# Why so many Temporal Logics Climb up the Trees?

Alexander Rabinovich and Shahar Maoz

School of Mathematical Science
Tel Aviv University
Tel Aviv, Israel 69978
{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 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 show that for every logic based on a finite set of modalities, the complexity of model checking is linear both in the size of structure and the size of formula.

## 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 the (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 and not at others. More complex properties are described by formulas built from the atoms using Boolean connectives and *Modalities* (temporal connectives): an $l$-place modality $C$ transforms statements $\varphi_1, \ldots, \varphi_l$ on points possibly other than the given point $t_0$ to a statement $C(\varphi_1, \ldots, \varphi_l)$ on the point $t_0$. The rule that specifies when is the statement

$C(\varphi_1, \ldots, \varphi_l)$ true for the given point is called *Truth Table*. The choice of the particular modalities with their truth table determines the different temporal logics. A Temporal Logic with modalities $M_1, \ldots M_r$ is denoted by $TL(M_1, \ldots, M_r)$.

The most basic modality is the one place modality $\mathbf{F}X$ saying "$X$ holds some time in the future". Its truth table is usually formalized by $\varphi_{\mathbf{F}}(t_0, X) \equiv (\exists t > t_0)X(t)$. This is a formula of the Monadic Logic of Order ($MLO$). $MLO$ is a fundamental formalism in Mathematical Logic. Its formulas are built using atomic propositions $X(t)$, 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 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 single 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 [13, 9] 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 that 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 future time temporal logics.

Milner and Park [16, 18] 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 computational 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 [14, 1]. 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(U) or of the form $\mathbf{A}$ "for every linear run"

followed by a formula in TL(U). $\mathbf{E}\varphi$ (respectively $A\varphi$) holds at a moment $t_0$ if for some path $\pi$ (respectively, for every path $\pi$) starting at $t_0$ the $TL(\mathbf{U})$ formula $\varphi$ holds along $\pi$. For example, one commonly used branching time logic is $CTL$ [1]. 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.

The logic $CTL^*$ suggested in [7] 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 [8, 4]) by imposing some syntactical restrictions on $CTL^*$ formulas. The lack of a yardstick was emphasized by Emerson in words very similar to the above [4, 5]:

> "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 known about comparisons of $BTL$s against each other. This contrasts with the reversed situation for Linear-Time Logics."

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 [5]. We examine two very natural fragments of $MLO$ and prove that there is no temporal logic over finite basis which is expressively equivalent over trees to each of these fragments. On the other hand, we show that for every finite set of modalities $M_1, M_2, \ldots M_r$, 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.

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 [17], $CTL^*$ was shown to be expressively equivalent to the bisimulation invariant fragment of monadic path logic [10]. 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 [17]. Thus at least $CTL^*$ represents some objectively quantified expressive power.

We describe a sequence $BTL_k$ ($k \in Nat$) 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 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 $\varphi$ is satisfiable in the computational 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 [2]. 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 ($\omega$) branches.

The paper is organized as follows. In the next section we review basic definitions about monadic logic of order and temporal logics. In Sect. 3 we introduce the sequence $BTL_k$ of temporal logics. Our main technical result is that $\{BTL\}_{k=1}^{\infty}$ contains a strict hierarchy. In Sect. 4 we show that $CTL^*$ has no finite basis and examine the expressive power of some commonly used branching time logics. In Sect. 5 we discuss the complexity of the model checking problem.

Because of space limitation the proofs are omitted. Detailed proofs can be found in [15].

## 2 Preliminaries

In this section we review basic definitions about computation trees, the monadic logic of order, temporal logics and modalities.

### 2.1 Computation Trees and Paths

A ***tree*** $T = (\mid T \mid, \leq)$ consists of a partially ordered set of nodes $\mid T \mid$ in which the predecessors of any given element $a \in \mid T \mid$ constitute a finite total order with a common minimal element $\epsilon_T$, referred to as the root of the tree. A ***computation tree*** is a structure $(\mid T \mid, \leq, P_1, P_2, \ldots)$, where $(\mid T \mid, \leq)$ is a tree, and $P_1, P_2, \ldots$ are subsets of $\mid T \mid$. We say that a node $a \in \mid T \mid$ is labeled by $P_i$ if $a \in P_i$. We often write $\boldsymbol{P}$ for the sequence $P_1, P_2, \ldots$; the length of $\boldsymbol{P}$ is denoted by $length(\boldsymbol{P})$.

