

“An $n!$ Lower Bound On Formula Size”

Micah Adler

Computer Science Dept.
UMass, Amherst, USA
<http://www.cs.umass.edu/~micah>

Neil Immerman*

Computer Science Dept.
UMass, Amherst, USA
<http://www.cs.umass.edu/~immerman>

Abstract

We introduce a new Ehrenfeucht-Fraïssé game for proving lower bounds on the size of first-order formulas. Up until now such games have only been used to prove bounds on the operator depth of formulas, not their size. We use this game to prove that the CTL^+ formula $\text{Occur}_n \equiv \mathbf{E}[\mathbf{F}p_1 \wedge \mathbf{F}p_2 \wedge \dots \wedge \mathbf{F}p_n]$ which says that there is a path along which the predicates p_1 through p_n occur in some order, requires size $n!$ to express in CTL. Our lower bound is optimal. It follows that the succinctness of CTL^+ with respect to CTL is exactly $\Theta(n)!$. Wilke had shown that the succinctness was at least exponential [Wil99].

We also use our games to prove an optimal $\Omega(n)$ lower bound on the number of boolean variables needed for a weak reachability logic (\mathcal{RL}^w) to polynomially embed the language LTL. The number of booleans needed for full reachability logic \mathcal{RL} and the transitive closure logic $\text{FO}^2(\text{TC})$ remain open [IV97, AI00].

1 Introduction

We introduce a new Ehrenfeucht-Fraïssé game for proving lower bounds on the size of first-order formulas. Previous such games only proved lower bounds on the quantifier depth of formulas.

We use this game to prove that the CTL^+ formula,

$$\text{Occur}_n \equiv \mathbf{E}[\mathbf{F}p_1 \wedge \mathbf{F}p_2 \wedge \dots \wedge \mathbf{F}p_n] \quad (1.1)$$

requires size $n!$ to express in CTL. The formula Occur_n says that there exists a path such that each of the predicates p_i occurs somewhere along this path. (\mathbf{E} is the existential path quantifier: there exists a maximal path starting from the current point. \mathbf{F} is the modal quantifier: somewhere now or in the future along the current path.)

This offers a quite different proof and improves the exponential lower bound on the succinctness of CTL compared

with CTL^+ [Wil99]. We thus prove that the succinctness of CTL^+ with respect to CTL is exactly $\Theta(n)!$.

We prove that the parse tree of any CTL formula expressing Occur_n has at least $n!$ leaves. This bound is exactly optimal because the following formula expresses Occur_n and has $n!$ leaves in its parse tree. Here we use $[n]$ to denote $\{1, 2, \dots, n\}$.

$$\varphi_n \equiv \bigvee_{i_1 \in [n]} \mathbf{EF} \left(p_{i_1} \wedge \bigvee_{i_2 \in [n] - \{i_1\}} \mathbf{EF} \left(p_{i_2} \wedge \bigvee_{i_3 \in [n] - \{i_1, i_2\}} \mathbf{EF} (\dots \wedge \mathbf{EF} p_{i_n}) \dots \right) \right)$$

The main contribution of these results is not so much the introduction of the new formula-size games, as their effective use proving a new and optimal result. Standard Ehrenfeucht-Fraïssé games are played on a single pair of structures \mathcal{A}, \mathcal{B} . They are used to prove lower bounds on the quantifier depth of a formula φ needed to distinguish \mathcal{A} from \mathcal{B} . Our new game works on a whole set of structures A, B where all of A satisfies φ and all of B satisfies $\neg\varphi$. In a standard game, the pair of structures \mathcal{A} and \mathcal{B} may differ on a disjunction: $\varphi \equiv \alpha \vee \beta$. In this case they differ on α or they differ on β and the “or” may be discarded. However, in the formula-size game, the set of structures A must be split into two portions: A_1 satisfying α and A_2 satisfying β . All of B satisfies $\neg\alpha$ and $\neg\beta$. Thus the game on (A, B) is shifted to a pair of games, (A_1, B) and (A_2, B) .

