BTL_2 and the expressive power of $ECTL^+$

Alexander Rabinovich

Dept. of Computer Science, Tel Aviv University
Ramat Aviv, Tel Aviv 69978, Israel

Philippe Schnoebelen

Lab. Spécification et Vérification (LSV)
ENS de Cachan & CNRS, France

Abstract

We show that $ECTL^+$, the classical extension of CTL with fairness properties, is expressively equivalent to BTL_2, a natural fragment of the monadic logic of order. BTL_2 is the branching-time logic with arbitrary quantification over paths, and where path formulae are restricted to quantifier depth 2 first-order formulae in the monadic logic of order. This result, linking $ECTL^+$ to a natural fragment of the monadic logic of order, provides a characterization that other branching-time logics, e.g., CTL, lack. We then go on to show that $ECTL^+$ and BTL_2 are not finitely based (i.e., they cannot be defined by a finite set of temporal modalities) and that their model-checking problems are of the same complexity.

Key words: Expressivity of branching-time temporal logic, Model checking

1 Introduction

Temporal Logic. Temporal logic is a popular formalism for reasoning about "reactive" systems, i.e., systems with (potentially) non-deterministic and non-terminating behavior [Eme90,MP92,MP95,CGP99]. What makes temporal logic attractive is its combination of good expressive power with feasible model checking [Eme96].

Email addresses: rabino@math.tau.ac.il (Alexander Rabinovich), phs@lsv.ens-cachan.fr (Philippe Schnoebelen).

In temporal logic, the properties of the system are described by *atomic propositions* that hold at some points in time but not at others. More complex properties are obtained by using Boolean connectives and *temporal modalities* that build up a statement on the current point by combining statements on points temporally related to it.

With a set $\{\mathsf{M}_1, \mathsf{M}_2, \ldots\}$ of modalities, one obtains a temporal logic denoted by $TL(\mathsf{M}_1, \mathsf{M}_2, \ldots)$. Choosing different modalities yields different temporal logics and the literature contains a large number of different proposals.

Expressivity. When it comes to arguing in favor of a given set of modalities, an important criterion is the expressive power of the resulting logics (see the survey [Rab02]). It is nice when a small set of modalities is provably sufficient for expressing all the properties from a natural and robust class.

For example, one of the most important results in the field is Kamp's Theorem [Kam68,GHR94], stating that $TL(\mathsf{U}, \mathsf{S})$, the temporal logic having only the modalities "*Until*" and "*Since*" [1], has the same expressive power over natural linear structures (e.g., $\langle \mathbb{Z}, \leq \rangle$, called *discrete time*, or $\langle \mathbb{R}, \leq \rangle$, called *real time*, or their positive segments) as *FOMLO*, the first-order logic of order with monadic predicates. If one replaces the binary U and S by the unary F and F^- ("*Future*" and "*Past*"), then $TL(\mathsf{F}, \mathsf{F}^-)$ has the same expressive power as the two-variable fragment of *FOMLO* [EVW02].

Branching time. Kamp's theorem is about temporal logics over linear structures, called *linear-time* logics, but many popular temporal logics, called *branching-time* logics [Lam80,EH86], view time as a tree-like set of time points, and are correspondingly interpreted over tree-like partially ordered structures.

Many branching-time logics have been proposed, starting with [Lam80,CE81,QS83,BPM83,EH85,EH86,EL87]. The basic modalities of these logics are obtained by combining a path quantifier "E" or "A" with a formula in $TL(\mathsf{U})$. The formula $\mathsf{E}\phi$ (respectively $\mathsf{A}\phi$) holds at time point t_0 if for some path (respectively, for every path) π starting at t_0 the $TL(\mathsf{U})$ formula ϕ holds along π. For example, a commonly used branching-time logic is *CTL* [CE81,CES86], based on the two binary modalities EU and AU.

Two extensions of *CTL*, namely *ECTL* and $ECTL^+$, have been proposed to deal with fairness properties [EH86]. *ECTL* is $TL(\mathsf{EU}, \mathsf{AU}, \mathsf{EF}^\infty)$ where $\mathsf{F}^\infty p$

[1] These are the *strict* versions of "*Until*" and "*Since*", for which the present is not included in the future. These versions allow expressing "*Next*" and agree with classical notions [Kam68,GPSS80,GHR94].

reads "p holds infinitely often in the future". $ECTL^+$ is more expressive since it allows $\mathsf{E}\phi$ for any formula ϕ in $TL(\mathsf{U}, \mathsf{F}^\infty)$ where modalities cannot be nested.

Finally, the logic CTL^*, from [EH86], is obtained by considering an infinite set of modalities: $\mathsf{E}\phi$ for any formula ϕ in $TL(\mathsf{U})$.

Expressive completeness. In contrast to Kamp's Theorem and the canonical linear models, we are not aware of any existing work proposing a natural predicate logic that corresponds to CTL, $ECTL$ or $ECTL^+$ over trees.

Regarding CTL^*, a recent result [MR03] is that this logic has the same expressive power as the bisimulation-invariant fragment of monadic path logic [GS85,HT87]. Thus at least CTL^* represents some objectively quantified expressive power (indeed, CTL^* is very close to the full monadic path logic [MR03]).

Finite bases. A temporal logic TL has a finite basis if it is built using only a finite set of modalities (such as CTL, $ECTL$, and $TL(\mathsf{U})$). For temporal logics such as CTL^* which are defined via an infinite, albeit "regular", set of modalities, a natural question is whether they could be defined with just finitely many modalities.

For example, CTL^+ is a temporal logic which is traditionally defined via an infinite set of modalities; however it is expressively equivalent to CTL [EH85] so that the infinite set of modalities only provides syntactic sugar (and succinctness [Wil99]) but is not strictly necessary. On the other hand, no finitely-based temporal logic is expressively equivalent to the mu-calculus over (linear) discrete time [BR02], or equivalent to the future fragment of $FOMLO$ over (linear) real time [HR03].

Regarding CTL^*, it was shown that its expressive power cannot be captured by a finite set of modalities, thus providing a partial explanation of why there is no general agreement as what should be the preferred set of modalities for branching-time logics [RM01]. In this paper, Rabinovich and Maoz introduce a sequence BTL_1, BTL_2, ... of temporal logics (where BTL_k has modalities $\mathsf{E}\phi$ for any $FOMLO$ formula ϕ of quantifier depth at most k) and show that there exists an infinite hierarchy (w.r.t. expressive power) among the sequence BTL_1, BTL_2, ... Since CTL^* is exactly as expressive as $BTL \stackrel{\text{def}}{=} \bigcup_k BTL_k$, and since any CTL^* modality is a BTL_k modality for some k, the existence of an infinite hierarchy among $\{BTL_k\}_{k=1,2,\ldots}$ entails that CTL^* has no finite basis.

Our contribution. We prove that $ECTL^+$ is exactly as expressive as BTL_2. This indicates that $ECTL^+$ corresponds to a natural level in expressive power. However, BTL_2 can be exponentially more succinct than $ECTL^+$.

Additionally, we prove that $ECTL^+$ and BTL_2 have no finite basis (unlike BTL_1 [RM01]). This shows that the definition of $ECTL^+$ via an infinite family of modalities is unavoidable, and partially answers the conjecture from [RM01] that no BTL_k for $k > 1$ admits a finite basis.

Finally we show that the model-checking problem for BTL_2 is $\mathbf{\Delta_2^p}$-complete. This shows that model checking is no harder for the more versatile BTL_2 than for $ECTL^+$, and gives a new example of a temporal logic for which model checking is $\mathbf{\Delta_2^p}$-complete.

Plan of the article. In section 2 we recall the necessary notions from Monadic logic of order (MLO). Section 3 recalls how temporal logics can be seen as fragments of MLO and defines the logics we study: $\{BTL_k\}_{k=1,2,\ldots}$, $ECTL^+$, etc. Section 4 proves that $ECTL^+$ and BTL_2 have the same expressive power but are not equally succinct. Finally section 5 proves that these two logics have no finite basis, and section 6 studies the complexity of their model-checking problems.

2 Preliminaries

In this section we review basic definitions and known results about computation trees, the monadic logic of order, and Kripke structures.

2.1 Computation trees and paths

A *tree* $T = (|T|, \leq)$ is a partially ordered set $|T|$ of *nodes* (sometimes also called *states*, or *time points*) in which the predecessors of any given element $a \in |T|$ constitute a finite total order with a common minimal element ε_T, referred to as the *root of the tree*. A *computation tree* is a structure $(|T|, \leq, P_1, P_2, \ldots)$, where $(|T|, \leq)$ is a tree, and P_1, P_2, … are subsets of $|T|$. We say that a node $s \in |T|$ is *labeled by* P_i if $s \in P_i$.

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 $|T_{\geq s}| \stackrel{\text{def}}{=} \{t : t \in |T| \text{ and } t \geq s\}$, and its relations are the corresponding restrictions of $\leq$, P_1, P_2, … from T.

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, ordered by $\leq$. A path π through T induces a substructure, denoted T_π, that is still a computation tree (where only the nodes occurring in π are kept).

2.2 Second-order monadic logic of order

The syntax of *MLO*, the *second-order monadic logic of order*, has in its vocabulary individual first-order variables x_0, x_1, x_2, $\ldots$ (representing nodes), second-order set variables X_0, X_1, X_2, $\ldots$ (representing sets of nodes), and set constants (monadic predicates) P_1, P_2, $\ldots$ Formulae ϕ, ψ, $\ldots$ are built up from atomic formulae of the form $x = x'$, $x \leq x'$, $x \in X$ and $x \in P$, using the Boolean connectives $\wedge$ and $\neg$, and the quantifiers $\exists x$ and $\exists X$. As usual, we use $\bot$, $\top$, $\phi \vee \psi$, $\phi \Rightarrow \psi$, $\phi \Leftrightarrow \psi$, $\forall x\ \phi$, $\forall X\ \phi$ as abbreviations for, respectively, $\exists x\ (x \in P_1 \wedge x \notin P_1)$, $\neg\bot$, $\neg(\neg\phi \wedge \neg\psi)$, $(\neg\phi) \vee \psi$, $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$, $\neg\exists x\ \neg\phi$, $\neg\exists X\ \neg\phi$, and we write $\phi(x_1, \ldots, x_k, X_1, \ldots, X_m)$ when we want to stress that the free variables of ϕ are among x_1, $\ldots$, x_k, X_1, $\ldots$, X_m.

