theorem similar to the previous ones again holds:

Theorem 6. *Suppose $\mathcal{L}$ is a class of downwards closed tree languages. Then for any $n \geq 0$ and trees $s, t \in T_\Sigma$, Duplicator has a winning strategy on (s, t) in the modified n-round* FTL($\mathcal{L}$)*-game if and only if s and t satisfy the same set of* FTL($\mathcal{L}$)*-formulas of depth at most n. Consequently, if $\mathcal{L}$ is finite, then for any tree language L, $L \in$* **FTL**($\mathcal{L}$) *if and only if there exists some $n \geq 0$ such that Spoiler has a winning strategy in the modified n-round* FTL($\mathcal{L}$)*-game on any pair (s, t) of trees with $s \in L$ and $t \notin L$.*

7. Examples

Recall the definition of the ranked alphabet Bool from Sec. 3, paragraph 2.

Example 3. Let L_{EF^+} and L_{EF^*} denote the Bool-tree languages of those trees having a *non-root* node labeled in UP, and *any* node labeled in UP, respectively. Then the logics FTL($\{L_{\mathrm{EF}^+}\}$) and FTL($\{L_{\mathrm{EF}^*}\}$) are related to the fragments of CTL[2] determined by the strict and non-strict existential future modalities. The modified n-round FTL($\{L_{\mathrm{EF}^+}\}$)-game and FTL($\{L_{\mathrm{EF}^*}\}$)-game have the same rules as the corresponding games described in [24]. (Observe that L_{EF^+} and L_{EF^*}

[2]CTL was originally introduced in [5] as a logic on Kripke structures, or infinite (unranked) trees. Regarding the definition of CTL on finite trees as used here, cf. [6].

are downwards closed.) It is shown in the papers [4], [9], [24] (using in part different arguments), that it is decidable for a regular tree language whether it is definable in these logics. For fragments of CTL involving the next modality and the strict or non-strict existential future modality, we refer to [4], [7].

Example 4. Let $L_{\rm EG} \subseteq T_{\rm Bool}$ consist of the Bool-trees having a maximal path p such that each node of p is labeled in UP. Then the logic FTL($\{L_{\rm EG}\}$) corresponds to the (non-strict) EG fragment of CTL. The modified n-round FTL($\{L_{\rm EG}\}$)-game characterizing this logic has the following rules, when played on the pair of trees (s, t):

(1) If Root(s) $\neq$ Root(t), Spoiler wins. Otherwise Step 2 follows.
(2) If $n = 0$, Duplicator wins. Otherwise Step 3 follows.
(3) Spoiler chooses one of the trees, say s, and a leaf node x of s (and thus selects a maximal path of s).
(4) Duplicator chooses a leaf node y of t.
(5) Spoiler chooses a (not necessarily strict) ancestor y' of y.
(6) Duplicator chooses a (not necessarily strict) ancestor x' of x.
(7) An $(n-1)$-round FTL($\{L_{\rm EG}\}$)-game is played on $(s|_{x'}, t|_{y'})$. The player winning the subgame also wins the whole game.

Example 5. Recall from Example 1 the definition of $L_{\rm even}$. This language is *not* downwards closed. Let $\mathcal{L} = \{L_{\rm even}\}$. The n-round FTL($\mathcal{L}$)-game characterizes the modular temporal logic FTL($\mathcal{L}$). The rules of this game on the pair (s, t) of trees are formulated as follows:

(1) If Root(s) $\neq$ Root(t), Spoiler wins. Otherwise Step 2 follows.
(2) If $n = 0$, Duplicator wins. Otherwise Step 3 follows.
(3) Spoiler marks an even number of nodes of one tree, and an odd number of nodes of the other tree. After that, Step 4 follows.
(4) Duplicator chooses a marked node x and an unmarked node y, either in the same tree or in different trees, and an $(n-1)$-round FTL($\mathcal{L}$)-game is played on the subtrees rooted in x and y. If he cannot do so, Spoiler wins. The player winning the subgame also wins the game.

The question whether **FTL**($\mathcal{L}$) is decidable when the rank type R contains an integer greater than 1 is open. For the classical case $R = \{0, 1\}$, see [2], [22].

Suppose Step 1 above gets replaced by the following:

(1') If for some $\sigma \in \Sigma$ exactly one of the trees contains an even number of nodes labeled σ, Spoiler wins. Otherwise Step 2 follows.

The resulting game characterizes the (weaker) modular logic XTL($\mathcal{L}$), where the root node is not distinguished from the other nodes.

Example 6. Consider the following n-round game on the pair of trees (s, t):

(1) If $\mathrm{Root}(s) \neq \mathrm{Root}(t)$, Spoiler wins. Otherwise Step 2 follows.

(2) If $n = 0$, Duplicator wins. Otherwise Step 3 follows.

(3) Spoiler chooses either to make an EX-move, in which case Step 4 follows, or an EU-move, in which case Step 5 follows.

(4) (EX-move.) Spoiler chooses one of the trees, say s, and a node x of s of depth one. If he cannot do so, Duplicator wins. Otherwise, Duplicator chooses a node y of t of depth one (if he cannot, he immediately loses), and an $(n-1)$-round game is played on the trees $(s|_x, t|_y)$. The player winning the subgame also wins the whole game.