There are extensive connections between the computational complexity of a problem and its descriptive complexity, i.e., how complex a formula is needed to describe the problem. Descriptive complexity is measured via the size, number of variables, operator depth, etc. of the requisite formulas as a function of the size of the input structures being described [Imm99].

The formula-size games introduced here generalize standard EF games. They are also related to the communication complexity games that Karchmer and Wigderson used

*Research supported by NSF grant CCR-9877078.

to prove lower bounds on the depth of monotone circuits [Kar89]¹. In the past, EF games have been useful in proving bounds on operator depth and number of variables, but they have not been used to prove lower bounds on the size of formulas. This has been a crucial lack, which the present paper takes a step in correcting.

The added complication of formula-size games means that we must build up considerable machinery to use them to prove lower bounds. Such lower bounds were heretofore unattainable for general first-order formulas. We believe that this game and the corresponding methods will have many applications.

In another application of formula-size games we show that $\Theta(n)$ booleans are needed to translate an LTL formula of size n to a polynomial-size formula of the reachability logic, \mathcal{RL}^w .

This paper is organized as follows: In §2 we provide the necessary background in logic including the introduction of transitive closure logic (FO(TC)) which provides the general setting for the games that we present. In §3 we review Ehrenfeucht-Fraïssé games and present the new formula-size games for FO(TC). In §4 we present the formula-size game for CTL. In §5 we define the graphs G_n over which we prove our lower bound. In §6 we prove our main result, the optimal $n!$ lower bound on the succinctness of CTL^+ with respect to CTL. In §7 we prove an $\Omega(n)$ lower bound on the number of boolean variables needed for \mathcal{RL}^w to express Occur_n in polynomial size. In Appendix A we describe the language CTL and in Appendix B we describe reachability logic (\mathcal{RL}).

2 Background

In this section we review some basic definitions concerning finite model theory and transitive closure logic [Imm99].

The language \mathcal{L} consists of first-order logic with unary relation symbols $\{p_n : n \in \mathbf{N}\}$, and binary relation symbol, R . By the *size* of a formula, we mean the number of nodes in its parse tree, i.e., the number of occurrences of logical connectives, quantifiers, operators, and atomic symbols.

For our purposes, a Kripke structure is a finite labeled graph:

$$\mathcal{K} = \langle S; p_n^{\mathcal{K}} : n \in \mathbf{N}; R^{\mathcal{K}} \rangle \quad (2.1)$$

¹Karchmer and Wigderson gave general games for proving lower bounds on circuit depth; but they proved lower bounds only using a monotone version of their games. They cast their games as a communication game in which two sets of structures differ on some property. Through successive bits of communication, each of which divides one of the sets of structures in half, eventually the sets are reduced to a collection of pairs where each pair differs on a particular bit. This is analogous to the closed nodes of our formula size game, in which each pair differs on a particular atomic formula.

where S is the set of states (vertices), each $p_n^{\mathcal{K}} \subseteq S$ is a unary relation on S , and $R^{\mathcal{K}} \subseteq S^2$ is the edge relation.

First-order logic \mathcal{L} does not suffice to express such simple formulas as,

$$\text{“There is a path from where we are } (x) \text{ to a vertex where } p_{17} \text{ holds.”} \quad (2.2)$$

For this reason we add a transitive closure operator to first-order logic to allow us to express reachability [Imm87].

Let the formula $\varphi(x_1, \dots, x_k, y_1, \dots, y_k)$ represent a binary relation on k -tuples. We express the reflexive, transitive closure of this relation using the transitive-closure operator (TC), as follows: $\text{TC}_{\bar{x}, \bar{y}}\varphi$. Let FO(TC) be the closure of first-order logic under the transitive-closure operator. For example, the following formula expresses Equation 2.2: $(\exists y)[(\text{TC}_{x,y}R(x,y))(x,y) \wedge p_{17}(y)]$.

3 Ehrenfeucht-Fraïssé Games

We assume that the reader is somewhat familiar with classical Ehrenfeucht-Fraïssé (EF) games [Ehr61, Fra54, Imm99]. Typically there is a pair of structures \mathcal{A}, \mathcal{B} and two players. Samson chooses vertices, trying to point out a difference between the two structures, and Delilah replies, trying to keep them looking the same. Typical games have a certain number of pebbles corresponding to variables, and rounds corresponding to the depth of nesting of quantifiers and other operators such as TC.