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

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

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

if the formula ϕ is satisfied in the tree T with x_i interpreted as s_i ($i = 1, \ldots, m$) and X_j interpreted as S_j ($j = 1, \ldots, n$).

2.3 Future formulae

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

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

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

In other words, a future formula is a formula with one free node variable x_0 whose value only depends on nodes higher than x_0 in the tree.

Observe that this is a semantic notion, not a syntactic one. However, it is possible to give a syntactic condition ensuring that a formula is a future formula. For this purpose it is convenient to extend the syntax of first-order monadic logic of order by the relativized (or *bounded*) quantifiers $(\exists x)_{\geq x_0}$ and $(\forall x)_{\geq x_0}$. The relativized quantification $(\exists x)_{\geq x_0}\phi$ (respectively $(\forall x)_{\geq x_0}\phi$) is a shorthand for $\exists x.\ x \geq x_0 \wedge \phi$ (respectively, $\forall x.\ x \geq x_0 \Rightarrow \phi$).

Definition 2.2 (Syntactic future formula) *An MLO formula $\phi(x_0, X_1, \ldots, X_k)$ is a* syntactic future formula *if all its quantifiers are of the form $(\exists x)_{\geq x_0}$ and $(\forall x)_{\geq x_0}$.*

The following is immediate.

Lemma 2.3 *Every syntactic future formula is a semantic future formula.*

With $\phi(x_0, X_1, \ldots, X_k)$, we associate a variant ϕ' obtained by replacing all first-order quantifiers "$\forall x$" and "$\exists x$" in ϕ with relativized versions "$(\forall x)_{\geq x_0}$" and "$(\exists x)_{\geq x_0}$". Then for any ϕ, the relativized ϕ' is a syntactic (and hence semantic) future formula. Moreover,

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

where, for $i = 1, \ldots, k$, S'_i is the restriction of S_i to $|T_{\geq s}|$. Hence, ϕ is a future formula iff ϕ and ϕ' are equivalent over trees, i.e., iff $\phi \Leftrightarrow \phi'$ is valid over trees. Incidentally, this implies that being a future formula is decidable since the validity of *MLO* formulae over trees is decidable [Rab69]. To sum up we have

Lemma 2.4 *1. Every future formula is equivalent to a syntactic future formula.*
2. It is decidable whether a formula is a future formula.

Since any future formula ϕ can be replaced by its relativized variant at no cost (same meaning, same free variables, linear increase in size), we assume that future formulae are *syntactic future*, i.e., have relativized quantifications, whenever we describe an algorithm that has "future formulae" as input.

2.4 Fragments of MLO

We denote by *FOMLO* the subset of *first-order formulae of MLO*, i.e., formulae where the second-order quantifier $\exists X$ does not occur.

We also consider *MPL*, the *monadic path logic* [HT87]: its syntax is the same as that of monadic second-order logic but the set variables X_1, X_2, ... range over *paths* rather than over arbitrary sets of nodes. Semantically *MPL* is very closely related to first-order logic [MR03].

Since "*X is a path*" can be expressed in *FOMLO*, *MPL* can be seen as a fragment of *MLO*.

2.5 Kripke structures

A *Kripke structure* is a structure $\mathcal{M} = \langle |\mathcal{M}|, R, P_1, P_2, \ldots \rangle$ where $|\mathcal{M}|$ is a set of nodes, the P_i are subsets of $|\mathcal{M}|$, and $R \subseteq |\mathcal{M}|^2$ is a binary *transition* relation. When $(s, s') \in R$, we say it is possible to move from s to s' in one step. A path π in $\mathcal{M}$ starting from s_0 is a maximal sequence s_0, s_1, ... s.t. $(s_i, s_{i+1}) \in R$ for all i. Maximality implies that a path is either infinite, or ends in a node with no R-successor.

For our purposes, Kripke structures are mainly another way of presenting computation trees: for a node s_0 of some $\mathcal{M}$, the tree $T_{\mathcal{M},s_0}$ (obtained by *unfolding* $\mathcal{M}$) is $\langle |T|, \leq, P'_1, P'_2, \ldots \rangle$ where $|T|$ is the set of all finite prefixes of paths from s_0, $\sigma \leq \sigma'$ iff σ is a prefix of σ', and $\sigma \in P'_i$ if the last node of σ is in P_i. Hence $\varepsilon_{T_{\mathcal{M},s_0}}$ is the sequence "s_0". A path starting from s in $\mathcal{M}$ directly yields a path in $T_{\mathcal{M},s}$ starting from the root.

Given a future *FOMLO* formula ϕ, we write $\mathcal{M}, s \models \phi$ when $T_{\mathcal{M},s}, s \models \phi$, agreeing with the standard interpretation of temporal logics over Kripke structures. We do not use these notions until section 5.

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, with notations adopted from [GHR94,RM01,HR04].

3.1 Temporal logics and modalities

The syntax of *Temporal Logic* (*TL*) has in its vocabulary a countably infinite set of *propositions* $\{q_1, q_2, \ldots\}$ and a possibly infinite set $B = \{\mathsf{H}_1^{l_1}, \mathsf{H}_2^{l_2}, \ldots\}$ of *modality names* (sometimes called "temporal connectives" or "temporal operators") with prescribed arity indicated as superscript (we usually omit the arity notation). *TL*(B) denotes the *temporal logic based on modality-set B* (and B is called the *basis* of $TL(B)$). Temporal formulae are built by combining atoms (the propositions q_i) and other formulae using Boolean connectives and modalities (with prescribed arity). Formally, the syntax of $TL(B)$ is given by the following grammar:

$$\phi \ ::= \ q_i \mid \phi_1 \wedge \phi_2 \mid \neg\phi_1 \mid \mathsf{H}_i(\phi_1, \phi_2, \ldots, \phi_{l_i})$$

The *nesting depth* (or *modal rank*) of a temporal formula ϕ, denoted by $\mathrm{nd}(\phi)$, is defined as usual: $\mathrm{nd}(q_i) = 0$; $\mathrm{nd}(\phi \wedge \phi') = \max(\mathrm{nd}(\phi), \mathrm{nd}(\phi'))$; $\mathrm{nd}(\neg\phi) = \mathrm{nd}(\phi)$; and $\mathrm{nd}(\mathsf{H}_i(\phi_1, \phi_2, \ldots, \phi_{l_i})) = 1 + \max\limits_{1 \leq j \leq l_i} (\mathrm{nd}(\phi_j))$.

Temporal formulae are interpreted over partially ordered sets with monadic predicates and, in particular, over computation trees, the only models we consider here. For this, every modality H comes with its semantics given in every tree T by a mapping $\mathsf{H}_T : 2^{|T|} \times \cdots \times 2^{|T|} \to 2^{|T|}$ which associates a set of nodes with any tuple of l sets of nodes. The idea is that if the S_i's are the sets of nodes where the ϕ_i's hold in T, then $\mathsf{H}_T(S_1, \ldots, S_l)$ is the set of nodes where $\mathsf{H}(\phi_1, \ldots, \phi_l)$ holds in T.

Formally, we define when a temporal formula ϕ holds at a node s of a computation tree $T = (|T|, \leq, P_1, P_2, \ldots)$, written $T, s \models \phi$, by the following inductive clauses:

$$T, s \models q_i \overset{\text{def}}{\Leftrightarrow} s \in P_i$$
$$T, s \models \mathsf{H}(\phi_1, \phi_2, \ldots, \phi_l) \overset{\text{def}}{\Leftrightarrow} s \in \mathsf{H}_T(S_{\phi_1}, S_{\phi_2}, \ldots, S_{\phi_l})$$

where $S_\phi \overset{\text{def}}{=} \{t \mid T, t \models \phi\}$. The usual clauses for Boolean connectives are omitted.

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

When a TL_1 formula ϕ is equivalent to some TL_2 formula ϕ', we say that ϕ *can be expressed in* TL_2. If ϕ has the form $\mathsf{H}(q_1, \ldots, q_l)$, we say that *the modality* H can be expressed in TL_2.

Remark 3.1 *A common situation is that two temporal logics TL_1 and TL_2 are expressively equivalent (they can express the same properties) but one is more succinct than the other (e.g., TL_1 formulae do not admit equivalent formulae* *in TL_2 whose size is bounded by a linear, or a polynomial, function of the size of the TL_1 formula).*

However, if TL_1 only uses a finite set of modalities, then $TL_1 \leq TL_2$ implies that there exists an effective polynomial-time translation from TL_1 to TL_2. Indeed, for every modality H_i in TL_1, let ψ_i be a TL_2 formula equivalent to $\mathsf{H}_i(q_1, \ldots, q_{l_i})$. We now define a translation $[\ ]'$ from TL_1 to TL_2 by structural induction:

$$[q_i]' \stackrel{\text{def}}{=} q_i \qquad [\phi_1 \land \phi_2]' \stackrel{\text{def}}{=} [\phi_1]' \land [\phi_2]'$$
$$[\neg\phi]' \stackrel{\text{def}}{=} \neg[\phi]' \qquad [\mathsf{H}_i(\phi_1, \ldots, \phi_{l_i})]' \stackrel{\text{def}}{=} \psi_i\{q_1 \mapsto [\phi_1]', \ldots, q_{l_i} \mapsto [\phi_{l_i}]'\}$$

where the notation "$\psi\{q \mapsto \phi, \ldots\}$" is used to denote variants where all occurrences of q in ψ have been replaced by ϕ. The length of $[\phi]'$ can be exponential in the length of ϕ but if we store formulae as dags [2]*, then the size of $[\phi]'$ is linear in the size of ϕ, the expansion factor being bounded by the size of the largest ψ_i.*

3.2 Defining modalities in MLO

In practice, most temporal modalities are defined in *MLO*. A *truth table* for an l-place modality H is an *MLO* formula $\psi_{\mathsf{H}}(x_0, X_1, \ldots, X_l)$ with one free first-order variable x_0 (and l free second-order variables) that defines H_T, i.e., such that for every tree T and subsets $S_1, \ldots, S_l$ of $|T|$:

$$\mathsf{H}_T(S_1, \ldots, S_l) \stackrel{\text{def}}{=} \{s \mid T, s, S_1, \ldots, S_l \models \psi_{\mathsf{H}}(x_0, X_1, \ldots, X_l)\}.$$

Abusing notation, we say that H has quantifier depth k if ψ_{H} has.

