An Infinite Hierarchy of Temporal Logics over Branching Time *

Alexander Rabinovich and Shahar Maoz
Department of Computer Science, Tel-Aviv University
Tel-Aviv, Israel
{rabino, maoz}@math.tau.ac.il

Abstract

Many temporal logics were suggested as branching time specification formalisms during the last 20 years. These logics were compared against each other for their expressive power, model checking complexity and succinctness. Yet, unlike the case for linear time logics, no canonical temporal logic of branching time was agreed upon. We offer an explanation for the multiplicity of temporal logics over branching time and provide an objective quantified 'yardstick' to measure these logics.

We define an infinite hierarchy BTL_k of temporal logics and prove its strictness. We examine the expressive power of commonly used branching time temporal logics. We show that CTL^* has no finite base, and that almost all of its many sub-logics suggested in the literature are inside the second level of our hierarchy.

We introduce new Ehrenfeucht-Fraissé games on trees, and use them as our main tool to prove inexpressibility.

1 Introduction

Various temporal logics have been proposed for reasoning about so-called "reactive" systems, computer hardware or software systems which exhibit (potentially) non-terminating and non-deterministic behavior. Such a system 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) is a convenient framework for the specification properties of systems. This made TL a popular subject in the Computer Science community and it enjoyed extensive research during the last 20 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 formulas 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 *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 formulas 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 $\forall X$. Practically 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 a convenient way to use fragments of MLO. MLO can also serve as a yardstick

*This is the extended and full version of a paper presented in MFCS 2000 [32].

by which to check 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 if two formulas are equivalent depends on the domain over which they are evaluated. Any 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 [23, 13] 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.

There is an important distinction between the future and the past. It is usually assumed that any particular point of time has one linear past, but perhaps various futures. This might be the reason why most of the temporal formalisms studied in Computer Science use only future time constructs. Fortunately, Kamp's theorem also implies that the TL with one modality $\mathbf{U}$ (until) has the same expressive power (over the canonical linear discrete model) as the future fragment of the first-order monadic logic. In this paper we will deal only with future fragments of MLO and with future time temporal logics.

Milner and Park [27, 29] pointed out that for the specification of concurrent systems we need a model finer than just the set of possible (linear) runs; this led to the computation tree model.

Of course, $TL(\mathbf{U})$ is interpreted not only over linear orders, but over arbitrary partial orders; in particular, over the trees. However, the expressive power of $TL(\mathbf{U})$ over the trees is very limited. For instance, a very basic property "for all paths that start at t_0, eventually p holds" is not expressible in $TL(\mathbf{U})$.

In order to reflect branching properties of computations many temporal logics were suggested starting from [25, 2]. The basic modalities of these logics (which are often called branching time logics) are either of the form $\mathbf{E}$ ("there exists a linear run") followed by a formula in $TL(\mathbf{U})$ or of the form $\mathbf{A}$ ("for every linear run") followed by a formula in $TL(\mathbf{U})$. $\mathbf{E}\varphi$ (respectively $\mathbf{A}\varphi$) holds at a moment t_0 if for some path π (respectively, for every path π) starting at t_0 the $TL(\mathbf{U})$ formula φ holds along π. For example, one commonly used branching time logic is CTL [2]. It is based on two binary modalities $\mathbf{EU}$ and $\mathbf{AU}$; $\mathbf{AU}(X,Y)$ (respectively $\mathbf{EU}(X,Y)$) holds at a current moment t_0 if "for all (respectively, for some) runs from the current moment, X until Y holds". In contrast to expressive completeness of $TL(\mathbf{U})$ over the canonical linear models, there is no natural predicate logic which corresponds to $TL(\mathbf{EU}, \mathbf{AU})$ (i.e., to CTL) over the trees. Moreover, it turns out that CTL cannot express many natural fairness properties [10].

The logic CTL^* suggested in [10] has the same expressive power as the temporal logic with infinite set of modalities $\{\mathbf{E}\varphi \ : \ \varphi \text{ is a formula of } TL(\mathbf{U})\} \cup \{\mathbf{A}\varphi \ : \ \varphi \text{ is a formula of } TL(\mathbf{U})\}$. Many temporal logics were suggested as branching time specification formalisms (see [10, 11, 7]) by imposing some syntactical restrictions on CTL^* formulas. The lack of a yardstick was emphasized by Emerson in words very similar to the above [7, 8]:

> "Hundreds perhaps thousands of papers developing the theory and application of temporal logic to reasoning about reactive systems were written. Dozens if not hundreds of systems of temporal logic have been investigated, both from the standpoint of basic theory and from the standpoint of applicability to practical problems . ... there is now a widespread consensus that some type of temporal logic constitutes a superior way to specify and reason about reactive systems. There is no universal agreement on just which logics are best ... While less is known about comparisons of Branching time logics against external "yardsticks", a great deal is knows about comparisons of BTLs against each other. This contrasts with the reversed situation for Linear-Time Logics." (page 44 in [8])

Our results offer an explanation for the multiplicity of temporal logics over branching time and suggest some yardsticks by which to measure these logics.

One popular equivalence between computation trees is that of bisimulation equivalence. This equivalence catches subtle differences between trees based on their branching structures. It is generally regarded as the finest behavioral equivalence of interest for concurrency (it is often argued that concurrent systems giving rise to bisimulation equivalent computation trees are indistinguishable for all reasonable notions of observation). In [28], CTL^* was shown to be expressively equivalent to the bisimulation invariant fragment of monadic path logic [15, 17]. The syntax of monadic path logic is the same as that of monadic second-order logic. The bound set (monadic) variables ranges over all the paths and semantically this logic is very closely related to the first-order logic [15, 28]. Thus at least CTL^* represents some objectively quantified expressive power.

We describe a sequence BTL_k ($k \in N$) of temporal logics. All these logics are sub-logics of CTL^* and their union has the same expressive power as CTL^*. Roughly speaking the modalities of BTL_k correspond to formulas with quantifier-depth at most k. However, for every m and k there is a BTL_k formula which is equivalent to no MLO formula with quantifier-depth $\leq m$. We show that BTL_{k+1} is strictly more expressive than BTL_k. Consequently, we obtain that there is no finite basis for CTL^* and hence there is no finite basis for the bisimulation invariant fragment of monadic path logic. Our proof also demonstrates that in contrast to the linear time case, there is no finite base temporal logic with the same expressive power (over the trees) as the bisimulation invariant fragment of first-order logic.

We examine the expressive power of commonly used branching time temporal logics. It turns out that almost all of these logics are inside the second level of our hierarchy. The modalities for these logics were suggested by a desire to formalize some pragmatic properties which often occur in specifications of hardware and software systems. It is interesting to observe that most of these properties can be formalized by formulas with quantifier depth at most two.

The problem whether a formula φ is satisfiable in the computation tree which corresponds to a finite state (Kripke) structure is known as the model checking problem. The model checking problem for CTL has $O(\mid K \mid \times \mid \varphi \mid)$ time complexity . Unlike CTL, the model checking problem for CTL^* is PSPACE complete [3]. We prove that the model checking problem has $O(\mid K \mid \times \mid \varphi \mid)$ time complexity for every temporal logic based on a finite set of modalities definable in CTL^*.

Finally, let us mention that all our results are valid not only for the class of all trees, but also for its interesting subclasses: the finite trees, trees obtained by unwinding finite Kripke structures, and trees with only infinite branches.

The paper is organized as follows. In the next section we review basic definitions about monadic logic of order and its fragments. In Section 3 we review basic definitions and known results about temporal logics and modalities. The logics BTL_k are also defined there. In Section 4 we introduce new Ehrenfeucht-Fraïssé games appropriate for BTL_k. These games are the main tool we use to prove the strictness of the hierarchy in Section 5. The techniques and results described here are of an independent interest. In Section 6 we show that CTL^* has no finite basis and examine the expressive power of some commonly used branching time logics. In Section 7 we give some strengthening to our main results and discuss the complexity of the model checking problem for finite base temporal logics. In the concluding section we present some open questions.

2 Preliminaries

In this section we recall basic definitions about the monadic logic of order.

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 $\mid A \mid$ for the universe of A and R^A for the interpretation of the relational symbol R in A. However, whenever there is no confusion we will also use A for the universe of A; sometimes, we use "$a \in A$" instead of "$a \in\mid A \mid$" 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 $\tau = \{R_i \ : \ i \in Ind\}$ we use notations $(\mid A \mid, \ldots, R_i^A, \ldots)$ which we also abbreviated to $(\mid A \mid, \ldots, R_i, \ldots)$ or to $(\mid A \mid, \vec{R})$, where a vector denotes a tuple of relations of appropriate length and arity.

We assume that logical languages contain the following logical symbols: equality, <u>true</u> and <u>false</u> with their standard interpretations.

2.1 Computation Trees

A structure $T = (\mid T \mid, \leq^T)$ is a *tree* if $\leq^T$ is a binary relation such that

1. The set $\mid T \mid$ is partially ordered by $\leq^T$.

2. There is a unique $\leq^T$ minimal element.

3. For every element $a \in\mid T \mid$ the set $\{b \in\mid T \mid \ : \ b \leq^T a\}$ is finite and $\leq^T$ is a linear order on this set.

The elements of $\mid T \mid$ are called *nodes* of the tree (sometimes we call them *states* or *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 $s' \leq^T s \wedge s' \neq s$ and there is no element between s and s'.

Let τ be a signature $\{\leq\} \cup \{P_i \ : \ i \in Ind\}$, where P_i are unary predicate symbols. We do not assume that the cardinality of the signature is finite or even countable. A structure $(\mid T \mid, \leq^T, \dots, P_i^T, \dots)$ for τ is a *computation tree* for a signature τ if $(\mid T \mid, \leq^T)$ is a tree. Note that P_i^T $(i \in Ind)$ are subsets of $\mid T \mid$. We say that a node $s \in\mid T \mid$ is *labeled by* P_i if $s \in P_i^T$.

Whenever a signature τ is clear from the context or is irrelevant we say "computation tree" instead of "computation tree for the signature τ".

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 $\mid T_{\geq s} \mid \triangleq \{t : t \in\mid T \mid \text{ and } t \geq s\}$, P_i is interpreted as $P_i^T \cap \mid T_{\geq s} \mid$ and $\leq$ is interpreted as $\leq^T \cap \mid T_{\geq s} \mid \times \mid T_{\geq s} \mid$.

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

2.2 Monadic Logic of Order

The syntax of the ***second-order Monadic Logic of Order (MLO)*** has in its vocabulary individual first-order variables $x_0, x_1, x_2, \dots$ (representing nodes, states or time points), set variables $X_0, X_1, X_2, \dots$ (representing sets of nodes), and set constants (monadic predicates names).

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

We define $\vee$, $\forall$, and $\rightarrow$ in terms of $\wedge$, $\neg$ and $\exists$ as usual. In addition, we use other standard abbreviations, such as $x_1 \neq x_2$ for $\neg x_1 = x_2$, $x_1 \geq x_2$ for $x_2 \leq x_1$ and $x_1 < x_2$ for $x_1 \leq x_2 \wedge x_1 \neq x_2$.

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

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

The semantics of *MLO* follows classical lines: if T is a computation tree, $s_1, \dots, s_m \in\mid T \mid$ are nodes of T and $S_1, \dots, S_n \subseteq\mid T \mid$ are sets of nodes, we write

$$(T, s_1, s_2, \dots, s_m, S_1, S_2, \dots, S_n) \models \varphi(x_1, x_2, \dots, x_m, X_1, X_2, \dots, X_n)$$

if the formula φ is satisfied in the tree T with x_i interpreted as s_i $(i = 1, \dots, m)$ and X_j interpreted as S_j $(j = 1, \dots, n)$. The definition is a standard one, so we just give three clauses of the definition here.

- $(T, s_1, s_2, \dots, s_m, S_1, S_2, \dots, S_n) \models x_j \in P_i$ iff $s_j \in P_i^T$.
- $(T, s_1, s_2, \dots, s_m, S_1, S_2, \dots, S_n) \models x_j \in X_k$ iff $s_j \in S_k$.
- $(T, s_1, s_2, \dots, s_m, S_1, S_2, \dots, S_n) \models \exists X_n \varphi(x_1, x_2, \dots, x_m, X_1, X_2, \dots, X_n)$ iff $(T, s_1, s_2, \dots, s_m, S_1, S_2, \dots, S_{n-1}, S) \models \varphi(x_1, x_2, \dots, x_m, X_1, X_2, \dots, X_n)$ for some subset S of $\mid T \mid$.

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

$$T, s, S_1, \dots, S_k \models \varphi \ \text{ iff } \ T_{\geq s}, s, S'_1, \dots, S'_k \models \varphi$$

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

In other words, a future formula is a formula with one free node 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.

Let $\varphi(x_0, X_1, \dots X_k)$ be a formula. Let $\tilde{\varphi}$ be obtained from φ by relativizing all first-order quantifiers to the elements greater than or equal to x_0, i.e., when "$\exists x.\ \dots$" and "$\forall x.\ \dots$" are replaced by "$\exists x(\ x \geq x_0 \wedge \dots)$" and by "$\forall x(x \geq x_0 \rightarrow \dots)$" respectively. Note that the formula $\tilde{\varphi}$ obtained in such a way is always a future formula. Moreover, φ is a future formula if and only if $\varphi \leftrightarrow \tilde{\varphi}$ is valid. The validity of *MLO* formulas is decidable [31]. Therefore, it is decidable whether a formula is a future formula.

2.3 Monadic Path Logic

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

We also consider *MPL*, the *monadic path logic* [17]: its syntax is the same as that of monadic second-order logic. However, unlike *MLO*, the bound set variables range over all the paths (not over arbitrary sets of nodes); the monadic predicate names (set constants) and the free set variables are interpreted by arbitrary sets of nodes.

Therefore, the corresponding clause for the second-order quantification is

$$(T, s_1, s_2, \dots, s_m, S_1, , \dots, S_n) \models \exists X_n \varphi(x_1, x_2, \dots, x_m, X_1, \dots, X_n) \text{ iff}$$

$$(T, s_1, s_2, \dots, s_m, S_1, , \dots, S_{n-1}, S) \models \varphi(x_1, x_2, \dots, x_m, X_1, \dots, X_n) \text{ for the set of nodes } S \text{ of a path in } T.$$

Since "X is a path" can be expressed in *MLO*, there is a meaning preserving translation from *MPL* into a fragment of *MLO*.

Though the syntax of *MPL* is the same as the syntax of *MLO*, the expressive power of *MPL* is very closely related to the expressive power of first-order logic [28]. In particular, for every *MPL* sentence φ there is a *FOMLO* sentence ψ such that for every finite tree T

$$T \models \varphi \text{ if and only if } T \models \psi.$$

3 Temporal Logics

In this section we review basic definitions and known results about temporal logics and modalities. We also introduce $\{BTL_k\}_{k=1}^{\infty}$, an infinite sequence of temporal logics; these logics provide natural yardsticks by which to measure the expressive power of temporal logics.