The typical fundamental theorem of EF games is that Delilah has a winning strategy for the k -pebble, m -move game on \mathcal{A}, \mathcal{B} iff \mathcal{A} and \mathcal{B} agree on all k -variable, depth- m formulas. EF games are used to show nonexpressivity of a property Φ as follows: Delilah chooses a pair of structures \mathcal{A}, \mathcal{B} that disagree on Φ but such that she has a winning strategy for the m -move, k -pebble game. It then follows that Φ is not expressible via a k -variable, depth- m formula.

We now present new games for proving lower bounds on formula size rather than depth. We first define the formula-size game for the language $\text{FO}^2(\text{TC})$ — first-order logic with 2 variables and the transitive closure operator. We chose this logic because it is simple, expressive, and quite general. It is easy to see how to generalize the game and its corresponding fundamental theorem to most reasonable logics by adding more variables and other operators. In the sequel we will specialize the $\text{FO}^2(\text{TC})$ game to a less general language, CTL, where we will prove our main results.

Definition 3.1 (FO²(TC) Formula-Size Game) In the formula-size game, Delilah starts by picking two finite sets of structures: A_0, B_0 . The root of the game tree is labeled A_0, B_0 . (The intuitive idea is that there is some property Φ such that every structure in A_0 satisfies Φ ($A_0 \models \Phi$) and no structure in B_0 satisfies Φ ($B_0 \models \neg\Phi$).)

At each move, Samson may play on any of the open leaves of the current game tree. (One of Samson’s possible moves will be to close a leaf.) Suppose that the leaf that Samson chooses to play on is labeled with the pair of sets A, B .

“not” move: Samson switches the two sets letting the current leaf have a unique child labeled B, A .

“or” move: Samson splits A into two sets: $A = A' \cup A''$. He lets the current leaf have two children labeled A', B and A'', B .

\exists move: Samson chooses a variable $v \in \{x, y\}$. He then assigns a value for v to every structure $\mathcal{A} \in A$. Delilah then answers by assigning a value for v to every structure $\mathcal{B} \in B$. Let A', B' be the two sets of structures with the new assignments for v . The current leaf is then given a child labeled A', B' .

TC move: Samson chooses a pair of previously assigned variables $v, v' \in \{x, y\}$. For every structure $\mathcal{A} \in A$, Samson then chooses a sequence of vertices from \mathcal{A} : $v^{\mathcal{A}} = a_0, a_1, a_2, \dots, a_t = v'^{\mathcal{A}}$. Delilah then answers by choosing for every structure $\mathcal{B} \in B$ a similar sequence, $v^{\mathcal{B}} = b_0, b_1, b_2, \dots, a_{t'} = v'^{\mathcal{B}}$. Samson then chooses a single consecutive pair b_i, b_{i+1} for each \mathcal{B} and assigns x to b_i and y to b_{i+1} . The current leaf is then given a child labeled A', B' where B' is the result of these new assignments for each structure in B . A' consists of multiple copies of each structure $\mathcal{A} \in A$, one for each consecutive pair a_j, a_{j+1} in the sequence for \mathcal{A} chosen by Samson and with x assigned to a_j and y assigned to a_{j+1} .

The idea behind this move is that Samson is asserting that every structure in A satisfies $\text{TC}_{x,y}(\delta)(v, v')$ and no structure in B does. He thus presents what he claims is a δ -path from v to v' for each structure \mathcal{A} in A . Delilah answers with a supposed δ path from v to v' for every \mathcal{B} in B . Samson must then challenge one pair b_i, b_{i+1} in each of Delilah’s supposed δ paths. He is in effect saying “ $\neg\delta(b_i, b_{i+1})$ ”. At the end of this move, every structure in A' should satisfy $\delta(x, y)$ and no structure in B' should.

atomic move: Samson chooses $v, v' \in \{x, y\}$ and an atomic formula $\alpha(v, v')$. (α can be $v = v', R(v, v')$ or $p_i(v)$.) Samson can only make this move if every structure in A satisfies $\alpha(v, v')$ and no structure in B does. In this case, the current leaf is *closed*.