Example 3.2 (Some common modalities and their truth tables)
The 1-place modalities F, G, X, F^∞ and the 2-place modalities U and S appear in many temporal logics. Informally $\mathsf{F}\phi$ reads "eventually ϕ"*, $\mathsf{G}\phi$ reads* "globally ϕ"*, $\mathsf{X}\phi$ reads* "in the next state ϕ"*, $\mathsf{F}^\infty\phi$ reads* "infinitely often ϕ"*, $\mathsf{U}(\phi_1, \phi_2)$ reads* "ϕ_1 until ϕ_2" *and $\mathsf{S}(\phi_1, \phi_2)$ reads* "ϕ_1 since ϕ_2"*. They all have*

[2] This amounts to defining the size of a formula as the number of its distinct subformulae.

FOMLO truth tables:

$$\begin{aligned}
\psi_{\mathsf{F}}(x_0, X) &\equiv \exists y(y > x_0 \land y \in X), \\
\psi_{\mathsf{G}}(x_0, X) &\equiv \forall y(y > x_0 \Rightarrow y \in X), \\
\psi_{\mathsf{X}}(x_0, X) &\equiv \exists y(y > x_0 \land y \in X \land \forall z(z > x_0 \Rightarrow z \geq y)), \\
\psi_{\mathsf{F}^\infty}(x_0, X) &\equiv \forall y(y > x_0 \Rightarrow \exists z(z > y \land z \in X)), \\
\psi_{\mathsf{U}}(x_0, X, Y) &\equiv \exists y(y > x_0 \land y \in Y \land \forall z(x_0 < z < y \Rightarrow z \in X)), \\
\psi_{\mathsf{S}}(x_0, X, Y) &\equiv \exists y(y < x_0 \land y \in Y \land \forall z(x_0 > z > y \Rightarrow z \in X)).
\end{aligned}$$

Notice that all these truth tables have quantifier depth at most 2 and, except for ψ_{S}, they are all future formulae.

Remark 3.3 *We adopted a "strict" definition of the until modality, where the present is not taken into account. In practical applications, a "non-strict" definition is often preferred for the until modality* [3] *: the "non-strict until"* U_{ns} *modality has truth table*

$$\psi_{\mathsf{U}_{\text{ns}}}(x_0, X, Y) \equiv \exists y(y \geq x_0 \land y \in Y \land \forall z(x_0 \leq z < y \Rightarrow z \in X)).$$

Clearly, U_{ns} *can be defined using* U*:* $\mathsf{U}_{\text{ns}}(\phi_1, \phi_2) \equiv \phi_2 \lor \Big(\phi_1 \land \mathsf{U}(\phi_1, \phi_2)\Big)$. *The nice thing with the strict definition of* U *is that it allows to express* X *by* $\mathsf{X}\phi \equiv \mathsf{U}(\bot, \phi)$.

Definition 3.4 (First-order future modality) *A temporal modality* H *is a* first-order future modality *if its truth table is a future formula of FOMLO.*

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

The famous *PLTL* logic for linear time is $TL(\mathsf{U}_{\text{ns}}, \mathsf{X})$, or equivalently $TL(\mathsf{U})$, interpreted over linear orders (of ω-type) with monadic predicates.

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 3.5 (Path modality) *Given a first-order future formula* $\phi(x_0, X_1, \ldots, X_l)$*,* $\mathsf{E}\phi$ *is the* l*-place modality such, that for all trees* T *and node* n*,* $T, n \models \mathsf{E}\phi(X_1, \ldots, X_l)$ *if and only if there is a path* π *from* n *in* T *with* $T_\pi, n \models \phi(x_0, X_1, \ldots, X_l)$.

[3] Similarly, there exist non-strict F, G and S.

$\mathsf{E}\phi$ is said to be the *path modality* which corresponds to ϕ.

Note that if $\phi(x_0, X_1, \ldots, X_l)$ is a first-order future formula, the truth table of the path modality $\mathsf{E}\phi$ is the *MPL* formula $\exists Y. x_0 \in Y \wedge \phi'(x_0, X_1, \ldots, X_l)$ where ϕ' is obtained from $\phi(x_0, X_1, \ldots, X_l)$, by relativizing all its quantifiers to Y. Thus path modalities have *MPL* truth tables.

When H is a first-order future modality with truth-table ψ_{H}, we write EH for the path modality $\mathsf{E}\psi_{\mathsf{H}}$. Another modality is AH, defined by the equivalence

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

Example 3.6 *CTL is usually defined as TL*$(\mathsf{EU}_{\mathrm{ns}}, \mathsf{AU}_{\mathrm{ns}}, \mathsf{EX}, \mathsf{AX})$*, which is expressively equivalent to TL*$(\mathsf{EU}, \mathsf{AU})$.

In the following, we use some special modalities Z_1, Z_2, … Informally $\mathsf{Z}_l(\phi, \phi', \phi_1, \ldots, \phi_l)$ means that ϕ holds at the present state, ϕ' holds at a future state, all states in-between satisfy $\bigvee_{i=1}^{l} \phi_i$, and every ϕ_i is satisfied at least once. This is formalized by the following truth table:

$$\psi_{\mathsf{Z}_l}(x_0, X, Y, X_1, \ldots, X_l) \stackrel{\text{def}}{=} \exists y \begin{pmatrix} x_0 < y \;\wedge\; x_0 \in X \;\wedge\; y \in Y \\ \wedge \forall z \Big(x_0 < z < y \Rightarrow \bigvee_{i=1}^{l} z \in X_i \Big) \\ \wedge \bigwedge_{i=1}^{l} \exists z \Big(x_0 < z < y \wedge z \in X_i \Big) \end{pmatrix}.$$

Thus Z_l is a first-order future modality.

Observe that $\mathsf{EU}(\phi_1, \phi_2)$ can be expressed as $\mathsf{EZ}_1(\top, \phi_2, \phi_1)$. More generally, the EZ_ls can be seen as abbreviations for complicated EU modalities:

Proposition 3.7 *Any formula in TL*$(\{\mathsf{EZ}_l\}_{l=1,2,\ldots})$ *is equivalent to a TL*(EU) *formula.*

PROOF. We adapt the translation from *CTL*$^+$ into *CTL* that appears in [EH85]. The difficulty when translating $\mathsf{EZ}_l(\psi, \psi', \phi_1, \ldots, \phi_l)$ into *TL*(EU), is that we have to consider all the possible orderings of the witnesses for the "every ϕ_i is satisfied at least once" part. Write Λ for the set of all permutations of $\{1, \ldots, l\}$. Then $\mathsf{EZ}_l(\psi, \psi', \phi_1, \ldots, \phi_l)$ is equivalent to

$$\bigvee_{\lambda \in \Lambda} \begin{pmatrix} \psi \wedge \mathsf{EU}\Big(\bot,\, \phi_{\lambda(1)} \wedge \mathsf{EU}\Big(\phi_{\lambda(1)},\, \phi_{\lambda(2)} \wedge \mathsf{EU}\Big(\cdots, \\ \ldots \wedge \mathsf{EU}\Big(\bigvee_{i=1}^{l-1} \phi_{\lambda(i)},\, \phi_{\lambda(l)} \wedge \mathsf{EU}\Big(\bigvee_{i=1}^{l} \phi_{\lambda(i)},\, \psi'\Big)\Big) \cdots \Big)\Big)\Big) \end{pmatrix}.$$

□

Observe that a $TL(\{\mathsf{EZ}_l\}_{l=1,2,\dots})$ formula of size n is translated into an equivalent $TL(\mathsf{EU})$ formula of size $2^{n^{O(1)}}$.

3.3 ECTL+ and $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$

$ECTL^+$ was introduced in [EH86] [4]. Its importance comes from the fact that it extends CTL with a rich set of fairness properties.

Definition 3.8 *$ECTL^+$ is the temporal logic where we allow all path modalities $\mathsf{E}\phi$ s.t. $\phi(x_0, X_1, \dots, X_l)$ is a Boolean combination of the $\psi_{\mathsf{F}^\infty}(x_0, X_i)$'s and the $\psi_{\mathsf{U}}(x_0, X_i, X_j)$'s.*

For our purposes, we introduce a fragment of $ECTL^+$. This fragment is built on special modalities M_1, M_2, ... defined as follows: for any $l = 1, 2, \dots$, M_l is an l-place modality s.t.

$$\mathsf{M}_l(\phi_1, \dots, \phi_l) \;\equiv\; \mathsf{F}^\infty\phi_1 \wedge \cdots \wedge \mathsf{F}^\infty\phi_l \wedge \mathsf{G}(\phi_1 \vee \cdots \vee \phi_l)$$

Thus M_l is a (first-order future) modality for a kind of fairness constraint: $\mathsf{EM}_l(\phi_1, \dots, \phi_l)$ states that there is a path along which every ϕ_i is satisfied infinitely often and where only nodes satisfying some of the ϕ_is are encountered.

Observe that $\mathsf{EM}_1\phi$ is very close to $\mathsf{EG}\phi$: the difference is that $\mathsf{EM}_1\phi$ requires that there exists an *infinite* path along which $\mathsf{G}\phi$ holds. Thus

$$\mathsf{EM}_1\phi \;\equiv\; \mathsf{EG}\Big(\phi \wedge \mathsf{EX}\top\Big),$$

showing that CTL is at least as expressive as $TL(\mathsf{EU}, \mathsf{EM}_1)$. In the other direction, one can define AU in terms of EU and EM_1:

$$\mathsf{AU}(\phi_1, \phi_2) \;\equiv\; \mathsf{EX}\top \;\wedge\; \neg\mathsf{EM}_1\neg\phi_2 \;\wedge\; \neg\mathsf{EU}\Big(\neg\phi_2, \neg\phi_2 \;\wedge (\neg\phi_1 \;\vee\; \neg\mathsf{EX}\top)\Big).$$

Thus $TL(\mathsf{EU}, \mathsf{EM}_1)$, $TL(\mathsf{EU}, \mathsf{AU})$ and CTL are expressively equivalent.

