Expressive Power of Temporal Logics

Alexander Rabinovich

School of Computer Science,
Tel Aviv University
Israel 69978
rabino@cs.tau.ac.il
http://www.cs.tau.ac.il/∼rabino

Abstract. The objectives of this paper is to survey classical and recent expressive completeness results and to provide some external yardsticks by which the expressive power of temporal logics can be measured.

1 Introduction

Various temporal logics have been proposed for reasoning about so-called "reactive" systems, computer hardware or software systems which exhibit (potentially) a non-terminating and a 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*) introduced to Computer Science by Pnueli in [28] is a convenient framework for the specification properties of "reactive" systems. This made temporal logics a popular subject in the Computer Science community and it enjoyed an 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 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 (partially) ordered set with monadic predicates is a model for *TL* and *MLO*, but the main, *canonical*, linear time intended models are the non-negative integers $\langle N, <\rangle$ for discrete time and the non-negative reals $\langle R^+, <\rangle$ for continuous time.

A major result concerning *TL* is Kamp's theorem [19, 12] 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.

The objectives of this paper is to survey classical and recent expressive completeness results and to provide some external yardsticks by which the expressive power of the temporal logics can be measured. We consider natural fragments of *MLO*, and for each of these fragment L we address the question whether there is a temporal logic with a finite set of modalities which has the same expressive power of L. It turns out that for most of these fragments the answer is negative.

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 about temporal logics and modalities. Section 4 provides the definition of expressive completeness. Section 5 surveys classical and recent results for expressive completeness over linear time models. The temporal logic with one modality **U** *"until"* is usually referred to as the standard linear time temporal logic (see e.g. survey [8]). The "linear time" appears in the name of this logic probably because when it was introduced by Pnueli its intended models were linear orders. The adjective "standard" is probably used due to the Kamp theorem which states that it is expressively equivalent to the first-order fragment of *MLO* - a very robust formalism.

Sections 6 and 7 deal with temporal logic interpreted over trees - branching time models. In contrast to the linear case many temporal logics over branching time models were suggested and investigated starting from [20, 3, 31, 9, 10]. In section 6 the expressive power of temporal logics over branching time models is compared with the fragments of *MLO* and some objective yardsticks by which to measure the expressive power are suggested. In Section 7 we offer an explanation for the multiplicity of temporal logics over branching time and discuss the complexity of the model checking problem for finite base temporal logics.

2 Monadic Logic of Order

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

2.1 Notations

We use standard notations and abbreviations. A (relational) signature is given by a set of relational symbols and their arity. Let A be a structure for a signature τ. We use $\mid A \mid$ 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 "$\langle a_1, \dots, a_n \rangle \in R$" instead of "$\langle a_1, \dots, a_n \rangle \in R^A$". For a structure A over a signature $\tau = \{\dots, R_i, \dots\}$ we use notations $\langle \mid A \mid, \dots, R_i^A, \dots \rangle$ which we also abbreviated to $\langle \mid A \mid, \dots, R_i, \dots \rangle$.

2.2 Syntax

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

The atomic formulas are of the form $x_1 = x_2, x_1 < x_2$, $x \in X$ and $x \in P$, where x_i (respectively, X and P) ranges over individual variables (respectively, set variables and monadic predicate 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 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 $\mathrm{qd}(\exists x \varphi) = \mathrm{qd}(\exists X \varphi) = 1 + \mathrm{qd}(\varphi)$.

2.3 Semantics

The semantics of *MLO* follows classical lines. A structure for *MLO* is a tuple $M = \langle \mid M \mid, <, \mathcal{P}_1, \dots, \mathcal{P}_n, \dots \rangle$ where $<$ is a partial order over a set $\mid M \mid$ and $\mathcal{P}_1, \dots, \mathcal{P}_n, \dots \subseteq \mid M \mid$ are monadic predicates. If M is such a structure, $s_1, \dots, s_m \in \mid M \mid$ are elements of M and $S_1, \dots, S_n \subseteq \mid M \mid$ are sets of elements, we write

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

if the formula φ is satisfied in the structure M 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 omit it.

