An Until Hierarchy and Other Applications of an Ehrenfeucht-Fraïssé Game for Temporal Logic*

Kousha Etessami

Bell Laboratories, Lucent Technologies

Murray Hill, New Jersey

kousha@research.bell-labs.com

Thomas Wilke

Lehrstuhl für Informatik VII

RWTH Aachen

52056 Aachen, Germany

wilke@informatik.rwth-aachen.de

*The research reported here was conducted while the authors were postdocs at DIMACS as part of the Special Year on Logic and Algorithms, and this paper was completed while the first author was at BRICS, Centre of the Danish National Research Foundation.

Summary

We prove there is a strict hierarchy of expressive power according to the Until depth of linear temporal logic (LTL) formulas: for each k, there is a natural property, based on quantitative fairness, that is not expressible with k nestings of Until operators, regardless of the number of applications of other operators, but is expressible by a formula with Until depth $k+1$. Our proof uses a new Ehrenfeucht-Fraïssé (EF) game designed specifically for LTL. These properties can all be expressed in first-order logic with quantifier depth and size $\mathcal{O}(\log k)$, and we use them to observe some interesting relationships between LTL and first-order expressibility. We note that our Until hierarchy proof for LTL carries over to the branching time logics, CTL and CTL*.

We then use the EF game in a novel way to effectively characterize (1) the LTL properties expressible without Until, as well as (2) those expressible without both Until and Next. By playing the game "on finite automata," we prove that the automata recognizing languages expressible in each of the two fragments have distinctive structural properties. The characterization for the first fragment was originally proved by Cohen, Perrin, and Pin using sophisticated semigroup-theoretic techniques. They asked whether such a characterization exists for the second fragment. The technique we develop is general and can potentially be applied in other contexts.

List of Symbols

$\mathcal{O}$ — big O, calligraphic

$\circ$ — circle

$\diamond$ — diamond

U, S — big U, big S, sans serif

$\equiv$, $\equiv_0$, $\equiv_{k,r}$, $\equiv_{k,r}^{+/-}$, $\equiv_k^{\diamond}$ — numbers or variables appear as subscripts, $\diamond$ and $+/-$ appear as superscripts

$\models$ — turnstile with double arms

$\sim_k$ — tilde, variable as subscript

$\mathcal{K}$ — big K, calligraphic

$\Gamma(\varphi)$ — big Gamma, with CTL* formula as argument

$\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$ — big A, big B, big C, German

$\succ$ — 1 and 2 appear as subscripts

$\rightsquigarrow$ — curved right arrow, + and $*$ appear as superscripts

1 Introduction

Temporal logic is a language for expressing relationships between the order of events occurring over time. A salient feature of the logic is the ability to build nested expressions using "Until". We establish a strict hierarchy of expressive power for temporal formulas based on Until nesting depth. To do this, we design a new Ehrenfeucht-Fraïssé (EF) game to specifically capture the power of linear temporal logic, LTL. (See (Ehrenfeucht, 1961) and (Fraïssé, 1954) for origins, and, e. g., (Immerman and Kozen, 1989) and (Thomas, 1993) for more recent use of EF games.) We use our game to show that, for each $k \geq 1$, there is a property FAIR_{k+1} that can not be expressed with $k-1$ nestings of Until operators, regardless of the number of applications of other operators. However, FAIR_{k+1} can be expressed with a formula that has Until depth k. FAIR_k is a simple and natural counting property which amounts to precisely a quantitative version of the notion of strong fairness, also known as "compassion" (Manna and Pnueli, 1992). In words, FAIR_k says: "there is no interval of time in which event a (e.g., a request by a process) occurs k times but event b (a response) does not occur." We observe that our Until hierarchy proof for LTL carries over easily to the branching time logics CTL and CTL*.

H. Kamp (1968) proved a long time ago that linear temporal logic is equivalent in expressive power to first-order logic on structures with unary predicates and a (Dedekind-complete) total ordering (see also (Gabbay *et al.*, 1980) and (Gabbay *et al.*, 1994)). We use the properties FAIR_k to draw some consequences about the relationship between the Until depth required to express a property in temporal logic and the first-order quantifier depth required to express the same property. FAIR_k is expressible via a first-order formula of quantifier depth $\mathcal{O}(\log k)$ using only three variables, as well as a formula of quantifier depth *and* size $\mathcal{O}(\log k)$ using 5 variables. Moreover, for all k, STAIR_k is expressible with a formula of alternation depth 2. Thus, there are properties whose expression requires exponentially more Until depth than first-order quantifier depth. These results complement some strong succinctness results that can be derived from a proof of Stockmeyer's (1974) combined with other considerations, as will be discussed in Section 5.

Since LTL formulas define regular sets, a natural question that has been asked is whether a given regular set is expressible within LTL or within one of its fragments. The work of Schützenberger (1965), Kamp (1968), McNaughton and Papert (1971), and Gabbay, Pnueli, Shelah, and Stavi (1980) yields an effective characterization of the finite automata recognizing LTL definable sets, while Cohen et al. (1993) provides an effective characterization for the LTL formulas that do not use the Until operator. Both those characterizations make heavy use of semigroup-theoretic techniques. We use the EF game in a novel way by "playing *on* finite automata" to simplify the proof of the second result and provide an effective characterization for LTL formulas that use Eventually as the only temporal operator. This answers a question raised by Cohen et al. and extends (Sistla and Zuck, 1993).

The outline of the paper is as follows. In Sect. 2 we review some basic notions; in Sect. 3 we introduce the EF game for LTL; in Sect. 4 we prove the Until hierarchy theorem and its extensions; in Sect. 5 we draw some consequences from the proof of the hierarchy theorem regarding the relationship between LTL and first-order logic; in Sect. 6 we use the game in a new way to provide automata-theoretic characterizations for fragments of LTL; and in Sect. 7 we make concluding remarks.

2 Terminology and Notation

For thorough coverage of definitions and background related to temporal logic please see the survey (Emerson, 1990) or the book (Gabbay *et al.*, 1994). We view linear temporal logic as expressing properties of words. As usual, we start indexing the positions of a word w at 0, and $|w|$ stands for the length of the word, with $|w| = \infty$ for ω-words. Given an alphabet $\Sigma = \{a_1, \ldots, a_n\}$, LTL formulas are composed of atomic propositions $p_{a_1}, \ldots, p_{a_n}$, the boolean connectives $\neg$, $\wedge$, $\vee$, and the temporal operators *Next* ($\bigcirc$), *Eventually* ($\Diamond$), and *Until* (U). We refer to the fragments of LTL without U, and without U and $\bigcirc$, by LTL($\bigcirc, \Diamond$) and LTL($\Diamond$), respectively.

Given a word $w = w_0 w_1 w_2 \ldots$, and given a position $i < |w|$ in w, we let $(w, i) \models \varphi$ denote the fact that φ is true on w at position i, and we define this inductively in the usual way, adopting the following standard conventions for the semantics of atomic formulas and temporal operators:

- $(w, i) \models p_{a_r}$ if $w_i = a_r$,
- $(w, i) \models \bigcirc\varphi$ if $i + 1 < |w|$ and $(w, i + 1) \models \varphi$,
- $(w, i) \models \Diamond\varphi$ if there exists j with $i \leq j < |w|$ such that $(w, j) \models \varphi$,
- $(w, i) \models \varphi \,\mathsf{U}\, \psi$ if there exists j with $i \leq j < |w|$ such that $(w, j) \models \psi$ and $(w, i') \models \varphi$ for all i' with $i \leq i' < j$.

We will be interested in the *depth* of a formula with respect to various operators. This is defined inductively in the natural way. We first give a general definition. Let $\Delta = \{\theta_1, \theta_2, \dots, \theta_k\}$ be a set of temporal operators, where $\theta_i(\varphi_1, \dots, \varphi_{r_i})$ denotes the arity r_i operator θ_i applied to the given sequence of formulas. (In this notation $\varphi \mathsf{U} \psi$ is written $\mathsf{U}(\varphi, \psi)$.) We define Δ-*depth*, inductively:

$$
\begin{aligned}
\Delta\text{-dp}(p_{a_j}) &= 0\,, \\
\Delta\text{-dp}(\neg\varphi) &= \Delta\text{-dp}(\varphi)\,, \\
\Delta\text{-dp}(\varphi \vee \psi) &= \max(\Delta\text{-dp}(\varphi), \Delta\text{-dp}(\psi))\,, \\
\Delta\text{-dp}(\theta'(\varphi_1, \dots, \varphi_{r'})) &= \max(\Delta\text{-dp}(\varphi_1), \dots, \Delta\text{-dp}(\varphi_{r'})) \text{ if } \theta' \notin \Delta\,, \\
\Delta\text{-dp}(\theta_i(\varphi_1, \dots, \varphi_{r_i})) &= \max(\Delta\text{-dp}(\varphi_1), \dots, \Delta\text{-dp}(\varphi_{r_i})) + 1 \text{ if } \theta_i \in \Delta\,.
\end{aligned}
$$

We will use *operator depth* to refer to $\{\bigcirc, \Diamond, \mathsf{U}\}$-depth, U-*depth* to refer to $\{\mathsf{U}\}$-depth, and *residual depth* to refer to $\{\bigcirc, \Diamond\}$-depth. It is worth mentioning here that the notion of U-depth is rather robust and is unaffected by the usual minor variations in the definition of temporal operators.