Note that for $l' > l$, $\mathsf{EM}_l(\phi_1, \dots, \phi_l)$ is equivalent to $\mathsf{EM}_{l'}(\phi_1, \dots, \phi_l, \phi_l, \dots)$. Therefore $TL(\mathsf{EU}, \mathsf{EM}_l)$ is expressively equivalent to $TL(\mathsf{EU}, \mathsf{EM}_1, \dots, \mathsf{EM}_l)$.

[4] But it is very similar to the logic CTF used in [EC80].

Definition 3.9 [RM01] *For $k = 1, 2, \ldots$, BTL_k is the temporal logic defined as $TL(B_k)$, where*

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

Note that, while any BTL_k *modality* is defined by a formula of bounded quantifier depth, it is possible to nest these modalities in BTL_k *formulae*. Hence BTL_k is not defined as a bounded quantifier-depth fragment in the usual sense.

We write BTL for the union $BTL_1 \cup BTL_2 \cup \cdots$. A corollary of Kamp's theorem is that the well-known temporal logic CTL^* (from [EH86]) has exactly the same expressive power as BTL. We refer to [RM01] for more motivations and results on these temporal logics, including a proof that the sequence $\{BTL_k\}_{k=1,2,\ldots}$ contains an infinite hierarchy w.r.t. expressive power. Here we are interested in the links between BTL_2 and $ECTL^+$.

4 $ECTL^+$ and BTL_2 are expressively equivalent

In this section we investigate the expressive power of $ECTL^+$. Our main result is the following theorem, providing a characterization in terms of a natural fragment of the monadic logic of order.

Theorem 4.1 *BTL_2, $ECTL^+$ and $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ have the same expressive power.*

The proof of Theorem 4.1 has two main steps. First, we provide a new characterization of when paths satisfy the same first-order future formulae of quantifier depth 2 (sections 4.1 and 4.2). This allows translating BTL_2 formulae into equivalent $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ formulae (Corollary 4.9).

One completes the proof by observing that $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ is defined as a fragment of $ECTL^+$, and that $ECTL^+$ can be seen as a fragment of BTL_2 since the path modalities it uses have truth-tables of quantifier depth at most 2 (Definition 3.8 and Example 3.2).

A final section considers succinctness issues and shows that BTL_2 is exponentially more succinct than $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ or $ECTL^+$.

For the sake of brevity, linearly ordered sets with monadic predicates will be called *labeled chains* or just *chains*. Hence, if π is a path in some T, then T_π is the chain that corresponds to π.

Definition 4.2 ($\equiv_k$ equivalence) *Given two chains C and C', and nodes $n \in |C|$ and $n' \in |C'|$, we write $(C, n) \equiv_k (C', n')$ iff for any first-order future formula $\phi(x_0)$ with $\mathrm{qd}(\phi) \leq k$ we have $C, n \models \phi(x_0)$ iff $C', n' \models \phi(x_0)$.*

In other words, $(C, n) \equiv_k (C', n')$ when the two structures cannot be distinguished by *FOMLO* future formulae of quantifier depth at most k. Clearly the $\equiv_k$'s are equivalence relations.

The equivalences $\equiv_k$ can be characterized in terms of the following Ehrenfeucht-Fraïssé game. Consider two chains C and C', and two nodes $n \in |C|$ and $n' \in |C'|$. Below, n is called the *reference node* in C (and n' is the reference in C'). The game has k rounds and is played by two players, Spoiler and Duplicator. Spoiler plays first. He chooses, in one of the two chains, a node which is greater than or equal to the reference node, after which Duplicator responds by choosing a node in the other chain, greater than or equal to the reference node, which she believes "matches" the node chosen by Spoiler. The game continues for k rounds: at every round Spoiler chooses in one of the two chains a node which is greater than or equal to the reference node, and Duplicator responds by choosing a node in the other chain.

After k rounds the game is completed. For $i = 1, \ldots, k$, let s_i and s'_i be the nodes selected in the ith round in chain C (resp. C'). Duplicator is deemed the winner if the mapping $[s_1 \mapsto s'_1, \ldots, s_k \mapsto s'_k, n \mapsto n']$ respects the relations $\leq, \in P_1, \in P_2, \ldots$ Note that if $k = 0$, no moves are played and Duplicator wins iff the reference nodes n and n' have the same labeling.

We say that (C, n) and (C', n') are *k-game equivalent*, and we write $(C, n) \sim_k^g (C', n')$, when Duplicator has a strategy that ensures she wins any k-round game played on (C, n) and (C', n').

Since the game only involves nodes greater than or equal to the reference nodes, one clearly has $(C, n) \sim_k^g (C_{\geq n}, n)$ for any C and n.

The following is a variant of Ehrenfeucht's theorem [Ehr61]:

Theorem 4.3 [RM01] *Given two chains C and C', and elements $n \in |C|$*

and $n' \in |C'|$,

$$(C, n) \sim_k^g (C', n') \text{ iff } (C, n) \equiv_k (C', n').$$

4.2 A characterization of $\equiv_2$

From now on, we consider chains $C = (|C|, \leq, P_1, \ldots, P_m, n)$ with only m predicates and where the reference node is the first node. It is convenient to view such a chain as a linearly ordered set labeled by letters from the alphabet $A \stackrel{\text{def}}{=} 2^{\{1,\ldots,m\}}$, i.e., a node $s \in |C|$ carries a letter $a_s \in A$ that tells for $i = 1, \ldots, m$, whether P_i labels s. Formally $a_s \stackrel{\text{def}}{=} \{i \mid s \in P_i\}$.

Additionally, if C has order type at most ω, we call it a *path*, since paths in computation trees give rise to such chains.

Assume $\Sigma, \Sigma' \subseteq A$ are two sub-alphabets, and $a \in A$ is a letter. We say that the triple $\tau = (\Sigma, a, \Sigma')$ is *realized at node s in chain C* if $a = a_s$, $\Sigma = \{a_t \mid t < s\}$ and $\Sigma' = \{a_t \mid t > s\}$ or, in other words, when a is the label of s and Σ (resp. Σ') is the set of letters that occur before s (resp. after s) in the chain. We say that a triple *occurs in* C if it is realized at some s in C.

Since A is finite, there is only a finite number of possible triples. We let $\tau(C)$ denote the set of all triples occurring in C, and call it the $\boldsymbol{\tau}$*-type* of C. The importance of $\boldsymbol{\tau}$-types comes from the following result.

Lemma 4.4 *$C \sim_2^g C'$ iff $\tau(C) = \tau(C')$.*

PROOF. ($\Rightarrow$:) We prove that $\tau(C) \neq \tau(C')$ implies $C \not\sim_2^g C'$. Assume, w.l.o.g., that $\tau(C)$ contains a triple $\tau = (\Sigma, a, \Sigma')$ that is not in $\tau(C')$. Then SPOILER has a winning strategy for 2-round games: he picks a node $s \in C$ that realizes τ. When DUPLICATOR answers and picks a $s' \in C'$, s' realizes some $\tau' = (\Sigma_2, a_2, \Sigma_2')$. Now $\tau \neq \tau'$ and there are several cases: if $a \neq a_2$ then SPOILER wins. If $\Sigma \neq \Sigma_2$, then there must exist a node on the left of s or s' carrying a letter that does not appear on the same side of the other node: SPOILER picks it and wins. Finally, if $\Sigma' \neq \Sigma_2'$, the same reasoning applies with a letter this time on the right of s or s'.

($\Leftarrow$:) We assume $\tau(C) = \tau(C')$ and show that DUPLICATOR has a winning strategy for 2-round games. Let SPOILER pick some s_1 in C or C'. The node s_1 realizes some triple $\tau = (\Sigma_1, a_1, \Sigma_1')$ and DUPLICATOR answers by picking in the other chain a node s_1' that also realizes τ. Such a node must exist because $\tau(C) = \tau(C')$. (Observe that if s_1 is the initial node of its chain, then DUPLICATOR must pick the initial node of the other chain since the initial nodes are the only nodes that realize a triple with empty Σ.)

When SPOILER picks a second node s_2, its label is in Σ_1 or Σ'_1 depending on whether s_2 lies to the left or the right of s_1 or s'_1. Then DUPLICATOR can pick in the other chain an s'_2 with the same label and on the same side of s_1 or s'_1. Additionally, if s_2 is the initial node, and only then, DUPLICATOR picks the initial node in the other chain. Finally the game is won by DUPLICATOR. □

Now let C be a path (i.e., a chain of order type ω or less). We say a node s of C is *limiting* if it is the first or the last occurrence (in C) of the letter a_s it carries. We consider the limiting nodes in the order they occur in C: they are $s_1 < s_2 < \cdots < s_p$. Note that s_1 is the initial node, and that p is at most twice the number of letters in A. For example, if C is the infinite word $abbabda(cb)^\omega$, then underlying its limiting nodes gives $\underline{a}\underline{b}bab\underline{d}\underline{a}\underline{c}b(cb)^\omega$.

With C we associate the sequence $\rho(C)$, of the form $a_1, \Sigma_1, a_2, \Sigma_2, \ldots, a_p, \Sigma_p$, where every a_i is the letter carried by s_i, the ith limiting node, and every Σ_i is the set of letters that occur at least once between s_i and s_{i+1} (Σ_p is the set of letters that occur after s_p, which must each occur infinitely often). Continuing our previous example, the path C seen above is associated with

$$\rho(C) \;=\; a,\ \{\},\ b,\ \{a,b\},\ d,\ \{\},\ a,\ \{\},\ c,\ \{b,c\}.$$

Note that $\rho(C)$ is entirely determined by C: we call it the $\boldsymbol{\rho}$*-type* of C.