The object of the game for Samson is to close the whole game tree while keeping it as small as possible. Delilah on the other hand wants to make the tree as large as possible.

Delilah may make multiple copies of the structures in B before any of her moves. For this reason, there is an obvious strategy for Delilah that is optimal, namely do everything: in answer to an existential move, make a copy of \mathcal{B} for each vertex in \mathcal{B} and reply with that vertex. Similarly, in answer to a TC-move, Delilah can make enough copies of \mathcal{B} and answer with all possible sequences without repetitions from v to v' . \square

The reason that Delilah is allowed to make multiple copies in the size game is that otherwise Samson need not play relevant parts of the minimal formula separating A and B . For example, suppose that $A = \{\mathcal{A}\}$ and $B = \{\mathcal{B}\}$ each consist of a single structure. Suppose that the smallest formula true of \mathcal{A} but not \mathcal{B} is,

$$\begin{aligned} \exists x \exists y (p_1(x) \leftrightarrow p_1(y) \wedge p_2(x) \leftrightarrow p_2(y) \wedge \\ \dots \wedge p_n(x) \leftrightarrow p_n(y)), \end{aligned}$$

i.e., \mathcal{A} has two points agreeing on all n predicate symbols, but \mathcal{B} does not. If Delilah could not make duplicates, then Samson could just choose the relevant x and y in \mathcal{A} and Delilah would have to answer with a single pair from \mathcal{B} . Then either the x ’s or the y ’s would differ on some predicate symbol p_i and Samson could close a game tree of size 3, rather than n .

The fundamental theorem of the formula-size game is:

Theorem 3.2 *Samson can close the game started at A_0, B_0 in a tree of size s iff there is a formula $\varphi \in \text{FO}^2(\text{TC})$ of size at most s such that every structure in A_0 satisfies φ and no structure in B_0 does.*

Proof: Suppose that φ of size s separates A_0 and B_0 . Then Samson can “play φ ” and a closed game tree of size s will result. Playing φ means the following. Suppose that $A \models \varphi$ and $B \models \neg\varphi$.

$\varphi = \neg\psi$: Samson plays the “not” move. In the resulting leaf $A' \models \psi$ and $B' \models \neg\psi$.

$\varphi = \psi \vee \rho$: Samson plays the “or” move letting A' be the subset of A satisfying ψ , and A'' the subset satisfying ρ . Thus one child differs on ψ and the other differs on ρ .

$\varphi = (\exists v)\psi$: Samson plays the \exists move assigning v to a witness for ψ in every structure of A . Thus, whatever Delilah answers we have that $A' \models \psi$ and $B' \models \neg\psi$.

$\varphi = \text{TC}_{x,y}(\delta)(v, v')$: Samson plays the TC move and as argued in the discussion after the definition of this move, $A' \models \delta$ and $B' \models \neg\delta$.

φ is atomic: Samson plays the atomic move, using φ and succeeds in closing this leaf.

Conversely, suppose that Samson has succeeded in closing the game in size s and that Delilah has played optimally. It follows that the resulting game tree is a size s formula satisfied by all of A_0 and none of B_0 .

This can be seen inductively from the leaves of the closed game tree. For closed leaf, (A, B) , $A \models \alpha$ and $B \models \neg\alpha$, where α is an atomic formula, i.e., has size one.

Inductively, assume that (A, B) has children (A_i, B_i) each differing on a formula ψ_i of size s_i where $i = 1$ for “not”, \exists and TC moves and $i = 1, 2$ for the “or” move. Here s_i is the size of the subtree rooted at (A_i, B_i) .

“not” move: $A \models \neg\psi_1, B \models \psi_1$ and thus they differ on a formula of size $s_1 + 1$.

“or” move: $A \models \psi_1 \vee \psi_2, B \models \neg(\psi_1 \vee \psi_2)$ and thus they differ on a formula of size $s_1 + s_2 + 1$.