3.1 Temporal Logics and Modalities

In this subsection, we recall the syntax and semantics of temporal logics with notations adopted from [18].

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 $B = \{\#_1^{l_1}, \#_2^{l_2}, \dots\}$ (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 formulas are just variables and other formulas are obtained from the atoms using Boolean connectives and applying the modalities. The syntax of *TL*(B) is given by the following grammar:

$$\varphi \ ::= \ P \ \mid \ \varphi_1 \wedge \varphi_2 \ \mid \ \neg\varphi_1 \ \mid \ \#_i^{l_i}(\varphi_1, \varphi_2, \dots, \varphi_{l_i}) \ \text{, where}$$

P ranges over the variable names.

The ***nesting-depth*** of a temporal formula φ, denoted by $\mathrm{nd}(\varphi)$, is defined as usual: $\mathrm{nd}(\varphi) = 0$ for atomic formulas; $\mathrm{nd}(\varphi \wedge \varphi') = \max(\mathrm{nd}(\varphi), \mathrm{nd}(\varphi'))$; $\mathrm{nd}(\neg\varphi) = \mathrm{nd}(\varphi)$; and $\mathrm{nd}(\#_i^{l_i}(\varphi_1, \varphi_2, \dots, \varphi_{l_i})) = 1 + \max\limits_{1 \leq j \leq l_i} (\mathrm{nd}(\varphi_j))$.

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

$\#^l_T : [\mathcal{P}(\mid T \mid)]^l \to \mathcal{P}(\mid T \mid)$ which assigns "the set of points where $\#^l(Q_1, \dots, Q_k)$ holds" to the $l-$tuple $\langle Q_1, \dots, Q_l \rangle$ (Here $\mathcal{P}$ is the power set notation, and $\mathcal{P}(\mid T \mid)$ denotes the set of all subsets of the universe of T.) Formally, the semantics of a formula $\varphi \in TL$ over a tree T is defined inductively as follows. For atomic formulas $T, s \models P_i$ iff $s \in P_i^T$; the semantics of Boolean combinations is defined as usual, and the semantics of modalities is defined by: $T, s \models \#^l(\varphi_1, \varphi_2, \dots, \varphi_l)$ iff $s \in \#^l_T(R_{\varphi_1}, R_{\varphi_2}, \dots, R_{\varphi_l})$ where $R_{\varphi_i} = \{a : T, a \models \varphi_i\}$ for all $i, 1 \leq i \leq l$.

Note. In temporal and modal logics, formulas 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.

3.2 Truth Tables

In this paper, we consider only temporal modalities which are defined in *MLO*: we assume that for every l-place modality $\#$ there is a formula $\bar{\#}(x_0, X_1, X_2, \dots, X_l)$ of *MLO* with one free first-order variable x_0 and l set variables, such that for every tree T and subsets $R_i \subseteq \mid T \mid$:

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

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

Example 3.1 (Some common modalities and their truth tables)

- *The one-place modality* $\mathbf{F}$ *("*eventually*"); its truth table is* $\varphi_{\mathbf{F}}(x_0, X_1) \triangleq \exists y(y > x_0 \ \wedge \ y \in X_1)$.
- *The one-place modality* $\mathbf{G}$ *("*globally*"); its truth table is* $\varphi_{\mathbf{G}}(x_0, X_1) \triangleq \forall y(y > x_0 \ \to \ y \in X_1)$.
- *The one-place modality* $\mathbf{X}$ *("*next*"); its truth table is*

$$\varphi_{\mathbf{X}}(x_0, X_1) \triangleq \exists y\big(y > x_0 \ \wedge \ y \in X_1 \ \wedge \ \forall z(z < y \ \to \ z \leq x_0)\big).$$

- *The two-place modality* $\mathbf{U}$ *("*until*"); its truth table is*

$$\varphi_{\mathbf{U}}(x_0, X_1, X_2) \triangleq \exists y\big(y > x_0 \ \wedge \ y \in X_2 \ \wedge \ \forall z(x_0 < z < y \ \to \ z \in X_1)\big).$$

  In the literature, sometimes a "nonstrict" definition of Until is given: the "nonstrict until" $\mathbf{U}_{ns}$ *modality has truth table* $\varphi_{\mathbf{U}_{ns}}(x_0, X_1, X_2) \triangleq \exists y\big(y \geq x_0 \ \wedge \ y \in X_2 \ \wedge \ \forall z(x_0 \leq z < y \ \to \ z \in X_1)\big)$. *Clearly,* $\mathbf{U}_{ns}$ *can be defined using* $\mathbf{U}$.

- *The one-place modality* $\mathbf{F}^\infty$ *("*infinitely often*"); its truth table is*

$$\varphi_{\mathbf{F}^\infty}(x_0, X_1) \triangleq \forall y\big(y > x_0 \ \to \ \exists z(z > y \ \wedge \ z \in X_1)\big).$$

- *The two-place modality* $\mathbf{S}$ *("*since*"); its truth table is*

$$\varphi_{\mathbf{S}}(x_0, X_1, X_2) \triangleq \exists y\big(y < x_0 \ \wedge \ y \in X_2 \ \wedge \ \forall z(x_0 > z > y \ \to \ z \in X_1)\big).$$

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* formulas (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 $\mathbf{F}$, $\mathbf{G}$, $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{F}^\infty$ are first-order future modalities; the modality $\mathbf{S}$ is not a future modality.

Definition 3.3 (Path Modalities) *For every first-order future formula $\varphi(x_0, X_1, \ldots, X_l)$, we define an l-place path modality $\mathsf{E}\varphi$ as follows: $T, a \models \mathsf{E}\varphi$ if and only if there is a path π from a in T, such that $T_\pi, a \models \varphi(x_0, X_1, \ldots, X_l)$. $\mathsf{E}\varphi$ is said to be the path modality which corresponds to $\varphi(x_0, X_1, \ldots, X_l)$.*

If $\varphi(x_0, X_1, \ldots, X_l)$ is a future *FOMLO* formula, the truth table of the path modality $\mathsf{E}\varphi$ is the *MPL* formula $\exists Y \big(x_0 \in Y \wedge \varphi'(x_0, X_1, \ldots, X_l, Y)\big)$ where $\varphi'(x_0, X_1, \ldots, X_l, Y)$ is obtained from $\varphi(x_0, X_1, \ldots, X_l)$ by relativising all its quantifiers to Y. Thus, the following proposition holds:

Proposition 3.4 *For every first-order future formula $\varphi(x_0, X_1, \ldots, X_l)$, the path modality $\mathsf{E}\varphi$ has an MPL truth table.*

3.3 Kamp's Theorem

Different temporal logics may be considered a convenient way to use fragments of *MLO*. The following are standard definitions and notations to discuss expressive power of temporal logics. A temporal logic formula φ is *equivalent* to an *MLO* formula $\psi(x_0)$ with a single free first-order variable (or to a temporal logic formula ψ) over a computation tree T iff for every $a \in\mid T \mid$:

$$T, a \models \varphi \text{ if and only if } T, a \models \psi.$$

φ is equivalent to ψ (over a set C of computation trees) iff for every T (respectively, for every $T \in C$), φ is equivalent to ψ over T.

Definition 3.5 *Let C be a set of structures. TL_1 is **less or equally expressive** than TL_2 over C (notation $TL_1 \preceq^C_{exp} TL_2$), if for every formula $\varphi_1 \in TL_1$ there is a formula $\varphi_2 \in TL_2$ which is equivalent to φ_1 over C. The relations **equally expressive** (notation $\equiv^C_{exp}$) and **strictly less expressive** (notation $\prec^C_{exp}$) are defined from the relation $\preceq^C_{exp}$ as expected. When C is the class of trees, we write $\preceq_{exp}$ for $\preceq^C_{exp}$. The relations $\equiv_{exp}$ and $\prec_{exp}$ are defined similarly.*

Definition 3.6 (Expressive Completeness) *Let L be a fragment of MLO and let C be a set of trees. A temporal logic TL_1 is **expressively complete** for L over C if for every formula $\psi(x_0) \in L$ with a single free first-order variable x_0 there is a formula $\varphi \in TL_1$ which is equivalent to ψ over C, and for every formula $\varphi \in TL_1$ there is a formula $\psi(x_0) \in L$ with a single free first-order variable x_0 which is equivalent to φ over C.*

Proposition 3.7
1. *If every modality of a temporal logic TL is defined by a FOMLO truth table then every formula φ of TL is equivalent to a FOMLO formula.*
2. *If every modality of a temporal logic TL is defined by a future FOMLO truth table then every formula φ of TL is equivalent to a future FOMLO formula.*
3. *Similar to (1) and (2) with "MPL" or "MLO" replacing "FOMLO".*

Proof Straightforward induction on the structure of *TL* formulas. □

Recall that a computation tree T is an ω-model if the underlying tree $(\mid T \mid, \leq^T)$ is isomorphic to the standard order of natural numbers (i.e., it is of ω-type). A major result concerning *TL* and *FOMLO* is Kamp's theorem:

Theorem 3.8 (Kamp [23, 13])
1. *The temporal logic with two modalities $\mathsf{U}(q_1, q_2)$ ("q_1 until q_2") and $\mathsf{S}(q_1, q_2)$ ("q_1 since q_2"), is expressively complete for FOMLO over all ω-models.*
2. *The temporal logic with the single modality $\mathsf{U}(q_1, q_2)$ is expressively complete for the future fragment of FOMLO over all ω-models.*

3.4 The Sequence BTL_k

In this subsection we describe a sequence $\{BTL_k\}_{k=1}^{\infty}$ of temporal logics. In Section 5 we will show that this sequence contains an infinite hierarchy, i.e., for every k, there exists $m > k$, such that BTL_m is strictly more expressive than BTL_k.

For every $k \geq 1$, let BTL_k be the temporal logic defined as $TL(M_k)$, where

$$M_k = \{\mathbf{E}\varphi : \mathrm{qd}(\varphi(x_0, X_1, \ldots, X_l)) \leq k \textit{ and } \varphi \text{ is a first-order future formula}\}.$$

Notes.

1. For every $k \geq 1$, BTL_k is based on an infinite set of modalities. However, for every m, the number of semantically distinct BTL_k modalities of arity m is finite.
2. The basic modalities of BTL_k are path modalities. Their truth tables can be defined by MPL formulas with only one path quantifier. However, formulas of BTL_k (as formulas of every temporal logic) permit arbitrarily deep nesting of modalities and can express "branching" properties which require many path quantifiers (see proof of Lemma 6.1).
3. The basic modalities of BTL_k have a bounded quantifier depth, but formulas of BTL_k permit arbitrarily deep nesting of modalities. The next proposition shows that properties of "an arbitrary quantifier depth" can be expressed in BTL_k.

Proposition 3.9 *For every $k \geq 1$ and $n \geq 0$, there is a BTL_k formula which is equivalent to no monadic second-order logic formula of quantifier-depth $\leq n$.*

Proof There is a BTL_1 formula which expresses "there is a path from the current moment t_0 that contains at least one point $t > t_0$ labelled by P". We denote it by $\mathbf{EF}P$. Hence, the property "there is a path from the current moment that contains at least m occurrences of P" is expressible by the BTL_1 formula

$$[P \wedge \underbrace{\mathbf{EF}(P \wedge \mathbf{EF}(P \wedge \ldots \mathbf{EF}}_{m-1}(P))] \vee \underbrace{\mathbf{EF}(P \wedge \mathbf{EF}(P \wedge \ldots \mathbf{EF}}_{m}(P))).$$

Hence, in order to prove the proposition it is sufficient to show that for every n, there is m such that no monadic second-order formula of quantifier-depth $\leq n$ can express the above property.

Consider MLO formulas of the form $\varphi(x_0, P)$, i.e., φ has only x_0 as a free individual variable, P is the only unary monadic predicate name in φ (φ may contain bound individual and bound monadic predicate variables). Observe that for every n there are only finitely many semantically distinct formulas of this form having quantifier-depth $\leq n$.

Let T_m be a computation tree $(N, \leq^N, \{1, \ldots, m\})$ for the signature $\{\leq, P\}$, where N is the set of natural numbers and $\leq^N$ is the standard order relation. The above observation implies that there are $m_1 < m_2$ such that for every MLO formula $\varphi(x_0, P)$ of quantifier-depth at most n:

$$(T_{m_1}, 0) \models \varphi \text{ iff } (T_{m_2}, 0) \models \varphi$$

Therefore, no MLO formula of quantifier-depth $\leq n$ can express "there is a path from the current moment that contains at least m_2 occurrences of P". This completes the proof. $\square$

4 Equivalence and Games

Ehrenfeucht-Fraissé games are a useful tool for showing that a given class of structures is not definable in predicate logics [5, 4, 20, 21, 34] or temporal logics [12]. In this section we introduce new Ehrenfeucht-Fraissé games appropriate for BTL_k. These games are the main tool we use to prove the strictness of the hierarchy $\{BTL_k\}_{k=1}^{\infty}$ in Section 5.

The techniques, notions and results described in this section are of an independent interest. However, we never use the notions introduced here in the statements of the theorems in the later sections. The reader can follow the results (but not all the proofs) of the paper even if she/he skips this section.

4.1 Games on Chains

For the sake of brevity, linearly ordered sets with monadic predicates will be called *labeled chains* or just *chains*. Therefore, a chain is a structure for a signature $\{\leq\} \cup \{P_i \ : \ i \in Ind\}$, where Ind is a set and P_i is a monadic predicate name. If π is a path in a tree T, let T_π denote the chain corresponding to π.

The following is an important equivalence relation on chains.

Definition 4.1 ($\equiv_k$-Equivalence) *Given two chains A and A', and elements $s \in \mid A \mid$ and $s' \in \mid A' \mid$, we write $(A,\ s) \equiv_k (A',\ s')$ iff for any first-order future formula $\varphi(t_0)$ with $\mathrm{qd}(\varphi) \leq k$ we have $(A,\ s) \models \varphi(t_0)$ iff $(A',\ s') \models \varphi(t_0)$.*

In other words, $(A,\ s) \equiv_k (A',\ s')$ if no first-order future formula of quantifier-depth at most k can distinguish between these structures.

Lemma 4.2 ($\equiv_k$ are Equivalence Relations) *The relations $\equiv_k$ are equivalence relations.*

The equivalences $\equiv_k$ have a characterization in terms of the following Ehrenfeucht-Fraissé game. Given are two chains A and A' for the same signature, and elements $s_0 \in \mid A \mid$ and $s'_0 \in \mid A' \mid$. The game is played by two players *Spoiler* and *Duplicator* on $(A,\ s_0)$ and $(A',\ s'_0)$ and involves k moves of each player; the nodes s_0 and s'_0 are called the *specified* nodes for this game. *Spoiler* plays first. He chooses a node which is greater than or equal to the specified node in one of the two chains, after which *Duplicator* is obliged to respond by choosing a node in the other chain, which is greater than or equal to the specified node of that chain and she believes 'matches' the node chosen by *Spoiler*. The game continues for k moves: in every move, *Spoiler* chooses a node which is greater than or equal to the specified node in one of the two chains, and *Duplicator* responds by choosing a node in the other chain which is greater than or equal to the specified node of that chain.