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

Chains. A structure $M = \langle| M |, <, \mathcal{P}_1, \ldots, \mathcal{P}_n, \ldots\rangle$ where $<$ is a linear order over $| M |$ and $\mathcal{P}_1, \ldots, \mathcal{P}_n, \ldots$ are monadic predicates is called a *colored (or labelled) chain*, or for brevity simply a *chain*.

The important examples of chains are $\langle R^+, <\rangle$ and $\langle N, <\rangle$ - the non-negative real line and the non-negative integers. A labelled chain over the non-negative integers is often called ω-model or ω-structure.

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

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

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

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

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

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

2.4 Fragments of Monadic Logic of Order

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* [16, 14]: 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

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

Since "X is a path" can be expressed in *FOMLO*, 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 [14]. In particular, for every *MPL* sentence φ there is a *FOMLO* sentence ψ such that for every finite tree T

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

Definition 2.1 (Future formula [13]). *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 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 individual variable x_0 whose value depends only on nodes higher than x_0 in the tree. Observe that this is a semantic notion, not a syntactic one.

Remark 2.2 (Syntactical conditions for the property to be a future formula). Let $\varphi(x_0, X_1, \dots X_k)$ 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, $\psi(x_0)$ is a future formula if and only if it is equivalent to a formula $\tilde{\varphi}(x_0)$ with all quantifies relativized to the elements greater than or equal to x_0.

3 Temporal Logics

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

3.1 Temporal Logics and Modalities

The syntax of ***Temporal Logic*** *(TL)* has in its vocabulary a set of variables (sometimes called propositional names) and a set B of ***modality names*** (sometimes called "temporal connectives" or "temporal operators") with prescribed arity $B = \{\#_1^{l_1}, \#_2^{l_2}, \dots\}$ (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. Formally, the syntax of *TL*(B) is given by the following grammar:

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

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 $\mathrm{nd}(\#_i^{l_i}(\varphi_1, \varphi_2, \ldots, \varphi_{l_i})) = 1 + \max_{1 \le j \le l_i} (\mathrm{nd}(\varphi_j))$.

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

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

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

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

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

$$T, s \models \#^l(\varphi_1, \varphi_2, \ldots, \varphi_l) \text{ iff } s \in \#_T^l(R_{\varphi_1}, R_{\varphi_2}, \ldots, R_{\varphi_l})$$

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

Notes.

1. 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.
2. Strictly speaking what we call temporal logic is called *one-dimensional temporal logic*. The syntax of k-dimensional temporal logic is the same as that of one-dimensional temporal logic. However, atomic formulas (variables) are interpreted as k-ary relations; accordingly, every l-place modality $\#^l$ is interpreted in every structure T as an operator which assigns a k-ary relation (over $\mid T \mid$) to every l-tuple of k-ary relations [12].

3.2 Truth Tables

There is not much interest in a modality # whose interpretation is arbitrary and is defined ad-hoc in each and every structure where the temporal logic is to be interpreted. To be of interest a modality needs to have a uniform description in some metalanguage that connects the set of points where $\#(\varphi_1, \dots, \varphi_k)$ holds to the sets of points where each of the φ_i holds. It is an empirical fact that all the temporal modalitities considered in the literature are defined in *MLO* in the following way: for every l-place modality $\#$ there is a formula $\bar{\#}(x_0, X_1, X_2, \dots, X_l)$ of *MLO* with one free first-order variable x_0 and l set variables, such that for every structure T and subsets $R_i \subseteq | T |$:

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

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

Example 3.1 (Some common modalities and their truth tables).

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

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

– The one-place modality G ("*globally*"); its truth table is

$$\varphi_{\mathsf{G}}(x_0, X_1) \triangleq \forall y(y > x_0 \ \rightarrow \ y \in X_1).$$

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

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

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

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

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

$$\varphi_{\mathsf{U}_{ns}}(x_0, X_1, X_2) \triangleq \exists y\big(y \geq x_0 \ \wedge \ y \in X_2 \ \wedge \ \forall z(x_0 \leq z < y \ \rightarrow \ z \in X_1)\big).$$

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

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

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

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

– The two-place modality S ("*since*"); its truth table is

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

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

Most of the temporal logics studied in Computer Science use only modalities having truth tables definable by future *MLO* 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.

