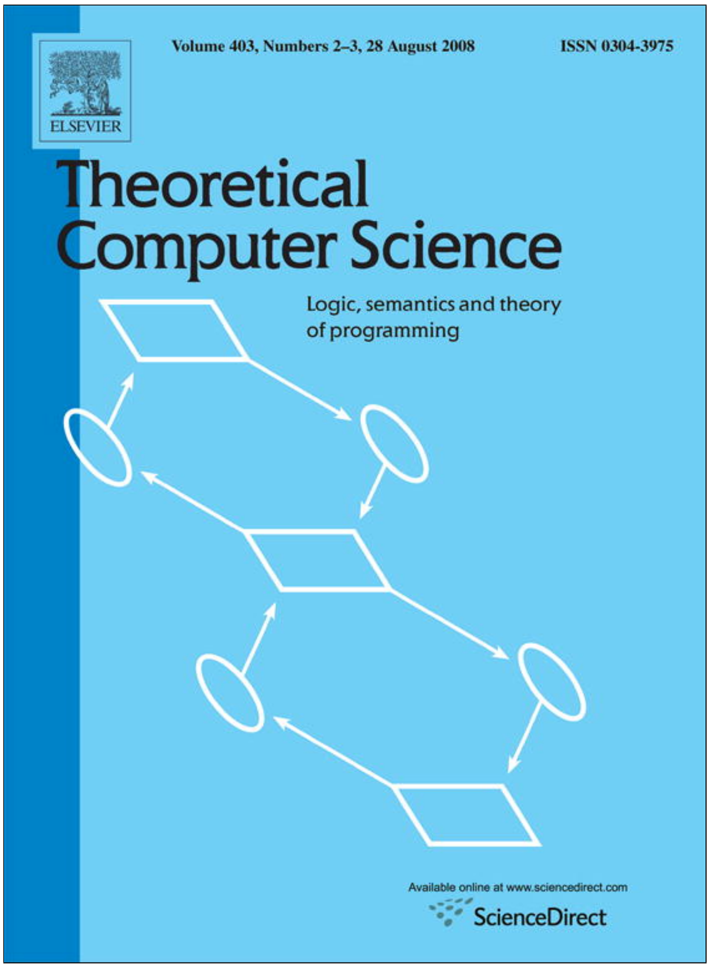
Volume 403, Numbers 2–3, 28 August 2008
ISSN 0304-3975
ELSEVIER
Theoretical Computer Science
Logic, semantics and theory of programming
Available online at www.sciencedirect.com
ScienceDirect

Theoretical Computer Science 403 (2008) 373–381

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Arity hierarchy for temporal logics

Alexander Rabinovich *

Department of Computer Science, Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, 69978, Israel

ARTICLE INFO

Article history:
Received 25 December 2007
Received in revised form 30 May 2008
Accepted 4 June 2008
Communicated by A. Avron

Keywords:
Temporal logic
Expressive power

ABSTRACT

A major result concerning temporal logics is Kamp's Theorem which states that the pair of modalities "until" and "since" is expressively complete for the first-order fragment of the monadic logic over the linear-time canonical model of naturals.

The paper concerns the expressive power of temporal logics over trees. The main result states that in contrast to Kamp's Theorem, for every n there is a modality of arity n definable by a monadic logic formula, which is not equivalent over trees to any temporal logic formula which uses modalities of arity less than n. Its proof takes advantage of an instance of Shelah's composition theorem.This result has interesting corollaries, for instance reproving that CTL^* and $ECTL^+$ have no finite basis.

1. Introduction

Various temporal logics have been proposed for reasoning about the so-called "reactive" systems, computer hardware or software systems which exhibit (potentially) a non-terminating and a non-deterministic behavior. A system of this kind is typically represented by (potentially) infinite sequences of computation states through which it may evolve, where we associate with each state the set of atomic propositions which are true in that state, along with the possible next state transitions to which it may evolve. Thus, its behavior is denoted by a (potentially) infinite rooted tree, with the initial state of the system represented by the root of the tree.

Temporal Logic (*TL*) introduced to Computer Science by Pnueli in [20] is a convenient framework for the specification properties of "reactive" systems. This made temporal logics a popular subject in the Computer Science community and it has enjoyed an extensive research in the last 30 years. In temporal logic the relevant properties of the system are described by *atomic propositions* that hold at some points in time, but not at others. More complex properties are described by the formulae built from the atoms using Boolean connectives and *Modalities* (temporal connectives): a k-place modality C transforms statements $\varphi_1, \ldots, \varphi_k$ on points possibly other than the given point t_0 to a statement $C(\varphi_1, \ldots, \varphi_k)$ on the point t_0. The rule that specifies when the statement $C(\varphi_1, \ldots, \varphi_k)$ is true for the given point is called a *Truth Table*. The choice of the particular modalities with their truth tables determines the different temporal logics. A Temporal Logic with modalities $M_1, \ldots, M_k$ is denoted by $TL(M_1, \ldots, M_k)$.

The most basic modality is the one-place modality $\mathsf{F}X$ saying: "X holds some time in the future". Its truth table is usually formalized by $\varphi_{\mathsf{F}}(t_0, X) \equiv (\exists t > t_0) t \in X$. This is a formula of the Monadic Logic of Order (*MLO*). The Monadic Logic of Order is a fundamental formalism in Mathematical Logic. Its formulae are built using atomic propositions $t \in X$, atomic relations between elements $t_1 = t_2$, $t_1 < t_2$, Boolean connectives, first-order quantifiers $\exists t$ and $\forall t$, and second-order (set) quantifiers $\exists X$ and $\forall X$. Nearly all the modalities used in the literature have their truth tables defined in *MLO*, and as a result every formula of a temporal logic translates directly into an equivalent formula of *MLO*. Therefore, the different temporal logics may be considered as a convenient way to use fragments of *MLO*. *MLO* can also serve as a yardstick by which to check

* Tel.: +972 36405388.
E-mail address: rabinoa@post.tau.ac.il.

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2008.06.006

the strength of the temporal logic chosen: a temporal logic is *expressively complete* for a fragment L of *MLO* if every formula of L with a single free variable t_0 is equivalent to a temporal formula.

Actually, the notion of expressive completeness refers to a temporal logic and to a model (or a class of models) since the question whether two formulae are equivalent depends on the domain over which they are evaluated. Any (partially) ordered set with monadic predicates is a model for *TL* and *MLO*, but the main, *canonical*, linear-time intended models are the non-negative integers $\langle N, <\rangle$ for discrete time and the non-negative reals $\langle R^+, <\rangle$ for continuous time.

A major result concerning *TL* is Kamp's theorem [16,12,11], which states that the pair of modalities "X *until* Y" and "X *since* Y" is expressively complete for the first-order fragment of *MLO* over the above two linear-time canonical models.

Kamp's theorem is about temporal logics over linear structures, called *linear-time* logics, but many popular temporal logics, called *branching-time* logics [17,9], view time as a tree-like set of time points, and are correspondingly interpreted over tree-like partially ordered structures.

Many branching-time logics have been proposed, starting with [17,2,3,21,4,6,8,9,29]. The basic modalities of these logics are obtained by combining a path quantifier "E" or "A" with a formula in $TL(\mathsf{U})$ ($TL(\mathsf{U})$ is the temporal logic with the until modality U). The formula $\mathsf{E}\phi$ (respectively $\mathsf{A}\phi$) holds at time point t_0 if for some path (respectively, for every path) π starting at t_0 the $TL(\mathsf{U})$ formula ϕ holds along π. For example, a commonly used branching-time logic is *CTL* [3,5], based on the two binary modalities EU and AU.