A ***path*** $\pi$ through $T$, starting at an element $s_1 \in \mid T \mid$, is a maximal linearly ordered sequence of successive nodes $\pi = \langle s_1, s_2, \ldots \rangle$ through $T$, ordered by $\leq$.

Let $T = (\mid T \mid, \leq, \boldsymbol{P})$ be a computation tree and let $a \in \mid T \mid$. We shall use $T_{\geq a}$ to denote the subtree of $T$ rooted at the node $a$. Formally, $T_{\geq a} = (\mid T_{\geq a} \mid, \leq, \boldsymbol{P'})$ where $\mid T_{\geq a} \mid = \{s : s \in \mid T \mid \textit{ and } s \geq a\}$ and $P'_i = P_i \cap \mid T_{\geq a} \mid$ for $i \leq length(\boldsymbol{P})$. We use $T_\pi$ for the substructure of $T$ over the nodes in the path $\pi$.

### 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 $x_0, x_1, x_2, \ldots$ (representing nodes, states or time points), set variables $X_0, X_1, X_2, \ldots$ (representing sets of nodes), and set constants (monadic predicates) $P_0, P_1, P_2, \ldots$. Formulas are built up from atomic formulas of the form $x_1 = x_2$, $x_1 < x_2$, $x \in X$ and $x \in P_i$, using the Propositional connectives $\wedge$ and $\neg$, and the quantifiers $\exists x$ and $\exists X$.

We shall write $\varphi(x_1, x_2, \ldots, x_m; X_1, X_2, \ldots, X_n)$ to indicate that the variables $x_1, x_2, \ldots, x_m; X_1, X_2, \ldots, X_n$ may appear free in $\varphi$.

The ***quantifier depth*** of a formula $\varphi$, denoted by $\mathrm{qd}(\varphi)$, is defined as usual. We write $(T, s_1, \ldots, s_m, S_1, \ldots, S_n) \models \varphi(x_1, \ldots, x_m, X_1, \ldots, X_n)$ if the formula $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n)$ is satisfied in the tree $T$ with $x_i$ interpreted by the node $s_i$ $(1 \leq i \leq m)$ and $X_j$ interpreted by the set of nodes $S_j$ $(1 \leq j \leq n)$.

We shall denote by $FOMLO$ the subset of first-order formulas of $MLO$ that do not have set quantification. We also consider Monadic Path Logic $MPL$ [10]. Its syntax is the same as that of monadic second-order logic. However, the bound set (monadic) variables range over all the paths (not over arbitrary sets of nodes) and semantically it is very closely related to the first-order logic [17].

**Definition 1 (Future MLO Formula).** *A formula $\varphi(x_0, X_1, X_2, \ldots, X_l)$ of $MLO$ with one free first-order variable $x_0$, is a future formula, if for every computation tree $T$ and node $a \in \mid T \mid$, and every $l$ subsets $S_1, S_2, \ldots, S_l$ of $\mid T \mid$, the following holds:*

$$T, a, S_1, \ldots, S_l \models \varphi(x_0, X_1, \ldots, X_l) \textit{ iff } \quad T_{\geq a}, a, S'_1, , \ldots, S'_l \models \varphi(x_0, X_1, \ldots, X_l)$$

*where $S'_i = \{s : s \in S_i \textit{ and } s \geq a\}$ for all $i$, $i = 1, 2, \ldots, l$. First-order future formulas are defined similarly.*

### 2.3 Temporal Logics and Modalities

In the following, we recall the syntax and semantics of temporal logics and how temporal modalities are defined using $MLO$ truth tables, with notation adopted from [9, 11].