If $k = 0$, no moves are played. In this case, *Duplicator* is deemed the winner if the specified nodes s_0 and s'_0 have the same labelling, i.e., $s_0 \in P_j^A$ iff $s'_0 \in P_j^{A'}$. Otherwise, for $i = 1, \ldots, k$, let s_i (respectively, s'_i) be the nodes selected in the ith move in the first (resp. second) chain. It is clear that $s_i \geq s_0$ and $s'_i \geq s'_0$ for $i = 1, \ldots, k$. *Duplicator* is deemed the winner if the mapping $s_i \mapsto s'_i$ $(i = 0, 1, \ldots, k)$ respects the relations $\leq$, and $\in P_m$, i.e., $s_i \leq^A s_j$ iff $s'_i \leq^{A'} s'_j$ and $s_i \in P_j^A$ iff $s'_i \in P_j^{A'}$.

A ***winning strategy***, for either player, is a strategy to follow, which will guarantee him or her a win, no matter what moves the other player chooses to play. If *Duplicator* (the second player) has a winning strategy, that is, a strategy to follow when choosing her responses to *Spoiler*'s moves which will guarantee her a win, then we say that $(A,\ s)$ and $(A',\ s')$ are ***k-game equivalent***, and we write $(A,\ s) \sim_k^g (A',\ s')$.

The following important theorem is a variant of Ehrenfeucht's theorem [5].

Theorem 4.3 ($\sim_k^g$ is the same as $\equiv_k$) *Given two chains A and A', and elements $a \in \mid A \mid$ and $a' \in \mid A' \mid$, we have:*

$$(A,\ a) \sim_k^g (A',\ a') \quad \textit{iff} \quad (A,\ a) \equiv_k (A',\ a')$$

Usually, Ehrenfeucht's games do not require that the player choose nodes which are greater than or equal to the specified nodes [5, 17]. The above extension for future formulas is simple.

4.2 Games on Trees

Let n, k be natural numbers. The (n, k)-games are defined as follows. The game is played by two players, *Spoiler* and *Duplicator*, on two computation trees T and T' (for the same signature). The game has n rounds. In each round a k-move game on chains is played. If $n = 0$ there are no rounds. In this case, *Duplicator* is deemed the winner if the roots of the two trees have the same labelling, i.e., $root \in P_j^T$ iff $root' \in P_j^{T'}$. Otherwise, if $n \geq 1$, each round starts by a move of *Spoiler* which chooses one of the two structures and a path from the root of this structure. *Duplicator* responds by choosing a path from the root of the other structure which she believes "matches" the path chosen by *Spoiler*. Let π (respectively π') be the path chosen in the first (respectively, second) tree. Let A (respectively A') be the chain T_π (respectively $T'_{\pi'}$).

The players play a k-game on the chains $(A, root)$ and $(A', root')$. If *Spoiler* wins in this k-game then he wins the (n,k)-game. Otherwise, let s_i and s'_i be the nodes chosen in A and A' in the ith move $(i = 1, \ldots, k)$. For the next round starting position, *Spoiler* can choose either the *root* (and then they start the new round on the trees T and T') or choose one pair $\langle s_i, s'_i \rangle$ (and they start the new round on the trees $T_{\geq s_i}$ and $T'_{\geq s'_i}$). *Duplicator* wins a game if *Spoiler* does not win it.

Notes.

1. If Spoiler chooses the pair of roots as the starting position of the next round, then the effect is the same as forgetting the last round. Hence, a smart *Spoiler* will never make such a move.

2. In Theorem 4.15 we will show that if *Duplicator* has a winning strategy in (n,k)-games on trees T and T', then T and T' are indistinguishable by BTL_k formulas of nesting n. Therefore, the number of rounds is related to the nesting of BTL_k formulas. Recall that the basic modalities of BTL_k are of the form $\mathbf{E}\varphi$, where φ is a future *FOMLO* formula. This is the reason why in every round *Spoiler* and *Duplicator* choose one path each and then play a k-move game on the corresponding chains.

We denote the (n,k)-games on two trees T and T' by $GAME_{n,k}(T, T')$.

Lemma 4.4 *For every $n \geq 0$ and $k \geq 0$, and every pair of trees T and T', either Duplicator or Spoiler has a winning strategy in $GAME_{n,k}(T, T')$.*

Proof (*Sketch*) The lemma follows directly from the fact that $GAME_{n,k}(T, T')$ is a finite game of perfect information [16, 20], and *Duplicator* wins a game if *Spoiler* does not win it. These games can be reduced to closed games under an appropriate definition of topology. The Gale-Stewart theorem [16] says that every closed game is determined, i.e., one of the players has a winning strategy.

Below, for the reader's convenience, a proof is sketched.

A position in this game records the information about the choices of nodes and paths made by *Duplicator* and *Spoiler* in each round and move starting at the beginning of the game until now; it also records who is moving next. A position is a *Spoiler* position (respectively a *Duplicator* position) if *Spoiler* (respectively, *Duplicator*) makes the next move. A position is said to be winning for a player if the player has a strategy which tells him or her how to proceed, once the play has reached this position, in such a way that he or she is guaranteed to win. The positions can be arranged into a game tree G. If there is a move which leads from a position Pos to another position Pos', we call Pos' a son of Pos. Observe that though the game tree may contain an infinite number of nodes, it has a finite depth (each player makes at most $(k+1) \times n$ moves, therefore the depth of the game tree is bounded by $2 \times (k+1) \times n$).

In order to slightly simplify our presentation we assume that no node or path is chosen twice. For a position Pos we denote by H_{Pos} the mapping which assigns to the path (or node) chosen in T at the i-th move the path (respectively, the node) chosen in T' in the i-th move; H_{Pos} also maps $root_T$ to $root_{T'}$. (Our assumption ensures that H_{Pos} is a well defined partial function; in a more general situation one has to replace H_{Pos} by a function that maps the index of a move to the associated node/path selected in T and T'.)

It is clear that a position Pos which is a leaf in G is a winning position for *Duplicator* if the H_{Pos} has the following properties: (1) $s \leq^T t$ iff $H_{Pos}(s) \leq^{T'} H_{Pos}(t)$ and (2) $s \in \pi$ iff $H_{Pos}(s) \in H_{Pos}(\pi)$ and (3) $s \in P^T$ iff $H_{Pos}(s) \in P^{T'}$, where s, t and π range over nodes and paths in the domain of H_{Pos}.

Let W be the minimal set of positions which satisfies the following conditions:

1. A leaf is in W iff it is a winning position.

2. A *Spoiler* (non-leaf) position is in W iff all its sons are in W.

3. A *Duplicator* (non-leaf) position is in W iff it has at least one son in W.

W is well-defined because G has finite depth. By induction it is easy to show that $Pos \in W$ iff *Duplicator* has a winning strategy from Pos, and $Pos \notin W$ iff *Spoiler* has a winning strategy from Pos. ☐

The (n,k)-games induce a relation on trees; we show in Lemma 4.6 that this relation is an equivalence relation.

Definition 4.5 (**(n,k)-Game-Equivalence**) *T and T' are (n,k)-**game equivalent** (notation $T \sim^g_{(n,k)} T'$) iff Duplicator has a **winning strategy** for $GAME_{n,k}(T,T')$.*

Lemma 4.6 *The relation $\sim^g_{(n,k)}$ is an equivalence relation on trees.*

Proof Reflexivity and symmetry are trivial from the definition. To prove transitivity, let $(T,\ a) \sim^g_{(n,k)} (T',\ a')$ and $(T',\ a') \sim^g_{(n,k)} (T'',\ a'')$, and let $W_{T,T'}$, $W_{T',T''}$ be the corresponding winning strategies for *Duplicator*. We will show how to construct a winning strategy $W_{T,T''}$ for *Duplicator* in $GAME_{n,k}(T,T'')$, from $W_{T,T'}$ and $W_{T',T''}$.

The proof proceeds by induction on n, the number of rounds in $GAME_{n,k}(T,T'')$, for all k simultaneously. The base $n = 0$ is trivial since if the roots of T and T' have the same labelling, and the roots of T' and T'' have the same labelling, then clearly the roots of T and T'' have the same labelling.

Inductive step: $n \mapsto n+1$. Given winning strategies for *Duplicator*, $W_{T,T'}$ in $GAME_{n+1,k}(T,T')$ and $W_{T',T''}$ in $GAME_{n+1,k}(T',T'')$ we construct the following winning strategy for *Duplicator* in $GAME_{n+1,k}(T,T'')$.

Let π be the path chosen by *Spoiler* in the beginning of the first round. Without a loss of generality, we can assume that π is in T. Let π' be the path in T', which is the response to the path π according to $W_{T,T'}$, in case it was chosen by *Spoiler* in the first round. Let π'' be the path in T'', which is the response to the path π' according to $W_{T',T''}$, in case it was chosen by *Spoiler* in the first round. *Duplicator* will use π'' as the response to π in the first round.

For every node $s_i \in T$, chosen by *Spoiler* during the first round, *Duplicator* will check the corresponding node $s'_i \in T'$, which should be the response to s_i according to $W_{T,T'}$. Then *Duplicator*'s response will be the node $s''_i \in T''$, such that s''_i is the response to s'_i according to $W_{T',T''}$. Similarly for nodes chosen by *Spoiler* from T''.

Since both strategies, $W_{T,T'}$ and $W_{T',T''}$, are winning strategies for the respective games, by the end of the first round, for every i ($i = 1, 2, \ldots, k$), we have

$$T_{\geq s_i} \sim^g_{n,k} T'_{\geq s'_i} \text{ and } T'_{\geq s'_i} \sim^g_{n,k} T''_{\geq s''_i} \tag{1}$$

Therefore, in particular, *Duplicator* will win the first round.

By the inductive hypothesis and (1), we have that for every i ($i = 1, 2, \ldots, k$), $T_{\geq s_i} \sim^g_{n,k} T''_{\geq s''_i}$. Hence, no matter what pair of nodes is chosen by *Spoiler* as a starting position for the remaining n-round game, *Duplicator* has a winning strategy for each of those rounds.

Therefore, the strategy described above is a winning strategy for *Duplicator* in $GAME_{n+1,k}(T,T'')$. $\square$

We will often use:

Lemma 4.7 *If T and T' are (n,k)-game equivalent, $m \leq n$ and $r \leq k$, then T and T' are (m,r)-game equivalent.*

4.3 Properties of (n,k)-Game-Equivalence

The relation $\sim^g_{(n,k)}$ splits the set of computation trees into equivalence classes. By the definition of the game, it is played on pairs of trees for the same signature. Hence, only trees of the same signature can be $\sim^g_{(n,k)}$-equivalent. Let C_i ($i \in I$) be the set of all $\sim^g_{(n,k)}$ -equivalence classes over computation trees for a signature τ. The signature τ and the number of the $\sim^g_{(n,k)}$-equivalence classes do not have to be finite or countable[1]. In the next definition we describe an expansion of computation trees for τ to computation trees for the signature $\tau \cup \{Q_i\ :\ i \in I\}$ (which has one new predicate name for every $\sim^g_{(n,k)}$-equivalence class).

Definition 4.8 (**(n,k)-Expansion**) *Let C_i ($i \in I$), be the equivalence classes of the relation $\sim^g_{(n,k)}$, and let $T = (\mid T\mid, \leq, \vec{P})$ be a tree. The (n,k) expansion of T (notation $EXPAN(n,k,T)$), is the tree $(\mid T\mid, \leq, \vec{P}, \ldots, Q_i, \ldots)$, where a is in Q_i ($i \in I$) if the subtree of T rooted at a is in C_i.*

[1]It can be proved that if the signature τ is finite then for every n and k the number of $\sim^g_{(n,k)}$-equivalence classes is finite. However we will not use this in what follows.

The relations $\sim^g_{(n,k)}$ enjoy the following important properties which reduces (n,k)-games on trees to the one-round games on their appropriate expansion.

Theorem 4.9 *$T \sim^g_{(n+1,k)} T'$ iff $EXPAN(n,k,T) \sim^g_{(1,k)} EXPAN(n,k,T')$.*

Proof By induction on n, the number of rounds in the game.

Basis. The basis $n = 0$ follows from the observation that T and $EXPAN(0,k,T)$ are "almost the same" computation trees. Let T and T' be computation trees for a signature $\tau = \{\leq\} \cup \{P_i : i \in Ind\}$. Observe that T and T' are $(0,k)$-equivalent iff their roots have the same labelling, i.e., $root \in P_i^T \leftrightarrow root' \in P_i^{T'}$ for $i \in Ind$. In particular, there is a one-one correspondence between the set of $(0,k)$-equivalence classes on the computation trees for the signature τ and the set of subsets of Ind. The labelling of a node $s \in\mid T \mid$ in $EXPAN(0,k,T)$ is completely determined from the labelling of s in T. Note also that for every n, the labelling of s in T is determined from the labelling of s in $EXPAN(n,k,T)$. Hence, a strategy is winning for the $(1,k)$-games on T and T' iff it is winning for the $(1,k)$-games on $EXPAN(0,k,T)$ and $EXPAN(0,k,T')$. Therefore, $T \sim^g_{(1,k)} T'$ iff $EXPAN(0,k,T) \sim^g_{(1,k)} EXPAN(0,k,T')$.

Inductive Step. For the "if" part of the theorem, assume that $EXPAN(n,k,T) \sim^g_{(1,k)} EXPAN(n,k,T')$. Let π be the path chosen by *Spoiler* in T at the beginning of the first round of $GAME_{n+1,k}(T,T')$. *Duplicator* has a one-round winning strategy in $GAME_{1,k}(EXPAN(n,k,T), EXPAN(n,k,T'))$. So, she can respond by choosing a path π' in $EXPAN(n,k,T')$ and win the k moves play on the paths π and π'. Let s_i and s'_i be the nodes chosen by *Spoiler* on π and π' after the first round. The respective subtrees of T and T' rooted at s_i and s'_i are in the same (n,k)-equivalence class. Indeed, assume that $T_{\geq s_i}$ is in an (n,k)-equivalence class C_j. Then s_i is labelled by Q_j in $EXPAN(n,k,T)$. Since *Duplicator* played according to a winning strategy in $GAME_{1,k}(EXPAN(n,k,T), EXPAN(n,k,T'))$, it follows that s'_i is labeled by Q'_j in $EXPAN(n,k,T')$ and therefore $T'_{\geq s'_i}$ is in the same (n,k)-equivalence class C_j. Hence, in the remaining n rounds of $GAME_{n+1,k}(T,T')$, *Duplicator* can follow the winning strategy for these subtrees. This completes the 'if' part of the theorem.