Two extensions of *CTL*, namely *ECTL* and $ECTL^+$, have been proposed to deal with fairness properties [9]. *ECTL* is $TL(\mathsf{EU}, \mathsf{AU}, \mathsf{EF}^\infty)$ where $\mathsf{F}^\infty p$ reads "p holds infinitely often in the future". $ECTL^+$ is more expressive since it allows $\mathsf{E}\phi$ for any formula ϕ with no nested modalities in $TL(\mathsf{U}, \mathsf{F}^\infty)$. The basic modalities of *CTL* and *ECTL* are unary or binary; however, the basic modalities of $ECTL^+$ have arbitrary arities. Finally, the logic CTL^*, from [9], is obtained by considering an infinite set of modalities: $\mathsf{E}\phi$ for any formula ϕ in $TL(\mathsf{U})$.

The expressive power of the first-order *MLO* over the trees is very limited. For instance, a very basic property "along all futures, eventually p" (that is, "p is inevitable") is not expressible in the first-order *MLO*. However all the modalities of the above logics have their truth tables defined in the second-order *MLO*.

Our main result states that in contrast to the Kamp theorem, for every n there is a modality of arity n (definable by an *MLO* formula) which is not equivalent over trees to any temporal logic formula which uses modalities of arity less than n.

The paper is organized as follows. In the next section we review the basic definitions about monadic logic of order and its fragments. In Section 3 we review the basic definitions about temporal logics and modalities. In Section 4 we state our results. Section 5 contains proofs. Section 6 compares our results with related works and states some open problems.

2. Monadic logic of order

Monadic logic of order is a fundamental formalism in mathematical logic and in the theory of computation. In this section we recall the basic definitions about monadic logic of order; we also define its important fragments.

2.1. Notations

We use standard notations and abbreviations. A (relational) signature is given by a set of relational symbols and their arity. Let A be a structure for a signature τ. We use $|A|$ for the universe of A and R^A for the interpretation of the relational symbol R in A. Whenever there is no confusion we will also use A for the universe of A; sometimes, we use "$a \in A$" instead of "$a \in |A|$" and "$\langle a_1, \ldots, a_n\rangle \in R$" instead of "$\langle a_1, \ldots, a_n\rangle \in R^A$". For a structure A over a signature $\tau = \{\ldots, R_i, \ldots\}$ we use notations $\langle |A|, \ldots, R_i^A, \ldots\rangle$ which we also abbreviated to $\langle |A|, \ldots, R_i, \ldots\rangle$.

2.2. Syntax

The syntax of the second-order Monadic Logic of Order (*MLO*) has in its vocabulary individual first-order variables $x_0, x_1, x_2, \ldots$, set variables $X_0, X_1, X_2, \ldots$, and set constants (monadic predicate's names).

The atomic formulae are of the form $x_1 = x_2$, $x_1 < x_2$, $x \in X$ and $x \in P$, where x_i (respectively, X and P) ranges over individual variables (respectively, set variables and monadic predicate's names). Formulae are built up from the atomic formulae using the propositional connectives $\wedge$ and $\neg$, and the quantifiers $\exists x$ and $\exists X$.

We denote by *FOMLO* the subset of *first-order formulae of MLO*, i.e., formulae where the second-order quantifier $\exists X$ does not occur. Note that the formulae of this fragment may contain free set variables.

We shall write $\phi(x_1, x_2, \ldots, x_k, X_1, X_2, \ldots, X_m)$ to indicate that the free variables of ϕ are among $x_1, x_2, \ldots, x_k, X_1, X_2, \ldots, X_m$.

The *quantifier-depth of a formula* ϕ, denoted by $\mathrm{qd}(\phi)$, is defined as usual: $\mathrm{qd}(\phi) = 0$ for atomic formulae; $\mathrm{qd}(\phi \wedge \phi') = \max(\mathrm{qd}(\phi), \mathrm{qd}(\phi'))$; $\mathrm{qd}(\neg\phi) = \mathrm{qd}(\phi)$; and $\mathrm{qd}(\exists x\phi) = \mathrm{qd}(\exists X\phi) = 1 + \mathrm{qd}(\phi)$.

2.3. Semantics

The semantics of *MLO* follows classical lines. A structure for *MLO* is a tuple $M = \langle |M|, <, \mathcal{P}_1, \ldots, \mathcal{P}_n, \ldots\rangle$ where $<$ is a partial order over a set $|M|$ and $\mathcal{P}_1, \ldots, \mathcal{P}_n, \ldots \subseteq |M|$ are monadic predicates. If M is such a structure, $s_1, \ldots, s_m \in |M|$

are elements of M and $S_1, \ldots, S_n \subseteq |M|$ are sets of elements, we write

$$(M, s_1, s_2, \ldots, s_m, S_1, S_2, \ldots, S_n) \models \phi(x_1, x_2, \ldots, x_m, X_1, X_2, \ldots, X_n)$$

if the formula ϕ is satisfied in the structure M with x_i interpreted as s_i ($i = 1, \ldots, m$) and X_j interpreted as S_j ($j = 1, \ldots, n$). The definition is a standard one, so we omit it.

We will be mainly interested in partial orders which are tree orders.

A structure $T = \langle |T|, <^T \rangle$ is a *tree* if $<^T$ is a binary relation such that

(1) The set $|T|$ is partially ordered by $<^T$.
(2) There is a unique $<^T$ minimal element.
(3) For every element $a \in |T|$ the set $\{b \in |T| \,:\, b <^T a\}$ is finite and $<^T$ is a linear order on this set.

The elements of $|T|$ are called *nodes* of the tree (sometimes we call them *time points*). The minimal element is denoted by ε_T or by $root_T$, and referred to as the *root of the tree*. A node s is an *ancestor* of a node s' in T if $s \leq^T s'$. A node s is a *successor* (in T) of a node s' if $s' <^T s$ and there is no element between s and s'.

A structure $(|T|, <^T, \ldots, P_i^T, \ldots)$ is a *labelled or computation tree* if $\langle |T|, <^T \rangle$ is a tree and P_i^T are unary predicates (subsets of $|T|$). We say that a node $s \in |T|$ is *labelled by P_i* if $s \in P_i^T$.

When s is a node in a computation tree T, we write $T_{\geq s}$ to denote the *subtree of T rooted at s*. Formally, the nodes of $T_{\geq s}$ are $|T_{\geq s}| \triangleq \{t : t \in |T| \text{ and } t \geq s\}$, P_i is interpreted as $P_i^T \cap |T_{\geq s}|$ and $<$ is interpreted as $<^T \cap |T_{\geq s}| \times |T_{\geq s}|$.

A *path through T starting at $s_1 \in |T|$* is a maximal linearly ordered sequence of successive nodes $\pi = \langle s_1, s_2, s_3, \ldots \rangle$ through the tree. A path π through T induces a substructure, denoted by T_π; the set of nodes of T_π is $\{s_1, s_2, \ldots\}$, s is labelled by P_i in T_π iff s is labelled by P_i in T, and s is an ancestor of s' in T_π iff $s \leq^T s'$.