The syntax of ***Temporal Logic*** $(TL)$ has in its vocabulary a set of ***predicate variables*** $\{q_1, q_2, \ldots\}$ and a set $B$ of ***modality names*** (sometimes called "temporal connectives" or "temporal operators") with prescribed arity $B = \{\#_1^{(l_1)}, \#_2^{(l_2)}, \ldots\}$ (we usually omit the arity notation). The set of modality names $B$ might be infinite. The syntax of $TL(B)$ is given by the following grammar:

$$\varphi \ ::= \ q_i \ \mid \ \varphi_1 \wedge \varphi_2 \ \mid \ \neg\varphi_1 \ \mid \ \#_i^{l_i}(\varphi_1, \varphi_2, \ldots, \varphi_{l_i})$$

Temporal formulas are interpreted over partially ordered sets, in particular over computation trees. Every modality $\#^{(l)}$ is interpreted in every tree $T$ as an operator $\#_T^{(l)} : [P(T)]^l \to P(T)$ which assigns "the set of points where $\#^{(l)}[P_1, \ldots, P_l]$ holds" to the $l-$tuple $\langle P_1, \ldots, P_l\rangle$. Formally, the semantics of a formula $\varphi \in TL$ over a tree $T = (\mid T \mid, \leq, \boldsymbol{Q})$ is defined inductively as follows. For atomic formulas $T, s \models q_i$ iff $s \in Q_i$; the semantics of Boolean combinations is defined as usual, and the semantics of modalities is defined by: $T, s \models \#^{(l)}(\varphi_1, \varphi_2, \ldots, \varphi_l)$ iff $s \in \#_T^{(l)}(R_{\varphi_1}, R_{\varphi_2}, \ldots, R_{\varphi_l})$ where $R_{\varphi_i} = \{a : T, a \models \varphi_i\}$ for all $i, 1 \leq i \leq l$.

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 (truth table) $\bar{\#}(x_0, X_1, X_2, \ldots, X_l)$ of $MLO$ with one free first-order variable $x_0$ and $l$ set variables, such that for every tree $T$ and subsets $Q_i \subseteq \mid T \mid$:

$$\#_T(Q_1, Q_2, \ldots, Q_l) = \{s : \mid T \mid, \leq, s, Q_1, Q_2, \ldots, Q_l \models \bar{\#}[x_0, X_1, X_2, \ldots, X_l]\}$$

*Example 2 (Some common modalities and their truth tables).*

- The one place modality $\mathbf{F}q$ ("*eventually* $q$"); its truth table is $\varphi(x_0, X) \equiv \exists y(y > x_0 \wedge y \in X)$.
- The one place modality $\mathbf{G}q$ ("*globally* $q$"); its truth table is $\varphi(x_0, X) \equiv \forall y(y > x_0) \to (y \in X)$.
- The two place modality $\mathbf{U}(q_1, q_2)$ ("$q_2$ *until* $q_1$"); its truth table is $\varphi(x_0, X, Y) \equiv \exists y(y > x_0 \wedge y \in X \wedge \forall z(x_0 < z < y) \to z \in Y)$. In the literature, some times "non strict" definition of Until is given: the "non-strict until" $\mathbf{U}_{ns}(q_1, q_2)$ modality has truth table $\varphi(x_0, X, Y) \equiv \exists y(y \geq x_0 \wedge y \in X \wedge \forall z(x_0 \leq z < y) \to z \in Y)$. Clearly, $\mathbf{U}_{ns}$ can be defined using $\mathbf{U}$.

The choice of the particular modalities with their truth tables determines the different temporal logics. The following are standard definitions and notation to discuss comparative expressive power between two temporal logics $TL_1$ and $TL_2$.

**Definition 3.** *Let $C$ be a set of structures. $TL_1$ is* ***less or equally expressive*** *than $TL_2$ over $C$ (notation $TL_1 \preceq_{exp}^{C} TL_2$), if for any formula $\varphi_1 \in TL_1$ there*

*is a formula $\varphi_2 \in TL_2$ which is equivalent to $\varphi_1$ 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 $\preceq^C_{exp}$. Similarly for $\equiv_{exp}$ and $\prec_{exp}$.*

**Definition 4 (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, $\mathbf{F}p$, $\mathbf{U}(p,q)$ and $F^\infty p$, are first-order future modalities.