\exists move: $A \models (\exists v)\alpha_1, B \models \neg(\exists v)\alpha_1$, and thus they differ on a formula of size $s_1 + 1$. Note that since Delilah plays optimally, if it were the case that some $\mathcal{B} \in B$ satisfies $(\exists v)\alpha_1$, then Delilah would have chosen the appropriate witness for this \mathcal{B} and it would not have been the case that $B_1 \models \neg\alpha_1$.

TC move: $A \models \text{TC}_{x,y}(\alpha_1)(v, v'), B \models \neg\text{TC}_{x,y}(\alpha_1)(v, v')$, and thus they differ on a formula of size $s_1 + 1$. By the definition of the TC move, since $A_1 \models \alpha_1$, we know that for every $\mathcal{A} \in A$, there is an α_1 -path from $v^{\mathcal{A}}$ to $v'^{\mathcal{A}}$. Furthermore, if there were an α_1 -path from $v^{\mathcal{B}}$ to $v'^{\mathcal{B}}$, for some $\mathcal{B} \in B$, then Delilah would have played it for one of her copies of \mathcal{B} . Therefore, no matter which consecutive pair in this path Samson challenged, it would satisfy α_1 .

Thus A_0 and B_0 differ on a formula of size s . \square

4 Definition of the CTL Game

For a definition of CTL see the appendix or [CGP99]. We now define the CTL formula-size game². This is a restriction of the FO²(TC) formula-size game (Definition 3.1) as follows.

- There is only a single pebble name: x .
- The “not” and “or” moves are unchanged.
- The atomic move is unchanged except that it is played only using atomic formulas p_i .

²It is easy to generalize this also to the CTL* formula-size game, but we leave this to the reader.

- The \exists and TC moves are replaced by the following, played on a leaf, ℓ , labeled with the pair of sets A, B ,

EX move: For each $\mathcal{A} \in A$, Samson reassigns x to a child of the current x . Delilah answers by first making as many copies of each $\mathcal{B} \in B$ as she wishes. For each copy $\mathcal{B} \in B$ she assigns x to a child of the current x . The resulting node labeled A', B' becomes the only child of ℓ .

EU move: For each $\mathcal{A} \in A$, Samson chooses a path of length zero or more: $x^{\mathcal{A}} = a_0, a_1, \dots, a_r$. Delilah answers as above with a path $x^{\mathcal{B}} = b_0, b_1, \dots, b_s$ for each copy she makes of each $\mathcal{B} \in B$. Samson is trying to assert that $(A, x) \models \mathbf{E}(\alpha\mathbf{U}\beta)$, i.e., that $(A, a_i) \models \alpha$ for $i < r$, and $(A, a_r) \models \beta$. Presumably this holds for all of Samson’s chosen paths and none of Delilah’s.

In the second half of the move, Samson divides the paths chosen by Delilah into two sets. For the first set he assigns x to some b_i with $i < s$ and puts these structures into B_1 . For the second set he assigns x to b_s and puts these structures into B_2 . Delilah answers by making enough copies so that she can assign x to each possible point in Samson’s paths. When she assigns x to the final point b_r in a path, she puts that structure in A_2 . When she assigns x to a non-final point she puts that structure into A_1 . The node ℓ now has two children labeled A_1, B_1 and A_2, B_2 respectively.

Intuitively what has happened in the second half of this move is that for those paths chosen by Delilah whose final points do not satisfy β , Samson chooses this point and puts the structure into B_2 . For those paths one of whose non-final points does not satisfy α , Samson chooses this point and puts the structure into B_1 . At the end of the move we have that $A_1 \models \alpha, B_1 \models \neg\alpha, A_2 \models \beta, B_2 \models \neg\beta$. If the set B_1 or B_2 should happen to be empty then that node is considered closed.

AU move: This is similar to the **EU** move except that the first half of the move now has two parts: (a) Samson chooses a maximal path for each structure in B , and Delilah makes copies and chooses a maximal path for each copy of each structure in A ; (b) Samson chooses a finite initial segment of each path chosen by Delilah and then Delilah chooses a finite initial segment of each path chosen by Samson. Delilah may make copies of the paths chosen by Samson in order to choose more than one initial segment from each path. The second half of the move is the same as for the **EU** move.