2.4. Future formula

Definition 2.1 (*Future Formula*). A formula $\phi(x_0, X_1, \ldots, X_k)$ of *MLO* with one free first-order variable x_0 is a *future formula* if for every tree T and node $s \in |T|$, and for all subsets $S_1, \ldots, S_k$ of $|T|$, the following holds:

$$T, s, S_1, \ldots, S_k \models \phi \quad \text{iff} \quad T_{\geq s}, s, S'_1, \ldots, S'_k \models \phi$$

where, for $i = 1, \ldots, k$, S'_i is the restriction of S_i to $|T_{\geq s}|$.

In other words, a future formula is a formula with one free individual variable x_0 whose value depends only on nodes higher than x_0 in the tree. Observe that this is a semantic notion, not a syntactic one.

Remark 2.2 (*Syntactical Conditions for the Property to be a Future Formula*). Let $\phi(X_1, \ldots, X_k)$ be a formula without free first-order variable. Let $\tilde{\phi}$ be obtained from ϕ by relativizing all first-order quantifiers to the elements greater than or equal to x_0, i.e., when "$\exists x. \ \ldots$" and "$\forall x. \ \ldots$" are replaced by "$\exists x(x \geq x_0 \wedge \cdots)$" and by "$\forall x(x \geq x_0 \rightarrow \cdots)$", respectively. Note that the formula $\tilde{\phi}$ obtained in such a way is always a future formula. Moreover, any $\psi(x_0)$ is a future formula if and only if it is equivalent to a formula $\tilde{\phi}(x_0)$ with all quantifiers relativized to the elements greater than or equal to x_0.

3. Temporal logics

In this section, we recall the syntax and semantics of temporal logics and how temporal modalities are defined using *MLO* truth tables.

3.1. Temporal logics and modalities

The syntax of Temporal Logic (*TL*) has in its vocabulary a set of variables (sometimes called propositional names) and a set B of modality names (sometimes called "temporal connectives" or "temporal operators") with prescribed arity $B = \{\#_1^{l_1}, \#_2^{l_2}, \ldots\}$ (we usually omit the arity notation). The set of modality names B might be infinite. A temporal logic based on a set of modalities B is denoted $TL(B)$; B is called the *basis* of $TL(B)$. Atomic temporal formulae are just variables and other formulae are obtained from the atoms using Boolean connectives and applying the modalities. Formally, the syntax of $TL(B)$ is given by the following grammar:

$$\phi \ ::= \ P \ \mid \ \phi_1 \wedge \phi_2 \ \mid \ \neg\phi_1 \ \mid \ \#_i^{l_i}(\phi_1, \phi_2, \ldots, \phi_{l_i}), \quad \text{where } P \text{ ranges over the variable names.}$$

The *nesting-depth* of a temporal formula ϕ, denoted by $\mathrm{nd}(\phi)$, is defined as usual: $\mathrm{nd}(\phi) = 0$ for atomic formulae; $\mathrm{nd}(\phi \wedge \phi') = \max(\mathrm{nd}(\phi), \mathrm{nd}(\phi'))$; $\mathrm{nd}(\neg\phi) = \mathrm{nd}(\phi)$; and $\mathrm{nd}(\#_i^{l_i}(\phi_1, \phi_2, \ldots, \phi_{l_i})) = 1 + \max_{1 \leq j \leq l_i}(\mathrm{nd}(\phi_j))$.

Temporal formulae are interpreted over partially ordered sets with monadic predicates, in particular over computation trees and over labelled chains. Every modality $\#^l$ is interpreted in every structure T as an operator

$$\#_T^l : [\mathcal{P}(|T|)]^l \rightarrow \mathcal{P}(|T|)$$

which assigns "the set of points where $\#^l(Q_1,\ldots,Q_l)$ holds" to the l-tuple $\langle Q_1,\ldots,Q_l\rangle$. (Here $\mathcal{P}$ is the power set notation, and $\mathcal{P}(|T|)$ denotes the set of all subsets of the universe of T.)

Formally, we define when a temporal formula ϕ holds in a node s of a structure $T = \langle |T|, \ <^T,\ P_1^T,\ldots,P_n^T,\ldots\rangle$, written $T, s \models \phi$, by the following inductive clauses:

(1) For atomic formulae $T, s \models P_i$ iff $s \in P_i^T$.
(2) The semantics of Boolean combinations is defined as usual, and
(3) The semantics of modalities is defined by:

$$T, s \models \#^l(\phi_1,\phi_2,\ldots,\phi_l) \quad \text{iff} \quad s \in \#^l_T(R_{\phi_1}, R_{\phi_2},\ldots,R_{\phi_l})$$

where $R_{\phi_i} = \{a : T, a \models \phi_i\}$ for all i, $1 \le i \le l$.

Notes.

(1) In temporal and modal logics, formulae are constructed from atoms by applying Boolean connectives and modalities. Formalisms like *MLO* and μ-calculus can specify properties of trees. However, they use binding, quantifiers, fixed-points; hence, they are not temporal logics according to our definition.
(2) Strictly speaking, what we call temporal logic is called *one-dimensional temporal logic*. The syntax of k-dimensional temporal logic is the same as that of one-dimensional temporal logic. However, atomic formulae (variables) are interpreted as k-ary relations; accordingly, every l-place modality $\#^l$ is interpreted in every structure T as an operator which assigns a k-ary relation (over $|T|$) to every l-tuple of k-ary relations [11].

For a class $\mathcal{C}$ of computation trees, we say that two temporal formulae ϕ_1 and ϕ_2 are equivalent over $\mathcal{C}$, written $\phi_1 \equiv_{\mathcal{C}} \phi_2$, when $T, s \models \phi_1$ iff $T, s \models \phi_2$ for all $T \in \mathcal{C}$ and $s \in |T|$. Given two temporal logics TL_1 and TL_2, we say that TL_1 is as expressive as TL_2 over $\mathcal{C}$, written $TL_2 \le_{\mathcal{C}} TL_1$, when every formula ϕ_2 in TL_2 has a $\mathcal{C}$-equivalent formula in TL_1. When both $TL_1 \le_{\mathcal{C}} TL_2$ and $TL_2 \le_{\mathcal{C}} TL_1$ hold, we say that the two logics are *expressively equivalent* over $\mathcal{C}$, written $TL_1 \equiv_{\mathcal{C}} TL_2$. We usually omit mentioning $\mathcal{C}$ when we consider the class of all computation trees.

When a TL_1 formula ϕ is equivalent to some TL_2 formula ϕ', we say that ϕ *can be expressed in* TL_2. If ϕ has the form $\mathsf{H}(P_1,\ldots,P_l)$, we say that *the modality* H can be expressed in TL_2.

We say that a temporal logic L *has* (or *admits*) *a finite basis* if there is a finite set of modalities $\#_1,\ldots,\#_k$ such that L is expressively equivalent to $TL(\#_1,\ldots,\#_k)$.