We say that (v, i) and (w, j) are k-*equivalent* iff they agree on all formulas of operator depth k or less. We denote this by $(v, i) \equiv_k (w, j)$. Note that $(v, i) \equiv_0 (w, j)$ if and only if the two words have the same character at positions i and j, respectively. Let $(v, i) \equiv_{k,r} (w, j)$ denote the fact that (v, i) and (w, j) agree on all formulas φ with U-depth k and residual depth r.

A *property* is just a set of strings or a set of ω-words, and we say a LTL formula φ *defines* or *expresses a property* $\mathcal{K}$ iff $\mathcal{K} = \{w \mid (w, 0) \models \varphi\}$.

When using first-order logic to express properties of words, we view a word as a first-order structure $(U, <, P_{a_1}, \ldots, P_{a_n})$, where $P_{a_1}, \ldots, P_{a_n}$ are unary predicates and U is an initial segment of the natural numbers which is well-ordered by the built-in relation $<$.

3 An EF Game for Temporal Logic

We now define the EF game for temporal logic. The *k-round LTL EF game* is played on a pair of words (w_0, w_1) and starts from a prespecified *initial configuration* (i_0, i_1), where a *configuration* is a pair of positions in w_0 and w_1, respectively.

The game is defined inductively as follows. In the 0-round game, there are no rounds to be played, and if $(w_0, i_0) \not\equiv_0 (w_1, i_1)$ Player I wins, otherwise Player II wins. In the $(k+1)$-round game, Player I wins if $(w_0, i_0) \not\equiv_0 (w_1, i_1)$. Otherwise, if $(w_0, i_0) \equiv_0 (w_1, i_1)$, one round is played, which either results in a win for Player I or a new configuration (i'_0, i'_1). In the second case, they then play a k-round game on (w_0, w_1) with initial configuration (i'_0, i'_1).

A round of the game proceeds as follows. At the beginning of a round, with the game in some configuration (i_0, i_1), Player I chooses one of three types of moves and they play according to this choice as follows.

[$\bigcirc$**-Move**]

> Player I may only choose this move if $i_0 + 1 < |w_0|$ or $i_1 + 1 < |w_1|$ (i. e., if we have not reached the end in one of the words). Player I then wins if $i_0 + 1 = |w_0|$ or $i_1 + 1 = |w_1|$ (i. e., if we are at the end of one word but not the other). Otherwise, the new configuration is $(i_0 + 1, i_1 + 1)$.

[$\Diamond$**-Move**]

1. Player I chooses w_δ, $\delta \in \{0,1\}$, and a position i'_δ in w_δ with $i_\delta \leq i'_\delta < |w_\delta|$.

2. Player II responds by choosing a position $i'_{1-\delta}$ in $w_{1-\delta}$ with $i_{1-\delta} \leq i'_{1-\delta} < |w_{1-\delta}|$.

3. The new configuration is (i'_0, i'_1).

[**U-Move**]

1. Player I chooses w_δ, $\delta \in \{0,1\}$, and a position i'_δ in w_δ with $i_\delta \leq i'_\delta < |w_\delta|$.

2. Player II responds by choosing a position $i'_{1-\delta}$ in $w_{1-\delta}$ with $i_{1-\delta} \leq i'_{1-\delta} < |w_{1-\delta}|$ such that if $i_\delta = i'_\delta$ then $i_{1-\delta} = i'_{1-\delta}$.

3. Player I chooses one of the following two steps.

   (a) Player I sets the new configuration to (i'_0, i'_1).

   (b) Player I chooses a position $i''_{1-\delta}$ in $w_{1-\delta}$ with $i_{1-\delta} \leq i''_{1-\delta} < i'_{1-\delta}$ and Player II responds with i''_δ in w_δ, $i_\delta \leq i''_\delta < i'_\delta$. The new configuration is set to (i''_0, i''_1).

We say Player II has a *winning strategy* in the k-round game on (v, i) and (w, j) if she can win with initial configuration (i, j) regardless of the moves made by Player I. We denote this by $(v, i) \sim_k (w, j)$. Otherwise, we say Player I has a winning strategy.

The intuition behind the game is: In the k-round game Player I is trying to establish that (v, i) and (w, j) are distinguished by some temporal formula of operator depth k. If the formula he has in mind is, for instance, $\gamma = \varphi \mathsf{U} \psi$, he plays the U-move: on the word (v, i) that satisfies γ he plays a point i' ahead such that ψ is satisfied at i' and φ is satisfied by all the points from i until i'. Player II does not know the formula γ that Player I has in mind, but she responds with a position j', claiming "any formulas ψ of operator depth $k-1$ satisfied by the point i' is satisfied by the point j', and furthermore the points until j' satisfy any $(k-1)$-depth formula φ satisfied by the points until i'." Player I then challenges one of the two assertions by II. For example, he pebbles a point j'' where $j \leq j'' < j'$, asserting: "You are lying. The point j'' does not satisfy φ." Player II then responds to the challenge at j'' by choosing a point i'' where $i \leq i'' < i'$, asserting that "any $(k-1)$-depth formula φ not satisfied by j'' is not satisfied by i'' either." They then proceed with the $(k-1)$-round game on (i'', j''). Other moves have a similar but simpler intuition.

The key fact about the game is the following.

Theorem 3.1 *For all $k \geq 0$,*

$$(v, i) \sim_k (w, j) \quad \textit{if and only if} \quad (v, i) \equiv_k (w, j)\,.$$

Proof. The proof is by induction on k. For $k = 0$, the proof is immediate. Suppose true for k, we will prove the theorem for $k+1$.

($\Rightarrow$) Suppose $(v, i) \sim_{k+1} (w, j)$, and let φ be an operator depth $k+1$ formula. We would like to show that $(v, i) \models \varphi$ iff $(w, j) \models \varphi$. We may assume that the outermost connective in φ is a temporal operator, for if (v, i) and (w, j) agree on all such operator depth $k+1$ formulas then they also agree on their boolean combinations.

There are several cases to consider, based on which temporal operator is the outermost in φ. We will focus on the most interesting case, namely when the U-operator is outermost. The other cases are similar but easier.

Let $\varphi = \psi \mathsf{U} \gamma$. Without loss of generality, suppose $(v, i) \models \varphi$, for if both do not satisfy φ then they agree on φ and we are done. Let Player I choose the U-move, and let him pick a witness point $i' \geq i$ on v for φ such that γ holds at i' and ψ holds at all i'' where $i \leq i'' < i'$. Player II responds according to its winning strategy with a j' on w.

Now in the case when Player I chooses to set the configuration to (i', j'), then we know by II's winning strategy that $(v, i') \sim_k (w, j')$. Thus by the inductive hypothesis $(v, i') \equiv_k (w, j')$, and hence $(w, j') \models \gamma$. On the other hand, when Player I chooses any j'', by II's winning strategy, II can choose an i'' such that $(v, i'') \sim_k (w, j'')$. Thus, again by the induction hypothesis $(w, j'') \models \psi$. Hence $(w, j) \models \psi \mathsf{U} \gamma$.

($\Leftarrow$) Suppose $(v, i) \not\sim_{k+1} (w, j)$. We will find a formula ψ of depth $k+1$ on which the two structures disagree. There are again several cases based on which move Player I plays first. We again focus on the most interesting case when Player I plays the U-move.

Suppose Player I's first move in his winning strategy is the U-move. Suppose, w. l. o. g., Player I chooses $i' \geq i$ in v. Consider the formula

$$\psi = (\varphi_i \vee \varphi_{i+1} \vee \ldots \vee \varphi_{i'-1}) \mathsf{U} \varphi_{i'}$$

where φ_l is the conjunction of all operator depth k formulas that hold true at (v, l). (As is the usual convention, when $i = i'$ the left hand side of the U-operator reduces to FALSE.) The formula ψ can itself be expressed as an operator depth $k+1$ formula because there are, up to equivalence, only a bounded number of operator depth k formulas (this can be proved easily by induction).