For the "only if" part, assume that it is not the case that $EXPAN(n,k,T) \sim^g_{(1,k)} EXPAN(n,k,T')$. Then, by Lemma 4.4, *Spoiler* has a winning strategy for $GAME_{1,k}(EXPAN(n,k,T), EXPAN(n,k,T'))$. So, in the first round of $GAME_{n+1,k}(T,T')$, *Spoiler* will follow his winning strategy on these expansions. If the nodes chosen by *Duplicator* do not respect the order relation, then she will lose in the first round. Otherwise, there is an i such that s_i and s'_i are in distinct (n,k)-equivalence classes. *Spoiler* will choose this i and by the inductive assumption he has a winning strategy in (n,k)-game on the subtrees rooted at s_i and s'_i. In the remaining n rounds he should follow this strategy.

This completes both parts of the theorem. □

Definition 4.10 (Grafting) *Let $T = (\mid T \mid, \leq_T, \ldots, P_i^T, \ldots)$ and $T_1 = (\mid T_1 \mid, \leq_{T_1}, \ldots, P_i^{T_1}, \ldots)$ be two computation trees over the same signature $\sigma = \{\leq\} \cup \{P_i : i \in Ind\}$, and let S be a subset of $\mid T \mid$. The grafting of T_1 on T at S (notation $Gr(T,S,T_1)$) is the tree $T' = (\mid T' \mid, \leq_{T'}, \ldots, P_i^{T'}, \ldots)$ over the same signature σ, defined as follows:*

Universe: $\mid T' \mid = \mid T \mid \cup \bigcup_{v \in S} \{\langle v, v_1\rangle : v_1 \in\mid T_1 \mid\}$

Interpretation of $<$: *For $a, b \in\mid T' \mid$, $a <_{T'} b$ if either: (1) $a, b \in\mid T \mid$ and $a <_T b$, or (2) $a = \langle v, u\rangle$ and $b = \langle v, u'\rangle$ and $u <_{T_1} u'$, or (3) $a \in\mid T \mid$ and $b = \langle v, u\rangle$ and $a \leq_T v$.*

Interpretation of P_i: *$a \in P_i^{T'}$ if either: (1) $a \in\mid T \mid$ and $a \in P_i^T$, or (2) $a = \langle v, u\rangle$ and $u \in P_i^{T_1}$.*

In other words, $Gr(T,S,T_1)$ is obtained from T by attaching to every node in S a tree isomorphic to T_1.

Lemma 4.11 (Grafting) *Assume that $T_1 \sim^g_{n,k} T_2$. Then $Gr(T,S,T_1) \sim^g_{n,k} Gr(T,S,T_2)$ for every T and $S \subseteq\mid T \mid$.*

Proof For every node $v \in T$, let $S_{\geq v} = \{u \in S : u \geq v\}$. Observe that the trees $Gr(T, S, T_1)_{\geq v}$ and $Gr(T, S, T_2)_{\geq v}$ are isomorphic to the trees $Gr(T_{\geq v}, S_{\geq v}, T_1)$ and $Gr(T_{\geq v}, S_{\geq v}, T_2)$ respectively.

The proof proceeds by induction on n, for all k simultaneously.

The base $n = 0$ is trivial since the roots of the trees have identical labelling.

Inductive step: $n \mapsto n+1$. Assume that in the first round of the $(n+1, k)$-game on $Gr(T, S, T_1)$ and $Gr(T, S, T_2)$ *Spoiler* has chosen a path π from the root of either tree, say $Gr(T, S, T_1)$. There are two possible cases to consider:

case 1 $\pi \subseteq\mid T \mid$. In this case *Duplicator* will choose the path $\pi' \subseteq\mid T \mid$ from the root of $Gr(T, S, T_2)$ such that π and π' are isomorphic. Then, for the k moves of the first round, *Duplicator* will use the identity function as a strategy. It is clear that in this way *Duplicator* wins the first round. Then *Spoiler* chooses a pair $\langle u, v\rangle$ of corresponding nodes as a starting position for the second round. Since in the first round *Duplicator* used the identity function to choose her responses to *Spoiler*'s moves, we have that $u = v$. The trees $Gr(T, S, T_1)_{\geq v}$ and $Gr(T, S, T_2)_{\geq u}$ are isomorphic to the trees $Gr(T_{\geq v}, S_{\geq v}, T_1)$ and $Gr(T_{\geq u}, S_{\geq u}, T_2)$ respectively. Clearly, for any two trees T' and T'', if $T' \sim^g_{n+1,k} T''$, then $T' \sim^g_{n,k} T''$. Hence, we can apply the inductive hypothesis to $T_{\geq v}$ and $S_{\geq v}$, to get $Gr(T_{\geq v}, S_{\geq v}, T_1) \sim^g_{n,k} Gr(T_{\geq v}, S_{\geq v}, T_2)$. Therefore, *Duplicator* has a winning strategy for the remaining n rounds of the game on $Gr(T, S, T_1)_{\geq v}$ and $Gr(T, S, T_2)_{\geq u}$.

case 2 π has a prefix π^p and a suffix π^s_1 such that (1) $\pi = \pi^p\pi^s_1$, there is a node $v \in T$ such that (2) π^p is the set of nodes between the root of T and v, and (3) $\pi^s_1 = \{\langle v, v_1\rangle \ : \ v_1 \in \pi_1\}$ for some path π_1 of T_1.

Let π_2 be the response to π_1 by *Duplicator*'s winning strategy WS_{n+1} for $GAME_{n+1,k}(T_1, T_2)$. *Duplicator*'s strategy is to choose the path π' in $Gr(T, S, T_2)$ such that (1) $\pi' = \pi^{p'}\pi^s_2$, (2) $\pi^{p'}$ is the set of nodes between the root of T and v (that is, $\pi^{p'}$ and π^p are isomorphic), and (3) $\pi^s_2 = \{\langle v, v_1\rangle \ : \ v_1 \in \pi_2\}$.

Then for the k moves of the first round, *Duplicator* will choose her responses according to the following strategy. For nodes chosen from π^p or $\pi^{p'}$, use the identity function on T. For nodes chosen from π^s_1 or π^s_2, use the winning strategy WS_{n+1} for the first round of the $(n+1, k)$-game on T_1 and T_2. This strategy is a winning strategy for the first round.

In the end of the first round, there will be k pairs of corresponding nodes from both paths. *Spoiler* will choose one of the pairs, $\langle w, z\rangle$ as a starting position for the second round. Now there are two subcases to consider:

subcase 2.1 $w \in T$. In this case $w = z$. This subcase is treated similarly to case 1 above.

subcase 2.2 $w = \langle v, u_1\rangle \in \pi^s_1$. In this case $z = \langle v, u_2\rangle \in \pi^s_2$. The two nodes were chosen according to the winning strategy WS_{n+1} for the first round of the $(n+1, k)$-game on T_1 and T_2, so $(T_1)_{\geq u_1} \sim^g_{n,k} (T_2)_{\geq u_2}$. The subtrees $Gr(T, S, T_1)_{\geq w}$ and $Gr(T, S, T_2)_{\geq z}$ are isomorphic to the subtrees $(T_1)_{\geq u_1}$ and $(T_2)_{\geq u_2}$. Therefore, there is a winning strategy for *Duplicator* for the remaining n rounds on the corresponding subtrees.

This completes the proof. □

The collapsed sum of two computation trees is obtained by gluing their roots (of course, the roots must have the same labelling). Formally:

Definition 4.12 (Collapsed Sum) *Let $T_1 = (\mid T_1 \mid, \leq_1, \ldots, P^1_j, \ldots)$ and $T_2 = (\mid T_2 \mid, \leq_2, \ldots, P^2_j, \ldots)$ be two computation trees over the same signature $\sigma = \{\leq\} \cup \{P_j \ : \ j \in Ind\}$, such that $root_1 \in P^1_j$ iff $root_2 \in P^2_j$ for every $j \in Ind$, where $root_1$ and $root_2$ are the roots of T_1 and T_2 respectively. The collapsed sum of T_1 and T_2 (notation $T_1 \oplus T_2$) is the tree $T = (\mid T \mid, \leq^T, \ldots, P^T_j, \ldots)$ over the same signature σ, defined as follows:*

Universe: $\mid T \mid = \{root\} \cup \{\langle 1, v\rangle \ : \ v \in \mid T_1 \mid \setminus \{root_1\}\} \cup \{\langle 2, v\rangle \ : \ v \in \mid T_2 \mid \setminus \{root_2\}\}$.

Interpretation of $\leq$: *For $a_1, a_2 \in \mid T \mid$, $a_1 \leq^T a_2$ iff $a_1 = root$ or $a_1 = \langle i, v_1\rangle$ and $a_2 = \langle i, v_2\rangle$ and $v_1 \leq_i v_2$ for $i = 1, 2$.*

Interpretation of P_j: *$a \in P_j^T$ iff either (1) $a = \langle 1, v\rangle$ and $P_j^1(v)$ or (2) $a = \langle 2, v\rangle$ and $P_j^2(v)$ or (3) $a = root$ and $P_j^1(root_1)$.*

Lemma 4.13 (Collapsed sum) *If T_1 and T_2 are compatible as described in Definition 4.12 and $T_1 \sim^g_{n,k} T_2$ then $T_1 \sim^g_{n,k} (T_1 \oplus T_2)$.*

Proof We show how to construct from a given winning strategy WS for *Duplicator* in $GAME_{n,k}(T_1, T_2)$, a winning strategy for *Duplicator* in $GAME_{n,k}(T_1, (T_1 \oplus T_2))$. The proof proceeds by induction on n, for all k simultaneously. The base $n = 0$ is trivial because the roots of T_1 and $T_1 \oplus T_2$ have the same labelling.

Inductive step: $n \mapsto n + 1$. Given a winning strategy WS for *Duplicator* in $GAME_{n+1,k}(T_1, T_2)$, we are going to construct a winning strategy for *Duplicator* in $GAME_{n+1,k}(T_1, (T_1 \oplus T_2))$.

In the first round, *Spoiler* chooses a path π from the root of either tree.

case 1 $\pi \in T_1$. Then *Duplicator* will respond by choosing the identical path in the T_1 subtree of $T_1 \oplus T_2$. *Duplicator* is guaranteed to win this round, by playing according to the identity function on T_1 (choosing the root as a response for the root, choosing $\langle 1, v\rangle$ as a response to $v \in T_1$, choosing $u \in T_1$ as a response to $\langle 1, u\rangle \in T_1 \oplus T_2$). After the first round, *Spoiler* will choose a pair of corresponding nodes as the starting position for the next round.

subcase 1.1 The chosen nodes are the roots of the trees. $T_1 \sim^g_{n+1,k} T_2$ implies $T_1 \sim^g_{n,k} T_2$ and therefore by the inductive hypothesis $T_1 \sim^g_{n,k} (T_1 \oplus T_2)$. Hence, *Duplicator* has a winning strategy for the remaining n rounds of the game.

subcase 1.2 Otherwise, both nodes are not the roots, so their corresponding subtrees are isomorphic. In this case, the identity strategy is a winning strategy for *Duplicator* for the remaining n rounds of the game.

case 2 $\pi \in (T_1 \oplus T_2)$ and all the nodes on this path, except the root, are of the form $\langle 1, v\rangle$. Then *Duplicator* will respond with the isomorphic path on T_1. This case is treated in the same way as case 1.

case 3 $\pi \in (T_1 \oplus T_2)$ and all the nodes on this path, except the root, are of the form $\langle 2, v\rangle$. Then *Duplicator* will reply according to the winning strategy WS for $GAME_{n+1,k}(T_1, T_2)$ to win the first round. For the next round starting position, *Spoiler* will choose a pair of corresponding nodes.

subcase 3.1 The case where both chosen nodes are the roots of the trees is treated exactly as subcase 1.1 above.

subcase 3.2 Otherwise, let $\langle 2, v\rangle$ and u be the pair of starting nodes chosen for the next round by *Spoiler* from $T_1 \oplus T_2$ and T_1 respectively. Observe that

$$(T_1)_{\geq u} \sim^g_{n,k} (T_2)_{\geq v} \tag{2}$$

because the first round was played according to the winning strategy for *Duplicator* in $GAME_{n+1,k}(T_1, T_2)$. Let WS' be the winning strategy for *Duplicator* in $GAME_{n,k}\big((T_1)_{\geq u}, (T_2)_{\geq v}\big)$ (by (2) such a strategy exists). The trees $(T_2)_{\geq v}$ and $(T_1 \oplus T_2)_{\geq \langle 2,v\rangle}$ are isomorphic. Hence, WS' is a winning strategy for *Duplicator* for the remaining n rounds of the game.

This completes the proof. □

4.4 Soundness

Recall the definition of the sequence $\{BTL_k\}_{k=1}^{\infty}$ in Section 3.4 . The following definition and theorem relate BTL_k with games on trees.

Definition 4.14 ($\equiv_{n,k}$-Equivalence) *Trees T and T' are equivalent modulo n, k (notation $\equiv_{n,k}$), if for every BTL_k formula φ of nesting-depth at most n*

$$T, root \models \varphi \quad \textit{if and only if} \quad T', root' \models \varphi$$

Theorem 4.15 **($\sim^g_{n,k}$-Soundness)** *Let $T = (| T |, \leq_T, \vec{P})$ and $T' = (| T' |, \leq_{T'}, \vec{P'})$ be two trees. If $T \sim^g_{n,k} T'$ then $T \equiv_{n,k} T'$.*

Proof We show that if a BTL_k formula φ of nesting-depth n distinguishes between $(T, root)$ and $(T', root')$, then *Spoiler* has a winning strategy in the (n, k)-game on T and T'. The proof proceeds by the structural induction on formulas. The induction base is trivial. The case of Boolean connectives is also immediate. The only non-trivial case is when φ has the form $m(\varphi_1, \varphi_2, \ldots \varphi_r)$ where m is a BTL_k modality. Assume that the theorem holds for n and let us show that it holds for $n + 1$.

Let $\varphi = m(\varphi_1, \varphi_2, \ldots \varphi_r)$ be a BTL_k formula of nesting $n + 1$ that distinguishes between T and T'. Without a loss of generality we can assume that

$$T, root \models \varphi \text{ and } T', root' \not\models \varphi \tag{3}$$

Assume that m is a modality that corresponds to a formula $\psi(x_0, X_0, X_1, \ldots X_r)$, i.e. ψ is a first-order future formula of quantifier-depth at most k and for every tree $T'' = (| T'' |, \leq_{T''}, \vec{R''})$,

$$T'', s \models m(X_1, \ldots, X_r) \text{ iff } T''_{\pi''}, s \models \psi(x_0, X_0, X_1, \ldots X_r) \text{ for some path } \pi'' \text{ starting at } s. \tag{4}$$

By (3) and (4), there is a path π in T such that

$$A_\pi, root \models \psi(x_0, X_0, X_1, \ldots X_r) \tag{5}$$

where A_π is the chain $(| \pi |, \leq_\pi, R_1, \ldots R_r)$ with $\leq_\pi$ inherited from $\leq_T$ and $s \in R_i$ iff $T_{\geq s}, s \models \varphi_i$ (for $i = 1, \ldots, r$). Similarly for every π' in T'

$$A'_{\pi'}, root' \not\models \psi(x_0, X_0, X_1, \ldots X_r) \tag{6}$$

where $A'_{\pi'}$ is the chain $(| \pi' |, \leq_{\pi'}, R'_1, \ldots R'_r)$ with $\leq_{\pi'}$ inherited from $\leq_{T'}$ and $s \in R'_i$ iff $T'_{\geq s}, s \models \varphi_i$ (for $i = 1, \ldots, r$).