**Definition 5 (Path Modalities).** *For every first-order future formula $\varphi(x_0, X_1, \ldots, X_l)$, we define an $l$ place path modality $\mathbf{E}\varphi$ as follows: $T, a \models \mathbf{E}\varphi$ if and only if there is a path $\pi$ from $a$ in $T$, such that $T_\pi, a \models \varphi(x_0, X_1, \ldots, X_l)$. $\mathbf{E}\varphi$ is said to be the path modality which corresponds to $\varphi(x_0, X_1, \ldots, X_l)$.*

**Proposition 6.** *For every first-order future formula $\varphi(x_0, X_1, \ldots, X_l)$, the path modality $\mathbf{E}\varphi$ has an $MPL$ truth table.*

## 3 A Hierarchy

For every $k \geq 1$, let $BTL_k$ be the temporal logic defined as $TL(M_k)$, where $M_k = \{\mathbf{E}\varphi : \mathrm{qd}(\varphi(x_0, X_1, \ldots, X_l)) \leq k$ *and* $\varphi$ is a first-order future formula$\}$.

Note that for every $k \geq 1$, $BTL_k$ is based on an infinite set of modalities. All the basic modalities of $BTL_k$ have a truth table of quantifier depth $\leq k+1$. However,

**Proposition 7.** *For every $n$ and $k > 1$ there is a $BTL_k$ formula which is equivalent to no $MLO$ formula of quantifier depth $\leq n$.*

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

**Definition 8 ($Block_k$).** *For $k \geq 1$ let $Block_k$ be a property of trees with two unary predicates $p, q$, defined as follows. $T \in Block_k$ iff there is a path $\pi$ starting at the root of $T$ such that: (1) There is a node $v \in \pi$ such that $v \in q$; (2) Each occurrence of $p$ between the root and $v$ is a part of a sequence of exactly $k$ consecutive $p$ labeled nodes on $\pi$; (3) There are no sequences of $k+1$ consecutive $p$ labeled nodes on $\pi$, between the root and $v$; (4) The root of $T$ is labeled $p$.*

**Proposition 9.** *For every $k \geq 1$, there is $k' \geq 1$ such that $Block_k$ is expressible by a formula $\varphi(p,q) \in BTL_{k'}$ (i.e., $T, root \models \varphi(p,q)$ iff $T$ has property $Block_k$).*

Our main inexpressibility result for $BTL_k$ is:

**Theorem 10.** *For all $k \geq 1$ and $m > 2^k$, there is no $BTL_k$ formula which expresses the property $Block_m$.*

The proof of Theorem 10 is based on a new Ehrenfeucht-Fraissé game on trees, appropriate for $BTL_k$ (see [15] for details). Theorem 10 and Proposition 9 implies:

**Theorem 11 (Hierarchy).** *For every $k \geq 1$, exists $k' > k$, such that $BTL_{k'}$ is strictly more expressive than $BTL_k$.*

## 4 On Temporal Logics over Branching Time

The infinite hierarchy of branching time logics $BTL_k$ defined in section 3, can serve as an external "yardstick" against which other future $BTL$s can be compared. Below we examine some commonly used branching time logics. These logics are based on different finite and infinite sets of future modalities. Recall that we use $TL(M)$ to indicate 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 $L$.