We claim that ψ holds on (v, i) but not on (w, j). That ψ holds on (v, i) is immediate from its definition. But ψ does not hold on (w, j), for otherwise II would have a winning strategy in reply to I's play at i' as follows. There would be a point j' at which 1) $\varphi_{i'}$ holds, and 2) for all points j'' with $j \leq j'' < j'$, $\varphi_i \vee \varphi_{i+1} \vee \ldots \vee \varphi_{i'-1}$ holds. In reply to Player I playing i', II would play j'. Then if Player I according to his winning strategy plays the endpoint i', the point j' satisfies $\varphi_{i'}$, and thus satisfies all operator depth k properties that the point i' does, and we are done because by the inductive hypothesis we have contradicted the fact that this is a winning strategy for Player I. Otherwise, if Player I plays a point j'' according to his strategy, then since j'' satisfies $\varphi_i \vee \varphi_{i+1} \vee \ldots \vee \varphi_{i'-1}$, it must satisfy some $\varphi_{i''}$. Thus, II plays the point i'', and by definition j'' satisfies all operator depth k properties of i''. Thus, again by the inductive hypothesis, this contradicts the fact that this is a winning strategy for Player I. ∎

Corollary 3.2 *Given a property $\mathcal{K}$, if for any pair $v \in \mathcal{K}$ and $w \notin \mathcal{K}$, Player I has a winning strategy in the k-round LTL EF game on $(v, 0)$ and $(w, 0)$, then there is a depth k temporal formula $\varphi_{\mathcal{K}}$ that defines the property $\mathcal{K}$.*

Proof. For $v \in \mathcal{K}$ and $w \notin \mathcal{K}$ Theorem 3.1 yields an operator-depth k formula $\psi_{v,w}$ that is true on $(v, 0)$ but false on $(w, 0)$. We define $\varphi_{\mathcal{K}}$ as follows.

$$\varphi_{\mathcal{K}} = \bigvee_{v \in \mathcal{K}} \bigwedge_{w \notin \mathcal{K}} \psi_{v,w} \,.$$

Up to equivalence, there are only finitely many formulas with operator depth k, thus the disjunction and conjunction in $\varphi_{\mathcal{K}}$ are finite, and $\varphi_{\mathcal{K}}$ properly defines a formula. The formula $\varphi_{\mathcal{K}}$ defines the property $\mathcal{K}$ because each $v \in \mathcal{K}$ satisfies the term $\bigwedge_{w \notin \mathcal{K}} \psi_{v,w}$, but no $w \notin \mathcal{K}$ satisfies any such term.

Let $(v, i) \sim_{k,r} (w, j)$ denote the fact that Player II has a winning strategy on (v, i) and (w, j) in the LTL game with k U-moves and r residual moves, interleaved as Player I wishes. The following is a straightforward refinement of Theorem 3.1.

Theorem 3.3 *For all $k, r \geq 0$,*

$$(v, i) \sim_{k,r} (w, j) \quad \textit{if and only if} \quad (v, i) \equiv_{k,r} (w, j)\,.$$

The proof proceeds by simultaneous induction on k and r, mimicking the proof of Theorem 3.1 except that the two resources are kept track of separately.

Remark 3.4 *Analogs of Theorem 3.1, Corollary 3.2, and Theorem 3.3 also hold for the LTL fragments LTL($\Diamond$) and LTL($\bigcirc, \Diamond$), where in each case the possible type of move is restricted to only the $\Diamond$-move or to the $\bigcirc$- and $\Diamond$-move, respectively.*

We note that it is easy to modify the moves in the game in order to capture variants of the temporal operators.

4 An Until Hierarchy

For $k \geq 1$, let FAIR_k define a property of words over the alphabet $\Sigma = \{a, b, c\}$ as follows:

$$\mathrm{FAIR}_k = \{v \mid \text{there is no substring of } v \text{ in which } a \text{ occurs } k \text{ times but no } b\text{'s occur}\}.$$

For pedagogical reasons, we will work with the complement of the property FAIR_k, which we call STAIR_k. We are going to use these simple properties to separate the levels of the "Until hierarchy". Clearly, FAIR_k is definable in a given level of the Until hierarchy if and only if STAIR_k is, because these levels are closed under complementation.

The main theorem of this subsection is the following.

Theorem 4.1 *For each $k \geq 1$, FAIR_{k+1}*

(i) is expressible with U-depth k, but

(ii) is not expressible with U-depth $k-1$ (even with arbitrary depth in the residual temporal operators).

Proof. Assertion (i) is easy. For $k \geq 0$, STAIR_{k+1} can be defined by a formula φ_k of U-depth k. Let $\varphi_k = \Diamond\, \gamma_k$, where γ_k is defined inductively:

$$\gamma_0 = p_a \quad \text{and} \quad \gamma_k = (p_a \wedge \bigcirc(p_c \,\mathsf{U}\, \gamma_{k-1}))\,. \tag{1}$$

FAIR_{k+1} is thus expressed by $\neg\varphi_k$. The interested reader may note that the definition of φ_k contains nestings of operators only on the right hand side of the U-operators.

To prove (ii) we define, for each k, r, a pair of words $v_{k,r}$ and $w_{k,r}$ such that $v_{k,r} \in \mathrm{STAIR}_{k+1}$ and $w_{k,r} \notin \mathrm{STAIR}_{k+1}$, and we show that Player II has a winning strategy on $(v_{k,r}, 0)$ and $(w_{k,r}, 0)$ in the EF game with $k-1$ U-moves and r residual moves, i. e., $(v_{k,r}, 0) \sim_{k-1,r} (w_{k,r}, 0)$. Thus, by Theorem 3.3, $(v_{k,r}, 0) \equiv_{k-1,r} (w_{k,r}, 0)$, and hence STAIR_{k+1} can not be defined by any formula with U-depth $k-1$ and residual depth r, for *any* r. Hence, nor can FAIR_{k+1}.

To simplify the presentation of our proof, we define $v_{k,r}$ and $w_{k,r}$ as ω-words from Σ^ω. The proof can be modified to work over finite strings as well by truncating the words $v_{k,r}$ and $w_{k,r}$ after a sufficient length and adjusting the winning strategy below for Player II.

For $k \geq 1$ and $r \geq 0$, let

$$\begin{aligned} v_{k,r} &= ac^{r+1} w_{k,r}\,, \\ w_{k,r} &= (ac^{r+1})^k (bc^{r+1}(ac^{r+1})^k)^\omega\,. \end{aligned}$$

Figure 1 provides a way to visualize $v_{k,r}$ (and $w_{k,r}$) as "staircases." The proof will essentially show that Player I can only force Player II to "climb the staircase" one *step* at a time, and he must use up one U-move for each step. Assertion (ii) is now established by Lemma 4.2. ∎

Lemma 4.2 *For $k \geq 1$ and $r \geq 0$,*

$$(v_{k,r}, 0) \sim_{k-1,r} (w_{k,r}, 0).$$

Proof. We need some definitions: Given positions x and x' in, say, $v_{k,r}$, where $x \leq x'$, let $v_{k,r}[x, x']$ denote the substring of $v_{k,r}$ defined by the positions

$x, x+1, \ldots, x'$. For a position x, we let $\mathrm{TD}(x)$ denote x's distance to the top of the current "staircase." More formally, for a position x of, say, $v_{k,r}$, let $e(x)$ be the smallest position $\geq x$ where a b occurs. Then $\mathrm{TD}(x)$ is the number of occurrences of a in $v_{k,r}[x, e(x)]$. For position x in Figure 1, $\mathrm{TD}(x) = 2$. Let $\mathrm{PD}(x)$ denote x's distance to the end of the current "plateau," i. e., to the next (or current) occurrence of an a or b. Thus for x in the figure, $\mathrm{PD}(x) = 3$, and for any position z were an a or b occurs, $\mathrm{PD}(z) = 0$. We will say that a position x' is exactly *one step ahead* of x if $x' = x + r + 2$ *and* $\mathrm{TD}(x') = \mathrm{TD}(x) - 1$. Note that $\mathrm{PD}(x) = \mathrm{PD}(x')$ for such x and x'. For example, position y in Figure 1 is exactly one step ahead of position x.

Observe that the $\Diamond$-move is useless on these pairs of words. Unless Player I repebbles the initial positions of $v_{k,r}$ (in which case Player II repebbles the initial position of $w_{k,r}$), Player II can respond so that the remaining ω-words are identical. Claim 4.3 now establishes a consistent strategy that yields a win for Player II.

Claim 4.3 *Player II can play in such a way that after k' ($k' \leq k-1$) U-moves and r' ($r' \leq r$) residual moves, with current configuration (i, j) on $v_{k,r}$ and $w_{k,r}$, the following conditions hold:*

1. $\mathrm{PD}(i) = \mathrm{PD}(j)$.

2. $\mathrm{TD}(i) = \mathrm{TD}(j)$ *or* $\mathrm{TD}(i) = \mathrm{TD}(j) + 1$.

3. *If* $\mathrm{TD}(i) = \mathrm{TD}(j) + 1$ *then*

   (a) $\mathrm{TD}(i) \geq k - k'$ *(and thus* $\mathrm{TD}(j) \geq k - k' - 1$*), and*

   (b) *if* $\mathrm{TD}(i) = k - k'$ *then* $\mathrm{PD}(i) \geq r - r' + 1$ *(and thus also* $\mathrm{PD}(j) \geq r - r' + 1$*).*

Proof. The claim is proved by simultaneous induction on k' and r'. Assuming the inductive claim, we extend the strategy it defines for Player II to one more round, maintaining the claim. For the *base case:* $k' = 0$, $r' = 0$, all conditions hold at initial configuration $(0, 0)$.