3.2. Truth tables

There is not much interest in a modality # whose interpretation is arbitrary and is defined ad hoc in each and every structure where the temporal logic is to be interpreted. To be of interest a modality needs to have a uniform description in some metalanguage that connects the set of points where $\#(\phi_1,\ldots,\phi_k)$ holds to the sets of points where each of the ϕ_i holds. It is an empirical fact that all the temporal modalities considered in the literature are defined in *MLO* in the following way: for every l-place modality # there is a formula $\bar{\#}(x_0, X_1, X_2,\ldots,X_l)$ of *MLO* with one free first-order variable x_0 and l set variables, such that for every structure T and subsets $R_i \subseteq |T|$:

$$\#_T(R_1, R_2,\ldots,R_l) = \{s : (T, s, R_1, R_2,\ldots,R_l) \models \bar{\#}(x_0, X_1, X_2,\ldots,X_l)\}.$$

The formula $\bar{\#}$ will be called the *truth table* of this modality $\#_T$. Let M be a temporal modality defined by a formula $\phi_M \in MLO$ serving as a truth table. We say that M has the quantifier-depth k if $\mathrm{qd}(\phi_M) = k$.

Example 3.1 (*Some Common Modalities and their Truth Tables*).

- The one-place modality F ("*eventually*"); its truth table is

$$\phi_{\mathsf{F}}(x_0, X_1) \triangleq \exists y(y > x_0 \ \wedge\ y \in X_1).$$

- The one-place modality X ("*next*"); its truth table is

$$\phi_{\mathsf{X}}(x_0, X_1) \triangleq \exists y\big(y > x_0 \ \wedge\ y \in X_1 \ \wedge\ \forall z(z < y \ \rightarrow\ z \le x_0)\big).$$

- The two-place modality U ("*until*"); its truth table is

$$\phi_{\mathsf{U}}(x_0, X_1, X_2) \triangleq \exists y\big(y > x_0 \ \wedge\ y \in X_2 \ \wedge\ \forall z(x_0 < z < y \ \rightarrow\ z \in X_1)\big).$$

In the literature a "nonstrict" definition of Until is sometimes given: the "nonstrict until" U_{ns} modality has the truth table

$$\phi_{\mathsf{U}_{\mathrm{ns}}}(x_0, X_1, X_2) \triangleq \exists y\big(y \ge x_0 \ \wedge\ y \in X_2 \ \wedge\ \forall z(x_0 \le z < y \ \rightarrow\ z \in X_1)\big).$$

Clearly, U_{ns} can be defined using U:

$$X_1 \mathsf{U}_{\mathrm{ns}} X_2 \leftrightarrow X_2 \vee (X_1 \wedge (X_1 \mathsf{U} X_2)).$$

- The one-place modality F^∞ ("*infinitely often*"); its truth table is

$$\phi_{\mathsf{F}^\infty}(x_0, X_1) \triangleq \forall y\big(y > x_0 \ \rightarrow \ \exists z(z > y \ \wedge \ z \in X_1)\big).$$

  Note that this formula expresses "X occurs infinitely often" on the natural numbers' flow, not on reals. For the reals there is another truth table which expresses "*infinitely often*".
- The two-place modality S ("*since*"); its truth table is

$$\phi_{\mathsf{S}}(x_0, X_1, X_2) \triangleq \exists y\big(y < x_0 \ \wedge \ y \in X_2 \ \wedge \ \forall z(x_0 > z > y \ \rightarrow \ z \in X_1)\big).$$

The choice of the particular modalities with their truth tables determines the different temporal logics.

Most of the temporal logics studied in Computer Science use only modalities having truth tables definable by future *MLO* formulae (see Definition 2.1).

Definition 3.2 (*First-Order Future Modality*). A temporal modality M is a first-order future modality if its truth table is a future formula of *FOMLO*.

Second-order future modalities are defined similarly. The modalities defined in the above example F, G, X, U and F^∞ are first-order future modalities; the modality S is not a future modality.

The rest of this section is not needed for the proof of our main results. However, it is needed for the statement of the consequences of the main results and for comparison with related works.

For reasoning about the branching structure of computation trees, so-called *branching-time* temporal logics have been introduced, with *CTL* and *CTL** as main representatives. These temporal logics use special modalities whose truth tables start with a path quantifier, as we now explain.

Definition 3.3 (*Path Modalities*). For every first-order future formula $\phi(x_0, X_1, \ldots, X_l)$, we define an l-place path modality $\mathsf{E}\phi$ as follows:

$$T, a \models \mathsf{E}\phi \quad \text{if and only if there is a path } \pi \text{ from } a \text{ in } T, \text{ such that}^1 T_\pi, a \models \phi(x_0, X_1, \ldots, X_l).$$

The modality $\mathsf{E}\phi$ is said to be the path modality which corresponds to $\phi(x_0, X_1, \ldots, X_l)$.

If $\phi(x_0, X_1, \ldots, X_l)$ is a future *FOMLO* formula, the truth table of the path modality $\mathsf{E}\phi$ is easily formalized in *MLO* (it is also easily formalized in the monadic path logic which is a small fragment of *MLO*). When H is a first-order future modality with truth table ψ_{H}, we write EH for the path modality $\mathsf{E}\psi_{\mathsf{H}}$. Another modality is AH, defined by the equivalence

$$\mathsf{AH}(\phi_1, \ldots, \phi_l) \ \equiv \ \neg\mathsf{E}\neg\psi_{\mathsf{H}}(\phi_1, \ldots, \phi_l).$$

Example 3.4. (1) *CTL* is usually defined as $TL(\mathsf{EU}_{ns}, \mathsf{AU}_{ns}, \mathsf{EX}, \mathsf{AX})$, which is expressively equivalent to $TL(\mathsf{EU}, \mathsf{AU})$.

(2) *CTL** is the temporal logic with modalities $\mathsf{E}\psi$, where ψ is a $TL(\mathsf{U})$ formula. *CTL** is expressively equivalent to the temporal logic defined as $TL(B)$, where

$$B \stackrel{\text{def}}{=} \{\mathsf{E}\phi \mid \phi(x_0, X_1, \ldots, X_l) \text{ is a first-order future formula}\}.$$

Note that the standard definition of *CTL** [9] uses an interplay between state formulae (which correspond to genuine modalities) and path formulae (which play an auxiliary role) in order to generate infinitely many modalities by a finite syntax. However, the standard *CTL** is expressively equivalent to the temporal logic defined above [7,23].

(3) For $k = 1, 2, \ldots$, BTL_k is the temporal logic defined as $TL(B_k)$, where

$$B_k \stackrel{\text{def}}{=} \{\mathsf{E}\phi \mid \phi(x_0, X_1, \ldots, X_l) \text{ is a first-order future formula with } \mathrm{qd}(\phi) \le k\}.$$

(4) *ECTL*$^+$ is the temporal logic with modalities $\mathsf{E}\phi$, where $\phi(x_0, X_1, \ldots, X_l)$ is a Boolean combination of the $\psi_{\mathsf{F}^\infty}(x_0, X_i)$'s and the $\psi_{\mathsf{U}}(x_0, X_i, X_j)$'s. *ECTL*$^+$ is expressively equivalent to BTL_2 [24].

4. Main results

Let $\mathbb{M}_k(X_1, \ldots, X_k)$ be the modality such that $T, s \models \mathbb{M}_k(X_1, \ldots, X_k)$ iff there is a path from s such that for every $I \subseteq \{1, \ldots, k\}$ the formula $\bigwedge_{i \in I} X_i \wedge \bigwedge_{j \notin I} \neg X_j$ is satisfied infinitely often along this path. It is clear that $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ has a truth table in *MLO*; it is also definable in *CTL** and even in *ECTL*$^+$.