It should be clear from the above definition and the proof of Theorem 3.2 that the following theorem holds:

Theorem 4.1 *Samson can close the CTL formula-size game started at A_0, B_0 in a tree of size s iff there is a formula $\varphi \in \text{CTL}$ of size at most s such that every structure in A_0 satisfies φ and no structure in B_0 does.*

5 Setting Up the Playing Field

In this section we describe the graphs on which we will play the CTL game to prove our main lower bound, Theorem 6.1. For each $n > 1$, we will build two sets of graphs A_0, B_0 such that $A_0 \models \text{Occur}_n$ and $B_0 \models \neg \text{Occur}_n$. For each of the $n!$ possible paths that might satisfy Occur_n , A_0 will include one graph that contains this path. Furthermore, we give each graph in A_0 and B_0 copies of all permutations of length $n - 1$. This will help make A_0 and B_0 very difficult to distinguish.

For any fixed $n > 1$ consider the following directed graph, $G_n = (V_n, E_n)$. Let $\Pi_{[n]}$ be the set of all permutations π on any nonempty subset of $[n]$ and let Π_n be the set of permutations on the full set $[n]$. The vertices of G_n consist of the union of two sets, $V_n = T_n \cup F_n$,

$$T_n = \{t_\pi \mid \pi \in \Pi_{[n]}\}; \quad F_n = \{f_\pi \mid \pi \in \Pi_{[n]}\}$$

We represent the permutation $\pi \in \Pi_{[n]}$ as a 1:1 map,

$$\pi : [\text{rng}(\pi)] \rightarrow \text{rng}(\pi) \subseteq [n].$$

For any such permutation π on at least two elements, define its tail, $\text{tail}(\pi) : [\text{rng}(\pi) - 1] \rightarrow \text{rng}(\pi) - \{\pi(1)\}$ where $\text{tail}(\pi)(i) = \pi(i + 1)$. For ease of notation, let $\pi^2 = \text{tail}(\pi)$, and in general, $\pi^{k+1} = \text{tail}^k(\pi)$, i.e., the permutation π starting from item $k + 1$.

For all $\pi \in \Pi_{[n]}$, the relation $p_{\pi(1)}$ holds of vertex t_π . Also, if π is a permutation on at least two elements then $p_{\pi(1)}$ holds of vertex f_π .

The node f_π has edges to the following successors nodes:

- $t_\sigma \in T_n$ where $\text{rng}(\sigma) \subseteq \text{rng}(\pi) - \{j\}$, for some $j \in \text{rng}(\pi), j \neq \pi(1)$
- $f_\sigma \in F_n$ where $\text{rng}(\sigma) \subseteq \text{rng}(\pi) - \{j\}$, for some $j \in \text{rng}(\pi)$

The node t_π has edges to all the successors of f_π together with the additional successor t_{π^2} . Furthermore, every vertex in V_n has an edge back to itself.

Consider the following sets of vertices and structures,

$$\begin{aligned} Y_n &= \{t_\pi \in T_n \mid \pi \in \Pi_n\} \\ N_n &= \{f_\pi \in F_n \mid \pi \in \Pi_n\} \\ A_0 &= \{(G_n, t_\pi) \mid t_\pi \in Y_n\} \\ B_0 &= \{(G_n, f_\pi) \mid f_\pi \in N_n\} \end{aligned}$$

The idea behind G_n is that for each $\pi \in \Pi_n, t_\pi$ and f_π are very difficult to distinguish. However, observe that,

Lemma 5.1 *For any $\pi \in \Pi_n$,*

$$(G_n, t_\pi) \models \text{Occur}_n; \quad \text{but} \quad (G_n, f_\pi) \models \neg \text{Occur}_n$$

6 Playing the CTL Game

In this section we prove the following,

Theorem 6.1 *The formula Occur_n (Equation 1.1) cannot be expressed in a CTL formula of size less than $n!$. Thus, there is a CTL^+ formula of size $O(n)$ whose smallest equivalent CTL formula has size $n!$.*