Inductive step on r'. Assume the claim for $(k', r'-1)$, we show it for (k',r'). As we have argued, we only need to consider O. Here, there is nothing for Player II to do. We just need to establish that the inductive claim remains true when we move ahead one position on both structures. We omit the calculation.

Inductive step on k'. Suppose the claim holds for $(k' - 1, r')$, we show it for (k', r'). We can assume that prior to the U-move we are at (i, j) and that $\mathrm{TD}(i) = \mathrm{TD}(j) + 1$, because otherwise $v_{k,r}[i, \infty) = w_{k,r}[j, \infty)$, which w. l. o. g. we assume is not the case. By the induction hypothesis, before the move $\mathrm{TD}(i) \geq k - k' + 1$, and if $\mathrm{TD}(i) = k - k' + 1$ then $\mathrm{PD}(i) \geq r - r' + 1$. There are two similar cases based on which word Player I plays on.

Case I: Player I plays a point i' *ahead on* $v_{k,r}$.

First sub-case: i' *is at least one* step *ahead of* i. Player II will now play the point j' such that $j' - j = i' - i - (r+2)$. In other words, Player II "catches up a step" with Player I. Thus, $\mathrm{TD}(i') = \mathrm{TD}(j')$ and $\mathrm{PD}(i') = \mathrm{PD}(j')$, i. e., $v_{k,r}[i', \infty) = w_{k,r}[j', \infty)$. Moreover, the substring $w_{k,r}[j, j']$ is now a suffix of $v_{k,r}[i, i']$, because we took exactly one less step than Player I. Hence, regardless of what Player I now chooses on $w_{k,r}[j, j']$, Player II can respond with the corresponding point in the suffix $v_{k,r}[i + r + 2, i']$ so that the two remaining ω-words are identical, and thus we are done.

Second sub-case: i' *is less than one step ahead of* i. Player II responds with a point j' ahead of j such that $j' - j = i' - i$. But we have thus moved ahead less than one step and the same amount on both words. We thus have identical substrings $v_{k,r}[i, i']$ and $w_{k,r}[j, j']$. Regardless of which point j'' in $w_{k,r}[j, j']$ Player I chooses subsequently, Player II chooses the corresponding point i'' in $v_{k,r}[i, i']$, so that we have $\mathrm{TD}(i'') = \mathrm{TD}(j'') + 1$ and $\mathrm{TD}(i'') \geq k - k'$. Moreover, if $\mathrm{TD}(i'') = k - k'$, then since we have moved ahead less than one whole step, we can not be any further to the right on the plateau, i. e., $\mathrm{PD}(i'') = \mathrm{PD}(j'') \geq r - r' + 1$.

Case II: Player I plays a point j' ahead on $w_{k,r}$: Clearly, since $\mathrm{TD}(i) = \mathrm{TD}(j) + 1$, Player II can "catch up a step" and respond with the point i' such that $i' - i = (j' - j) + r + 2$, and $v_{k,r}[i', \infty) = w_{k,r}[j', \infty)$. Furthermore, note that $w_{k,r}[j, j']$ is a suffix of $v_{k,r}[i, i']$. Thus, by the analysis in the first case, we only need to consider the case when Player I subsequently plays a point $i'' \in [i, i+r+2]$. Otherwise, Player II would respond with the corresponding point j'' such that $i' - i'' = j' - j''$, and we would have $v_{k,r}[i'', \infty) = w_{k,r}[j'', \infty)$.

However, if Player I does play $i'' \in [i, i + r + 2]$, i.e., a point that is at most one whole step ahead of i, then Player II responds with a point j'' that is the same distance ahead of j, and by the same analysis as in first case again, we are done. ■

The lemma follows from the claim. After k' U-moves, $k' \leq k - 1$, and r' residual moves, $r' \leq r$, by (1) $\mathrm{PD}(i) = \mathrm{PD}(j)$. If $\mathrm{TD}(i) = \mathrm{TD}(j)$, then $v_k[i, \infty) = w_k[j, \infty)$, and we are done. Otherwise, by (2) $\mathrm{TD}(i) = \mathrm{TD}(j) + 1$. Thus by (3.$a$), either $\mathrm{TD}(i) > k-(k-1) = 1$, in which case we are reading the same symbol on both words (an a if $\mathrm{PD}(i) = \mathrm{PD}(j) = 0$, and a c otherwise), or else $\mathrm{TD}(i) = 1$, in which case by (3.b), $\mathrm{PD}(i) = \mathrm{PD}(j) \geq r - r + 1 = 1$,

i. e., we are reading a c on both words. Hence, $(v, i) \equiv_0 (w, j)$. ■

The standard temporal operators *Next* ($\bigcirc$), *Eventually* ($\Diamond$) and *Until* (U) each have past-time counterparts: *Previously* ($\ominus$), *Eventually in the past* ($\diamondminus$), and *Since* (S). Their definitions are obvious modifications of their counterparts. For example, we define Since as follows: $(w, i) \models \varphi \mathsf{~S~} \psi$ if there exists j, $0 \leq j \leq i$ such that $(w, j) \models \psi$ and $(w, i') \models \varphi$ for all i' with $j < i' \leq i$. We define $(v, i) \equiv_{k,r}^{+/-} (w, j)$ to mean that (v, i) and (w, j) agree on all formulas with $\{\mathsf{U}, \mathsf{S}\}$-depth k and residual depth r (which now includes depth in past operators $\diamondminus$ and $\ominus$ as well). We also define $(v, i) \sim_{k,r}^{+/-} (w, j)$ to mean that Player II has a winning strategy in the game with both past and future moves where, moreover, there are a total of k uses of S- and U-moves, while there are r uses of the other moves. It is not more difficult to show that Theorem 3.3 also holds when $\equiv_{k,r}^{+/-}$ and $\sim_{k,r}^{+/-}$ are substituted for $\equiv_{k,r}$ and $\sim_{k,r}$, respectively. This allows us to extend Theorem 4.1 to include past operators as well:

Theorem 4.4 *For each $k \geq 1$,* STAIR_{2k+1}

(i) is expressible with $\{\mathsf{U}, \mathsf{S}\}$-depth k, but

(ii) is not expressible with $\{\mathsf{U}, \mathsf{S}\}$-depth $k - 1$.

To prove (i) it is sufficient to note that staircases of height $2k + 1$ are expressed with the $\{\mathsf{U}, \mathsf{S}\}$-depth k formula $\Diamond(\delta_k \wedge \gamma_k)$ where

$$\delta_0 = \text{TRUE} \quad \text{and} \quad \delta_k = \ominus(p_c \mathsf{~S~} (p_a \wedge \delta_{k-1}))$$

and γ_k is as in (1). To prove (ii) we set

$$\begin{aligned}
u_r &= (ac^{2(r+1)})\,, \\
v'_{k,r} &= (u_r^{2k}bc^{2(r+1)})^{k+r}u_r^{2k+1}b(c^{2(r+1)}u_r^{2k}b)^{k+r}\,, \\
w'_{k,r} &= (u_r^{2k}bc^{2(r+1)})^{k+r}u_r^{2k}b(c^{2(r+1)}u_r^{2k}b)^{k+r}\,,
\end{aligned}$$

and show just as before:

Lemma 4.5 *For $k \geq 1$ and $r \geq 0$,*

$$(v'_{k,r}, 0) \sim^{+/-}_{k-1,r} (w'_{k,r}, 0)\,.$$

For the proof here we need to maintain a more complicated invariant than the one in Claim 4.3, where rather than being concerned only with the "topdistance" we consider the minimum of "topdistance" and "bottomdistance" and make sure that, if we are lagging behind "one step" on the $v'_{k,r}$ structure, this distance is not below the current threshold (depending on the round we are on). Moreover, since we have defined $v'_{k,r}$ and $w'_{k,r}$ as finite words in this case, we can not ignore $\Diamond$-moves (both past and future) and need to keep track of our distance to the end of a string and make sure that it is also not below a certain threshold. This is the reason why the strings are "padded" with $k+r$ identical staircases of height k on both sides of the middle staircase where the discrepancy in height occurs. We omit the details.

Branching Time

In this section we observe that the Until hierarchy proof for LTL, Theorem 4.1, carries over virtually unchanged to the branching time logics CTL and

CTL*. (Please see, e.g., (Emerson, 1990) for background on these branching-time logics.)

Consider the branching property $\exists\mathrm{STAIR}_k$ = "there exists a path on which STAIR_k holds." The property $\exists\mathrm{STAIR}_k$ is expressible by a CTL formula as follows. For $k \geq 0$, let $\rho_k = \mathsf{E}\Diamond\psi_k$,[1] where ψ_k is defined inductively:

$$\psi_0 = p_a \quad \text{and} \quad \psi_k = (p_a \wedge \mathsf{E}\bigcirc\mathsf{E}[p_c \,\mathsf{U}\, \psi_{k-1}])\,. \tag{2}$$

then ρ_k expresses exactly $\exists\mathrm{STAIR}_{k+1}$.