Here we prove that if a temporal logic uses only modalities of arity at most r definable by future *MLO* formulae, then $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ cannot be expressed in this temporal logic.

Definition 4.1 (*Logic TLAR*$_k$). For every $k \ge 1$, let $TLAR_k$ be the temporal logic based on all modalities of arity at most k defined by *MLO* future formulae.

---

1 Recall that T_π is the substructure of T over the set of nodes π (see Section 2.3).

Lemma 4.2 (*Main Lemma*). *For every k, the modality $\mathbb{M}_{k+1}$ is not expressible in $TLAR_k$.*

A top-down proof of the lemma is presented in Section 5.

The following consequences of the main lemma are immediate:

Theorem 4.3 (*Arity Hierarchy*). *$TLAR_{k+1}$ is strictly more expressive than $TLAR_k$.*

Corollary 1. (1) *Let M_k be the set of all modalities of arity at most k, which are definable by CTL* formulae. $TL(M_k)$ is strictly less expressive than $TL(M_{k+1})$ for every k.*
(2) *Assume that for infinitely many values of $k \in \mathbb{N}$ the modality $\mathbb{M}_{k+1}$ is expressible in a temporal logic L with all its modalities definable by future MLO formulae. Then L has no finite base.*

5. Proof of Lemma 4.2

It will be convenient to represent trees by Kripke structures.

A *Kripke structure* is a structure $\mathcal{M} = \langle |\mathcal{M}|, R, P_1, P_2, \ldots\rangle$ where $|\mathcal{M}|$ is a set of nodes, the P_i are subsets of $|\mathcal{M}|$, and $R \subseteq |\mathcal{M}|^2$ is a binary *transition* relation. When $(s, s') \in R$, we say that it is possible to move from s to s' in one step. A path π in $\mathcal{M}$ starting from s_0 is a maximal sequence $s_0, s_1, \ldots$ s.t. $(s_i, s_{i+1}) \in R$ for all i.

For our purposes, Kripke structures are mainly another way of presenting computation trees.

Definition 5.1 (*Unfolding*). For a node s_0 of a Kripke structure $\mathcal{M} = \langle |\mathcal{M}|, R, P_1, P_2, \ldots\rangle$, the tree $T_{\mathcal{M},s_0}$ obtained by *unfolding* $\mathcal{M}$ from s_0 is $\langle |T|, \leq, P'_1, P'_2, \ldots\rangle$ where

(1) $|T| = \{\sigma, \ldots\}$ is the set of all finite prefixes of paths from s_0,
(2) $\sigma \leq \sigma'$ iff σ is a prefix of σ', and
(3) $\sigma \in P'_i$ if the last node of σ is in P_i.

Hence $root_{T_{\mathcal{M},s_0}}$ is the sequence "s_0". A path starting from s in $\mathcal{M}$ directly yields a path in $T_{\mathcal{M},s}$ starting from the root.

Let ϕ be a future *MLO* formula. For every pair of nodes s_0 and s_1 and every σ which starts at s_0 and ends at s_1

$$T_{\mathcal{M},s_0}, \sigma \models \phi \quad \text{iff} \quad T_{\mathcal{M},s_1}, s_1 \models \phi.$$

Accordingly, for a future (*MLO* or a temporal logic) formula ϕ, we write $\mathcal{M}, s \models \phi$ when $T_{\mathcal{M},s}, s \models \phi$, agreeing with the standard interpretation of future temporal logics over Kripke structures.

Now we are going to describe a Kripke structure $\mathcal{M}$ which will be used in our proof that $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ cannot be expressed in $TLAR_r$.

Let Δ_0 be a set $\{q_0, \ldots, q_{2^{r+1}-1}\}$ of 2^{r+1} elements, and let $\Delta_1, \Delta_2, \ldots, \Delta_{2^{r+1}}$ be all subsets of Δ_0 with $2^{r+1} - 1$ elements.

We now define a Kripke structure $\mathcal{M} = \langle |\mathcal{M}|, \to, V_1, \ldots, V_{r+1}\rangle$:

(1) the nodes in $|\mathcal{M}|$ are all $\langle q, \Delta, l, j\rangle$ where Δ is a subset of Δ_0 with at least $2^{r+1} - 1$ elements, $q \in \Delta$ and $l, j \in \mathbb{N}$. Δ is the *support*, l is the *level* of $\langle q, \Delta, l, j\rangle$.
(2) In $\mathcal{M}$, node $\langle q_p, \Delta, l, j\rangle$ is in V_i if the ith digit in the binary representation of p is one.
(3) The transitions in $\mathcal{M}$ are all $\langle q, \Delta, l, j\rangle \to \langle q', \Delta', l', j'\rangle$ such that (1) $\Delta = \Delta'$ and $l = l'$, or (2) $l' < l$ and $\Delta' \neq \Delta_0$.

Note that the transitions of type (1) create cliques where Δ and l do not change.

Transitions of type (2) connect the cliques: from level $l > 0$ one can move to any clique at level $< l$ except Δ_0-cliques.

Observe that the Δ_0-cliques are the only ones that carry all 2^{r+1} different elements $\{q_0, \ldots, q_{2^{r+1}-1}\}$ and the only ones that cannot be reached from another clique. Hence, we have:

Fact 5.2. $\mathcal{M}, \langle q, \Delta, l, j\rangle \models \mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ *iff* $\Delta = \Delta_0$.

Also observe that

Fact 5.3. *The trees obtained by unfolding $\mathcal{M}$ from $\langle q, \Delta, l, j\rangle$ and from $\langle q, \Delta, l, j'\rangle$ are isomorphic. Therefore these nodes satisfy the same future MLO formulae and if a future MLO formula is satisfied at a node u, then it is satisfied at infinitely many nodes in the clique of u.*

Below we study how $TLAR_r$ formulae are satisfied in $\mathcal{M}$ in order to prove that they cannot express $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$.

Let B be a finite set of modalities definable by future *MLO* formulae of arity r. The next lemma states that whether $\langle q, \Delta, l, j\rangle$ satisfies $\phi \in TL(B)$ formula does not depend on Δ, j if l is sufficiently large:

Lemma 5.4. *Let B be a finite set of modalities definable by future MLO formulae of arity r. For every formula $\phi \in TL(B)$ there is $k_\phi \in \mathbb{N}$ such that for all $l, l' \geq k_\phi$ and all Δ, Δ', j, j' and for all $q \in \Delta \cap \Delta'$, we have*

$$\mathcal{M}, \langle q, \Delta, l, j\rangle \models \phi \quad \textit{iff} \quad \mathcal{M}, \langle q, \Delta', l', j'\rangle \models \phi. \tag{1}$$

The proof of the lemma is given in Section 5.2, after providing some elementary background about Composition method in Section 5.1.