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

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 $\mathbf{E}\varphi$ as follows:*

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

The modality $\mathbf{E}\varphi$ 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 $\mathbf{E}\varphi$ 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 $\mathbf{E}\varphi$ has an MPL truth table.*

4 Expressive Completeness - Basic Definitions

Different temporal logics may be considered a convenient way to use fragments of *MLO*. The following are standard definitions and notations to discuss the expressive power of temporal logics. A temporal logic formula φ is *equivalent* to an *MLO* formula $\psi(x_0, X_1, \ldots, X_l)$ with a single free first-order variable (or to a temporal logic formula ψ) over a structure 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 structures) iff for every T (respectively, for every $T \in C$), φ is equivalent to ψ over T.

[1] Recall that T_π is the substructure of T over the set of nodes π (see Subsection 2.3)

Definition 4.1. *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 all trees, we write $\preceq_{exp}$ for $\preceq^C_{exp}$. The relations $\equiv_{exp}$ and $\prec_{exp}$ are defined similarly.*

Definition 4.2 (Expressive Completeness). *Let L be a fragment of MLO and let C be a set of structures. 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.*

By a straightforward induction on the structure of *TL* formulas one can show

Proposition 4.3. *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".

We say that a fragment L of *MLO* (or a temporal logic L) *has* (or *admits*) a **finite basis** over a class C of structures if there is a finite set of modalities $M_1, \dots, M_k$ such that L is expressively equivalent over C to $TL(M_1, \dots, M_k)$.

5 Linear Time

In the next subsection we recall classical theorems which show that over linear time models *FOMLO* is expressively equivalent to a finite base temporal logic (i.e., temporal logic with a finite set of modalities). In the second subsection we show that there is no finite base temporal logic which is expressively equivalent to *MLO*. The third subsection shows that there is no finite base temporal logic which is expressively equivalent over the reals to the future fragment of *FOMLO*.

5.1 Kamp's and Stavi's Theorems

A major result concerning *TL* and *FOMLO* is Kamp's theorem which implies that $TL(\mathbf{U}, \mathbf{S})$, the temporal logic having "*Until*" and "*Since*" as the only modalities, is expressively complete for *FOMLO* over the two canonical models - real numbers and natural numbers. There is a technical property which unifies in the general case the treatment of real and discrete time: recall that a chain is Dedekind complete iff every non empty bounded subset has a lowest upper bound. The natural numbers are trivially Dedekind complete since every bounded set is finite and has a maximal element. The real line is also Dedekind complete. Kamp's result applies to both these models:

Theorem 5.1 (Kamp [19, 12]). *The temporal logic with two modalities* U *("until") and* S *("since") is expressively complete for FOMLO over the class of Dedekind complete chains.*

Observe that the rationals are not Dedekind complete. It turns out that the temporal logic with U and S modalities is less expressive than *FOMLO* over the rationals [34, 12]. Stavi introduced two more modalities U^{Stavi} and S^{Stavi} and proved

Theorem 5.2 (Stavi [34, 12]). *The temporal logic with four modalities* U, S, U^{Stavi} *and* S^{Stavi} *is expressively complete for FOMLO over the class of all linear orders.*

5.2 No Finite Base for *MLO*

Some natural properties are not expressible in the first-order logic [23]. For example, in the first-order fragment of *MLO* it is impossible to specify any of the following properties $Mult_p(x_0, X)$ $(1 < p \in \mathbb{N})$:

$Mult_p(x_0, X)$: $Mult_p(x_0, X)$ holds at point x_0 iff there is exactly one $x > x_0$ such that X holds at x, and the number of elements between x_0 and this x is a multiple of p plus one.

However, for every p, the property $Mult_p(x_0, X)$ is easily expressible in the future fragment of MLO.

In [1] we addressed the question of existence of a finite base temporal logic which is expressively equivalent (over the naturals) to MLO.