Lemma 4.5 *The $\boldsymbol{\tau}$-type of a path can be computed from its $\boldsymbol{\rho}$-type.*

PROOF. Assume $\rho(C)$ is $a_1, \Sigma_1, \ldots, a_p, \Sigma_p$. Then for $i = 1, \ldots, p$, there is a triple τ_i realized by s_i, and for every $a \in \Sigma_i$ there is a triple τ_i^a realized by the non-limiting nodes:

$$\tau_i = \Big(\{a_j \mid j < i\},\ a_i,\ \{a_j \mid j > i\} \cup \bigcup_{j \geq i} \Sigma_j\Big),$$

$$\tau_i^a = \Big(\{a_j \mid j \leq i\},\ a,\ \{a_j \mid j > i\} \cup \bigcup_{j \geq i} \Sigma_j\Big).$$

Finally, $\tau(C)$ contains no other triples. □

In the other direction, $\tau(C)$ contains enough information to reconstruct $\rho(C)$, but explaining this requires some notations. We say a triple (Σ, a, Σ') is *limiting* if $a \notin \Sigma \cap \Sigma'$: a node s in C is limiting iff it realizes a limiting triple.

For two triples $\tau_1 = (\Sigma_1, a_1, \Sigma'_1)$ and $\tau_2 = (\Sigma_2, a_2, \Sigma'_2)$, we write $\tau_1 \sqsubseteq \tau_2$ when $\Sigma_1 \subseteq \Sigma_2$ and $\Sigma'_1 \supseteq \Sigma'_2$: observe that $\sqsubseteq$ is only a quasi-ordering in general (since we may have $a_1 \neq a_2$ while $\tau_1 \sqsubseteq \tau_2 \sqsubseteq \tau_1$).

If now s_1 and s_2 are two nodes of C that realize τ_1 and τ_2 respectively, then $s_1 \leq s_2$ implies $\tau_1 \sqsubseteq \tau_2$.

Lemma 4.6 *The $\boldsymbol{\rho}$-type of a path can be computed from its $\boldsymbol{\tau}$-type.*

PROOF. (Idea) Assume $\tau(C)$ is known. The limiting triples in $\tau(C)$ are linearly ordered by $\sqsubseteq$, so that we get a sequence $\tau_1 \sqsubseteq \tau_2 \sqsubseteq \cdots \sqsubseteq \tau_p$. W.r.t. $\sqsubseteq$, a non-limiting triple in $\tau(C)$ falls between two consecutive limiting triples (or to the right of τ_p). We obtain a list of the following general form

$$\tau_1,\ \{\tau_1^1, \ldots, \tau_1^{n_1}\},\ \tau_2,\ \{\tau_2^1, \ldots, \tau_2^{n_2}\},\ \ldots,\ \tau_p,\ \{\tau_p^1, \ldots, \tau_p^{n_p}\}.$$

Given such a list, one obtains $\rho(C)$ by replacing every triple (Σ, a, Σ') by the letter a it witnesses. $\square$

Summing up Theorem 4.3 and Lemmas 4.4, 4.5, 4.6 we get

Corollary 4.7 *For any two paths C and C', $C \equiv_2 C'$ iff $C \sim_2^g C'$ iff $\tau(C) = \tau(C')$ iff $\rho(C) = \rho(C')$.*

4.3 From BTL$_2$ to TL($\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots}$)

The nice thing with $\boldsymbol{\rho}$-types is that having a path with a given $\boldsymbol{\rho}$-type can be written in *TL*($\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots}$):

Lemma 4.8 *For any $\boldsymbol{\rho}$-type ρ, there exists a formula ψ_ρ in TL($\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots}$) s.t. for any tree $T = (|T|, \leq, P_1, \ldots, P_m)$ and node n of T, $T, n \models \psi_\rho$ iff there exists a path π in T starting from n such that $\rho(T_\pi) = \rho$. Furthermore, ψ_ρ has size $2^{|\rho|^{O(1)}}$.*

PROOF. For ρ having the form $a_1, \Sigma_1, \ldots, a_p, \Sigma_p$, we express what it means to have $\boldsymbol{\rho}$-type ρ with

$$\mathsf{EZ}(a_1, \mathsf{EZ}(a_2, \ldots \mathsf{EZ}(a_p, \mathsf{EM}(\Sigma_p)) \ldots, \Sigma_2), \Sigma_1) \qquad (\theta_\rho)$$

where, for $\Sigma = \{a_1, \ldots, a_l\}$, $\mathsf{EZ}(a, b, \Sigma)$ and $\mathsf{EM}(\Sigma)$ are short for, respectively, $\mathsf{EZ}_l(a, b, a_1, \ldots, a_l)$ and $\mathsf{EM}_l(a_1, \ldots, a_l)$.

Now Proposition 3.7 entails that θ_ρ can be expressed by some ψ_ρ in $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$. Since θ_ρ has size $O(|\rho|)$, we end up with $|\psi_\rho|$ in $2^{|\rho|^{O(1)}}$. $\square$

Corollary 4.9 *Every BTL_2 modality can be expressed in $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$.*

PROOF. Let $\mathsf{E}\phi$ be a BTL_2 path modality, induced by some first-order future formula $\phi(x_0, X_1, \dots, X_l)$, and let $\rho(\phi)$ be the set $\{\rho(C) \mid C \models \phi\}$. Since there are only a finite number of possible $\boldsymbol{\rho}$-types for a given set of letters, $\rho(\phi)$ is finite and, by Lemma 4.8, there exists a $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$ formula ψ (e.g., $\psi \stackrel{\text{def}}{=} \bigvee_{\rho\in\rho(\phi)} \psi_\rho$) such that $T, n \models \psi$ iff T has a path starting from n with $\boldsymbol{\rho}$-type in $\rho(\phi)$. Now if ϕ has quantifier depth 2, a path having $\boldsymbol{\rho}$-type in $\rho(\phi)$ satisfies ϕ (by Corollary 4.7). Hence $\psi \equiv \mathsf{E}\phi(q_1, \dots, q_l)$. $\square$

Hence BTL_2 is not more expressive than $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$.

4.4 The succinctness of BTL_2

Here we investigate succinctness issues for the translations that underly our proof that BTL_2, $ECTL^+$ and $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$ are expressively equivalent.

We start with upper bounds. Let $\phi(x_0, X_1, \dots X_m)$ be a first-order future formula. The corresponding alphabet Σ has size $|\Sigma| = n = 2^m$ so that the number of $\boldsymbol{\rho}$-types over Σ is bounded by $r = (2n)! \times 2^{n(2n+1)}$ which is $2^{n^{O(1)}}$. In Corollary 4.9 we constructed a $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\dots})$ formula ψ which is equivalent to the BTL_2 path modality $\mathsf{E}\phi$. The size of ψ is bounded by 2^r. Hence, when translating from BTL_2 to $ECTL^+$, an upper bound on the size of resulting formulae is $2^{2^{2^{O(|\phi|)}}}$.

Regarding lower bounds, BTL_2 can be exponentially more succinct than $ECTL^+$. Indeed, consider the following first-order future formula:

$$\phi_n(x_0, X_1, \dots, X_n, Y) \stackrel{\text{def}}{=} \forall y, y' > x_0 \ \Big(\bigwedge_{i=1}^{n} y \in X_i \Leftrightarrow y' \in X_i\Big) \Rightarrow (y \in Y \Leftrightarrow y' \in Y)$$

stating that all future states that agree on $X_1, \dots, X_n$ agree on Y as well. It has quantifier depth 2. The BTL_2 formula $\mathsf{E}\phi_n(q_1, \dots, q_n, q_0)$ can be expressed by the following $ECTL^+$ formula

$$\psi \stackrel{\text{def}}{=} \mathsf{E} \bigwedge_{v\subseteq\{0,1,\ldots,n\}} \mathsf{G}\Big(\big[\bigwedge_{i=1}^{n} q_i \Leftrightarrow (i \in v)\big] \Rightarrow \big[q_0 \Leftrightarrow (0 \in v)\big]\Big)$$

where all possible valuations for the atomic propositions have been accounted for by the outermost conjunction. (The "$i \in v$" subformulae in ψ stand for the Boolean constants $\top$ or $\bot$, depending on i and v.)

ψ has exponential size but this is essentially the best possible: Etessami *et al.* prove that the $TL(\mathsf{U},\mathsf{S})$ formulae that are equivalent to ϕ_n over chains have size $2^{\Omega(n)}$ [EVW02]. Since removing the path quantifiers in an $ECTL^+$ formula yields a linear-sized $TL(\mathsf{U})$ formula that is equivalent over chains, the smallest $ECTL^+$ formulae equivalent to $\mathsf{E}\phi_n$ must have size $2^{\Omega(n)}$.

There also exists an exponential succinctness gap between $ECTL^+$ and $TL(\mathsf{EU},\{\mathsf{EM}_l\}_{l=1,2,\ldots})$: the $ECTL^+$ formulae $\psi_n \stackrel{\text{def}}{=} \mathsf{E}(\mathsf{F}q_1 \wedge \cdots \wedge \mathsf{F}q_n)$ can be expressed by $TL(\mathsf{EU},\{\mathsf{EM}_l\}_{l=1,2,\ldots})$ formulae of size $O(n!)$ (along the lines of the proof of Proposition 3.7). Wilke [Wil99] (see also [AI01]) proved that CTL formulae expressing ψ_n have size $2^{\Omega(n)}$ and his proof applies even if one considers "equivalence over finite trees" as the equivalence criterion. Assume a $TL(\mathsf{EU},\{\mathsf{EM}_l\}_{l=1,2,\ldots})$ formula ϕ is equivalent to ψ_n. ϕ can be transformed into a shorter CTL formula ϕ' that is equivalent over finite trees: one simply replaces any $\mathsf{EM}_l(\phi_1,\ldots,\phi_l)$ by $\bot$. We deduce that ϕ', and therefore ϕ, must have size in $2^{\Omega(n)}$.