Now we are ready to prove Lemma 4.2:

Proof of Lemma 4.2. From Lemma 5.4 we can derive Lemma 4.2 as follows. Assume that $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ is equivalent to some $\phi \in TLAR_r$. Let B be the (finite) set of $TLAR_r$ modalities that appear in ϕ. Let k_ϕ be as in Lemma 5.4 and let $l \geq k_\phi$. Then, for any Δ of size $2^{r+1} - 1$ and $q \in \Delta$: $\mathcal{M}, \langle q, \Delta, l, j\rangle \models \phi$ iff $\mathcal{M}, \langle q, \Delta_0, l, j\rangle \models \phi$, contradicting Fact 5.2. $\square$

5.1. Elements of composition method

Composition theorems are tools which reduce sentences about some compound structures to sentences about their parts. A seminal example of such a result is the Feferman–Vaught Theorem [10] which reduces the first-order theory of the generalized product to the first-order theory of its factors. Composition theorems for theories of orderings were first explored by Läuchli [18], and subsequently developed by Shelah [27]. Shelah [27] used the composition theorem for linear orders as one of the main tools for obtaining very strong decidability results for the monadic theory of linear orders. The technique was used in a series of papers by Shelah (see, e.g., [14,15,19]), and outlined in survey expositions by Gurevich [13] and Thomas [28].

In this section definitions and lemmas which will be used later are collected. They are adaptations of more general results proved by Shelah [27]. The proofs of the theorems stated here can be easily extracted from the results in [27,19].

Given two computation trees T and T', we write $T \equiv_n T'$ if no *MLO* sentence of quantifier-depth n can distinguish between these trees. Formally, $T \equiv_n T'$ if and only if for any *MLO* sentence φ with $\mathrm{qd}(\varphi) \leq n$ we have $T \models \varphi$ iff $T' \models \varphi$. Equally, we write $(T, s) \equiv_n (T', s')$ if no *MLO* formula $\varphi(x)$ with $\mathrm{qd}(\varphi) \leq n$ can distinguish between these trees with specified nodes.

The relations $\equiv_n$ are clearly equivalence relations over trees and over trees with specified nodes and they enjoy the following important properties.

Lemma 5.5. *Let Σ be a finite set of monadic predicate names.*

(1) *For each n, the relation $\equiv_n$ defines finitely many equivalence classes $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_m$ of trees over Σ; that is, $T \equiv_n T'$ iff $T, T' \in \mathbb{T}_i$ for some $i \in \{1, 2, \ldots, m\}$.*
(2) *For each equivalence class $\mathbb{T}_i$ there is a MLO sentence β_i with $\mathrm{qd}(\beta_i) \leq n$ which characterises it; that is, $T \in \mathbb{T}_i$ iff $T \models \beta_i$.*
(3) *Every MLO sentence φ with $\mathrm{qd}(\varphi) \leq n$ is equivalent to a (finite) disjunction of the characterising sentences β_i.*

The proof of the above lemma is easy once you realize that there are only finitely many semantically-distinct formulae with at most one free variable of a fixed quantifier-depth n. This fact itself can be shown easily by induction on quantifier-depth.

Referring to Lemma 5.5, with n fixed, we can fix m as well as the equivalence classes $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_m$ and sentences $\beta_1, \beta_2, \ldots, \beta_m$ as given in the lemma. We then define the extended alphabet[2]

$$\Sigma' = \Sigma \cup \{1, 2, \ldots, m\}.$$

Lemma 5.6 below states that in order to find to which $\equiv_n$ class a computation tree T belongs, it is enough to know the labels of the root and to which $\equiv_n$ equivalence classes the sons of the root belong.

Given a computation tree T, we denote by $\alpha(T)$ the computation tree over Σ' whose nodes are the root of T and its sons; the $<$ relation in $\alpha(T)$ is inherited from T, i.e, $root_T$ is the root of $\alpha(T)$ and $root_T$ and $root_{\alpha(T)}$ have the same sons. Hence, the leaves of $\alpha(T)$ are at level one.

The labelling of $\alpha(T)$ is defined as follows: the root $root_{\alpha(T)}$ of $\alpha(T)$ has the same label as the root of T and $v \neq root_{\alpha(T)}$ is labelled by i if the subtree of T rooted at v is in the equivalence class $\mathbb{T}_i$. The importance of $\alpha(T)$ is that it captures the whole of T with respect to the distinguishing power of *MLO* formulae of quantifier-depth n.

Lemma 5.6. *For every n there is n_B such that for every tree T and a tree T':*

(1) *if $\alpha(T) \equiv_{n_B} \alpha(T')$ then $T \equiv_n T'$;*
(2) *In particular, if $\alpha(T)$ and $\alpha(T')$ are isomorphic then $T \equiv_n T'$.*

This lemma can be easily proved using Ehrenfeucht–Fraissé games. It is also a simple instance of Shelah's composition theorem for the generalized sum (see Theorems 2, 4 in [27] or Th 1.12 in [19]).

5.2. Proof of Lemma 5.4

Let B be a finite set of future *MLO* modalities of arity at most r. Let n be an upper bound on the quantifier-depth of the truth tables for modalities in B. Let $\mathbb{T}_i$ $(i = 1, \ldots, m)$ be the $\equiv_n$ equivalence classes and let β_i $(i = 1, \ldots, m)$ be the corresponding characteristic sentences over monadic predicate names $P_1, \ldots, P_r$. Let $\hat{\beta}_i(x_0, X_1, \ldots, X_r)$ be formulae obtained by relativizing all first-order quantifiers of β_i to $\geq x_0$ and by replacing the predicate names P_i by monadic variables X_i $(i = 1, \ldots, r)$. From Lemma 5.5 and Remark 2.2 it follows that every modality in B has a truth table which is a (finite) disjunction of formulae from $\{\hat{\beta}_i : i = 1, \ldots, m\}$. Hence, every formula in *TL*(B) is equivalent to a formula in *TL*($\{\hat{\beta}_i : i = 1, \ldots, m\}$). Therefore, w.l.o.g., it is sufficient to prove Lemma 5.4 for the case when $B = \{\hat{\beta}_i : i = 1, \ldots, m\}$.

The proof proceeds by induction over the structure of ϕ. The cases where ϕ is an atomic proposition, or a Boolean combination of subformulae are obvious. The only interesting case is the case when $\phi = \hat{\beta}(\phi_1, \ldots, \phi_r)$, where $\hat{\beta}$ is modality and $\phi_1, \ldots, \phi_r$ are formulae.

[2] W.l.o.g. we assume that Σ and $\{1, 2, \ldots, m\}$ are disjoint.

For $i = 1, \ldots, r$ let P_i be the set of nodes in $\mathcal{M}$ that satisfy ϕ_i. Let $\mathcal{M}'$ be the Kripke structure with the same nodes as $\mathcal{M}$, the same accessibility relations as $\mathcal{M}$ and with r unary predicates $P_1, \ldots, P_r$.

Let $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_m$ be the $\equiv_n$ equivalence classes of computational trees with r monadic predicates.

Let $F = \{i$: there is $\langle q, \Delta, l, j\rangle$ such that $\Delta \neq \Delta_0$ and the tree obtained by unfolding $\mathcal{M}'$ from $\langle q, \Delta, l, j\rangle$ is in $\mathbb{T}_i\}$.

Let k_ϕ be such that $k_\phi \geq k_{\phi_i}$ (for $i = 1, \ldots, r$) and if $i \in F$ then there is $\langle q', \Delta', l', j'\rangle$ such that $\Delta' \neq \Delta_0$ and the level $l' < k_\phi$ and the tree obtained by unfolding $\mathcal{M}'$ from $\langle q', \Delta', l', j'\rangle$ is in $\mathbb{T}_i$. Note that k_ϕ exists because F and B are finite sets.