The main technical lemma implies

Lemma 5.3 ([1]). *For every finite set of modalities* $\{M_1, \dots M_k\}$ *with truth tables in MLO there is* $p \in \mathbb{N}$ *such that the property* $Mult_p(x_0, X)$ *is not expressible (over the naturals) in* $TL(M_1, \dots M_k)$.

As a consequence of this lemma we obtain

Theorem 5.4 ([1]). *There is no finite base temporal logic which is expressively equivalent (over the naturals) to monadic logic of order.*

The above theorem can be strengthened as follows:

Theorem 5.5. *Let* C *be a set of linear orders which is either infinite or contains an infinite linear order. There is no finite set of modalities such that the temporal logic over this set of modalities is expressively equivalent over* C *to monadic logic of order.*

5.3 No Finite Base for the Future Fragment of *FOMLO*

Most of the temporal formalisms studied in Computer Science concern only future-time modalities (see [21] for the discussion of past-time modalities in specification formalisms). Is there also a natural temporal logic that is expressively complete for the fragment of future *FOMLO* formulas with respect to the canonical models? Is it simply $TL(\mathbf{U})$, and if not, can we add some naturally defined modalities to obtain an expressively complete future logic?

Theorem 5.6 ([13]). *The temporal logic with the single modality* $\mathbf{U}$ *("until") is expressively equivalent (over the naturals) to the future fragment of FOMLO.*

Thus, in the discrete case $TL(\mathbf{U})$ is as strong as the future fragment of *FOMLO*. The following theorem from [17] shows that for continuous time the situation is radically different.

Theorem 5.7 ([17]). *There is no finite base temporal logic which is expressively equivalent (over the reals) to the future fragment of FOMLO.*

Let $\langle A, <\rangle$ be a linear order and let $P \subseteq A$. A point x is a left limit point of P if for every $y < x$, the interval (y, x) contains a point from P. Let $P^{(1)}$ be the set of left limit points of P and let $P^{(m+1)}$ be the set of left limit points of $P^{(m)}$. It is easy to express in *FOMLO* the following property of an unary predicate P.

$Prop_m(P)$**:** "there is in the future a point in $P^{(m)}$.

The main technical lemma in [17] shows that if a temporal logic L has modalities definable by future *FOMLO* formulas of quantifier depth at most m, then $Prop_m(P)$ is not expressible in L. Theorem 5.7 is an easy consequence of this lemma.

6 Branching Time

Milner and Park [24, 27] 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. The logic $TL(\mathbf{U})$ is usually referred to as the standard linear time temporal logic (see e.g., survey [8]). 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 used 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.

Recall that $TL(\mathbf{U})$ as any logic with modalities defined in *MLO* is interpreted not only over linear orders, but over arbitrary partial orders (according to the truth table of $\mathbf{U}$); 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 "along all futures, eventually p" (that is, "p is inevitable") is not expressible in $TL(\mathbf{U})$.

In order to reflect branching properties of computations many temporal logics were suggested starting from [20, 3, 31, 9, 10]. 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 a 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 [3]. 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 the 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, 35]) 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]:

> "... 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." (page 44 in [8]).

In Section 7 we offer an explanation for the multiplicity of temporal logics over branching time. In Section 6 we suggest some yardsticks by which to measure the expressive power of 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 [25], CTL^* was shown to be expressively equivalent to the bisimulation invariant fragment of monadic path logic [14, 16]. Thus, at least CTL^* represents an objectively quantified expressive power.

In Subsection 6.3 we describe a sequence TLQ_k ($k \in N$) of temporal logics. All these logics are (expressively equivalent to) fragments of CTL^* and their union has the same expressive power as CTL^*. Roughly speaking, the modalities of TLQ_k correspond to formulas with quantifier-depth at most k. However, for every m and k there is a TLQ_k formula which is equivalent to no MLO formula with quantifier-depth $\leq m$. It turns out that the sequence TLQ_k ($k \in N$) of temporal logics forms a hierarchy according to the expressive power [32], i.e., TLQ_{k+1} is strictly more expressive than TLQ_k. Consequently, we obtain that there is no finite base temporal logic which is expressively equivalent to CTL^*, and hence there is no finite base temporal logic which is expressively equivalent to the bisimulation invariant fragment of monadic path logic.