We do not know whether these last two results add up to a doubly-exponential succinctness gap between BTL_2 and $TL(\mathsf{EU},\{\mathsf{EM}_l\}_{l=1,2,\ldots})$, nor how one can reduce the gap between these lower bounds and the triply exponential upper bound.

5 No finite bases for BTL_2 and $ECTL^+$

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

Example 5.1 (Some temporal logics with a finite basis)
– *CTL is defined as $TL(\mathsf{EU}_{\mathrm{ns}},\mathsf{AU}_{\mathrm{ns}},\mathsf{EX})$, and is expressively equivalent to $TL(\mathsf{EU},\mathsf{AU})$. Hence it has a finite basis.*

– *BTL_1 is expressively equivalent to $TL(\mathsf{EY})$, where $\mathsf{Y}(\phi_1,\phi_2) \equiv (\mathsf{F}\phi_1 \wedge$*

$\mathsf{G}\phi_2$) [RM01]. *Hence it has a finite basis.*

– *ECTL is defined as* $TL(\mathsf{EU}_{\mathrm{ns}}, \mathsf{AU}_{\mathrm{ns}}, \mathsf{EX}, \mathsf{EF}^{\infty})$ *and hence has a finite basis.*

Finding bases answers questions about which temporal modalities are essential and which are just convenient abbreviations. For temporal logics like CTL^* that are defined via an infinite set of modalities, finding a finite basis is a way of providing a simpler definition.

A major result from [RM01] is that BTL, and thus CTL^*, do not admit a finite basis. The same article also conjectures that no BTL_k logic for $k > 1$ admits a finite basis. In the rest of this section, we partially prove this conjecture by showing that BTL_2, and thus $ECTL^+$, do not admit a finite basis.

5.1 An infinite hierarchy inside $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$

We already mentioned that $TL(\mathsf{EU}, \mathsf{EM}_1)$ is expressively equivalent to CTL. The fact that $\mathsf{E}(\mathsf{G}\phi \wedge \mathsf{F}^{\infty}\psi)$ cannot be expressed in $ECTL$ [Lar94, p. 34] shows that $TL(\mathsf{EU}, \mathsf{EM}_2)$ is already strictly more expressive than $ECTL$.

In this subsection we prove that, for any n, $\mathsf{EM}_n(q_1, \ldots, q_n)$ cannot be expressed with only EU and EM_{n-1}, so that $TL(\mathsf{EU}, \mathsf{EM}_n)$ is strictly more expressive than $TL(\mathsf{EU}, \mathsf{EM}_{n-1})$.

Let P be a family $\{q_1, \ldots, q_n\}$ of $n \geq 2$ atomic propositions, and let $S = \{P_0, \ldots, P_n\}$ be the set of all subsets of P with at least $n-1$ elements, defined by $P_0 \stackrel{\text{def}}{=} P$ and, for $i > 0$, $P_i \stackrel{\text{def}}{=} \{q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_n\}$.

We now define a Kripke structure $\mathcal{M}$: the nodes in $|\mathcal{M}|$ are all $\langle q, \Sigma, m\rangle$ with $\Sigma \in S$, $q \in \Sigma$ and $m \in \mathbb{N}$. In $\mathcal{M}$, every node $\langle q, \Sigma, m\rangle$ is labeled with q, called the *visible value* of the node (Σ is the *support*, m is the *level*).

The transitions in $\mathcal{M}$ are all $\langle q, \Sigma, m\rangle \to \langle q', \Sigma', m'\rangle$ s.t. (1) $\Sigma = \Sigma'$ and $m = m'$, or (2) $m' = m - 1$ and $\Sigma' \neq P_0$. Transitions of type (1) create cliques where Σ and m do not change. Inside a (Σ, m)-clique, each of the $n-1$ nodes (or n if $\Sigma = P_0 = P$) carries a different visible value from Σ.

Transitions of type (2) connect the cliques as illustrated by Figure 1: from level $m > 0$ one can move to any clique at level $m - 1$ except $(P_0, m - 1)$. Hence the cliques are also strongly connected components.

Observe that the (P_0, m)-cliques are the only ones that carry all n different propositions from P, and the only ones that cannot be reached from any other

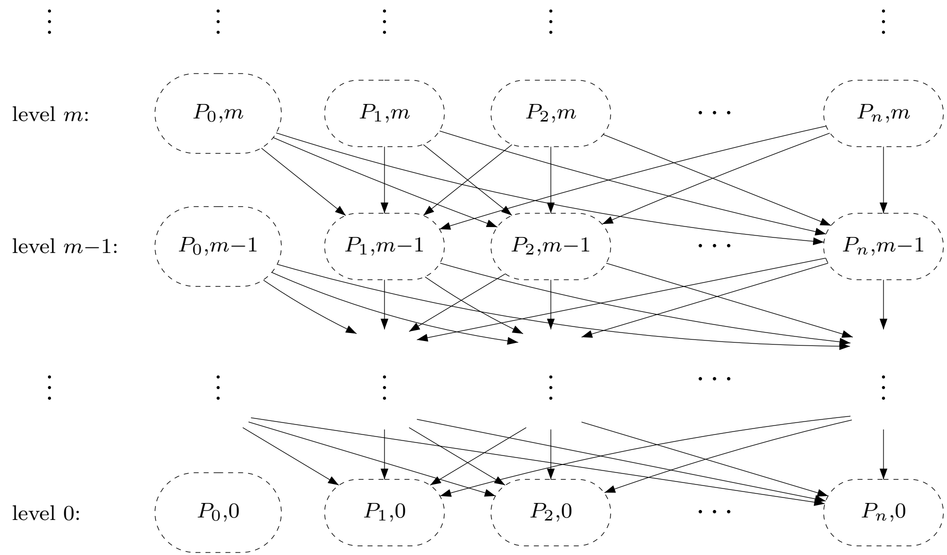

Fig. 1. The transitions between cliques in $\mathcal{M}$

clique. Hence we have:

Fact 5.2 $\langle q, \Sigma, m\rangle \models \mathsf{EM}_n(q_1, \ldots, q_n)$ *iff* $\Sigma = P_0 = P$.

In the following we study how $TL(\mathsf{EU}, \mathsf{EM}_{n-1})$ formulae are satisfied in $\mathcal{M}$ in order to prove that they cannot express $\mathsf{EM}_n(q_1, \ldots, q_n)$.

The next lemma states that whether $\langle q, \Sigma, m\rangle$ satisfies $\phi \in TL(\mathsf{EU}, \mathsf{EM}_{l-1})$ does not depend on Σ, m if m is greater than or equal to $\mathrm{nd}(\phi)$, the nesting depth of ϕ:

Lemma 5.3 *Let ϕ be a $TL(\mathsf{EU}, \mathsf{EM}_{n-1})$ formula. For all $k \geq \mathrm{nd}(\phi)$, for all $\Sigma, \Sigma' \in S$, for all $q \in \Sigma \cap \Sigma'$, we have*

$$\langle q, \Sigma, k\rangle \models \phi \;\; \textit{iff} \;\; \langle q, \Sigma', k+1\rangle \models \phi. \qquad (*)$$

PROOF. First observe that if Lemma 5.3 holds for a given ϕ, then for all $k, k' \geq \mathrm{nd}(\phi)$, for all $\Sigma, \Sigma' \in S$, for all $q \in \Sigma \cap \Sigma'$, $\langle q, \Sigma, k\rangle \models \phi$ iff $\langle q, \Sigma', k'\rangle \models \phi$.

We write s_0 for $\langle q, \Sigma, k\rangle$, s'_0 for $\langle q, \Sigma', k+1\rangle$, and prove $(*)$ by induction on the structure of ϕ. The cases where ϕ is an atomic proposition, or a Boolean combination of subformulae are obvious and there remain two cases.

1: ϕ is $\mathsf{EU}(\phi_1, \phi_2)$:

(⇒:) If $s_0 \models \phi$ then there is a path $\pi = s_0, s_1, \ldots$ and an $r \geq 1$ s.t. $s_r \models \phi_2$, and $s_i \models \phi_1$ for $0 < i < r$. We write $\langle q^i, \Sigma^i, m^i \rangle$ for s_i.

1a: If $m^r \geq k-1$ then, by ind. hyp., $\langle q^r, \Sigma'', k \rangle \models \phi_2$ for any Σ'' containing q^r. Pick a Σ'' different from P_0 and there is a transition $s'_0 \to \langle q^r, \Sigma'', k \rangle$, proving $s'_0 \models \phi$.

1b: If $m^r < k-1$ then $r > 1$ and $m^i = k-1$ for some $0 < i < r$. $s_i \models \phi_1$ and, by ind. hyp., $\langle q^i, \Sigma^i, k \rangle \models \phi_1$. Since $\Sigma^i \neq P_0$, we can construct a path $\pi' = s'_0, \langle q^i, \Sigma^i, k \rangle, s_i, s_{i+1}, \ldots$ proving $s'_0 \models \phi$.

($\Leftarrow$:) If $s'_0 \models \phi$ then there is a path $\pi' = s'_0, s'_1, \ldots$ and an $r \geq 1$ s.t. $s'_r \models \phi_2$, and $s'_i \models \phi_1$ for $0 < i < r$. We write $\langle q^i, \Sigma^i, m^i \rangle$ for s'_i.

1c: If $m^r \geq k$ then, by ind. hyp., $\langle q^r, \Sigma'', k-1 \rangle \models \phi_2$ for any Σ'' containing q^r. If we pick $\Sigma'' \neq P_0$, we have a transition $s_0 \to \langle q^r, \Sigma'', k-1 \rangle$ proving $s_0 \models \phi$.

1d: If $m^r \leq k-1$ then $m^i = k-1$ for some $0 < i \leq r$ and $s_0, s'_i, s'_{i+1}, \ldots$ is a path proving $s_0 \models \phi$.

2: ϕ is $\mathsf{EM}_{n-1}(\phi_1, \ldots, \phi_{n-1})$:

($\Rightarrow$:) If $s_0 \models \phi$ then there is an infinite path $\pi = s_0, s_1, \ldots$ witnessing $s_0 \models \phi$. We write $\langle q^i, \Sigma^i, m^i \rangle$ for s_i.