From Theorem 4.3, (5) and (6), it follows that *Spoiler* has a winning strategy $W_{\pi,\pi'}$ for the k-game on chains $(A_\pi, root)$ and $(A'_{\pi'}, root')$. We are going to construct a winning strategy for *Spoiler* in the $(1, k)$-game on $EXPAN(n, k, T)$ and $EXPAN(n, k, T')$. This together with Theorem 4.9 will imply the desired result that not $T \sim^g_{n+1,k} T'$.

Spoiler's strategy is as follows. In the first move, he chooses a path π in T that satisfies (5). Let π' be the path chosen by *Duplicator*. Now *Spoiler* will follow the strategy $W_{\pi,\pi'}$ given for the k-game on chains $(A_\pi, root)$ and $(A'_{\pi'}, root')$. Let s_i (respectively, s'_i) be the nodes chosen in π (resp. π') after the first round. Since $W_{\pi,\pi'}$ is *Spoiler*'s winning strategy for the k-game on chains $(A_\pi, root)$ and $(A'_{\pi'}, root')$, it follows that either

case 1 The mapping $s_i \mapsto s'_i$ does not preserve the order relation. In this case *Spoiler* wins the game on $EXPAN(n, k, T)$ and $EXPAN(n, k, T')$, or

case 2 The mapping $s_i \mapsto s'_i$ does not preserve the labelling of the chains. In this case there is $l \leq k$ and h such that either $s_l \in R_h$ while $s'_l \notin R'_h$ or $s_l \notin R_h$ while $s'_l \in R'_h$. Without a loss of generality, we can assume that $s_l \in R_h$ while $s'_l \notin R'_h$. Hence,

$$T_{\geq s_l}, s_l \models \varphi_h \text{ and } T'_{\geq s'_l}, s'_l \not\models \varphi_h.$$

The nesting-depth of φ_h is at most n. Hence, by the induction hypothesis $T_{\geq s_l}$ and $T'_{\geq s'_l}$ are not $\sim^g_{n,k}$-equivalent. Let C_j (respectively, $C_{j'}$ for $j' \neq j$) be the $\sim^g_{n,k}$-equivalence class of $T_{\geq s_l}$ (respectively, of $T'_{\geq s'_l}$). In $EXPAN(n, k, T)$ the node s_l is labelled by Q_j and s_l is not labelled by $Q_{j'}$. In $EXPAN(n, k, T')$ the node s'_l is labelled by $Q_{j'}$ and s'_l is not labelled by Q_j. Therefore, the mapping $s_i \mapsto s'_i$ does not respect labelling in $EXPAN(n, k, T)$ and $EXPAN(n, k, T')$. Hence *Spoiler* wins also in this case.

This completes the proof. $\square$

5 A Hierarchy

The main result of this section is:

Theorem 5.1 (Hierarchy) *For every $k \geq 1$, there exists $k' > k$, such that $BTL_{k'}$ is strictly more expressive than BTL_k.*

We use the following simple property to show that $\{BTL_k\}_{k=1}^{\infty}$ contains a true infinite hierarchy.

Definition 5.2 ($Block_k$) *For $k \geq 1$ let $Block_k$ be a property of trees with two unary predicates P and Q, defined as follows. $T \in Block_k$ iff there is a path π starting at the root of T such that:*

(1) There is a node $v \in \pi$ such that $v \in Q$;

(2) For every node $u \leq v$ such that $u \in P$ there is a sequence $v_1, v_2 \ldots, v_k$ of k consecutive nodes such that $v_i \leq v \wedge v_i \in P$ (for $i = 1, \ldots, k$) and $u = v_j$ for some $j \in \{1, \ldots, k\}$;

(3) There is no sequence of $k+1$ consecutive P-labeled nodes on π between the root and v;

(4) The root of T is labeled by P.

We say that a formula φ *expresses* $Block_k$ (or $Block_k$ is expressed by φ), whenever φ is such that:

$$T, root \models \varphi \text{ iff } T \text{ has the property } Block_k.$$

The following lemma is immediate.

Lemma 5.3 *For every $k \geq 1$, there is $k' \geq 1$ such that $Block_k$ is expressible in $BTL_{k'}$.*

Theorem 5.1 follows from Lemma 5.3 and the following inexpressiveness result for BTL_k.

Theorem 5.4 *Let $m_k = \lfloor log_2(k+1) \rfloor$ for $k \geq 1$. Then there is no BTL_{m_k} formula which expresses the property $Block_k$.*

From our proof of Theorem 5.1 we obtain a stronger result:

Theorem 5.5 *For all $m \geq 1$, BTL_m is strictly less expressive than BTL_{m+3}.*

The proof of Theorem 5.5 will be given in subsection 5.2.

5.1 Proof of Theorem 5.4

The proof technique for theorem 5.4 is standard [10]. For every $k \geq 1$, we inductively define two sequences $M_1^k, M_2^k, M_3^k, \ldots$ and $N_1^k, N_2^k, N_3^k, \ldots$ of trees (see below). All the trees in the sequence N_i^k have the property $Block_k$, while all the trees in the sequence M_i^k do not have it. However, we show in lemma 5.9 that each formula ψ of $BTL_{\lfloor log_2(k+1) \rfloor}$ of nesting-depth at most j, cannot distinguish between M_j^k and N_j^k; i.e., $M_j^k, root \models \psi$ iff $N_j^k, root \models \psi$ for every formula ψ of BTL_{m_k} of nesting-depth at most j. Therefore, no BTL_{m_k} formula can express the property $Block_k$.

The proof is based on the following constructions and lemmas.

For every $k \geq 1$, we inductively define two sequences $M_1^k, M_2^k, M_3^k, \ldots$ and $N_1^k, N_2^k, N_3^k, \ldots$ of computation trees with two monadic predicates P and Q as follows. Trees M_1^k and N_1^k are described in figure 1. Both M_1^k and N_1^k are paths: M_1^k is of length $k+2$ and N_1^k is of length $k+1$. The last node (denoted c) in M_1^k (resp. N_1^k) is the only Q-labeled node in these trees. The node c is also labeled by $\neg P$, while all other nodes are labeled by P. Note that the difference between M_1^k and N_1^k is that M_1^k has $k+1$ nodes $a_1^1, a_1^2, \ldots, a_1^k, a_1^{k+1}$ labeled by P, while N_1^k has only k nodes $a_1^1, a_1^2, \ldots, a_1^{k-1}, a_1^k$ labeled by P.

Figure 2 shows how M_{i+1}^k (resp. N_{i+1}^k) is constructed from M_i^k and N_i^k. Both trees are constructed from an infinite path of P-labeled nodes $a_{i+1}^1, a_{i+1}^2, \ldots$ (these nodes are also labeled with $\neg Q$).

For every $i \geq 0$ and $j \geq 1$, the node a_{i+1}^j is connected to nodes b_{i+1}^j labeled by $\neg P$ and $\neg Q$.

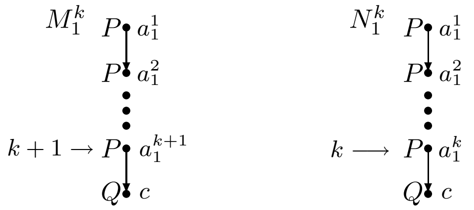

Figure 1: M_1^k and N_1^k

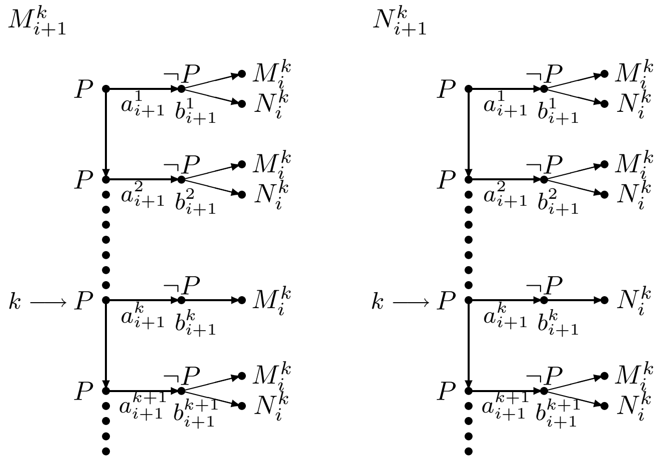

Figure 2: M_{i+1}^k and N_{i+1}^k

The node b_{i+1}^k has one son. The subtree rooted at the unique son of b_{i+1}^k in M_{i+1}^k is isomorphic to M_i^k, while the subtree rooted at the unique son b_{i+1}^k in N_{i+1}^k is isomorphic to N_i^k.

For $j \neq k$, every b_{i+1}^j has two sons. The subtree rooted at one son is isomorphic to M_i^k and the subtree rooted at the second son is isomorphic to N_i^k.

The following Lemma is a simple observation on the tree sequences described above.

Lemma 5.6 *For every $k \geq 1$ and $i \geq 1$, the tree N_i^k has the $Block_k$ property, while M_i^k does not have it.*

The following is well known (e.g., see [4, 34]):

Lemma 5.7 *Duplicator has a winning strategy in games on chains with at most $m_k = \lfloor log_2(k+1) \rfloor$ moves, played on two unlabelled linear orders of lengths $\geq k$.*

Denote the well known strategy from the above lemma by W. We use W to obtain the following winning strategy for *Duplicator* in the $(1, m_k)$-game on M_1^k and N_1^k: if *Spoiler* chooses a node a_1^j, *Duplicator* replies according to W by choosing a node a_1^i in the other structure. Otherwise, *Spoiler* chooses the node c and *Duplicator* replies by choosing c in the other structure. Note that the above strategy applied by *Duplicator* always obtains agreement on labels. Therefore, by the Soundness Theorem 4.15, we have:

Lemma 5.8 *$M_1^k \equiv_{1,m_k} N_1^k$ for every $k \geq 1$,*

Lemma 5.9 *$M_i^k \equiv_{i,m_k} N_i^k$ for every $k \geq 1$ and $i \geq 1$.*

Proof We show that M_i^k and N_i^k are (i, m_k)-game equivalent. Therefore, by the Soundness Theorem 4.15, we obtain that $M_i^k \equiv_{i,m_k} N_i^k$. The proof proceeds by induction on i, for all k simultaneously. The base $i = 1$ is just Lemma 5.8.

Inductive step: $i \mapsto i+1$. Given a winning strategy for *Duplicator* for the (i, m_k)-game on M_i^k and N_i^k, we are going to construct a winning strategy for *Duplicator* when playing a game of $i+1$ rounds on M_{i+1}^k and N_{i+1}^k with at most m_k moves in each round.

By the inductive hypothesis we have that

$$M_i^k \sim^g_{i,m_k} N_i^k \tag{7}$$

Therefore, from the grafting Lemma 4.11 and the collapsed sum Lemma 4.13, we have that for every $h \geq 1$ and $h' \geq 1$:

$$(M^k_{i+1})_{\geq b^h_{i+1}} \sim^g_{i,m_k} (N^k_{i+1})_{\geq b^{h'}_{i+1}} \tag{8}$$

$$(M^k_{i+1})_{\geq a^h_{i+1}} \sim^g_{i,m_k} (N^k_{i+1})_{\geq a^{h'}_{i+1}} \tag{9}$$

case 1 In the first round *Spoiler* has chosen a path π in either tree, which does not include the node b^k_{i+1}. In this case, *Duplicator* replies with the isomorphic path in the other tree. For the rest of the round, *Duplicator* will use the identity function on the two paths. Clearly, *Duplicator* wins the first round. *Spoiler* chooses a pair $\langle u, v\rangle$ of corresponding nodes as the starting position for the next round. The following subcases are possible:

subcase 1.1 $u \geq b^h_{i+1}$ and $v \geq b^h_{i+1}$. In this case the corresponding subtrees $(M^k_{i+1})_{\geq u}$ and $(N^k_{i+1})_{\geq v}$ are isomorphic. Therefore, *Duplicator* will use the identity function as a winning strategy for the remaining i rounds of the game.

subcase 1.2 $u = a^h_{i+1}$ and $v = a^h_{i+1}$. In this case the corresponding subtrees are not necessarily isomorphic, but according to (9), they are (i, m_k)-game equivalent. Therefore, *Duplicator* has a winning strategy for the remaining i rounds of the game.

case 2 *Spoiler* has chosen in the first round a path π which includes b^k_{i+1} (in either trees), say M^k_{i+1}. Let π_1 (resp. π_2) be the prefix (suffix) of π, which ends at b^k_{i+1} (begins after b^k_{i+1}). Let π'_2 be the path isomorphic to π_2, which begins after the node b^{k+1}_{i+1} in N^k_{i+1}. *Duplicator*'s strategy will be to choose the path $\pi' = \pi'_1\pi'_2$ in N^k_{i+1}, where $\pi'_1 = \langle a^1_{i+1}, a^2_{i+1}, \ldots a^{k+1}_{i+1}, b^{k+1}_{i+1}\rangle$.

The only difference between π and π' is that π starts with k consecutive P-labeled nodes, while π' starts with $k+1$ consecutive P-labeled nodes. Thus, *Duplicator*'s strategy for the first round will be to choose her responses to nodes chosen by *Spoiler* as follows: for nodes chosen in π_2 or π'_2, use the identity function; for b^k_{i+1} reply by choosing b^{k+1}_{i+1}, and for other nodes chosen in π_1 or π'_1, use the well known strategy W (introduced in Lemma 5.7).

Clearly, this strategy is a winning strategy for *Duplicator* in the first round. In the end of the first round, *Spoiler* will choose a pair of corresponding nodes $\langle u, v\rangle$, as the starting position for the next round. Since *Duplicator* has played the first round according to the strategy described above, only the following cases are possible:

subcase 2.1 $u \in \pi_2$ and $v \in \pi'_2$. In this case, the subtrees rooted at these nodes, $(M^k_{i+1})_{\geq u}$ and $(N^k_{i+1})_{\geq v}$ are isomorphic, therefore (i, m_k)-game equivalent.

subcase 2.2 $u = b^k_{i+1}$ and $v = b^{k+1}_{i+1}$. By (8) the corresponding subtrees are (i, m_k)-game equivalent.

subcase 2.3 $u \in \pi_1$ and $v \in \pi'_1$. By (9) the subtrees rooted at these nodes, $(M^k_{i+1})_{\geq u}$ and $(N^k_{i+1})_{\geq v}$, are (i, m_k)-game equivalent.