In Subsection 6.4 and in Subsection 6.5 we present two additional hierarchies of temporal logics. The first hierarchy is based on the alternation depth of the truth tables for the logical modalities and the second one is based on the arity of the logical modalities.

6.1 *CTL**

Computation Tree Logic (CTL) was introduced in [3]. 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 the expressive completeness of $TL(\mathbf{U})$ over ω-models (the canonical discrete-time 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) to generate infinitely many modalities by a finite syntax. CTL^* is expressively equivalent to the temporal logic based on the infinite set of modalities $\{E\varphi : \varphi$ is a formula of $TL(\mathbf{U})\} \cup \{A\varphi : \varphi$ is a formula of $TL(\mathbf{U})\}$ [7, 32]. $\mathbf{A}\varphi$ (respectively $\mathbf{E}\varphi$) holds at a current moment t_0 if "for all (respectively, for some) paths from the current moment, φ holds". Formally, $T, t_0 \models \mathbf{E}\varphi$ iff $T_\pi, t_0 \models \varphi$ for some path π starting at t_0.

The modalities of the form $\mathbf{A}\varphi$ are expressible from $\mathbf{E}\varphi$ since for every φ in $TL(\mathbf{U})$, $\mathbf{A}\varphi$ means the same as $\neg\mathbf{E}\neg\varphi$. Thus, the set of modalities $\{\mathbf{E}\varphi : \varphi \in TL(\mathbf{U})\}$ is a base for CTL^*.

Observe that if $\varphi(X_1,\dots,X_l)$ is a $TL(\mathbf{U})$ formula, then by Proposition 4.3(1) there is a future $FOMLO$ formula $\psi(x_0,X_1,\dots,X_l)$ 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

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

Let H be the set of modalities

$$H \triangleq \{\mathbf{E}\psi : \psi \text{ is } FOMLO \text{ future formula}\}.$$

From the above observation it follows that $TL(M) \preceq_{exp} TL(H)$. Theorem 5.6 implies that for every formula $\psi(x_0, X, \dots, X_l)$ 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 $TL(H)$ are expressively equivalent.

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

Lemma 6.1. *CTL^* is expressively equivalent to $TL(\{\mathbf{E}\varphi : \varphi \in TL(\mathbf{U})\}) \equiv_{exp}$ $TL(\{\mathbf{E}\psi : \psi$ is FOMLO future formula$\}$.*

6.2 CTL* versus Monadic Path Logic

Bisimulation equivalence [24, 27] 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 (future *MLO*) 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.

Theorem 6.2 ([25]). *CTL^* is expressively equivalent (over trees) to the bisimulation invariant future fragment of MPL.*

Thus, CTL^* represents an objectively quantified expressive power.

The counting operator $\mathbf{D}^n X$ with $n>0$ expresses "*for at least $n>0$ different successors* have property X". The truth table for $\mathbf{D}^n$ is

$$\exists y_1 \ldots \exists y_n (\bigwedge_{i \neq j} y_i \neq y_j) \wedge (\bigwedge_{i=1}^{n} (y_i \in X \wedge Succ(x_0,\ y_i)),\ where$$

$Succ(x, y)$ (y is a successor of x) is $x < y \wedge \neg\exists z(x < z < y)$. One can easily describe bisimilar trees T and T' such that $T, root \models \mathbf{D}^2 X$, but $T', root' \not\models \mathbf{D}^2 X$. Hence, modalities $\mathbf{D}^n$ are not bisimulation invariant.

Theorem 6.3 ([26]). *The extension of CTL^* by the counting operators $\{\mathbf{D}^n\}_{n=1}^{\infty}$ is expressively equivalent (over trees) to the future fragment of MPL.*

Theorem 6.4. *The extension of CTL^* by the counting operators $\{\mathbf{D}^n\}_{n=1}^{\infty}$ and by $\mathbf{S}$ ("since") is expressively equivalent (over trees) to MPL.*

6.3 Quantifier Hierarchy

The quantifier depth of a modality (i.e. of its truth table) is a natural measure of its complexity.