### 4.1 $CTL^*$ Has No Finite Base

The definition of $CTL^*$ [7] uses an interplay between state formulas (which correspond to genuine modalities) and path formulas to generate infinitely many modalities by a finite syntax. Nevertheless, $CTL^*$ can be represented as the temporal logic based on the infinite set of modalities $\{E\varphi \ : \ \varphi \text{ is a formula of TL(U)}\} \cup \{A\varphi \ : \ \varphi \text{ is a formula of TL(U)}\}$ [5]. $\mathbf{A}\varphi$ (respectively $\mathbf{E}\varphi$) holds at a current moment $t_0$ if "for all (respectively, for some) paths from the current moment, $\varphi$ holds". Formally, $T, t_0 \models \mathbf{E}\varphi$ iff $T_\pi, t_0 \models \varphi$ for some $\pi$ starting at $t_0$. The modalities of the form $\mathbf{A}\varphi$ are redundant since for every $\varphi$ in $TL(\mathbf{U})$, $\mathbf{A}\varphi$ means the same as $\neg\mathbf{E}\neg\varphi$. Thus, the set of modalities $\{\mathbf{E}\varphi : \varphi \in TL(U)\}$ is a base for $CTL^*$. Kamp's theorem implies that the set $\{\mathbf{E}\varphi : \varphi \in TL(U)\}$ of modalities is semantically equivalent to the set of modalities $\{\mathbf{E}\varphi : \varphi$ is a future formula of $FOMLO\}$. Therefore:

**Lemma 12.** $\bigcup_{k=1}^{\infty} BTL_k \equiv_{exp} CTL^*$

Since $\{BTL\}_{k=1}^{\infty}$ is a true infinite hierarchy (Theorem 11), we obtain:

**Theorem 13.** *$CTL^*$ has no finite base.*

In [17], $CTL^*$ was shown to be expressively equivalent to the bisimulation invariant fragment of monadic path logic. Bisimulation equivalence plays a very important role in concurrency. (Recall that $\varphi(x_0)$ is bisimulation invariant if $T, root \models \varphi(x_0)$ implies $T', root' \models \varphi(x_0)$ whenever $T$ and $T'$ are bisimulation equivalent.) Thus, $CTL^*$ represents some objectively quantified expressive power. From the fact that the property $Block_k$ is expressible in $FOMLO$ we obtain the following theorem:

**Theorem 14.** *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 of linear time, where the temporal logic based on the single modality $\mathbf{U}$ is expressively equivalent to the future fragment of first-order monadic logic of order [13]. We believe that this is the reason for the multiplicity of temporal logics over branching time.

### 4.2 $BTL_k$ vs. Commonly Used Branching Time Logics

Many temporal logics were suggested as branching time specification formalisms (see [8, 4]) by imposing some syntactical 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 2). $\mathbf{F}^\infty p$ 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'.

**$B(\mathbf{F})$ (see [14, 4]).** 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 [4], 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 $B(\mathbf{F}) \prec_{exp} BTL_1$.

**$CTL$ and $CTL^+$ (see [1]).** $CTL$ 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 $\Phi_1$ be the set of $TL(\mathbf{U})$ formulas of nesting depth $\leq 1$. 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 $CTL^+ \preceq_{exp} BTL_2$.

**$ECTL$ and $ECTL^+$ (see [4]).** $ECTL$ and $ECTL^+$ allow specification of fairness properties. $ECTL$ 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 $\Phi_2$ be the set of $TL(\mathbf{X}, \mathbf{U}, \mathbf{F}^\infty, \mathbf{G}^\infty)$ formulas of nesting depth $\leq 1$. Let $M_4$ be the infinite set $\{\mathbf{E}\varphi : \varphi \in \Phi_2\}$ of path modalities, then $ECTL^+$ 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, $ECTL \preceq_{exp} BTL_2$ and $ECTL^+ \preceq_{exp} BTL_2$.

In [7], the formula $\mathbf{AF}(p \wedge \mathbf{X}p)$ was provided as an example for a $CTL^*$ formula which is not expressible in $ECTL^+$. 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 $BTL_2$.

It is an open question whether $ECTL^+$ is strictly less expressive than $BTL_2$.

## 5 Complexity of Model Checking

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 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 [2]. The next Theorem shows that for a temporal logic based on a finite set of modalities, the model checking problem has low complexity.

Recall that modal $\mu$ calculus is equivalent to the bisimulation invariant fragment of (future) monadic second-order logic [12].

**Theorem 15 (Complexity of Model Checking).** *Let $TL(M_1, M_2, \ldots, M_r)$ be a $TL$ based on a finite set of modalities.*

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

*Proof.* (Sketch) (1) Is easily reducible to (2).