Since in all cases the starting position for the second round involves (i, m_k)-game equivalent subtrees, *Duplicator* has a winning strategy for the remaining i rounds of the game.

This completes the proof of Lemma 5.9. □

Finally, Theorem 5.4 follows from Lemma 5.6 and Lemma 5.9.

5.2 BTL_m is strictly less expressive than BTL_{m+3}

We use the following simple property to separate between BTL_m and BTL_{m+3}.

Definition 5.10 ($Path_k$) *For $k \geq 1$, let $Path_k$ be a property of trees with one unary predicate P defined as follows: $T \in Path_k$ iff there is a path π starting at the root of T such that*

(1) For every node $u \in \pi$ such that $u \in P$ there is a sequence $v_1, v_2 \dots, v_k$ of k consecutive nodes such that $v_i \in \pi \wedge v_i \in P$ (for $i = 1, \dots, k$) and $u = v_j$ for some $j \in \{1, \dots, k\}$;

(2) There is no sequence of $k+1$ consecutive P-labeled nodes on π;

(3) The root of T is labeled by P.

It is instructive to observe that the property $Path_k$ is more "complex" than $Block_k$ (See def. 5.2). $Block_k$ is expressible in *FOMLO*, while $Path_k$ is not [26].

Lemma 5.11 *1. For every $k \geq 1$, the property $Block_{2^k}$ is expressible in BTL_{k+4}.*

2. For every $k \geq 1$, the property $Path_{2^k}$ is expressible in BTL_{k+3}.

Proof It is well-known that there are *FOMLO* formulas $dist_{2^k}(x_0, x_1)$ of quantifier-depth $k+1$ which express the property "$x_0 \leq x_1$ and there are at most $2^k - 1$ nodes between x_0 and x_1".

These formulas are inductively defined as follows:

$$dist_1(x_0, x_1) \triangleq x_0 < x_1 \wedge \forall x(x < x_1 \rightarrow x \leq x_0)$$

$$dist_{2^{k+1}}(x_0, x_1) \triangleq \exists x\big(dist_{2^k}(x_0, x) \wedge dist_{2^k}(x, x_1)\big)$$

For $k \in N$ we define $H_{2^k}(x_0, x_1)$ as

$$H_{2^k}(x_0, x_1) \triangleq dist_{2^k}(x_0, x_1) \wedge \forall x(\ x_0 < x \leq x_1 \rightarrow x \in P)$$

The property $Block_{2^k}$ is expressible by the following *FOMLO* formula:

$$\alpha_k(x_0, P, Q) \triangleq x_0 \in P \wedge \exists x_1 \left(x_1 \in Q \wedge \left[\begin{array}{c} x_0 < x_1 \wedge \big(\forall x\big)_{x_0}^{x_1}(x \in P \rightarrow \neg\exists x_3 \leq x_1.\ H_{2^k}(x, x_3)) \\ \bigwedge \\ (\forall x)_{x_0}^{x_1}\big((x \notin P \wedge \exists x_2 \leq x_1. H_1(x, x_2)) \rightarrow \exists x_3 \leq x_1.\ H_{2^k}(x, x_3)\big) \end{array} \right] \right)$$

(Here "$(\forall x)_{x_0}^{x_1}\varphi$" abbreviates "$\forall x.(x_0 \leq x \leq x_1) \rightarrow \varphi$".)

Finally, observe that α_k has quantifier-depth $k+4$ and it is equivalent to a BTL_{k+4} basic modality $\mathbf{E}\alpha_k$. This completes the proof of (1).

(2) Observe that the formula $\mathbf{E}\beta_k$, where β_k is the following formula of quantifier-depth $k+3$, expresses $Path_{2^k}$:

$$\beta_k(x_0, P) \triangleq x_0 \in P \wedge \left[\begin{array}{c} \forall x\big((x_0 \leq x \wedge x \in P) \rightarrow \neg\exists x_3.\ H_{2^k}(x, x_3)\big) \\ \bigwedge \\ \forall x\big((x_0 \leq x \wedge x \notin P \wedge \exists x_2 H_1(x, x_2)) \rightarrow \exists x_3.\ H_{2^k}(x, x_3)\big) \end{array} \right]$$

$\square$

To complete the proof that BTL_m is strictly less expressive than BTL_{m+3} we proceed as follows.
Proof (of Theorem 5.5) In the proof of Theorem 5.4, we have defined two sequences of computation trees N_i^k and M_i^k (see figures 1 and 2). Observe that for every $k \geq 1$ and $i \geq 1$, the tree N_i^k has the $Path_k$ property, while M_i^k does not have it. Moreover, by Lemma 5.9, $M_i^{2^k} \equiv_{i,k} N_i^{2^k}$ for every $k \geq 1$ and $i \geq 1$. Hence, no BTL_k formula can express the property $Path_{2^k}$ which is expressible in BTL_{k+3} by Lemma 5.11. This completes the proof. $\square$

6 On Temporal Logics over Branching Time

The infinite hierarchy of temporal logics BTL_k defined in Section 3.4 can serve as an external "yardstick" against which other temporal logics can be compared. Below we examine some commonly used temporal logics. These logics are based on different finite and infinite sets of future modalities. Recall from Section 3.1 that $TL(M)$ is the temporal logic based on the set of modalities M. A set of modalities M is a ***base*** for a temporal logic L if $L \equiv_{exp} TL(M)$. Note that the modalities in M do not have to be basic modalities of L.

6.1 PLTL

The Propositional Linear time Temporal Logic ($PLTL$) [30] is usually referred to as the standard linear time temporal logic (see e.g. survey [8]). By definition $PLTL$ is just $TL(\mathbf{U})$. Of course, $TL(\mathbf{U})$ is interpreted not only over linear orders, but also over arbitrary partial orders; in particular, over trees. The "linear time" appears in the name of this logic probably because when it was introduced by Pnueli its intended models were linear orders; more precisely, ω-models. The adjective "standard" is probably due to the Kamp theorem which states that it is expressively equivalent (over the ω-models) to (the future fragment of) $FOMLO$ - a very robust formalism.

Observe that $\mathbf{U}$ has a first-order truth table $\varphi_{\mathbf{U}}$ of quantifier depth two. Moreover, $\varphi_{\mathbf{U}}$ is equivalent over trees to $\mathbf{E}\varphi_{\mathbf{U}}$. Therefore, $PLTL$ is expressively equivalent to a fragment of BTL_2.

6.2 CTL^*

Computational Tree Logic (CTL) was introduced in [2]. It is based on two binary modalities $\mathbf{EU}$ and $\mathbf{AU}$; $\mathbf{AU}(X, Y)$ (respectively $\mathbf{EU}(X, Y)$) holds at a current moment t_0 if "for all (respectively, for some) paths from the current moment, X until Y holds". The modality $\mathbf{EU}$ is equivalent to $\mathbf{U}$. The truth table for $\mathbf{AU}(X, Y)$ is $\neg\mathbf{E}\neg\varphi_{\mathbf{U}}$, where $\varphi_{\mathbf{U}}$ is the truth table of $\mathbf{U}$.

In contrast to expressive completeness of $TL(\mathbf{U})$ over ω-models (the canonical linear models), there is no natural predicate logic which corresponds to CTL ($= TL(\mathbf{EU}, \mathbf{AU}) \equiv_{exp} TL(\mathbf{U}, \mathbf{E}\neg\varphi_{\mathbf{U}})$) over the class of trees. Moreover, it turns out that CTL cannot express many natural properties over trees [10].

The logic CTL^* suggested in [10] is much more expressive. The definition of CTL^* [10] uses an interplay between state formulas (which correspond to genuine modalities) and path formulas (which play an auxiliary role). Below we recall the syntax and the semantics of CTL^* and then show that it is expressively equivalent to the union of BTL_k.

The syntax of CTL^* inductively defines a class of state formulas and a class of path formulas using the rules below:

S1 Each atomic formula is a state formula.

S2 if p and q are state formulas then so are $p \wedge q$ and $\neg q$.

S3 If p is a path formula then $\mathbf{E}p$ is a state formula.

P1 Each state formula is also a path formula.

P2 if p and q are path formulas then so are $p \wedge q$ and $\neg q$.

P3 If p and q are path formulas then so is $p\mathbf{U}q$.

In addition to the standard abbreviation for propositional connectives, $\mathbf{A}p$ ("*for all paths, p*") abbreviates $\neg\mathbf{E}\neg p$.

CTL^* formulas are interpreted over trees. Given a tree T, a node s in this tree and a path π through this tree, we write $(T, s) \models q$ to mean that state formula q is true at node s in the tree T; we write $(T, \pi) \models p$ to mean that path formula p is true at the path π in T.

These relations are defined inductively as follows:

S1 $(T, s) \models P$ iff $s \in P^T$

S2 The standard rules for $\wedge$ and $\neg$.

S3 $(T,s) \models \mathbf{E}p$ iff there is a path π in T,which starts at s such that $(T,\pi) \models p$.

P1 $(T,\pi) \models q$ iff $(T,s) \models q$, where s is the first state of π.

P2 The standard rules for $\wedge$ and $\neg$.

P3 $(T,\pi) \models p\mathbf{U}q$ iff there is i with $1 \leq i \leq \mathrm{length}(\pi)$ such that $(T,\pi^i) \models q$ and $(T,\pi^k) \models p$ whenever $1 \leq k < i$ (for a path $\pi = \langle s_0, s_1, \ldots s_j, s_{j+1}, \ldots \rangle$, π^j denotes the path $\langle s_j, s_{j+1}, \ldots \rangle$).

Observe that if $\varphi(X_1,\ldots,X_l)$ is a $TL(\mathbf{U})$ formula then $\mathbf{E}\varphi$ is a CTL^* state formula. We can associate with $\mathbf{E}\varphi$ the following l-place temporal operator $\#$: for every tree T and subsets $R_i \subseteq \mid T \mid$:

$$\#(R_1,R_2,\ldots,R_l) = \{s \in \mid T \mid \ : \ ((T,R_1,R_2,\ldots,R_l),s) \models \mathbf{E}\varphi(X_1,X_2,\ldots,X_l)\}$$

By the abuse of notations we denote this operator by $\mathbf{E}\varphi$. Observe that by Proposition 3.7(1), there is a *FOMLO* formula $\psi(x_0,X_1,\ldots,X_l)$ which is equivalent to φ. Moreover, the path modality $\mathbf{E}\psi$ (see Definition 3.3) defines the same temporal operators as $\mathbf{E}\varphi$. Let M be the set of modalities

$$M \triangleq \{\mathbf{E}\varphi : \varphi \in TL(\mathbf{U})\}.$$

From the above observation it follows that $TL(M) \preceq_{exp} \bigcup_{k=1}^{\infty} BTL_k$. Kamp's theorem implies that for every formula $\psi(x_0,X,\ldots,X_l)$ in the future fragment of *FOMLO* there is φ in $TL(\mathbf{U})$ such that the temporal operators $\mathbf{E}\varphi$ and $\mathbf{E}\psi$ are the same. Therefore, the temporal logics $TL(M)$ and $\bigcup_{k=1}^{\infty} BTL_k$ are expressively equivalent.

The following result is due to Emerson (see the last paragraph of section 2.4 in[8]).

Lemma 6.1 *CTL^* is expressively equivalent to $TL(M)$.*

Proof (Sketch) The direction $TL(M) \preceq_{exp} CTL^*$ follows from the observation that every modality in M is defined by a CTL^* formula.

We illustrate the proof of the other direction by a generic example. Consider the following CTL^* formula:

$$\alpha \triangleq \mathbf{E}[P_1\mathbf{U}P_2 \wedge (\neg\mathbf{E}(P_3\mathbf{U}(P_2\mathbf{U}(\mathbf{E}P_2\mathbf{U}(P_2 \wedge P_3)))))].$$

We will translate it into an equivalent $TL(M)$ formula starting from the innermost occurrences of $\mathbf{E}$.

Let $m_1(X_1,X_2)$ be the modality $\mathbf{E}X_1\mathbf{U}(X_1 \wedge X_2)$ which "corresponds" to the innermost occurrence of $\mathbf{E}$ in the above CTL^* formula. Note that $m_1(P_2,P_3)$ is equivalent to the formula which follows the innermost occurrence of $\mathbf{E}$. Let $m_2(X_1,X_2,X_3)$ be the modality $\mathbf{E}(X_1\mathbf{U}(X_2\mathbf{U}X_3))$. Observe that $\mathbf{E}(P_3\mathbf{U}(P_2\mathbf{U}(\mathbf{E}P_2\mathbf{U}(P_2 \wedge P_3))))$ is equivalent to $m_2(P_3,P_2,m_1(P_2,P_3))$. Finally let $m_3(X_1,X_2,X_3)$ be $\mathbf{E}(X_1UX_2 \wedge \neg X_3)$. Observe that the $TL(M)$ formula $m_3(P_1,P_2,m_2(P_3,P_2,m_1(P_2,P_3)))$ is equivalent to α. $\square$

To summarize, we have demonstrated that the following temporal logics are expressively equivalent:

Lemma 6.2 $\bigcup_{k=1}^{\infty} BTL_k \equiv_{exp} CTL^* \equiv_{exp} TL(\{\mathbf{E}\varphi : \varphi \in TL(U)\})$.

6.3 CTL^* Has No Finite Base

Recall that a temporal logic L has a finite base iff there is a finite set M of modalities such that L is expressively equivalent to $TL(M)$.

Since $\{BTL_k\}_{k=1}^{\infty}$ contains a true infinite hierarchy (Theorem 5.1), from Lemma 6.2 we obtain:

Theorem 6.3 *CTL^* has no finite base.*

The following Theorem was proved in [28].

Theorem 6.4 *CTL^* is expressively equivalent to the bisimulation invariant fragment of monadic path logic.*

Bisimulation equivalence plays a very important role in concurrency. This equivalence catches subtle differences between trees based on their branching structures. It is generally regarded as the finest behavioral equivalence of interest for concurrency. A formula $\varphi(x_0, X_1, \ldots, X_l)$ is bisimulation invariant if $T, root \models \varphi(x_0, X_1, \ldots, X_l)$ implies $T', root' \models \varphi(x_0, X_1, \ldots, X_l)$ whenever T and T' are bisimulation equivalent. Thus, CTL^* represents some objectively quantified expressive power.

It is easy to see that the property $Block_k$ is expressible in $FOMLO$. In addition, since $Block_k$ is expressible in CTL^*, it is clear that $Block_k$ is a bisimulation invariant property. Therefore, we obtain the following theorem:

Theorem 6.5 *The bisimulation invariant fragment of future first-order monadic logic of order has no finite base.*

Hence, the situation for temporal logics over trees (branching time models) is completely different than the situation for temporal logics over linear time, where the temporal logic based on the single modality $\mathbf{U}$ is expressively equivalent (over ω-chains), to the future fragment of first-order monadic logic of order [23, 14]. We believe that this is the reason for the multiplicity of temporal logics over branching time.