Definition 6.5 (Logic TLQ_k). *For every $k \geq 1$, let TLQ_k be the temporal logic with the set of basic modalities defined by the following set Q_k of truth tables:*

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

Notes.

1. The basic modalities of TLQ_k are path modalities. Their truth tables can be defined by *MPL* formulas with only one path quantifier. However, formulas of TLQ_k (as formulas of every temporal logic) permit arbitrarily deep nesting of modalities and can express "branching" properties which require many path quantifiers.

2. The basic modalities of TLQ_k have a bounded quantifier depth, but formulas of TLQ_k permit arbitrarily deep nesting of modalities. Properties of "an arbitrary quantifier depth" can be expressed even in TLQ_1. More precisely, for every n, there is a TLQ_1 formula φ_n which is equivalent to no MLO formula of quantifier-depth $\leq n$ [32].
3. Observe that Lemma 6.1 implies that $\bigcup\limits_{k=1}^{\infty} TLQ_k \equiv_{exp} CTL^*$.

Theorem 6.6 (Quantifier Depth Hierarchy [32]). *TLQ_{k+1} is strictly more expressive than TLQ_k.*

On the expressive power of TLQ_2. Many temporal logics were suggested as branching time specification formalisms (see [11, 7]) by imposing some syntactic restrictions on CTL^* formulas. In [32] we examined the expressive power of commonly used branching time temporal logics. It turns out that almost all of these logics are inside TLQ_2 - the second level of our hierarchy. The modalities for these logics were suggested by the 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 TLQ_2 formulas constructed from basic modalities of quantifier-depth two.

In [33] it was shown that $B(\mathsf{U}, \mathsf{F}, \mathsf{F}^\infty, \mathsf{G}^\infty, \wedge, \neg)$, the classical extension of CTL with fairness properties [10], is expressively equivalent to TLQ_2. This result, linking $B(\mathsf{U}, \mathsf{F}, \mathsf{F}^\infty, \mathsf{G}^\infty, \wedge, \neg)$ to a natural fragment of the monadic logic of order, provides a characterization that other branching time logics, e.g., CTL, lack.

It was also shown in [33] that TLQ_2 does not admit a finite base (i.e., it is expressively equivalent to no temporal logic with a finite set of modalities).

6.4 Alternation Depth Hierarchy

An alternation type of a first-order formula is a finite sequence of positive natural numbers. We define below when a formula φ has an alternation type $\langle n_1, \dots n_k\rangle$. However, unlike the quantifier depth, a formula has many alternation types.

1. If $\bar{n}$ is an alternation type and φ is atomic then $\bar{n}$ is an alternation type of φ.
2. If $\bar{n}$ is an alternation type of φ_1 and of φ_2 then $\bar{n}$ is an alternation type of $\neg\varphi_1$,of $\varphi_1 \wedge \varphi_2$ and of $\varphi_1 \vee \varphi_2$.
3. If $\langle n_1, \dots n_k\rangle$ is an alternation type of φ then $\langle n_0, n_1, \dots n_k\rangle$ is an alternation type of $\forall u_1 \dots \forall u_m \varphi$ and of $\exists u_1 \dots \exists u_m \varphi$ for $n_0 \geq m$.

For example, if φ is in prenex normal form with the quantifier prefix $\forall^2\exists^3$ then $\langle 2, 3\rangle$, $\langle 1, 1, 1, 1, 1\rangle$ and $\langle 2, 2, 1\rangle$ are alternation types of φ, but $\langle 1, 3, 1\rangle$ is not one of its alternation types. Finally, we say that φ has an alternation depth (or rank) k if φ has an alternation type of the length k.

Definition 6.7 (Logic $TLAL_k$). *For every $k \geq 1$, let $TLAL_k$ be the temporal logic with the set of basic modalities defined by the following set AL_k of truth tables:*

$$AL_K = \{\mathbf{E}\varphi : \textit{alternation depth of } \varphi(x_0, X_1, \dots, X_l) \leq k \\ \textit{and } \varphi \textit{ is a first-order future formula}\}.$$

Theorem 6.8 (Alternation Hierarchy). *$TLAL_{k+1}$ is strictly more expressive than $TLAL_k$.*

6.5 Arity Hierarchies

Let $Fair_k(X_1, \dots, X_k)$ be the modality such that $T, s \models Fair_k(X_1, \dots, X_k)$ iff there is a path from s along which every X_i is satisfied infinitely often and where only nodes satisfying some of the X_is are encountered.