First we show that

(A) for every $\langle q, \Delta, l, j\rangle$, $\langle q, \Delta', l', j'\rangle$ at level $\geq k_\phi$ if $\Delta \neq \Delta_0 \neq \Delta'$ then

$$\mathcal{M}, \langle q, \Delta, l, j\rangle \models \phi \quad \text{iff} \quad \mathcal{M}, \langle q, \Delta', l', j'\rangle \models \phi.$$

Let T (respectively, T') be the tree obtained by unfolding $\mathcal{M}'$ from $\langle q, \Delta, l, j\rangle$ (respectively, from $\langle q, \Delta', l', j'\rangle$). Since $l, l' \geq k_\phi$, from the definition of $\mathcal{M}'$ it follows that for every $i \in F$ both the root of T and the root of T' have infinitely many sons in the $\equiv_n$ equivalence class $\mathbb{T}_i$. For every $i \notin F$ neither the root of T nor the root of T' have a son in $\mathbb{T}_i$. By the inductive assertion $\mathcal{M}, \langle q, \Delta, l, j\rangle \models \phi_i$ iff $\mathcal{M}, \langle q, \Delta', l', j'\rangle \models \phi_i$ for $i = 1, \ldots, r$. Therefore, the roots of T and of T' have the same labels. Hence, $\alpha(T)$ is isomorphic to $\alpha(T')$ and by Lemma 5.6 we obtain that $T \equiv_n T'$. This implies (A) because $\phi = \hat{\beta}(\phi_1, \ldots, \phi_r)$ and $\hat{\beta}$ is a future *MLO* modality of quantifier-depth at most n.

Now let $U = \{\langle q_0, \Delta_0, l, 1\rangle, \langle q_1, \Delta_0, l, 1\rangle, \ldots, \langle q_{2^{r+1}-1}, \Delta_0, l, 1\rangle\}$, where $l \geq k_\phi$. Since the cardinality of U is $2^{r+1} > 2^r$, there are $j_1 \neq j_2$ such that $\mathcal{M}, \langle q_{j_1}, \Delta_0, l, 1\rangle \models \phi_i$ iff $\mathcal{M}, \langle q_{j_2}, \Delta_0, l, 1\rangle \models \phi_i$ for $i = 1, \ldots, r$. Let $\Delta' = \Delta_0 \setminus \{q_{j_1}\}$ and $\Delta'' = \Delta_0 \setminus \{q_{j_2}\}$. For every $q \in \Delta'$ (respectively, $q \in \Delta''$) the tree obtained by unfolding $\mathcal{M}'$ from $\langle q, \Delta_0, l, j\rangle$ is isomorphic to the tree obtained by unfolding $\mathcal{M}'$ from $\langle q, \Delta', l, j\rangle$ (respectively, from $\langle q, \Delta'', l, j\rangle$). Therefore, for every $q \in \Delta'$ (respectively, $q \in \Delta''$) $\mathcal{M}', \langle q, \Delta_0, l, j\rangle \models \hat{\beta}(P_1, \ldots, P_r)$ iff $\mathcal{M}', \langle q, \Delta', l, j\rangle \models \hat{\beta}(P_1, \ldots, P_r)$ (respectively, $\mathcal{M}', \langle q, \Delta'', l, j\rangle \models \hat{\beta}(P_1, \ldots, P_r)$). This together with (A) and the definition of $\mathcal{M}'$ imply that $\mathcal{M}, \langle q, \Delta, l, j\rangle \models \phi$ iff $\mathcal{M}, \langle q, \Delta', l', j'\rangle \models \phi$ for every $\langle q, \Delta, l, j\rangle$, $\langle q, \Delta', l', j'\rangle$ at level $\geq k_\phi$.

This completes the proof of the lemma.

6. Conclusion and related results

The paper concerns the expressive power of temporal logics over trees. Our main result states: for every n, there is an n-ary modality $\mathbb{M}_n$ that is not equivalent over trees to any temporal formula built over modalities of arity strictly less than n. Its proof takes advantage of an instance of Shelah's composition theorem. Besides, this result has interesting corollaries for instance reproving the known facts that CTL^* and $ECTL^+$ have no finite basis.

The modality $\mathbb{M}_k(X_1, \ldots, X_k)$ expresses a "natural fairness property": $\mathbb{M}_k(X_1, \ldots, X_k)$ holds at s iff there is a path from s such that for every $I \subseteq \{1, \ldots, k\}$ the formula $\bigwedge_{i\in I} X_i \wedge \bigwedge_{j\notin I} \neg X_j$ is satisfied infinitely often along this path.

As one of the consequences of the main lemma, we obtained the arity hierarchy for the temporal logics with the modalities definable by future *MLO* formulae — Theorem 4.3.

The main lemma also implies that neither CTL^* nor $ECTL^+$ has finite base.

Regarding CTL^*, it was shown that its expressive power cannot be captured by a finite set of modalities, thus providing a partial explanation of why there is no general agreement as to what should be the preferred set of modalities for branching-time logics [22,23]. In [23], we introduced a sequence $BTL_1, BTL_2, \ldots$ of temporal logics where BTL_k has the modalities $\mathsf{E}\phi$ for any *FOMLO* formula ϕ of quantifier-depth at most k (see Example 3.4(3)) and showed that there exists an infinite hierarchy (w.r.t. expressive power) among $BTL_1, BTL_2, \ldots$. Since CTL^* is exactly as expressive as $BTL \overset{\text{def}}{=} \bigcup_k BTL_k$, and since any CTL^* modality is a BTL_k modality for some k, the existence of an infinite hierarchy among $\{BTL_k\}_{k=1,2,\ldots}$ entails that CTL^* has no finite basis. Note that for every $k > 1$, the modalities $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ are in BTL_k. Therefore, by Corollary 1(2), for $k > 1$ there is no finite base for BTL_k.

In [24] it was proved that $ECTL^+$ has no finite basis. The modalities EM_l were introduced for a kind of fairness constraint: $\mathsf{EM}_l(X_1, \ldots, X_l)$ states that there is a path along which every X_i is satisfied infinitely often and where only the nodes satisfying some of the X_is are encountered. In [24], the proof that $ECTL^+$ has no finite base was derived from the following results:

(1) the temporal logics $ECTL^+$ and BTL_2 and $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ are expressively equivalent and
(2) for all $k \in \mathbb{N}$ the logic $TL(\mathsf{EU}, \{\mathsf{EM}_1, \ldots, \mathsf{EM}_k\})$ is strictly less expressive than $TL(\mathsf{EU}, \{\mathsf{EM}_1, \ldots, \mathsf{EM}_k, \mathsf{EM}_{k+1}\})$.

The construction of a Kripke structure that witnesses the lack of expressive power stated in our main result is strongly inspired from a similar construction in [24] (cf. Sect. 5.1) and the n-ary modality $\mathbb{M}_n$ used to prove the main result is a variant of a modality EM_n.

Our main lemma is much stronger than (2). Our proof is quite different from the proof in [24] and requires the replacement of a simple induction by more subtle arguments based on the composition method.

In our proof we used trees with infinite branching. However, this is not essential. The proof can be easily modified to show that $\mathbb{M}_{r+1}$ is not expressible in $TLAR_r$ over the class of trees with finite branching.

Let us conclude with some open questions:

Arity Hierarchy over finite trees: Is $TLAR_r$ less expressive than $TLAR_{r+1}$ over the class of finite trees?