2a: If $m^r = k-1$ for some r, then $\pi' = s'_0, \langle q^r, \Sigma^r, k \rangle, s_r, s_{r+1}, \ldots$ is a path proving $s'_0 \models \phi$ since, by ind. hyp., $\langle q^r, \Sigma^r, k \rangle \models \bigvee_i \phi_i$.

2b: Otherwise $m^r = k$ for all r and π stays inside one clique. Let $r_1, \ldots, r_{n-1}$ be indexes s.t. $s_{r_i} \models \phi_i$ (and $r_i > 0$). Let $\Sigma'' \in S$ be some support containing all q^{r_i}'s. We can pick $\Sigma'' \neq P_0$ since there are at most $n-1$ values to accommodate. Defining $s''_i = \langle q^{r_i}, \Sigma'', k \rangle$, we have $s''_i \models \phi_i$ (ind. hyp.) so that $s'_0, s''_1, s''_2, \ldots, s''_{n-1}, s''_1, \ldots$ is a path proving $s'_0 \models \phi$.

($\Leftarrow$:) If $s'_0 \models \phi$ then there is an infinite path $\pi' = s'_0, s'_1, \ldots$ witnessing $s'_0 \models \phi$. We write $\langle q^i, \Sigma^i, m^i \rangle$ for s'_i.

If $m^r = k-1$ for some r, then $s_0, s'_r, s'_{r+1}, \ldots$ is a path proving $s_0 \models \phi$.

Otherwise $m^r \geq k$ for all r and we proceed as in case **2b**. With $s_i \stackrel{\text{def}}{=} \langle q^{r_i}, \Sigma'', k-1 \rangle$, we build a path $s_0, s_1, \ldots, s_{n-1}, s_1, \ldots$ proving $s_0 \models \phi$. $\square$

Lemma 5.4 $\mathsf{EM}_n(q_1, \ldots, q_n)$ *cannot be expressed in* $TL(\mathsf{EU}, \mathsf{EM}_{n-1})$.

PROOF. Assume $\mathsf{EM}_n(q_1, \ldots, q_n)$ is equivalent to some $\phi \in TL(\mathsf{EU}, \mathsf{EM}_{n-1})$ and let $k \geq \mathrm{nd}(\phi)$. Then, for any $\Sigma \in S$ and for all $q \in \Sigma$, $\langle q, \Sigma, k \rangle \models \phi$ iff $\langle q, \Sigma_0, k \rangle \models \phi$ (Lemma 5.3), contradicting Fact 5.2. $\square$

This can be seen as a generalization of the result (from [EH86]) that $\mathsf{E}(\mathsf{F}^\infty q_1 \wedge \mathsf{F}^\infty q_2)$ cannot be expressed in *ECTL*. Our Kripke structure shows that $\mathsf{E}(\mathsf{F}^\infty q_1 \wedge \cdots \wedge \mathsf{F}^\infty q_n)$ cannot be expressed in a fragment of $ECTL^+$ where only $n-1$-ary conjunctions of F^∞ modalities are allowed under an existential path quantifier.

5.2 BTL_2 and $ECTL^+$ have no finite basis

A corollary of Lemma 5.4 is:

Corollary 5.5 *With regards to their expressive power, the logics $TL(\mathsf{EU}, \mathsf{EM}_1)$, $TL(\mathsf{EU}, \mathsf{EM}_2)$, ..., $TL(\mathsf{EU}, \mathsf{EM}_n)$, ... form an infinite hierarchy inside $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$.*

We can now conclude with the following result.

Theorem 5.6 *BTL_2, $ECTL^+$, and $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ have no finite basis.*

PROOF. Assume $\mathsf{H}_1, \ldots, \mathsf{H}_k$ are $ECTL^+$ (or, equivalently, BTL_2) modalities. Then every H_i can be defined as some $TL(\mathsf{EU}, \mathsf{EM}_{n_i})$ formula (Theorem 4.1) so that $TL(\mathsf{H}_1, \ldots, \mathsf{H}_k)$ is not more expressive than $TL(\mathsf{EU}, \mathsf{EM}_{\max(n_i)})$. Thus, by Corollary 5.5, $TL(\mathsf{H}_1, \ldots, \mathsf{H}_k)$ is strictly less expressive than $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ and, by Theorem 4.1, than BTL_2 and $ECTL^+$. □

6 Model checking

In this section, we study the model-checking problem for BTL_2 and $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$.

Recall that the *model-checking problem* for a temporal logic L is as follows: Given a finite Kripke structure $\mathcal{M}$, a node s of $\mathcal{M}$, and a formula $\phi \in L$, determine whether $T_{\mathcal{M},s}, s \models \phi$, where $T_{\mathcal{M},s}$ is the tree obtained by unfolding $\mathcal{M}$ from its node s (see section 2.5).

While it is well known that model checking is **P**-complete for *CTL* and **PSPACE**-complete for CTL^*, the precise complexity of model checking $ECTL^+$ has only been recently characterized.

Theorem 6.1 [LMS01] *The model-checking problem for $ECTL^+$ is $\mathbf{\Delta_2^P}$-complete.*

Here $\mathbf{\Delta_2^p}$, from the polynomial-time hierarchy, is the class of decision problems for which there is an algorithm in $\mathbf{P^{NP}}$. It lies "between" **NP** $\cup$ **coNP** and **PSPACE** [Sto76,Pap94].

Considering the model-checking problem for BTL_2 allows to further compare $ECTL^+$ and BTL_2. Indeed, $ECTL^+$ and BTL_2 have the same expressive power but BTL_2 can be (at least) exponentially more succinct than $ECTL^+$. Hence model checking could well be thought to be harder for BTL_2 than for $ECTL^+$. Recall that, in the case of CTL^+ and CTL, the succinctness gap translates into a complexity gap for model checking and satisfiability [LMS01,JL03].

6.1 Periodic paths and BTL_2 modalities

Throughout this section we consider a given finite Kripke structure $\mathcal{M} = \langle |\mathcal{M}|, R, P_1, \ldots \rangle$ and write n for the number of nodes in $\mathcal{M}$.

A path $\pi = s_0, s_1, \ldots$ in $\mathcal{M}$ is *ultimately periodic* (or succinctly *periodic*) if there are some k and k' s.t. $s_{i+k'} = s_i$ for every $i \geq k$ (assuming $s_{i+k'}$ exists, hence finite paths are periodic). Thus a periodic path consists of a finite prefix followed by a repeated loop (if the path is infinite). We define $|\pi|$, the *size of* π, as $k + k'$ since, computationally, π can be described by a sequence of $k + k'$ nodes.

(Small) periodic paths are what we are looking for when model checking BTL_2 path modalities:

Lemma 6.2 (Small witnesses for BTL_2) *Let $\mathsf{E}\phi$ be a BTL_2 path modality with arity l. If there exists in $\mathcal{M}$ a path π starting from s_0 s.t. $T_\pi, s_0 \models \phi(x_0, P_1, \ldots, P_l)$, then there exists such a path that is periodic, and has size $O(n^3)$.*

PROOF. Assume π is $s_0, s_1, \ldots$ and let $\rho = a_1, \Sigma_1, \ldots, a_p, \Sigma_p$ be its $\boldsymbol{\rho}$-type. Since $\mathcal{M}$ has n states, only n different letters can appear in ρ, and thus $p \leq 2n$.

We build a periodic path π' out of π by keeping s_0, all s_i's that are limiting occurrences in π, and for each letter $b \in \Sigma_i$ one state witnessing that b occurs at least once between the corresponding limiting occurrences. Between these selected states, we keep additional states from π ensuring the connectivity of the sequence (and ensuring a final loop visiting the witnesses from Σ_p). The result is a periodic path π' with the same $\boldsymbol{\rho}$-type as π, hence $T_{\pi'}, s_0 \models \phi(x_0, P_1, \ldots, P_l)$ by Corollary 4.7. Because we only selected $O(n^2)$ states and because at most $n-1$ states are needed to ensure the connectivity between any two states along π, the path π' has size $O(n^3)$. $\square$

Model checking periodic paths is easy:

Lemma 6.3 (Model checking over periodic paths) *Given a periodic path π starting from s_0 in $\mathcal{M}$, and a first-order future formula $\phi(x_0, X_1, \ldots, X_l)$ with $\mathrm{qd}(\phi) \leq 2$, checking whether $T_\pi, s_0 \models \phi(x_0, P_1, \ldots, P_l)$ can be done in deterministic time $O(|\pi|^2 \times |\phi|)$.*

PROOF. Assume $\pi = s_0, s_1, \ldots$ is such that $s_{i+k'} = s_i$ for $i \geq k$ and let $m : \mathbb{N} \to \{0, 1, \ldots, k+k'-1\}$ project every position $i \in \mathbb{N}$ to its representative: we have $m(i) = i$ if $i < k + k'$ and $m(i) = m(i - k')$ otherwise (we assume $k > 0$ so that $m(i) = 0$ iff $i = 0$).

For every subformula $\psi(x_0, x, y, X_1, \ldots, X_l)$ of quantifier depth 0 that occurs inside ϕ, we build a table $\mathtt{T}^\psi$ that says, given i and j, whether $T_\pi, s_0, s_i, s_j \models \psi(x_0, x, y, P_1, \ldots, P_l)$. Observe that ψ is a Boolean combination of atoms of the form $z \in X$ or $z < z'$ so that knowing $m(i)$, $m(j)$ and the position of j relative to i (j can be *before*, *at*, or *after* i) is enough to say whether $T_\pi, s_0, s_i, s_j \models \psi(x_0, x, y, P_1, \ldots, P_l)$. Therefore it is enough to build tables $\mathtt{T}^\psi$'s with (less than) $3 \times (k + k')^2$ entries and all these tables can be filled in time $O(|\pi|^2 \times |\phi|)$.