Given a CTL* formula φ, define $\Gamma(\varphi)$ to be the LTL formula obtained from φ by eliminating all occurences of path quantifiers, E and A, in φ. We let the U-depth of a CTL* formula φ be defined as the U-depth of $\Gamma(\varphi)$. We have just demonstrated that $\exists\mathrm{STAIR}_{k+1}$ is expressible in CTL with U-depth k. It is, however, not expressible with smaller U-depth, not even in CTL*:

Theorem 4.6 *For $k \geq 0$, the property $\exists\mathrm{STAIR}_{k+1}$ is expressible in CTL with U-depth k, but is not expressible, even in CTL*, and even over finite structures, with U-depth $\leq k-1$.*

Proof. Having already established the first claim, we have to show that $\exists\mathrm{STAIR}_{k+1}$ is not expressible in CTL* with until depth $k-1$. This will follow as a corollary of our proof for the until hierarchy in LTL, Theorem 4.1.

Note that we may view any ω-word $u = vw^\omega$, where v and w are finite strings, as a finite branching structure with just *one* branch. In other words,

[1] Note that our notation deviates slightly from Emerson (1990), e.g., we write $\mathsf{E}\Diamond$ instead of EF.

given such a word $u = vw^\omega$, we may define a corresponding finite Kripke structure $K(u)$ which has an initial linear chain of states whose labels are given by the word v, followed by a single "loop" chain whose labels are given by the word w. Let s_0^u be the initial state on the linear chain in $K(u)$.

Intuitively, it is clear that path quantifiers are irrelevant on a structure without branching, and indeed it is easy to establish that for any CTL* formula φ, and any ω-word $u = vw^\omega$:

$$(K(u), s_0^u) \models \varphi \qquad \text{iff} \qquad (u, 0) \models \Gamma(\varphi) \ . \tag{3}$$

Now, consider the ω-words $v_{k,r}$ and $w_{k,r}$ from the proof of Theorem 4.1. Both words are of the form vw^ω, and hence the finite Kripke structures $K(v_{k,r})$ and $K(w_{k,r})$ are well-defined. It is easy to verify that $(K(v_{k,r}), s_0^{v_{k,r}}) \in \exists\mathrm{STAIR}_{k+1}$, while $(K(w_{k,r}), s_0^{w_{k,r}}) \notin \exists\mathrm{STAIR}_{k+1}$. Thus the two pointed Kripke structures disagree on the property $\exists\mathrm{STAIR}_{k+1}$.

Now, consider any CTL* formula φ with U-depth $\leq k-1$. We claim that such a formula can not distinguish $(K(v_{k,r}), s_0^{v_{k,r}})$ from $(K(w_{k,r}), s_0^{w_{k,r}})$, and therefore can not define $\exists\mathrm{STAIR}_{k+1}$.

Note that since $\Gamma(\varphi)$ has U-depth $\leq k-1$, it follows from Lemma 4.2 that $(v_{k,r}, 0) \models \Gamma(\varphi)$ iff $(w_{k,r}, 0) \models \Gamma(\varphi)$. But then by (3), it follows that

$$(K(v_{k,r}), s_0^{v_{k,r}}) \models \varphi \qquad \text{iff} \qquad (K(w_{k,r}), s_0^{w_{k,r}}) \models \varphi \ ,$$

which establishes our claim. ■

5 First-order Logic versus LTL

L. Stockmeyer (1974) proved that any decision procedure for satisfiability of first-order formulas over words requires non-elementary computational complexity. A close examination of his proof (using his Lemma 4.26, p. 107) reveals that there are first-order sentences over words, which use only 3 variables, and which have quantifier depth k and size $\mathcal{O}(k)$, such that the minimum length string satisfying this sentence has size non-elementary in k.[2] A corollary of this is that there is a non-elementary gap between the succinctness of first-order logic versus that of temporal logic, because it is known (see, e.g., (Sistla and Clarke, 1985)) that every satisfiable temporal formula must have models that are of size at most exponential in the size of the formula.

We can use the FAIR_k properties and Theorem 4.1 to better understand the interrelationships between first-order quantifier depth, first-order alternation depth, temporal logic operator depth, and Until depth:

Proposition 5.1 *For every k, there are first-order sentences ψ_k, ψ'_k, and ψ''_k that define STAIR_k and satisfy the following properties.*

1. *The sentences ψ_k use only five variables and are of quantifier depth and size $\mathcal{O}(\log k)$.*
2. *The sentences ψ'_k use only three variables and are of quantifier depth $\mathcal{O}(\log k)$ and size $\mathcal{O}(k)$.*
3. *The sentences ψ''_k are of quantifier alternation depth two and size $\mathcal{O}(k)$.*

[2] A function $f(k)$ is *non-elementary* in k if it grows faster than $tower_c(k)$ for any number $c \geq 0$, where, $tower_0(k) = k$, and inductively, $tower_i(k) = 2^{tower_{i-1}(k)}$.

(Note that the negation of these sentences define FAIR_k *and satisfy the same properties.)*

Proof. We define ψ_k for $k = 2^d + 1$ via a formula ρ_d defined by induction. A straightforward modification works for arbitrary k. The formula ρ_d is a formula in x_1 and x_2 that means "positions x_1 and x_2 have a's, and there are $2^{d-1} - 1$ occurrences of a in between them, and no b's ". (The idea of the expression is repeated doubling and renaming of variables.)

$$\begin{aligned}
\psi_{2^d+1} &= \exists x_1 \exists x_2 (x_1 < x_2 \wedge \rho_d) \\
\rho_d &= \exists x_3 (x_1 < x_3 < x_2 \wedge \forall x_4 \forall x_5 (\eta \rightarrow \forall x_1 \forall x_2 (\mu \rightarrow \rho_{d-1}))) \\
\eta &= (x_4 = x_1 \wedge x_5 = x_3) \vee (x_4 = x_3 \wedge x_5 = x_2) \\
\mu &= (x_1 = x_4 \wedge x_2 = x_5) \\
\rho_1 &= P_a(x_1) \wedge P_a(x_2) \wedge \forall x_3 (x_1 < x_3 < x_2 \rightarrow P_c(x_3))
\end{aligned}$$

It is even easier to define the ψ'_k's, i. e., to show that 3 variables suffice if we are only interested in maintaining quantifier depth $\mathcal{O}(\log k)$ but do not care about size; we just do not need to rename variables in the above. Defining the ψ''_k's is also trivial: we just existentially quantify the k distinct points where a occurs and assert that for all points between them b does not occur:

$$\psi''_k = \exists x_1 \ldots \exists x_k (\bigwedge_{1 \leq i < k} x_i < x_{i+1} \wedge \bigwedge_{1 \leq i \leq k} P_a(x_i) \wedge \forall z (x_1 < z < x_k \rightarrow \neg P_b(z)))$$

■

Of course, every LTL formula with operator depth k does have an equivalent first-order expression with quantifier depth $\mathcal{O}(k)$, as can readily be seen from the definition of the LTL operators.

A. Rabinovich (1996) has pointed out to us that if one is only interested in proving that the Until hierarchy is strict (without obtaining explicit properties that separate the levels) there is an alternative proof that (for the future only hierarchy) achieves this. His proof uses a number of syntactic case distinctions to show that every LTL($\Diamond$,$\bigcirc$) formula can be written as a boolean combination of Σ_2 first-order formulas, and that every U-depth k formula can thereby be written in Σ_{4k+2}. He then uses the strictness of the first-order quantifier alternation hierarchy (Thomas, 1982) and the equivalence of first-order and temporal logic (Kamp, 1968) to prove the strictness of the future Until hierarchy.

Observe that even though every U-depth k property is expressible in Σ_{4k+2} the FAIR$_k$ properties can all be expressed in Σ_2 (see Proposition 5.1.3). Thus, the necessity of large Until depth does not imply the necessity of large alternation depth.

6 Effective Characterizations of Fragments of Temporal Logic

We now use the EF game of Section 3 to fully characterize the structure of the automata recognizing properties definable in LTL($\Diamond$). Our technique is general, and we demonstrate this by also characterizing the automata recognizing LTL($\bigcirc, \Diamond$)-definable properties, reproving the characterization for LTL($\bigcirc, \Diamond$) first obtained in (Cohen *et al.*, 1993), where semigroup-theoretic techniques were used.

We want to show that automata with certain structural characteristics recognize exactly those properties of strings definable in a given logic. To do so, for one direction we show that for every such automaton and any pair of strings that lead to distinct states in that automaton, there is a winning strategy for Player I in the game for the respective logic (where the number of rounds is bounded by a function of the number of states of the automaton). The fact that the set of strings recognized by such an automaton is definable in the logic then follows from Corollary 3.2 together with Remark 3.4. Player I will exploit the structural characteristics of the automaton to win the game. For the other direction, we show that for any automaton lacking the given structural properties there are, for each k, two strings only one of which belongs to the language of the automaton, but on which Player II has a winning strategy in the k round game.