Note that over the class of finite trees $\mathbb{M}_r(X_1, \ldots, X_r)$ is equivalent to false; hence, it cannot distinguish between $TLAR_r$ and $TLAR_{r+1}$. The results of [25,26] imply that any future first-order *MLO* formula is equivalent to a $TL(\mathrm{U}, \{\mathbf{D}^n X\}_{n=1,2,\ldots})$ formula, where $\mathbf{D}^n X$ are counting modalities; $\mathbf{D}^n X$ holds at t if X holds *for (at least)* $n>0$ *different successors* of t. Therefore, every modality definable by a future first-order *MLO* formula is equivalent to a $TLAR_2$ formula and cannot be used to show that the logics $\{TLAR_r : r \in \mathbb{N}\}$ form a hierarchy.

Arity Hierarchy over ω: Is $TLAR_r$ less expressive than $TLAR_{r+1}$ over the class of ω-chains?

Note that over the class of ω-chains $\mathbb{M}_r(X_1, \ldots, X_r)$ is equivalent to a future first-order *MLO* formula. By the Kamp theorem every future first-order *MLO* formula is equivalent to a $TL(\mathrm{U})$ formula, and therefore, it cannot be used to show that the logics $\{TLAR_r : r \in \mathbb{N}\}$ form a hierarchy. In [1], it was shown that there is no finite base temporal logic which has the same expressive power as the *MLO*. However, the separation lemma in this proof used modalities of arity one.

In our proof it was essential that we only deal with modalities definable by *future MLO* formulae. We believe that our results can be extended to modalities definable by arbitrary *MLO* formulae. In particular, we conjecture that $\mathbb{M}_{r+1}(X_1, \ldots, X_{r+1})$ cannot be expressed in any temporal logic with modalities of arity at most r definable by *MLO* formulae (not necessarily future ones).

References

[1] D. Beauquier, A. Rabinovich, Monadic logic of order over naturals has no finite base, Journal on Logic and Computation 12 (2) (2002) 243–253.
[2] M. Ben-Ari, Z. Manna, A. Pnueli, The temporal logic of branching time, in: Proceedings of the 8th Annual ACM Symposium on Principles of Programming Languages, ACM, New York, 1981, pp. 164–176.
[3] E.M. Clarke, E.A. Emerson, Design and verification of synchronous skeletons using branching time temporal logic, in: Lecture Notes in Computer Science, vol. 131, Springer-Verlag, 1981, pp. 52–71.
[4] E.M. Clarke, E.A. Emerson, A.P. Sistla, Automatic verification of finite state concurrent system using temporal logic, in: 10th ACM Symp. on Principles of Prog. Lang., 1983, pp. 117–126.
[5] E.M. Clarke, E.A. Emerson, A.P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications, ACM Transactions on Programming Languages and Systems 8 (2) (1986) 244–263.
[6] E.A. Emerson, E.M. Clarke, Characterizing correctness properties of parallel programs using fixpoints, in: Proc. 7th Coll. Automata, Languages and Programming, ICALP'80, Noordwijkerhout, NL, in: Lecture Notes in Computer Science, vol. 85, Springer-Verlag, 1980, pp. 169–181.
[7] E.A. Emerson, Temporal and modal logic, in: J. van Leeuwen (Ed.), in: Handbook of Theoretical Computer Science, vol. B., Elsevier, Amsterdam, 1990.
[8] E.A. Emerson, J.Y. Halpern, Decision procedures and expressiveness in the temporal logic of branching time, in: Proceedings of the 14th Annual ACM Symposium of Theory of Computing, ACM, New York, 1982, pp. 169–180.
[9] E.A. Emerson, J.Y. Halpern, 'Sometimes' and 'not never' revisited: On branching versus linear time temporal logics, Journal of the ACM 33 (1) (1986) 151–178.
[10] S. Feferman, R.L. Vaught, The first-order properties of products of algebraic systems, Fundamenta Mathematica 47 (1959) 57–103.
[11] D. Gabbay, I. Hodkinson, M. Reynolds, Temporal Logic, Oxford University Press, 1994.
[12] D. Gabbay, A. Pnueli, S. Shelah, J. Stavi, On the temporal analysis of fairness, in: 7th ACM Symp. on Principles of Prog. Lang., 1980, pp. 163–173.
[13] Y. Gurevich, Monadic second-order theories, in: J. Barwise, S. Feferman (Eds.), Model-Theoretic Logics, Springer-Verlag, 1985, pp. 479–506.
[14] Y. Gurevich, S. Shelah, To the decision problem for branching time logic, in: P. Weingartner, G. Dold (Eds.), Foundations of Logic and Linguistics: Problems and their Solutions, Plenum, 1985, pp. 181–198.
[15] Y. Gurevich, S. Shelah, The decision problem for branching time logic, Journal of Symbolic Logic 50 (3) (1985) 668–681.
[16] H.W. Kamp, Tense logic and the theory of linear order, Ph.D. Thesis, University of California, Los Angeles, 1968.
[17] L. Lamport, "Sometimes" is sometimes "not never" — On the temporal logic of programs, in: Proceedings of the 7th Annual ACM Symposium on Principles of Programming Languages, ACM, New York, 1980, pp. 174–185.
[18] H. Läuchli, A decision procedure for the weak second-order theory of linear order, in: Contributions to Mathematical Logic, Proceedings of Logic Colloquium Hanover 1966, North-Holland, 1968, pp. 189–197.
[19] S. Lifsches, S. Shelah, Uniformization, choice functions and well orders in the class of trees, Journal of Symbolic Logic 61 (3) (1996) 1206–1227.
[20] A. Pnueli, The temporal logic of programs, in: Proc. IEEE 18th Annu. Symp. on Found. Comput. Sci., IEEE, New York, 1977, pp. 46–57.
[21] J.-P. Queille, J. Sifakis, Fairness and related properties in transition systems. A temporal logic to deal with fairness, Acta Informatica 19 (3) (1983) 195–220.
[22] A. Rabinovich, Expressive power of temporal logics, in: Proc. 13th Int. Conf. Concurrency Theory, CONCUR 2002, Brno, Czech Republic, August 2002, in: Lecture Notes in Computer Science, vol. 2421, Springer-Verlag, 2002, pp. 57–75.
[23] A. Rabinovich, S. Maoz, An infinite hierarchy of temporal logics over branching time, Information and Computation 171 (2001) 306–332.
[24] A. Rabinovich, Ph. Schnoebelen, BTL_2 and expressive completeness for $ECTL^+$, Information and Computation 204 (7) (2006) 1023–1044.
[25] B. Schlingloff, On the expressive power of modal logics on trees, in: Proceedings of Symposium on Logical Foundations of Computer Science, in: Lecture Notes in Computer Science, vol. 620, Springer-Verlag, 1992, pp. 441–451.
[26] B. Schlingloff, Expressive completeness of temporal logic over trees, Journal of Applied Non-classical Logics 2 (1992) 157–180.
[27] S. Shelah, The monadic theory of order, Annals of Mathematics 102 (1975) 379–419.
[28] W. Thomas, Ehrenfeucht games, the composition method, and the monadic theory of ordinal words, in: Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht, in: Lecture Notes in Computer Science, vol. 1261, Springer-Verlag, 1997, pp. 118–143.
[29] M. Vardi, P. Wolper, Yet another process logic, in: Logics of Programs, in: Lecture Notes in Computer Science, vol. 164, Springer-Verlag, 1984, pp. 501–512.