Corollary 6.2 *The succinctness of CTL^+ with respect to CTL is exactly $\Theta(n)!$.³*

By Lemma 5.1 we have that $A_0 \models \text{Occur}_n$ and $B_0 \models \neg \text{Occur}_n$. To prove Theorem 6.1 it suffices to show the following,

Lemma 6.3 *Samson cannot close the CTL-game on (A_0, B_0) in a game tree with fewer than $n!$ leaves.*

We will prove Lemma 6.3 through a series of additional lemmas. Since there is only one structure namely G_n on which we are playing and the only thing that matters is where x is assigned, we will abbreviate the structure \mathcal{A} for which $x^{\mathcal{A}} = a$ by the point a . Thus a tree node will be labeled A, B with A and B both sets of vertices from G_n .

We say that a pair $\langle a, b \rangle$ occurs at a node v of a game tree if v is labeled (A, B) and $a \in A, b \in B$. The following lemma is obvious but useful:

Lemma 6.4 *If a pair $\langle a, a \rangle$ occurs anywhere in a game tree, then that tree can never be closed.*

Let \mathcal{T} be a closed game tree whose root is labeled (Y_n, N_n) and on which Delilah and Samson have both played perfectly. We will argue that \mathcal{T} has at least $n!$ leaves.

Lemma 6.5 *Let $\pi \in \Pi_n$. Then there is a branch in \mathcal{T} from root to leaf along which the following pairs occur (in this order),*

$$\langle t_\pi, f_\pi \rangle, \langle t_{\pi^2}, f_{\pi^2} \rangle, \langle t_{\pi^3}, f_{\pi^3} \rangle, \dots, \langle t_{\pi^n}, f_{\pi^n} \rangle$$

Proof: By definition of $Y_n, N_n, \langle t_\pi, f_\pi \rangle$ occurs at the root. Suppose inductively that $\langle t_{\pi^k}, f_{\pi^k} \rangle$ occurs at node v_k (and is preceded by $\langle t_{\pi^j}, f_{\pi^j} \rangle$ for all $j < k$); and v_k is the lowest node at which $\langle t_{\pi^k}, f_{\pi^k} \rangle$ occurs. If $k = n$, then the lemma is proved. Suppose that $k < n$. In this case, v_k is an

³See Emerson and Halpern [EH85] for the upper bound.

open node since t_{π^k} and f_{π^k} both satisfy the same predicate symbol, $p_{\pi(k)}$.

From now on, let us assume that there are no “not” moves, but that instead Samson may play on the left or on the right. This may slightly decrease the size of \mathcal{T} by removing “not” moves, but the number of leaves is unchanged. Note that an “or” move on the right is really an “and” move, and an **E** move on the right is really an **A** move.

Observe that if Samson plays an “or” move at v_k , then the pair $\langle t_{\pi^k}, f_{\pi^k} \rangle$ would still occur at one of v_k ’s children. Furthermore, Samson cannot close v_k . Thus, Samson must play one of the following moves: **EX**, **EU**, **AU**.

Recall that every path from f_{π} is also a path from t_{π} . Thus if Samson plays on the right, stepping off f_{π} to some descendant d , then t_{π} has the identical descendant d which Delilah will play. It follows from Lemma 6.4 that, Samson must play on the left at v_k .

If Samson plays **EX** then he must move from t_{π^k} to one of its successors. The only successor of t_{π^k} that is not a successor of f_{π^k} is $t_{\pi^{k+1}}$. Thus, Samson must move to $t_{\pi^{k+1}}$ and Delilah will move to all successors of f_{π^k} , including $f_{\pi^{k+1}}$. Thus $\langle t_{\pi^{k+1}}, f_{\pi^{k+1}} \rangle$ occurs in the child of v_k as desired.