It is clear that $Fair_k$ has a truth table in Monadic path logic; moreover, $Fair_k$ is TLQ_2 modality. It was shown in [33]

Lemma 6.9 ([33]). *The modality $Fair_{k+1}$ is not expressible in $TL(\mathbf{U}, \{Fair_n\}_{n=1}^{k})$ for every k.*

Definition 6.10 (Logic $TLAR_k$). *For every $k \geq 1$, let $TLAR_k$ be the temporal logic based on all modalities of arity at most k defined by MLO future formulas.*

Lemma 6.9 can be strengthened as follows:

Lemma 6.11. *The modality $Fair_{k+1}$ is not expressible in $TLAR_k$ for every k.*

As a consequence one obtains

Theorem 6.12 (Arity Hierarchy). *$TLAR_{k+1}$ is strictly more expressive than $TLAR_k$.*

Corollary 6.13. *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.*

7 Why so many Temporal Logics climb up the trees

Two of the most important characteristics of a TL are (1) its expressive power and (2) the complexity of its model checking problem [8]. Theorem 7.1 shows that for natural fragments of *MLO* there is no expressive equivalent finite base temporal logics. On the other hand, Theorem 7.2 below shows that for every finite set of modalities $M_1, M_2, \dots, M_r$, the complexity of model checking for $TL(M_1, M_2, \dots, M_r)$ 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.

7.1 There is no natural finite base TL over Trees

Theorem 7.1. *There are no finite base temporal logics which are expressive equivalent over trees to the the following fragments of MLO.*

1. *FOMLO and the future fragment of FOMLO.*
2. *Like in (1) with "MPL" or "MLO" replacing "FOMLO".*
3. *The future bisimulation invariant fragment*[2] *of MPL.*
4. *Like in (3) with "FOMLO" or "MLO" replacing "MPL".*

About proofs: (1) Recall that $\mathbf{D}^n$ is the counting operator and $\mathbf{D}^n X$ says "*at least n different successors* have property X". The truth table for $\mathbf{D}^n$ (see Sect. 6.2) uses n different bound variables. Moreover, any *FOMLO* formula which is equivalent (over trees) to $\mathbf{D}^n X$ should use at least $n+1$ different variables. From this observation it is derived in [12] (see Section 1.8) that there is no finite base temporal logic which is expressive equivalent over trees to *FOMLO* [12]. The arguments for the future fragment of *FOMLO* and for (2) are similar to (1).

(3)-(4) can be derived from the Hierarchy Theorems (Theorems 6.6, 6.8, 6.12) or their proofs. For example, to prove (3) recall that CTL^* is equivalent to $\bigcup TLQ_k$ and by Theorem 6.6 $\{TLQ_k\}_{k=1}^{\infty}$ contains a strict hierarchy. Hence, CTL^* cannot have a finite base. This together with 6.2 establish (3).

7.2 Complexity of Model Checking

The model checking problem for a logic L is as follows [3, 31, 7, 5]. 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 [4]. 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].

Theorem 7.2 (Complexity of Model Checking [32]). *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* $O(\mid K \mid \times \mid \varphi \mid)$.

[2] This fragment is equivalent to CTL^*

Acknowledgments

I am grateful to Danièle Beauquier, Yoram Hirshfeld, Shahar Maoz, Faron Moller, Philippe Schnoebelen and Boris Trakhtenbrot. Many results reported in this paper were obtained in our joint work.