(5) (EU-move.) Spoiler chooses one of the trees, say s, and a node x of s. After that, Duplicator chooses a node y of t. Then, Spoiler again can make a decision to continue the game either with the pair of trees $(s|_x, t|_y)$, or with $(s|_{x'}, t|_{y'})$, where x' is a strict ancestor of x and y' is a strict ancestor of y.

(6) In the first case, an $(n-1)$-round game is played on $(s|_x, t|_y)$ and the winner of the subgame wins the game.

(7) In the second case, Spoiler chooses a strict ancestor y' of y, after which Duplicator chooses a strict ancestor x' of x. (If someone cannot choose such a node, the other player wins.) Then, an $(n-1)$-round game is played on $(s|_{x'}, t|_{y'})$. The winner of the subgame also wins the whole game.

This game (resulting from Theorem 6) characterizes the temporal logic CTL: a tree language L is definable in CTL if and only if there exists an integer $n \geq 0$ such that Spoiler wins the n-round game on any pair (s, t) of trees with $s \in L$ and $t \notin L$.

When $R = \{0, 1\}$, our game is similar to the one described in [12] for words. (See also [15] for a similar game for Mazurkiewicz traces). It is also closely related to the game developed for full CTL (over Kripke structures) in [1].

References

[1] M. Adler and N. Immerman, An $n!$ lower bound on formula size, *ACM Trans. Comput. Log.* **4** (2003), 296–314.

[2] A. Baziramwabo, P. McKenzie and D. Thérien, Modular Temporal Logic, In: 14th Symposium on Logic in Computer Science (Trento, 1999), 344–351, *IEEE Computer Soc., Los Alamitos, CA*, 1999.

[3] M. Benedikt and L. Segoufin, Regular Tree Languages Definable in FO, STACS 2005, Lecture Notes in Comput. Sci., 3404, *Springer, Berlin*, 2005.

[4] M. Bojańczyk and I. Walukiewicz, Characterizing EF and EX tree logics, *Theoret. Comput. Sci.* **358**, no. 2–3 (2006), 255–272.

[5] E. A. Emerson and E. M. Clarke, Using branching time temporal logic to synthesize synchronization skeletons, *Sci. Comput. Programming* **2**, no. 3 (1982), 241–266.

[6] Z. Ésik, Characterizing CTL-like logics on finite trees, *Theoret. Comput. Sci.* **356**, no. 1–2 (2006), 136–152.

[7] Z. Ésik, An Algebraic Characterization of Temporal Logics on Finite Trees, Parts 1, 2, 3, In: 1st International Conference on Algebraic Informatics, Thessaloniki, *Aristotle University of Thessaloniki*, 2005.

[8] Z. Ésik and Sz. Iván, Products of tree automata with an application to temporal logic, *Fund. Inform.* **82**, no. 1–2 (2008), 61–78.

[9] Z. Ésik and Sz. Iván, Some varieties of finite tree automata related to restricted temporal logics, *Fund. Inform.* **82**, no. 1–2 (2008), 79–103.

[10] Z. Ésik and Sz. Iván, Games for Temporal Logics on Trees, In: CIAA 2008, proceedings, Lecture Notes in Comput. Sci. 5148, *Springer*, 2008.

[11] Z. Ésik and P. Weil, Algebraic characterization of logically defined tree languages, Int. J. Algebra and Computation, (An extended abstract appeared as: Z. Ésik and P. Weil. On logically defined recognizable tree languages, in FST TCS 2003 Lecture Notes in Comput. Sci. 2914, 195–207, Springer, 2003) (*to appear*).

[12] K. Etessami and Th. Wilke, An Until Hierarchy for Temporal Logic, In: 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, *IEEE Computer Society Press*, 1996.

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

[14] U. Heuter, First-order properties of trees, star-free expressions, and aperiodicity, In: STACS 88, Bordeaux, Lecture Notes in Comput. Sci. 294, *Springer*, 1988.

[15] J. G. Henriksen, An expressive extension of TLC, *Internat. J. Found. Comput. Sci.* **13**, no. 3 (2002), 341–360.

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

[17] A. Potthoff, Modulo-counting quantifiers over finite trees, *Theoret. Comput. Sci.* **126**, no. 1 (1994), 97–112.

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

[19] M. Steinby, Syntactic Algebras and Varieties of Recognizable Sets, In: Les arbres en algèbre et en programmation (4éme Colloq., 1979), Lille, *Univ. Lille I*, 1979.

[20] M. Steinby, General varieties of tree languages, *Theoret. Comput. Sci.* **205**, no. 1–2 (1998), 1–43.

[21] S. Salehi and M. Steinby, Tree algebras and varieties of tree languages, *Theoret. Comput. Sci.* **377**, no. 1–3 (2007), 1–24.

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

[23] W. Thomas, Logical aspects in the study of tree languages, In: Ninth colloquium on trees in algebra and programming, Bordeaux, *Cambridge University Press, Cambridge*, 1984.

[24] Z. Wu, A note on the characterization of TL(EF), *Inform. Process. Lett.* **102** (2007), 48–54.

ZOLTÁN ÉSIK
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SZEGED
HUNGARY

E-mail: ze@inf.u-szeged.hu

SZABOLCS IVÁN
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SZEGED
HUNGARY

E-mail: szabivan@inf.u-szeged.hu

(Received August 30, 2009; revised January 31, 2010)