A deterministic finite automaton $\mathfrak{A} = (\Sigma, Q, q_I, \delta, F)$ can be viewed as a labeled digraph in a natural way. We refer to this graph as the *transition graph of* $\mathfrak{A}$ and denote it by $G_{\mathfrak{A}}$. We write δ^* for the extended transition function of $\mathfrak{A}$. The *closure graph of* $\mathfrak{A}$ is the labeled digraph whose vertex set is Q and which has an edge from q to q' labeled with a non-empty word u if $\delta^*(q, u) = q'$. This graph is denoted by $H_{\mathfrak{A}}$. (Note that this graph has an infinite edge set. To answer the decidability questions we are interested in, it will, however, not be necessary to construct this graph.) The language recognized by $\mathfrak{A}$ is denoted by $L(\mathfrak{A})$.

For a given string $u = u_0 u_1 \ldots u_{n-1}$, its *reverse* is denoted by u^ρ and defined as $u_{n-1} u_{n-2} \ldots u_0$. The reverse of a set of strings L is the set $\{u^\rho \mid u \in L\}$.

The characterization of LTL($\Diamond$) is stated in the following theorem.

Theorem 6.1 *For a minimized deterministic finite automaton $\mathfrak{A}$, the following are equivalent:*

(A) $\mathfrak{A}$ *recognizes the reverse of a LTL($\Diamond$)-definable set of strings.*

(B) The closure graph $H_{\mathfrak{A}}$ does not have subgraphs of either of the types (i) and (ii) displayed in Figure 2, where a denotes an arbitrary letter, x and y arbitrary non-empty strings, p and p' arbitrary states, and q and q' distinct states.

Note that $H_{\mathfrak{A}}$ has a subgraph of type (ii) if and only if there exist two edges in $G_{\mathfrak{A}}$ that have the same label, sources in the same strongly connected component (SCC), and distinct targets. Thus checking whether for a given automaton $\mathfrak{A}$, the (infinite) graph $H_{\mathfrak{A}}$ has a subgraph of type (ii) can be done effectively, in fact, in time $\mathcal{O}(|Q| \times |A|)$.

Note also that if $H_{\mathfrak{A}}$ does not have a subgraph of type (i) or (ii), certain other subgraphs will not occur either:

Lemma 6.2 *If $\mathfrak{A}$ is an automaton such that $H_{\mathfrak{A}}$ does not have subgraphs of type (i) or (ii), then $H_{\mathfrak{A}}$ does not have subgraphs of the types (iii) and (iv) displayed in Figure 3, where q and q' again denote distinct states.*

Proof. Suppose $H_{\mathfrak{A}}$ has no subgraph of type (i) or (ii) but a subgraph of type (iii). Fix such a subgraph and let q and q' denote the respective states in $H_{\mathfrak{A}}$. Since $\mathfrak{A}$ is complete, there exists a transitions starting from q and labeled with a. Since $H_{\mathfrak{A}}$ does not have a subgraph of type (i), this transition leads to q, i. e., it is a self loop. For the same reason, there is a self loop labeled with a around q'. We have a situation as in (ii), with the roles of the transitions from p to q and from p' to q' in (ii) being played by the self loops around q and q' respectively—a contradiction.

The argument is similar for (iv). ∎

The characterization of LTL($\bigcirc, \Diamond$) is stated in the following theorem.

Theorem 6.3 (Cohen *et al.*, 1993) *For a minimized deterministic finite automaton $\mathfrak{A}$, the following are equivalent:*

(A) $\mathfrak{A}$ recognizes the reverse of a LTL($\bigcirc, \Diamond$)-definable set of finite strings.

(B) The closure graph $H_{\mathfrak{A}}$ does not have a subgraph of type (v) displayed in Figure 4, where x, y, and z denote arbitrary non-empty strings, and p and q distinct states.

The proofs of Theorems 6.1 and 6.3 are given below.

As was observed already by Cohen et al., it can be effectively determined whether $H_{\mathfrak{A}}$ has a subgraph of type (v): Consider the labeled digraph G' whose vertex set is $Q \times Q$ and which has an edge labeled a from (p, q) to (p', q') if $\delta(p, a) = p'$ and $\delta(q, a) = q'$. The graph $H_{\mathfrak{A}}$ has a subgraph of type (v) if and only if there are distinct states p and q belonging to the same SCC of $G_{\mathfrak{A}}$ and such that (p, q) is part of a cycle in G'. This can be checked in time $\mathcal{O}((|Q| \times |A|)^2)$.

Corollary 6.4 *For LTL formulas over strings, there is an algorithm that given a LTL formula outputs*

1. *an equivalent LTL($\Diamond$) formula if there is one and "no" otherwise, and*
2. *an equivalent LTL($\bigcirc, \Diamond$) formula if there is one and "no" otherwise (Cohen* et al.*, 1993).*

Proof. To decide whether a given LTL formula φ is equivalent to a LTL($\Diamond$) or LTL($\bigcirc, \Diamond$) formula, we proceed as follows: first, we construct an automaton $\mathfrak{A}$ that accepts the strings satisfying φ (see, e. g., (Vardi and Wolper, 1986)); second, we reverse the transitions in $\mathfrak{A}$ and exchange initial and final states to obtain a non-deterministic automaton $\mathfrak{B}$ recognizing the reverse of $L(\mathfrak{A})$; third, we apply a subset construction and a minimization procedure to $\mathfrak{B}$ to get a minimized deterministic finite automaton $\mathfrak{C}$ recognizing the reverse of the set of strings defined by φ; finally, we check whether $H_{\mathfrak{C}}$ has a subgraph of type (i) or (ii), or (iii), respectively.

To see that an appropriate LTL($\Diamond$) or LTL($\bigcirc, \Diamond$) formula can be computed, note that all LTL($\Diamond$) and LTL($\bigcirc, \Diamond$) formulas can be enumerated effectively and that the equivalence of LTL formulas is decidable (see, e. g., (Sistla and Clarke, 1985)). A trivial algorithm for constructing an equivalent formula is thus given by the following rules: on input a LTL formula φ, first check whether φ is equivalent to a LTL($\Diamond$) (respectively LTL($\bigcirc, \Diamond$)) formula. If this is not the case, answer "no". Else enumerate all LTL($\Diamond$) (respectively LTL($\bigcirc, \Diamond$)) formulas and for each such formula ψ check whether φ and ψ are equivalent and if this is so, output ψ. ∎

Theorem 6.1 can alternatively be viewed as a theorem about LTL($\Diamond$)-formula equivalence (game equivalence). Let k be a fixed natural number. Let $u \equiv_k^{\Diamond} v$ mean that u and v satisfy the same LTL($\Diamond$)-formulas of operator depth at most k.

The relation $\equiv_k^{\Diamond}$ is an equivalence relation and is in fact a left congruence: if $u \equiv_k^{\Diamond} v$ then $wu \equiv_k^{\Diamond} wv$, for every word w. We can thus define a labeled digraph $G_k^{\Diamond}$ whose vertices are the equivalence classes of $\equiv_k^{\Diamond}$ and which has, for every node $u/{\equiv_k^\Diamond}$ and every letter a, an edge labeled a from $u/{\equiv_k^\Diamond}$ to $(au)/{\equiv_k^\Diamond}$. Theorem 6.1 can now be viewed as saying that $G_k^\Diamond$ has no subgraphs of type (i) or (ii) and that every graph without subgraphs of type (i) or (ii) is a homomorphic image of $G_k^\Diamond$, for some k. An analog statement based on Theorem 6.3 holds for LTL($\bigcirc, \Diamond$).

Before we get to the details of the proofs of Theorems 6.1 and 6.3, we explain the general strategy. In order to show the (more difficult) implications (B) $\Rightarrow$ (A) in Theorems 6.1 and 6.3, assume $\mathfrak{A} = (Q, q_I, \delta, F)$ is a finite automaton that recognizes the reverse of a property L we are interested in and u and v are such that $\delta^*(q_I, u^\rho) \neq \delta^*(q_I, v^\rho)$. We want to show that Player I has a winning strategy in the respective game (either the LTL($\Diamond$) or LTL($\bigcirc, \Diamond$) game) on u and v, where the number of rounds in the game depends only on the size of $\mathfrak{A}$. Player I looks at the runs of $\mathfrak{A}$ on u^ρ and v^ρ, which we will think of as paths through $G_\mathfrak{A}$, say $p_m \stackrel{e_m}{\to} p_{m-1} \dots p_1 \stackrel{e_1}{\to} p_0$ is the run of $\mathfrak{A}$ on u^ρ and $q_n \stackrel{f_n}{\to} q_{n-1} \dots q_1 \stackrel{f_1}{\to} q_0$ is the run of $\mathfrak{A}$ on v^ρ.

We say that a pair (e, f) of edges is *separating* if their target states are distinct and their labels are the same. Similarly, a configuration (i, j) of the game on u and v is called separating if (e_i, f_j) is separating. We will define a partial order relation $\succ$ (more precisely: one for the proof of Theorem 6.1 and another one for the proof of Theorem 6.3) on all separating pairs and show that Player I has a strategy satisfying:

(*) If (i, j) is a separating configuration and (i', j') is a configuration reachable from (i, j) in a round in which Player I plays according to his strategy, then (i', j') is separating and $(e_i, f_j) \succ (e_{i'}, f_{j'})$.