References

1. D. Beauquier and A. Rabinovich (2002). Monadic Logic of Order over Naturals Has no Finite Base. Journal on Logic and Computations **12**:(2) pp 243-253.
2. 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, pp. 164-176.
3. 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.
4. E.M. Clarke, E.A. Emerson and A.P. Sistla (1983). Automatic verification of finite state concurrent system using temporal logic. In *Proceedings of the 10th Annual ACM Symposium on Principles of Programming Languages.*
5. E. Clarke and B.H. Schlingloff (1999). Model checking. In A. Robinson and A. Voronkov, editors, *Handbook on Automated Reasoning*, pages 3–157. Elsevier Science Publishers B.V.
6. E. A. Emerson and E. M. Clarke (1980). Characterizing correctness properties of parallel programs using fixpoints. In *Proc. 7th Coll. Automata, Languages and Programming (ICALP'80), Noordwijkerhout, NL,* , 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. D. Gabbay, I. Hodkinson and M. Reynolds (1994). *Temporal Logic.* Oxford University Press.
13. D. Gabbay, A. Pnueli, S. Shelah and J. Stavi (1980). On the temporal analysis of fairness. *7th ACM Symp. on Principles of Prog. Lang.*, pp. 163-173, 1980.
14. Y. Gurevich and S. Shelah, "To the decision problem for branching time logic", In "Foundations of Logic and Linguistics: Problems and their Solutions" (ed. P. Weingartner and G. Dold), Plenum, 1985, 181–198.
15. Y. Gurevich and S. Shelah (1985). The decision problem for branching time logic. *Journal of Symbolic Logic* **50**(3):668–681.

16. 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.
17. Y. Hirshfeld and A. Rabinovich (2000). Future temporal logic needs infinitely many modalities. *Technical Report*, Tel-Aviv University.
18. 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.
19. H. W. Kamp (1968). *Tense logic and the theory of linear order.* PhD Thesis, University of California, Los Angeles.
20. 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.
21. F. Laroussinie and Ph. Schnoebelen(2000). Specification in CTL+Past for verification in CTL. Information and Computation, 156(1/2):236-263.
22. S. Maoz (2000). Infinite Hierarchy of Temporal Logics over Branching Time Models. M.Sc. Thesis, Tel-Aviv University.
23. R. McNaughton and S. Papert (1971). Counter-Free Automata. *MIT Press*, 1971.
24. R. Milner (1989). *Communication and Concurrency.* Prentice-Hall.
25. F. Moller and A. Rabinovich (1999). On the expressive power of CTL*. *Proceedings of fourteenth IEEE Symposium on Logic in Computer Science*, 360-369.
26. F. Moller and A. Rabinovich (2001). Counting on CTL*: On the Expressive power of Monadic Path Logic. Submitted.
27. D.M.R. Park (1981). Concurrency and automata on infinite sequences. *Lecture Notes in Computer Science* **104**:168–183.
28. A. Pnueli (1977). The temporal logic of programs. In *Proc. IEEE 18th Annu. Symp. on Found. Comput. Sci.*, pages 46–57, New York, 1977. IEEE.
29. A. Prior (1967). Past, Present and Future. Claredon Press, Oxford.
30. M. O. Rabin (1969). Decidability of second order theories and automata on infinite trees. In *Trans. Amer. Math. Soc.*,**141**,pp 1-35.
31. J.-P. Queille and J. Sifakis (1983). Fairness and related properties in transition systems. A temporal logic to deal with fairness. *Acta Informatica*, 19(3):195–220.
32. A. Rabinovich and S. Maoz (2001). An Infinite Hierarchy of Temporal Logics over Branching Time. Information and Computation, 171:306-332, 2001.
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. Stavi (1979). Functional Completeness over the rationals. Unpublished, Bar Ilan University, Israel, 1979.
35. C. Stirling (1992). Modal and temporal logics. In *Handbook of Logic in Computer Science*, Oxford University press, 476-567.
36. W. Thomas 1990). Automata on Infinite Objects. In J. van Leeuwen editor *Handbook of Theoretical Computer Science*, The MIT Press.
37. W. Thomas (1997). Languages, Automata, and Logic. In Handbook of Formal Language Theory (G. Rozenberg, A. Salomaa, Eds.),Vol. III, Springer-Verlag , New York, pp. 389-455.
38. M. Vardi and P. Wolper (1984). Yet another process logic. *In Logics of Programs, Lecture Notes in Computer Science* **164**, 501-512, Springer-Verlag.