6.4 BTL_k vs. Commonly Used Branching Time Logics

Many temporal logics were suggested as branching time specification formalisms (see [11, 7]) by imposing some syntactic restrictions on CTL^* formulas. We examine the expressive power of commonly used branching time temporal logics. It turns out that almost all of these logics are inside the second level of our hierarchy. The modalities for these logics were suggested by desire to formalize some pragmatic properties which often occur in specifications of hardware and software systems. It is instructive to observe that most of these properties can be formalized by BTL_2 formulas constructed from basic modalities of quantifier-depth two.

In the following list we use the symbols $\mathbf{U}, \mathbf{F}, \mathbf{G}$ to indicate the non-strict versions of the respective temporal operators (see Example 3.1). $\mathbf{F}^\infty p$ abbreviates $\mathbf{GF}p$ and $\mathbf{G}^\infty p$ abbreviates $\neg\mathbf{F}^\infty\neg p$; its meaning on linear orders is '*almost everywhere p*'. The meaning of the modality $\mathbf{X}p$ is 'next time p'. Recall also that $\mathbf{A}\varphi$ ("*for all paths, φ*") abbreviates $\neg\mathbf{E}\neg\varphi$.

$B(\mathbf{F})$ **(see [25, 7]).** Let $M_1 = \{\mathbf{EG}, \mathbf{EF}, \mathbf{AG}, \mathbf{AF}\}$, then $B(\mathbf{F})$ can be defined as $TL(M_1)$. Since the truth tables of $\mathbf{F}$ and $\mathbf{G}$ have quantifier-depth 1 and for every formula p, $\mathbf{AF}p = \neg\mathbf{EG}\neg p$ and $\mathbf{AG}p = \neg\mathbf{EF}\neg p$, $B(\mathbf{F}) \preceq_{exp} BTL_1$. According to [7], the formula $\mathbf{E}(\mathbf{F}p \wedge \mathbf{G}q)$ is not expressible in $B(\mathbf{F})$. Since this formula is expressible in BTL_1, it follows that $B(\mathbf{F}) \prec_{exp} BTL_1$.

UB **(see [1]).** UB can be defined as $TL(\mathbf{EF}, \mathbf{EG}, \mathbf{EX})$. Since $\mathbf{AX}p = \neg\mathbf{EX}\neg p$, adding $\mathbf{AX}$ as a basic modality will not increase the expressive power of UB. The truth table of $\mathbf{X}$ has quantifier-depth 2, hence $UB \preceq_{exp} BTL_2$.

CTL **and** CTL^+ **(see [2]).** CTL can be defined as $TL(\mathbf{EX}, \mathbf{AX}, \mathbf{EU}, \mathbf{AU})$. Since the truth table of the $\mathbf{U}$ operator has quantifier-depth 2, we have $CTL \preceq_{exp} BTL_2$. Let Φ_1 be the set of $TL(\mathbf{U})$ formulas of nesting-depth ≤ 1. Let M_2 be the infinite set $\{\mathbf{E}\varphi : \varphi \in \Phi_1\}$ of path modalities, then CTL^+ is defined as $TL(M_2)$ and hence $CTL^+ \preceq_{exp} BTL_2$.

CTL^2 **(see [24]).** The syntax of CTL^2 contains two kinds of auxiliary formulas: path formulas of degree one and path formulas of degree two. However, CTL^2 is expressively equivalent to the extension of CTL by the binary modality $\mathbf{E}(\mathbf{G}(X_1\mathbf{U}X_2))$. Observe that $\mathbf{G}(X_1\mathbf{U}X_2)$ is equivalent over chains to

$$\varphi(x_0, X_1, X_2) \triangleq \big(\forall x \geq x_0(x \in X_1 \vee x \in X_2)\big) \wedge \big(\forall x \geq x_0. \exists x_1(x_1 \geq x \wedge x_1 \in X_2)\big).$$

The formula $\varphi(x_0, X_1, X_2)$ has quantifier-depth 2. Hence, $CTL^2 \preceq_{exp} BTL_2$.

$B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty)$ **and** $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg)$ **(see [10, 7]).** $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty)$ and $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg)$ allow the specification of fairness properties. $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg)$ is very similar to CTF used in [6]. $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty)$ can be defined as $TL(\mathbf{EX},\mathbf{AX},\mathbf{EU},\mathbf{AU},\mathbf{EF}^\infty,\mathbf{EG}^\infty,\mathbf{AF}^\infty,\mathbf{AG}^\infty)$. Let Φ_2 be the set of $TL(\mathbf{X},\mathbf{U},\mathbf{F}^\infty,\mathbf{G}^\infty)$ formulas of nesting-depth ≤ 1. Let M_4 be the infinite set $\{\mathbf{E}\varphi : \varphi \in \Phi_2\}$ of path modalities, then $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg)$ can be defined as $TL(M_4)$. The truth tables of $\mathbf{F}^\infty p$ and $\mathbf{G}^\infty p$ are both of quantifier-depth 2. Therefore, $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty) \preceq_{exp} BTL_2$ and $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg) \preceq_{exp} BTL_2$.

In [10], the formula $\mathbf{AF}(p \wedge \mathbf{X}p)$ was provided as an example for a CTL^* formula which is not expressible in $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg)$. The formula $\mathbf{AF}(p \wedge \mathbf{X}p)$ is expressible in BTL_3. As far as we know, this is the only modality discussed in the literature which is not definable in BTL_2.

Recently, it was shown in [33] that $B(\mathbf{U},\mathbf{F},\mathbf{F}^\infty,\mathbf{G}^\infty,\wedge,\neg)$ is expressively equivalent to BTL_2.

6.5 BTL_1 Has a Finite Base

Observe that BTL_1 was defined as a temporal logic with an infinite set of basic modalities. However, we prove below that BTL_1 has a finite base of modalities.

Theorem 6.6 *There is a two-place modality which is a base for BTL_1.*

In the remainder of this section, the proof of Theorem 6.6 is given. Let $m(X_1, X_2)$ be the path modality $\mathbf{E}\alpha$, where $\alpha(x_0, X_1, X_2)$ is defined as

$$\alpha(x_0, X_1, X_2) \triangleq \exists y(y > x_0 \wedge y \in X_1) \wedge \forall y\big((y > x_0) \rightarrow (y \in X_2)\big).$$

$\mathbf{E}\alpha(p,q)$ is expressible by the CTL^* formula $\mathbf{E}(\mathbf{F}p \wedge \mathbf{G}q)$. We are going to prove that for every future $FOMLO$ formula $\varphi(x_0, Y_1, \dots, Y_n)$ of quantifier-depth one, there is a $TL(m)$ formula ψ such that $E\varphi$ is equivalent over trees to ψ. This will imply Theorem 6.6.

First we define formulas $\varphi_{\sigma,\rho}$ which will satisfy Lemma 6.7 stated below.

For a subset σ of $\{1, \dots, n\}$ we denote by $\varphi_\sigma(x_0, Y_1, \dots, Y_n)$ the formula

$$\bigwedge_{i\in\sigma} x_0 \in Y_i \wedge \bigwedge_{i\notin\sigma} x_0 \notin Y_i$$

Note that if $\sigma \neq \sigma'$ then $\varphi_\sigma(x_0, Y_1, \dots, Y_n) \wedge \varphi_{\sigma'}(x_0, Y_1, \dots, Y_n)$ is unsatisfiable.

For a sequence $\rho = \sigma_1\sigma_2, \dots, \sigma_k$ of ($k \geq 1$) distinct subsets of $\{1, \dots, n\}$ and a subset σ of $\{1, \dots, n\}$ we denote by $\varphi_{\sigma,\rho}(x_0, Y_1, \dots, Y_n)$ the formula

$$\varphi_\sigma \wedge \exists x_1 \exists x_2 \dots \exists x_k \big(x_0 < x_1 < \dots < x_{k-1} < x_k \wedge \bigwedge_{i=1}^{k} \varphi_{\sigma_i}(x_i)\big) \wedge \forall x > x_0 .\ \bigvee_{i=1}^{k} \varphi_{\sigma_i}(x)$$

If ρ is the empty sequence, then $\varphi_{\sigma,\rho}$ is defined as $\varphi_\sigma \wedge \forall x.\ x \leq x_0$.

Some explanations might be helpful to understand the above formulas. We say that an element s of a chain $A = (\mid A \mid, \leq, P_1, P_2, \dots, P_n)$ is tagged by $\sigma \subseteq \{1, \dots, n\}$ if $s \in P_i \leftrightarrow i \in \sigma$. Note that each $s \in\mid A \mid$ is uniquely tagged by some subset σ. For a sequence $\rho = \sigma_1\sigma_2, \dots, \sigma_k$ of ($k \geq 1$) subsets of $\{1, \dots, n\}$ and a subset σ of $\{1, \dots, n\}$, the formula $\varphi_{\sigma,\rho}$ holds in A at s if s is tagged by σ and there are elements $s < s_1 < \dots < s_k$ such that s_i is tagged by σ_i (for $i = 1, \dots, k$) and every node greater than s in the chain is tagged by one of σ_i ($i = 1, \dots, k$).

Lemma 6.7 *1. For every labeled chain $A = (\mid A \mid, \leq, P_1, P_2, \dots, P_n)$ and $a \in\mid A \mid$ there are σ and ρ such that $A, a \models \varphi_{\sigma,\rho}$.*

2. Let $A = (\mid A \mid, \leq_A, P_1^A, P_2^A, \dots, P_n^A)$ and $B = (\mid B \mid, \leq_B, P_1^B, P_2^B, \dots, P_n^B)$ be labeled chains and let $a \in\mid A \mid$ and $b \in\mid B \mid$. If $A, a \models \varphi_{\sigma,\rho}$ and $B, b \models \varphi_{\sigma,\rho}$ then for every future FOMLO formula $\alpha(x_0, Y_1, \dots, Y_n)$ of quantifier-depth one:

$$A, a \models \alpha \textit{ if and only if } B, b \models \alpha$$

3. *Every future FOMLO formula $\alpha(x_0, Y_1, \dots, Y_n)$ of quantifier-depth one is equivalent over the class of all chains to a (finite) disjunction of formulas of the form $\varphi_{\sigma,\rho}$.*

4. *For every σ and ρ there is a $TL(m)$ formula which is equivalent (over trees) to $E\varphi_{\sigma,\rho}$.*

Proof (1) is immediate.

(2) Observe that if $A, a \models \varphi_{\sigma,\rho}$ and $B, b \models \varphi_{\sigma,\rho}$ then *Duplicator* has a winning strategy in one-move chain games over (A, a) and (B, b) (see Section 4.1 for the description of games on chains). This observation and Theorem 4.3 implies that for every future *FOMLO* formula $\alpha(x_0, Y_1, \dots, Y_n)$ of quantifier-depth one: $A, a \models \alpha$ if and only if $B, b \models \alpha$.

(3) Let $\alpha(x_0, Y_1, \dots, Y_n)$ be a future *FOMLO* formula of quantifier-depth one. Observe that there are only finitely many sequences of distinct subsets of $\{1, \dots, n\}$. Hence, there are only finitely many formulas of the form $\varphi_{\sigma,\rho}$. By (2) it follows that for every σ and ρ, either $\varphi_{\sigma,\rho} \to \alpha$ holds (over chains) or $\varphi_{\sigma,\rho} \to \neg\alpha$ holds (over chains). This together with (1) implies that α is equivalent (over chains) to

$$\bigvee\{\varphi_{\sigma,\rho} \ : \ \varphi_{\sigma,\rho} \to \alpha \text{ holds over the class of all chains}\}.$$

This is a finite disjunction of formulas of the form $\varphi_{\sigma,\rho}$.

(4) For a subset σ of $\{1, \dots, n\}$ we denote by ψ_σ the TL formula

$$\bigwedge_{i\in\sigma} Y_i \wedge \bigwedge_{i\notin\sigma} \neg Y_i$$

For a sequence $\rho = \sigma_1\sigma_2 \dots, \sigma_k$ of $(k \geq 1)$ distinct subsets of $\{1, \dots, n\}$ we denote by ψ_ρ the TL formula

$$\bigvee_{i=1}^{k} \psi_{\sigma_i}$$

Let $H_1(X)$ be an abbreviation for $m(X, \psi_\rho)$ (i.e. it abbreviates $\mathbf{E}(\mathbf{F}X \wedge \mathbf{G}\psi_\rho)$). One can easily check that $\mathbf{E}\varphi_{\sigma,\rho}$ is equivalent over trees to the following $TL(m)$ formula

$$\psi_\sigma \wedge H_1(\psi_{\sigma_1} \wedge H_1(\psi_{\sigma_2} \wedge \dots H_1(\psi_{\sigma_k})))$$

□

Let $\mathbf{E}\alpha$ be a BTL_1 modality. By Lemma 6.7(3), there are σ_i and ρ_i $(i = 1, \dots, m)$ such that α is equivalent over chains to $\bigvee_{i=1}^{m} \varphi_{\sigma_i,\rho_i}$. Therefore, $\mathbf{E}\alpha$ is equivalent over trees to $\bigvee_{i=1}^{m} E\varphi_{\sigma_i,\rho_i}$. This together with Lemma 6.7(4) implies that $E\alpha$ is equivalent (over trees) to a $TL(m)$ formula. Since every BTL_1 modality is equivalent to a $TL(m)$ formula and m is a basic BTL_1 formula, it follows that the set $\{m\}$ is a basis for BTL_1. This completes the proof for Theorem 6.6.

7 Further Results

7.1 Strengthening the Main Results

We show that the $\{BTL_k\}_{k=1}^{\infty}$ hierarchy, as well as the results presented in Section 6, are valid not only for arbitrary trees, but extend to specific classes of trees. Below we consider the following classes of trees: trees with a bounded number of immediate successors at each node, trees with no finite branches, finite trees, and trees obtained by unwinding finite Kripke structures.

Corollary 7.1 1. *CTL^* has no finite basis over trees of bounded degree.*

2. *CTL^* has no finite base over trees with no leaves.*

3. *CTL^* has no finite base over the class of finite trees.*

4. CTL^* has no finite base over finite Kripke structures.

Proof (Sketch)

(1) follows from the observation that the proof of Theorem 5.4 involves only trees with out-degree at most two.

(2) In the proof of Theorem 5.4, change the definition of M_1^k and N_1^k (Figure 1) by adding an ω-path of nodes labeled by $\neg P$ to the leaves of both trees.

(3) In the proof of Theorem 5.4, change the inductive definition of M_{i+1}^k and N_{i+1}^k (Figure 2) by cutting the infinite path $\langle a_{i+1}^1, a_{i+1}^2, \ldots\rangle$ after $2k+1$ nodes, in both trees. Let $\hat{M}_{i+1}^k$ and $\hat{N}_{i+1}^k$ be the modified trees and let $m_k = \lfloor log_2(k+1)\rfloor$. Observe that $\hat{M}_{i+1}^k$ and $\hat{N}_{i+1}^k$ are finite trees.