This ensures that Player I wins the game in at most as many rounds as the number of edges in $G_{\mathfrak{A}}$ squared. More precisely, Player I wins the game in at most as many rounds as the length of a longest chain with respect to $\succ$.

In order to prove the other (simpler) implications (A) $\Rightarrow$ (B) we will show that given a minimized automaton that does not have the specified structural characteristics and an arbitrary number k, there exist two words u_k and v_k such that $u_k \in L$ and $v_k \notin L$, but Player II wins the k-round game of the respective game on u_k and v_k.

We need some graph-theoretic notation and terminology. The source state, target state, and label of an edge e are denoted by σe, τe, and λe. We write $s \overset{+}{\rightsquigarrow} t$ if state t is reachable from state s, and $s \overset{*}{\rightsquigarrow} t$ if $s \overset{+}{\rightsquigarrow} t$ or $s = t$. Similarly, we write $e \overset{+}{\rightsquigarrow} f$ if $\tau e \overset{*}{\rightsquigarrow} \sigma f$, and $e \overset{*}{\rightsquigarrow} f$ if $e \overset{+}{\rightsquigarrow} f$ or $e = f$. We write $p \approx q$ if $p \overset{*}{\rightsquigarrow} q$ and $q \overset{*}{\rightsquigarrow} p$, i. e., if p and q belong to the same SCC.

Proof of Theorem 6.1. *(A) $\Rightarrow$ (B).* Write L for the reverse of $L(\mathfrak{A})$, the property we are interested in.

Assume first that $H_{\mathfrak{A}}$ has a subgraph of type (i). Since $\mathfrak{A}$ is minimized, there exists a string w_1 with $\delta^*(q_0, w_1) = p$ and a string w_2 such that either

- $\delta^*(q, w_2) \in F$ and $\delta^*(q', w_2) \notin F$, or
- $\delta^*(q, w_2) \notin F$ and $\delta^*(q', w_2) \in F$.

We choose (independently of k) $u_k = (w_1 a w_2)^\rho$ and $v_k = (w_1 a a w_2)^\rho$. Obviously, $u_k \in L$ and $v_k \notin L$, or $u_k \notin L$ and $v_k \in L$. We claim that Player II wins the k-round game with $\Diamond$-moves only on u_k and v_k. For this, we prove that Player II can play in such a way that after $k' \leq k$ rounds played the remaining strings u and v are either identical or there exists w such that $u = w a w_1^\rho$ and $v = w a a w_1^\rho$ (which proves the claim).

The claim is obviously true at the beginning of the game, i. e., when $k' = 0$. There is nothing to show when u and v are identical. So, for the rest, suppose after $k' < k$ rounds played the remaining strings u and v are of the form $u = waw_1^\rho$ and $v = waaw_1^\rho$. We distinguish several cases based on what Player I does in the next round. If Player I plays a point i in the common prefix w of u and v, Player II will respond by simply playing the same point in the other string. Similarly, if Player I plays a point in the common suffix w_1^ρ of u and v, Player II responds by playing the corresponding point in the other string. If Player I plays point $|w|$ on u, Player II will respond by playing $|w| + 1$ on v, and if Player I plays point $|w|$ or $|w| + 1$ on v, Player II will respond by playing point $|w|$ on u. This strategy clearly works.

Assume now that $H_{\mathfrak{A}}$ has a subgraph of type (ii). We can find w_1 and w_2 as above, and we choose $u_k = (w_1(xy)^k aw_2)^\rho$ and $v_k = (w_1(xy)^k xaw_2)^\rho$ (now depending on k). Again, it is obvious that $u_k \in L$ and $v_k \notin L$, or $u_k \notin L$ and $v_k \in L$. We claim that Player II wins the k-round game with $\Diamond$-moves only on u_k and v_k. To prove that, we show that Player II can play in such a way that after $k' \leq k$ rounds played the remaining strings u and v are identical or satisfy one of the following conditions:

(a) there exists a string w such that $u = w(w_1(xy)^k)^\rho$ and $v = w(w_1(xy)^k x)^\rho$, or

(b) there exists a number k'' and a suffix w of $(yx)^\rho$ such that $k'' \leq k'$ and $u = w(w_1(xy)^{k-k''})^\rho$ and $v = w(w_1(xy)^{(k-k'')+1})^\rho$.

As above the only interesting case is when $k' < k$ and the remaining strings u and v are not identical.

First, suppose u and v satisfy (a). If Player I plays a point in the common prefix w, Player II will respond by playing the corresponding point in the other word, and the remaining strings will again satisfy (a). If Player I plays any point $\geq |w|$ in u, Player II can respond in such a way that the remaining strings are identical. Also, if Player I plays a point $\geq |w|+|x|$ in v, Player II can respond in such a way that the remaining strings are identical. The case which is left is where Player I plays a point i in u with $|w| \leq i < |w|+|x|$. In this case, Player II will respond by playing the point $|w|+|y|+|x|$, resulting in remaining strings as described in (b).

Second, suppose u and v satisfy (b). Only if Player I plays a point $< |xy|$ on v, there is no obvious way for Player II to respond in such a way that the remaining strings are identical. But in this case, Player II simply responds by playing the same point on u, resulting in two remaining strings as in (b) with $k'-1$ instead of k' and $k''-1$ instead of k''.

(B) ⇒ (A). The partial order we use here, denoted $\succ_1$, is is defined as follows. We set $(e,f) \succ_1 (e',f')$ if

- $e' \overset{*}{\rightsquigarrow} e$, $f' \overset{*}{\rightsquigarrow} f$, and
- $\tau e \not\approx \tau e'$, $\sigma e \not\approx \sigma e'$, $\tau f \not\approx \tau f'$, or $\sigma f \not\approx \sigma f'$.

Obviously, $\succ_1$ is a partial order.

To complete the proof, we only have to find a strategy for Player I such that (*) holds (with $\succ_1$ instead of $\succ$).

Assume (i,j) is an arbitrary separating configuration. Let $s \geq i$ be minimal such that there exists a $t' \geq j$ with $p_s \approx q_{t'}$. Symmetrically, let $t \geq j$ be minimal such that there exists an $s' \geq i$ with $q_t \approx p_{s'}$. Obviously,

$p_s \approx q_t$. If we had both $s \leq i+1$ and $t \leq j+1$, we would have a situation exactly as depicted in one of (ii), (iii), or (iv). (View e_i and f_j as the edges labeled with a.) So $i+1 < s$ or $j+1 < t$. If $i+1 < s$, Player I plays position $s-1$; if $i+1 \geq s$, he plays position $t-1$. In either case, (*) is satisfied. ∎

Proof of Theorem 6.3. *(A) ⇒ (B).* Assume $H_{\mathfrak{A}}$ has a subgraph of type (v). Since $\mathfrak{A}$ is minimized, there exists a strings w_1 with $\delta^*(q_0, w_1) = p$ and a string w_2 such that either

- $\delta^*(p, w_2) \in F$ and $\delta^*(q, w_2) \notin F$, or
- $\delta^*(p, w_2) \notin F$ and $\delta^*(q, w_2) \in F$.

We choose u_k and v_k as follows:

$$\begin{aligned} u_k &= w_1(z^k x z^k y)^k z^k w_2^\rho \,, \\ v_k &= w_1(z^k x z^k y)^k z^k x z^k w_2^\rho \,. \end{aligned}$$

The claim is that Player II can play on u_k and v_k in such a way that after $k' \leq k$ rounds played, with configuration (i, j) reached, one of the following conditions holds:

(a) $i = j < |w_2| + k'|z|$, or

(b) $i, j \geq |w_2|$ and $|u_k| - i = |v_k| - j - |x| - |y| - 2|z| \geq |w_1| + (k-k')(|x| + |y| + 2|z|)$, or

(c) $|u_k| - i = |v_k| - j \leq |w_1| + k(|x| + |y| + 2|z|) + |z|$, i. e., the remaining strings are identical.

There is nothing to show if (c) holds. Before the game starts, i. e., when $k' = 0$, (a) holds. As long as Player I does not choose an $\Diamond$-move to a position $\geq |w_2|$, (a) can obviously be maintained. If (a) or (b) holds and Player I chooses an $\Diamond$-move to a position $\geq |w_2|$ in u_k, Player II can respond by playing in such a way that the remaining strings are identical, i. e., such that (c) holds. Player II can do so as well, if (a) holds and Player I moves to a position $\geq |w_2| + |x| + k|z|$ in v_k, or if (b) holds and Player I moves at least $|x| + |y| + k|z|$ positions ahead on v_k. The remaining case is when (b) holds and Player I moves less than $|x| + |y| + k|z|$ positions ahead on v_k. In this case, Player II will simply move the same number of positions ahead on u_k, maintaining (b). (Observe that this also captures the case where Player I chooses a next move.)