Then, for every subformula $\psi'(x_0, x, X_1, \ldots, X_l)$ of quantifier depth 1 that occurs inside ϕ, we build a table $\mathtt{T}^{\psi'}$ that says, given i, whether $T_\pi, s_0, s_i \models \psi'(x_0, x, P_1, \ldots, P_l)$. This only depends on $m(i)$ and the position of i relative to $k + k'$. To see this, imagine that ψ' is $\exists y\ \psi$: knowing $m(i)$ and the position of i relative to $k + k'$ allows to enumerate all $m(j)$ for j before i, and all $m(j)$ for j after i. The table $\mathtt{T}^\psi$ is then used to check if $T_\pi, s_0, s_i, s_j \models \psi(x_0, x, y, P_1, \ldots, P_l)$ for one of these cases (the case $i = j$ must be also be considered), that is to check whether $T_\pi, s_0, s_i \models \psi'(x_0, x, P_1, \ldots, P_l)$. Therefore, the tables for the $\mathtt{T}^{\psi'}$'s only need to have $k + 2k'$ entries and they can be filled in time $O(|\pi|^2 \times |\phi|)$.

Finally, once the $\mathtt{T}^{\psi'}$'s tables are built, evaluating whether $T_\pi, s_0 \models \phi(x_0, X_1, \ldots, X_l)$ can be done with additional time $O(|\pi| \times |\phi|)$. $\square$

Remark 6.4 *More generally, model checking periodic paths with an arbitrary FOMLO formula ϕ can be done in deterministic time $O(|\pi|^{\mathrm{qd}(\phi)} \times |\phi|)$, and is* **PSPACE**-*complete [MS03].*

Proposition 6.5 *The problem of deciding, for a finite Kripke structure $\mathcal{M}$, a node $s_0 \in |\mathcal{M}|$, and a BTL_2 path modality $\mathsf{E}\phi$, whether $s_0 \models \mathsf{E}\phi(q_1, \ldots, q_l)$ is* **NP***-complete.*

PROOF. Membership in **NP** is shown by the following non-deterministic algorithm: guess a periodic path π of size $O(n^3)$ and check $\pi \models \phi(q_1, \ldots, q_l)$ in polynomial time (Lemma 6.3). This algorithm is correct by Lemma 6.2.

NP-hardness is well-known and already appears with BTL_1 modalities, e.g., with formulae of the form $\mathsf{E} \bigwedge_i (\bigvee_j \mathsf{F} q_{n_{i,j}})$ [SC85,DS02]. □

The important corollary is

Theorem 6.6 *The model-checking problem for BTL_2 is $\mathbf{\Delta_2^p}$-complete.*

PROOF. Since $ECTL^+$ can be seen as a fragment of BTL_2, $\mathbf{\Delta_2^p}$-hardness follows from Theorem 6.1.

Membership in $\mathbf{\Delta_2^p}$ is a corollary of Proposition 6.5: given a Kripke structure $\mathcal{M}$ with n nodes and a BTL_2 formula ϕ with m path quantifiers, a model-checking algorithm along the lines of [Eme90, Theorem 6.26] will compute, for each node n in $\mathcal{M}$ and each subformula ψ of ϕ, whether $\mathcal{M}, n \models \psi$. By considering subformulae in order of increasing size, the algorithm only needs nm invocations of an **NP**-oracle for BTL_2 path modalities and then belongs to $\mathbf{P}^{\mathbf{NP}}$. □

6.3 Model checking $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$

Theorem 6.7 *The model-checking problem for $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ is* **P***-complete.*

PROOF. (Idea) The classic algorithm for model checking CTL with fairness [CES86, section 4] is easily adapted to deal with EM_n modalities, yielding a $O(|\mathcal{M}| \times |\phi|)$ running time.

That **P**-hardness already appears with $TL(\mathsf{EX})$ is a folk result (for a proof, see the survey [Sch03]). □

Thus it seems that $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ is a good compromise between high expressive power and low model-checking complexity.

7 Conclusion

We proved that $ECTL^+$ and BTL_2 are expressively equivalent. Since BTL_2 is a natural fragment of MLO, the second-order monadic logic of order, our result provides an informative characterization of the expressive power of $ECTL^+$. The lack of similar results for CTL and other branching-time logics is one of the reasons why there is no clear consensus on what should be the branching-time logics of choice.

Then we proved that $ECTL^+$ and BTL_2 do not admit a finite basis. This negative result complements a similar result for CTL^* [RM01], explaining why these temporal logics are not presented in the usual form $TL(\mathsf{H}_1, \ldots, \mathsf{H}_k)$ of a logic built with a finite set of natural and independent modalities.

A side result of our study is that the fragment $TL(\mathsf{EU}, \{\mathsf{EM}_l\}_{l=1,2,\ldots})$ is enough to express all $ECTL^+$ formulae, but has a much lower model-checking complexity.

References

[AI01] M. Adler and N. Immerman. An $n!$ lower bound on formula size. In *Proc. 16th IEEE Symp. Logic in Computer Science (LICS 2001), Boston, MA, USA, June 2001*, pages 197–206. IEEE Comp. Soc. Press, 2001.

[BPM83] M. Ben-Ari, A. Pnueli, and Z. Manna. The temporal logic of branching time. *Acta Informatica*, 20:207–226, 1983.

[BR02] D. Beauquier and A. Rabinovich. Monadic logic of order over naturals has no finite base. *Journal of Logic and Computation*, 12(2):243–253, 2002.

[CE81] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. Logics of Programs Workshop, Yorktown Heights, New York, May 1981*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[CGP99] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* MIT Press, 1999.

[DS02] S. Demri and Ph. Schnoebelen. The complexity of propositional linear temporal logics in simple cases. *Information and Computation*, 174(1):84–103, 2002.

[EC80] 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.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

[EH86] E. A. Emerson and J. Y. Halpern. "Sometimes" and "Not Never" revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.

[Ehr61] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[EL87] E. A. Emerson and Chin-Laung Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8(3):275–306, 1987.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science, 1990.

[Eme96] E. A. Emerson. Automated temporal reasoning about reactive systems. In *Logics for Concurrency: Structure Versus Automata*, volume 1043 of *Lecture Notes in Computer Science*, pages 41–101. Springer, 1996.

[EVW02] K. Etessami, M. Y. Vardi, and T. Wilke. First order logic with two variables and unary temporal logic. *Information and Computation*, 179(2):279–295, 2002.

[GHR94] D. M. Gabbay, I. M. Hodkinson, and M. A. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, vol. 1*, volume 28 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1994.

[GPSS80] D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proc. 7th ACM Symp. Principles of Programming Languages (POPL '80), Las Vegas, NV, USA, Jan. 1980*, pages 163–173, 1980.

[GS85] Y. Gurevich and S. Shelah. The decision problem for branching time logic. *The Journal of Symbolic Logic*, 50(3):668–681, 1985.

[HR03] Y. Hirshfeld and A. Rabinovich. Future temporal logic needs infinitely many modalities. *Information and Computation*, 187(2):196–208, 2003.

[HR04] Y. Hirshfeld and A. Rabinovich. Logics for real time: Decidability and complexity. *Fundamenta Informaticae*, 62(1):1–28, 2004.

[HT87] T. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *Proc. 14th Int. Coll. Automata, Languages, and Programming (ICALP '87), Karlsruhe, FRG, July 1987*, volume 267 of *Lecture Notes in Computer Science*, pages 269–279. Springer, 1987.

[JL03] J. Johannsen and M. Lange. CTL$^+$ is complete for double exponential time. In *Proc. 30th Int. Coll. Automata, Languages, and Programming (ICALP 2003), Eindhoven, NL, July 2003*, volume 2719 of *Lecture Notes in Computer Science*. Springer, 2003.

[Kam68] J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, UCLA, Los Angeles, CA, USA, 1968.

[Lam80] L. Lamport. "Sometimes" is sometimes "Not Never". In *Proc. 7th ACM Symp. Principles of Programming Languages (POPL '80), Las Vegas, NV, USA, Jan. 1980*, pages 174–185, 1980.

[Lar94] F. Laroussinie. Logique temporelle avec passé pour la spécification et la vérification des systèmes réactifs. Thèse de Doctorat, I.N.P. de Grenoble, France, November 1994.

[LMS01] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking CTL^+ and $FCTL$ is hard. In *Proc. 4th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS 2001), Genova, Italy, Apr. 2001*, volume 2030 of *Lecture Notes in Computer Science*, pages 318–331. Springer, 2001.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, 1992.

[MP95] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer, 1995.

[MR03] F. Moller and A. Rabinovich. Counting on CTL*: On the expressive power of monadic path logic. *Information and Computation*, 184(1):147–159, 2003.

[MS03] N. Markey and Ph. Schnoebelen. Model checking a path (preliminary report). In *Proc. 14th Int. Conf. Concurrency Theory (CONCUR 2003), Marseille, France, Sep. 2003*, volume 2761 of *Lecture Notes in Computer Science*, pages 251–265. Springer, 2003.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[QS83] J.-P. Queille and J. Sifakis. Fairness and related properties in transition systems. A temporal logic to deal with fairness. *Acta Informatica*, 19(3):195–220, 1983.

[Rab69] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[Rab02] A. Rabinovich. Expressive power of temporal logics. In *Proc. 13th Int. Conf. Concurrency Theory (CONCUR 2002), Brno, Czech Republic, Aug. 2002*, volume 2421 of *Lecture Notes in Computer Science*, pages 57–75. Springer, 2002.

[RM01] A. Rabinovich and S. Maoz. An infinite hierarchy of temporal logics over branching time. *Information and Computation*, 171(2):306–332, 2001.

[SC85] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[Sch03] Ph. Schnoebelen. The complexity of temporal logic model checking. In *Advances in Modal Logic, vol. 4, selected papers from 4th Conf. Advances in Modal Logic (AiML 2002), Sep.-Oct. 2002, Toulouse, France*, pages 437–459. King's College Publication, 2003.

[Sto76] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Wil99] T. Wilke. CTL^+ is exponentially more succint than CTL. In *Proc. 19th Conf. Found. of Software Technology and Theor. Comp. Sci. (FST&TCS '99), Chennai, India, Dec. 1999*, volume 1738 of *Lecture Notes in Computer Science*, pages 110–121. Springer, 1999.