Recall that in the proof of Theorem 5.4 we showed that for every $h \geq 1$ and $h' \geq 1$:

$$(M_{i+1}^k)_{\geq a_{i+1}^h} \sim_{i,m_k}^g (N_{i+1}^k)_{\geq a_{i+1}^{h'}} \tag{10}$$

One can show that for every $1 \leq h \leq k+1$ and $1 \leq h' \leq k+1$:

$$(\hat{M}_{i+1}^k)_{\geq a_{i+1}^h} \sim_{i,m_k}^g (\hat{N}_{i+1}^k)_{\geq a_{i+1}^{h'}} \tag{11}$$

Hence the corresponding nodes in $\hat{M}_{i+1}^k$ and $\hat{N}_{i+1}^k$ are still (i, m_k)-game equivalent. The rest of the arguments are exactly like for Theorem 6.3 (a detailed proof can be found in [26]).

(4) This follows from (3) or directly from the observation that every computation tree in the sequences M_i^k and N_i^k (in the proof of Theorem 5.4) has a finite and indeed very succinct representation as a Kripke structure. Therefore, CTL^* has no finite base over finite Kripke structures. $\square$

7.2 Complexity of Model Checking

There is a tradeoff between the expressive power, the succinctness and the complexity of verification of specification formalisms. In this subsection we prove that the model checking problem has linear time complexity for every temporal logic based on a finite set of modalities. Hence, the complexity theory cannot provide us a sharp criterion for the choice of a finite base temporal logic.

The model checking problem for a logic L is as follows. Given a finite Kripke structure K and a formula $\varphi \in L$, determine whether $T_K, root \models \varphi$, where T_K is the tree that corresponds to the unwinding of K from its initial state.

CTL is based on four modalities. The model checking problem for CTL has the linear time complexity $O(\mid K \mid \times \mid \varphi \mid)$. CTL^* is based on an infinite set of modalities. Unlike CTL, the model checking problem for CTL^* is PSPACE complete [3]. The next Theorem shows that for a temporal logic based on a finite set of modalities, the model checking problem has a low complexity. Its proof is based on techniques from [11].

Recall that modal μ calculus is equivalent to the bisimulation invariant fragment of (future) monadic second-order logic [22].

Theorem 7.2 (Complexity of Model Checking) *Let $TL(M_1, M_2, \ldots, M_k)$ be a TL based on a finite set of modalities.*

1. *Assume that M_i ($i = 1, 2, \ldots, k$) are definable by CTL^* formulas. Then the model checking problem for $TL(M_1, M_2, \ldots, M_k)$ has time complexity $O(\mid K \mid \times \mid \varphi \mid)$.*
2. *Assume that M_i ($i = 1, 2, \ldots, k$) are of the form $E\varphi$, where φ is a future monadic second-order formula. Then the model checking problem for $TL(M_1, M_2, \ldots, M_k)$ has time complexity $O(\mid K \mid \times \mid \varphi \mid)$.*
3. *Assume that M_i ($i = 1, 2, \ldots, k$) are definable by μ-formulas. Then the model checking problem for $TL(M_1, M_2, \ldots, M_k)$ is in PTIME.*

Proof (Sketch) (1) Is easily reducible to (2). Indeed, by Lemma 6.2, for every CTL^* definable modality M_i we can find a finite set $\{N_i^1, \dots, N_i^{r_i}\}$ of modalities of the from $\mathsf{E}\varphi$, where φ is a future $FOMLO$ formula, such that M_i is expressible in $TL(N_i^1, \dots, N_i^{r_i})$. Let

$$\Delta \triangleq \bigcup_{i=1}^{k} \{N_i^1, \dots, N_i^{r_i}\}.$$

$TL(M_1, \dots M_k) \preceq_{exp} TL(\Delta)$. Moreover, there is a linear time meaning preserving translation from $TL(M_1, \dots M_k)$ into $TL(\Delta)$. Therefore, the model checking problem for $TL(M_1, \dots M_k)$ is reducible in linear time to the model checking problem for $TL(\Delta)$. This completes the reduction of (1) to (2).

The model checking algorithm for (2) and (3) uses standard techniques [8]. Given a structure K and a formula φ the algorithm takes the sub-formulas of φ starting with the innermost ones, and iteratively labels with each sub-formula χ the states of K that satisfy χ. The only nontrivial case is when χ has the form $M(\varphi_1, \dots \varphi_l)$, where M is a modality. This case is explained below for (2) and (3).

(2) First, for every basic modality $M_i = E\psi_i$ $(i = 1, \dots, r)$ of the logic, construct a Buchi automaton A_i such that A_i accepts the ω-language definable by ψ_i. Let b be an upper bound on the size of these automata.

In order to find the states of K which should be labeled by $M_i(\varphi_1, \dots, \varphi_l)$ construct the product Pr of A_i and K and then find in Pr the set of states S from which there is an accepting run; the states of K which should be labeled by $M_i(\varphi_1, \dots, \varphi_l)$ are easily extracted from S. Namely, $s \in \mid K \mid$ should be labeled by $M_i(\varphi_1, \dots, \varphi_l)$ iff $\langle q_0, s\rangle \in S$, where q_0 is the initial state of A_i.

The time complexity of this step is bounded by $\mid K \mid \times \mid A_i \mid$. Since the number of sub-formulas of φ is at most $\mid \varphi \mid$, the time complexity of the algorithm is bounded by $\mid K \mid \times \mid \varphi \mid \times b = O(\mid K \mid \times \mid \varphi \mid)$.

(3) In this case the basic modalities M_i $(i = 1, \dots, r)$ are μ formulas. The set of states which should be labeled by $M_i(\varphi_1, \dots, \varphi_l)$ can be found by applying the Tarski-Kanster algorithms. A naive implementation of the algorithm runs in time $\leq \ \mid K \mid^{d_i} \times \mid M_i \mid^{d_i}$, where d_i is the nesting-depth of the fixed point operators in M_i. Hence, if $d = max(d_i)$ and $b = max(\mid M_i \mid^{d_i})$, the time complexity of the model checking algorithm for $TL(M_1, \dots, M_r)$ is bounded by $\mid K \mid^d \times \mid \varphi \mid \times b = O(\mid K \mid^d \times \mid \varphi \mid)$. $\square$

Notes.

1. The assumption in Theorem 7.2(2) can be replaced by "M_i $(i = 1, \dots, k)$ are definable by formulas in $TL(W)$, where $W = \{\mathsf{E}\varphi \ : \ \varphi \text{ is a future } MLO \text{ formula}\}$". The reduction from this more general form to (2) is the same as the reduction from (1) to (2). The logic $TL(W)$ is expressively equivalent to $ECTL^*$ from [35]. It is also expressively equivalent to the future bisimulation invariant fragment of the monadic chain logic [17] (the monadic chain logic is obtained from the monadic second-order logic when all the bound set variables are restricted to range over linearly ordered subsets of trees.).

2. Observe that Theorem 7.2 does not cover BTL_k logics because they are based on infinite sets of modalities. In [33], it was proved that the model checking problem for BTL_2 is $\mathbf{P}^{\mathbf{NP}}$ complete, where $\mathbf{P}^{\mathbf{NP}}$ is the class of decision problems for which there is an algorithm in $\mathbf{P}$ with an oracle in $\mathbf{NP}$.

   It is an open question what is the complexity of the model checking problem for BTL_k $(k \neq 2)$.

8 Conclusion

Our results offer an explanation for the multiplicity of temporal logics over branching time and suggest some yardsticks by which to measure these logics.

Two of the most important characteristics of a TL are (1) its expressive power and (2) the complexity of its model checking problem [8]. We examined two very natural fragments of MLO: CTL^* which is equivalent to the bisimulation invariant fragment of future monadic path logic [28], and the bisimulation invariant fragment of future first-order monadic logic. We proved that there is no temporal logic over a finite base which is expressively equivalent over the trees to each of these fragments. On the other hand, we showed that for every finite set of modalities $M_1, M_2, \dots, M_r$, the complexity of model checking for

$TL(M_1, M_2, \ldots, M_r)$ is linear both in the size of structure and the size of formula. We believe, therefore, that these are the reasons for so many suggestions for temporal logics over branching time models.

We defined a sequence BTL_k of temporal logics and proved that it contains a strict hierarchy. This was sufficient to show that CTL^* has no finite base. It can be shown that BTL_{k+1} is strictly more expressive than BTL_k, however, we have not succeeded to show that properties $Path_i$ and $Block_i$ separate between BTL_k and BTL_{k+1}; the proof of this separation result uses a more complex property.

We examined many sub-logics of CTL^* suggested in the literature and showed that they are inside the second level of our hierarchy.

In this work only modalities defined in the future fragment of *MLO* were considered. The situation with temporal logics based on future and past modalities can be quite different. For example, the full version of Kamp's theorem [23, 13] implies that the temporal logic based on two modalities **U** ("until") and **S** ("since") is expressively equivalent over the reals to the *FOMLO*. However, until is not expressively equivalent (over the reals) to the future fragment of *FOMLO*. Moreover, it was shown in [19] that there is no finite set of future modalities which is expressively equivalent over the reals to the future fragment of *FOMLO*. We believe that the temporal logic with two modalities until and since is expressively equivalent over the trees to the bisimulation invariant fragment of *FOMLO*.

We conclude with two conjectures.

Conjecture 8.1 *There is no finite base for BTL_k for $k > 1$.*

Conjecture 8.2 (Arity Hierarchy) *Let M_k be the set of all modalities of arity at most k, which are definable by CTL^* formulas. $TL(M_k)$ is strictly less expressive than $TL(M_{k+1})$ for every k.*

Acknowledgements

We would like to thank Yoram Hirshfeld for his insights that influenced this research. We are grateful to the anonymous referees for providing numerous suggestions for improvements.

References

[1] M. Ben-Ari, Z. Manna and A. Pnueli (1981). 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.

[2] E.M. Clarke and E.A. Emerson (1981). Design and verification of synchronous skeletons using branching time temporal logic. *Lecture Notes in Computer Science* **131**:52–71.

[3] E.M. Clarke, E.A. Emerson and A.P. Sistla. Automatic verification of finite state concurrent system using temporal logic. In *Proceedings of the 10th Annual ACM Symposium on Principles of Programming Languages*, 1983.

[4] H.D. Ebbinghaus and J. Flum (1995). *Finite Model Theory.* Springer Perspectives in Mathematical Theory.

[5] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. Fundamenta mathematicae 49,129-141,1961.

[6] E. A. Emerson and E. M. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In *Proc. 7th Coll. Automata, Languages and Programming (ICALP'80), Noordwijkerhout, NL, Jul. 1980*, volume 85 of *Lecture Notes in Computer Science*, pages 169–181. Springer, 1980.

[7] E.A. Emerson (1990). Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. Elsevier, Amsterdam, 1990.

[8] E.A. Emerson (1996). Automated temporal reasoning about reactive systems. *Logics for Concurrency: Structure versus Automata*, F. Moller and G. Birtwistle (eds.), Springer LNCS no. 1043, pp. 41–101, 1996.

[9] E.A. Emerson and J.Y. Halpern (1982). 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.

[10] E.A. Emerson and J.Y. Halpern (1986). 'Sometimes' and 'not never' revisited: on branching versus linear time temporal logics. *Journal of the ACM* **33**(1):151–178.

[11] E.A. Emerson and C.L. Lei. Modalities for model checking: branching time strikes back. *12th ACM Symp. on Principles of Prog. Lang.*, pp. 84-96, 1985.

[12] K. Etessami and T. Wilke. An until hierarchy for temporal logic. In Proceedings 11th Annual IEEE Symposium on Logic in Computer Science, pages 108-117, 1996.

[13] D. Gabbay, I. Hodkinson and M. Reynolds (1994). *Temporal Logic.* Oxford University Press.

[14] D. Gabbay, A. Pnueli, S. Shelah and J. Stavi. On the temporal analysis of fairness. *7th ACM Symp. on Principles of Prog. Lang.*, pp. 163-173, 1980.

[15] Y. Gurevich and S. Shelah (1985). The decision problem for branching time logic. *Journal of Symbolic Logic* **50**(3):668–681.

[16] D. Gale and F. Stewart (1953). Infinite games with perfect information. *Ann. Math. Studies*, **28**:245-266.

[17] T. Hafer and W. Thomas (1987). Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *Proceedings of ICALP'87: International Colloquium on Automata, Languages and Programming*, *Lecture Notes in Computer Science* **267**:269–279, Springer-Verlag.

[18] Y. Hirshfeld and A. Rabinovich (1999). Quantitative Temporal Logic. In Computer Science Logic 1999, *Lecture Notes in Computer Science* **1683**:172-187, Springer-Verlag.

[19] Y. Hirshfeld and A. Rabinovich (2000). On the Expressive Power of Temporal Logics over the Reals. *Technical Report*, Tel-Aviv University.

[20] W. Hodges (1997). *A Shorter Model Theory.* Cambridge University Press.

[21] N. Immerman (1998).*Descriptive Complexity*, Springer Verlag, 1998.

[22] D. Janin and I. Walukiewicz (1996). On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of CONCUR'96: International Conference on Concurrency Theory*, *Lecture Notes in Computer Science* **1119**:263–277, Springer-Verlag.

[23] H. W. Kamp (1968). *Tense logic and the theory of linear order.* PhD Thesis, University of California, Los Angeles.

[24] O. Kupferman and O. Grumberg (1996). Buy one, get one free!!! Journal of Logic and Computation, 6:523-539.

[25] L. Lamport (1980). "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.

[26] S. Maoz (2000). Infinite Hierarchy of Temporal Logics over Branching Time Models. M.Sc. Thesis, Tel-Aviv University.

[27] R. Milner (1989). *Communication and Concurrency.* Prentice-Hall.

[28] F. Moller and A. Rabinovich (1999). On the expressive power of CTL*. *Proceedings of fourteenth IEEE Symposium on Logic in Computer Science*, 360-369.

[29] D.M.R. Park (1981). Concurrency and automata on infinite sequences. *Lecture Notes in Computer Science* **104**:168–183.

[30] A. Pnueli (1977). The temporal logic of programs. In *Proceedings of eighteenth IEEE Symposium on Foundations of Computer Science*, 46-57.

[31] M. O. Rabin (1969). Decidability of second order theories and automata on infinite trees. In *Trans. Amer. Math. Soc.*,**141**,pp 1-35.

[32] A. Rabinovich and S. Maoz (2000). Why so many temporal logics climb up the trees? In *Proceedings of MFCS 2000, Lecture Notes in Computer Science* **1893**, Springer-Verlag.

[33] A. Rabinovich and Ph. Schnoebelen (2000). BTL_2 and expressive completeness for $ECTL^+$. Research Report LSV-00-8, Lab. Specification and Verification, ENS de Cachan, Cachan, France, October 2000.

[34] J. G. Rosenstein (1982). *Linear orderings*. Academic Press, New York.

[35] M. Vardi and P. Wolper (1984). Yet another process logic. *In Logics of Programs, Lecture Notes in Computer Science* **164**, 501-512, Springer-Verlag.