(B) ⇒ (A). The partial order we use here, denoted $\succ_2$, is defined as the transitive closure of $\succ_1 \cup \rightarrow$ where $\succ_1$ is the partial order relation from the previous proof and $\rightarrow$ is defined by: $(e, f) \rightarrow (e', f')$ if

- $\lambda e' = \lambda f'$, $\sigma e = \tau e'$ and $\sigma f = \tau f'$, and
- there exist $p \in \{\sigma e, \tau e\}$ and $q \in \{\sigma f, \tau f\}$ such that $p \approx q$.

The transitive closure of $\succ_1 \cup \rightarrow$ is in fact a partial order: if $\succ_2$ had a cycle, there would be a sequence $(e^1, f^1) \rightarrow \cdots \rightarrow (e^l, f^l) \rightarrow (e^1, f^1)$, which would imply a situation as depicted in (v), with $p = \tau e^1$, $q = \tau f^1$, and $z = \lambda e^l \lambda e^{l-1} \ldots \lambda e^1$.

In order to establish (*) assume that (i, j) is a separating configuration, and let s and t be defined as in the proof of Theorem 6.1. If $i + 1 < s$ or $j + 1 < t$, Player I makes an $\Diamond$-move just as in the case for LTL($\Diamond$). If

$s \leq i+1$ and $t \leq j+1$, Player I makes a $\bigcirc$-move. This is possible as we would otherwise have $p_{i+1} = q_I = q_{j+1}$—a contradiction to that $\mathfrak{A}$ is deterministic. By definition of $\succ_2$, it is clear that (*) is satisfied in both cases. ∎

It is easy to see that $2|Q|$ and $|Q|^2$ are upper bounds for chains with respect to $\succ_1$ and $\succ_2$. We therefore have:

Corollary 6.5 *Let $\mathfrak{A}$ be a deterministic automaton with k states and L the reverse of $L(\mathfrak{A})$.*

1. *If $H_{\mathfrak{A}}$ does not have subgraphs of type (i) and (ii), then L is definable by a LTL($\Diamond$) formula of operator depth at most $2k$.*

2. *If $H_{\mathfrak{A}}$ does not have subgraphs of type (v), then L is definable by a LTL($\bigcirc, \Diamond$) formula of operator depth at most k^2.*

We can use Corollary 6.5 to determine an explicit bound on the running time of the algorithm sketched in the proof of Corollary 6.4, and the winning strategies constructed in the proofs of Theorems 6.1 and 6.3 can be used to improve the efficiency of the algorithm itself.

7 Conclusion

We have defined an EF game for temporal logic, and we have used it to prove a strict hierarchy theorem for the Until depth of temporal properties, independent of the depth in other temporal operators. We have seen that an EF game designed explicitly for temporal logic permits much simpler and much finer analysis of expressibility in the logic than does the common practice of translating from temporal to first-order formulas and using results about first-order expressibility.

We have used the EF games in a new way on finite automata to show decidability of expressibility in fragments of temporal logic. Recently, Thérien and Wilke (1996) have extended these results to show that the minimal Until depth required for expressing a given temporal property on strings or ω-words is decidable. Their proof for strings uses sophisticated semigroup-theoretic techniques, while the extension to ω-words makes crucial use of our EF game for LTL.

An interesting question is whether the techniques of Section 6 can be used to provide an alternative proof, without using semigroups, that counter-free automata accept LTL definable languages (Schützenberger, 1965, McNaughton and Papert, 1971, and Kamp, 1968). We would also like to know whether the technique for finite automata developed in Section 6 can be adapted to work for ω-automata.

Acknowledgment. Thanks to Eric Allender, Neil Immerman, Alexander Rabinovich, Wolfgang Thomas, and Moshe Vardi for helpful discussions and comments.

References

Cohen, J., Perrin, D., and Pin, J.-E. (1993), On the expressive power of temporal logic, *J. Comput. System Sci.* **46**(3), 271–294.

Ehrenfeucht, A. (1961), An application of games to the completeness problem of formalized theories, *Fund. Math.* **49** 129–141, 1961.

Emerson, E. A. (1990) Temporal and modal logics, *in* "Handbook of Theoret. Comput. Sci.," vol. B (J. v. Leeuwen, Eds.), pp. 995–1072, Elsevier, Amsterdam.

Fraïssé, R. (1954), Sur quelques classifications des systèmes de relations, *Publ. Sci. Univ. Alger. I.* **1** 35–182.

Gabbay, D., Hodkinson, I., and Reynolds, M. (1994), "Temporal Logic," vol. 1, Clarendon Press, Oxford.

Gabbay, D., Pnueli, A., Shelah, S., and Stavi, J. (1980), On the temporal analysis of fairness, *in* "Proceedings, 7th Annual ACM Symposium on Principles in Programming Languages," pp. 163–173.

Immerman, N. and Kozen, D. (1989), Definability with bounded number of bound variables, *Inform. and Comput.* **83** 121–139.

Kamp, H. (1968), "Tense logic and the theory of linear order," Ph. D. thesis, Univ. of California, Los Angeles.

Manna, Z. and Pnueli, A. (1992), "The Temporal Logic of Reactive and Concurrent Systems," Springer, New York.

McNaughton, R. and Papert, S. (1971), "Counter-Free Automata,' MIT Press, Cambridge, Mass.

Rabinovich, A. (1996), personal communication.

SCHÜTZENBERGER, M. P. (1965), On finite monoids having only trivial subgroups, *Inform. and Control* **8** 190–194.

SISTLA, A. P. AND CLARKE, E. M. (1985), The complexity of propositional linear temporal logics, *J. ACM* **32** (3) 733–749.

SISTLA, A. P. AND ZUCK, L. D. (1993), Reasoning in a restricted temporal logic, *Inform. and Comput.* **102** 167–195.

STOCKMEYER, L. J. (1974), "The complexity of decision problems in automata theory and logic," Ph. D. thesis, MIT, Cambridge, Mass.

THÉRIEN, D. AND WILKE, TH. (1996), Temporal logic and semidirect products: An effective characterization of the until hierarchy, *in* "Proceedings, 37th Annual Symposium on Foundations of Computer Science," pp. 256–263.

THOMAS, W. (1982), Classifying regular events in symbolic logic, *J. Comput. System Sci.* **25** 360–376.

THOMAS, W. (1993), On the Ehrenfeucht-Fraïssé game in theoretical computer science, *in* "Tapsoft '93," (M. Gaudel and J. P. Jouannaud, Eds.), Lect. Notes Comput. Sci., vol. 668, Springer, Berlin, pp. 559–568.

VARDI, M. Y. AND WOLPER, P. (1986), An automata-theoretic approach to automatic program verification, *in* "Proceedings, First IEEE Symposium in Logic in Comput. Sci.," pp. 322–331.

Figure 1

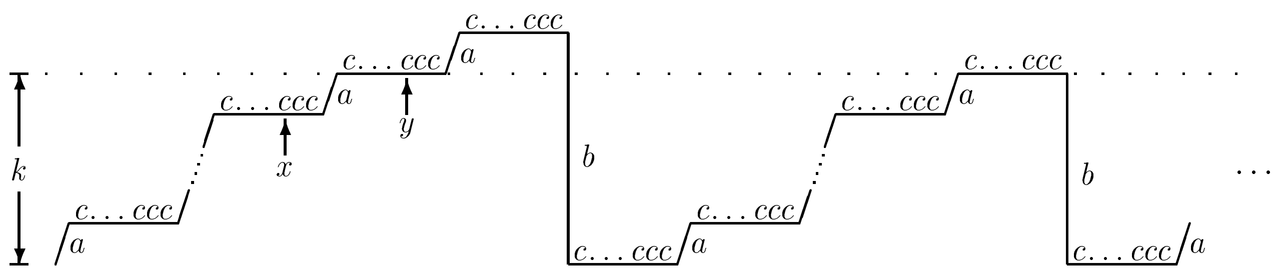


Figure 2

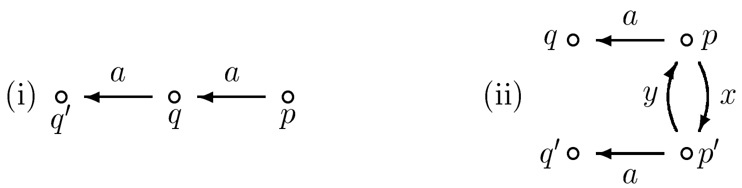
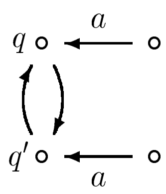


Figure 3

(iii)



(iv)

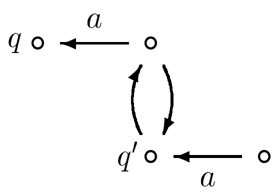
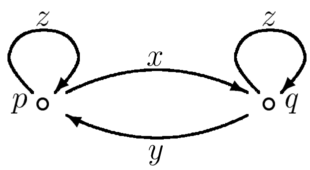


Figure 4

(v) 

Captions for Figures

caption for Figure 1: "staircase" word $v_{k,r}$

caption for Figure 2: forbidden subgraphs for LTL($\diamond$)

no caption for Figure 3

caption for Figure 4: forbidden subgraphs for LTL($\bigcirc, \diamond$)