The model checking algorithm uses standard techniques [5]. Given a structure $K$ and a formula $\varphi$ the algorithm takes the sub-formulas of $\varphi$ starting with the innermost ones, and iteratively labels with each sub-formula $\chi$ the states of $K$ that satisfy $\chi$. The only nontrivial case is when $\chi$ has the form $M(\varphi_1, \ldots \varphi_l)$, where $M$ is a modality. This case is explained below for 2 and 3.

(2) First, for every basic modality $M_i = E\psi_i$ $(i = 1, \ldots, r)$ of the logic, construct a Buchi automaton $A_i$ such that $A_i$ accepts the $\omega$-language definable by $\psi_i$. 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 $M_i(\varphi_1, \ldots, \varphi_l)$ 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 $M_i(\varphi_1, \ldots, \varphi_l)$ are easily extracted from $S$.

The time complexity of this step is bounded by $\mid K \mid \times \mid A_i \mid$. Since the number of sub-formulas of $\varphi$ 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 $M_i$ $(i = 1, \ldots, r)$ are $\mu$ formulas. The set of states which should be labeled by $M_i(\varphi_1, \ldots, \varphi_l)$ can be found by applying the Tarski-Kanster algorithms. A naive implementation of the algorithm runs in time $\leq \quad \mid K \mid^{d_i} \times \mid M_i \mid^{d_i}$, 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 $TL(M_1, \ldots, M_r)$ is bounded by $\mid K \mid^d \times \mid \varphi \mid \times b = O(\mid K \mid^d \times \mid \varphi \mid)$.

## Acknowledgments

We would like to thank Yoram Hirshfeld for his insights that influenced this research.

## References


1. E.M. Clarke and E.A. Emerson (1981). Design and verification of synchronous skeletons using branching time temporal logic. *LNCS* **131**:52–71.
2. E.M. Clarke, E.A. Emerson and A.P. Sistla. Automatic verification of finite state concurrent system using temporal logic. In: *POPL*, 1983.
3. A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. Fundamenta mathematicae 49,129-141,1961.
4. E.A. Emerson (1990). Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. Elsevier, Amsterdam, 1990.
5. E.A. Emerson (1996). *Automated Temporal Reasoning about Reactive Systems.* LNCS vol. 1043, pp. 41-101, Springer Verlag 1996.
6. E.A. Emerson and J.Y. Halpern (1982). Decision procedures and expressiveness in the temporal logic of branching time. In *STOC'82.* pp. 169-180.
7. 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.
8. E.A. Emerson and C.L. Lei. Modalities for model checking: branching time strikes back. *12th ACM Symp. on POPL*, pp. 84-96, 1985.
9. D. Gabbay, I. Hodkinson and M. Reynolds (1994). *Temporal Logic.* Oxford University Press.
10. T. Hafer and W. Thomas (1987). Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *ICALP'87*, *LNCS* **267**:269–279, Springer-Verlag.
11. Y. Hirshfeld and A. Rabinovich(1999). Quantitative Temporal Logic. *LNCS* **1683**:172-187, Springer-Verlag.
12. D. Janin and I. Walukiewicz (1996). On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *CONCUR'96*, *LNCS* **1119**:263–277, Springer-Verlag.
13. H.W. Kamp (1968). Tense logic and the theory of linear order. PhD Thesis, University of California, Los Angeles.
14. L. Lamport (1980). "Sometimes" is sometimes "not never" - On the temporal logic of programs. In *POPL'80.* pp. 174-185.
15. S. Maoz (2000). Infinite Hierarchy of Temporal Logics over Branching Time Models. M.Sc. Thesis, Tel-Aviv University.
16. R. Milner (1989). *Communication and Concurrency.* Prentice-Hall.
17. F. Moller and A. Rabinovich (1999). On the expressive power of CTL*. *Proceedings of fourteenth IEEE Symposium on Logic in Computer Science*, 360-369.
18. D.M.R. Park (1981). Concurrency and automata on infinite sequences. *LNCS* **104**:168–183.