Suppose that Samson plays **AU**. Samson starts by choosing a maximal path for each structure on the left. Delilah answers by choosing the infinite loop on the current vertex for each structure on the right. Recall that G_n has a self-loop at each vertex. Now, Samson chooses an initial segment of each infinite self-loop. Delilah responds by choosing the initial segments of length zero from Samson’s paths. The right child of v_k is thus labeled exactly the same as v_k . Thus it is not useful for Samson to play **AU**.

Finally, suppose that Samson plays **EU**. He chooses a path from t_{π^k} to some descendant d . Note that if $d \neq t_{\pi^{k+1}}$ then d is also a descendant of f_{π^k} . Thus Delilah will answer with the path consisting of a single step from f_{π^k} to d . If Samson challenges f_{π^k} then we have made no progress. If Samson challenges d , then the right child of v_k contains the pair $\langle d, d \rangle$ and thus Delilah wins. Thus, Samson must play the path from t_{π^k} to $t_{\pi^{k+1}}$. Delilah will answer among others with the path from f_{π^k} to $f_{\pi^{k+1}}$ and $\langle t_{\pi^{k+1}}, f_{\pi^{k+1}} \rangle$ occurs at a child of v_k as desired. \square

The path of permutation π which is guaranteed by Lemma 6.5 to occur along at least one branch of \mathcal{T} may in fact occur along several branches. For each permutation π we would like to choose a particular branch as the representative branch of π . If $\langle t_{\pi^k}, f_{\pi^k} \rangle$ occurs at v along this branch, and $\langle t_{\pi^k}, f_{\pi^k} \rangle$ still occurs at one of v ’s children, then we follow this child, i.e., we take a branch that avoids making progress if possible. If both steps make progress, or neither do, we follow the left child.

Let π, σ be distinct elements of Π_n . In the next lemma we prove that the branches of π and σ must diverge at some

point in \mathcal{T} . By this we mean that the branches start together at the root, but eventually separate and end at distinct leaves. It will then follow that there are at least as many leaves of \mathcal{T} as elements of Y_n and Lemma 6.3 and Theorem 6.1 thus follow.

Lemma 6.6 *Let π, σ be distinct elements of Π_n . Then the branches of π and σ diverge.*

Proof: Let us assume for the sake of a contradiction that the branches of π and σ coincide. Let k be the first place that π and σ differ, i.e., $\pi(i) = \sigma(i)$ for $i < k$ and $\pi(k) \neq \sigma(k)$. We know that $\langle t_{\pi}, f_{\pi} \rangle$ and $\langle t_{\sigma}, f_{\sigma} \rangle$ both occur at the root.

The branches for π and σ may be moving down in lock step, i.e., $\langle t_{\pi^i}, f_{\pi^i} \rangle$ occurs at the same node as $\langle t_{\sigma^i}, f_{\sigma^i} \rangle$ or one may be ahead of the other, e.g., $\langle t_{\pi^{i+1}}, f_{\pi^{i+1}} \rangle$ occurs at the same node as $\langle t_{\sigma^i}, f_{\sigma^i} \rangle$. Let us assume that they are in lock step, or π is ahead of σ when $\langle t_{\sigma^{k+1}}, f_{\sigma^{k+1}} \rangle$ first occurs. Let v_k be the lowest node on the branch at which $\langle t_{\sigma^k}, f_{\sigma^k} \rangle$ occurs. Since $\langle t_{\sigma^k}, f_{\sigma^k} \rangle$ does not occur as a child of v_k , Samson must play either **EX** or **EU** at the node v_k . There are two cases.

Case 1: $\langle t_{\pi^k}, f_{\pi^k} \rangle$ also occurs at v_k . Thus Samson must step from t_{π^k} to $t_{\pi^{k+1}}$ and from t_{σ^k} to $t_{\sigma^{k+1}}$ at this step. Since $\pi(k) \neq \sigma(k)$, $t_{\pi^{k+1}}$ is a descendant of f_{σ^k} (and $t_{\sigma^{k+1}}$ is a descendant of f_{π^k}). If Samson challenges either of these descendants, then we have the same point on both sides of a node in \mathcal{T} and Delilah wins. If Samson challenges neither, then $\langle t_{\sigma^k}, f_{\sigma^k} \rangle$ occurs at a proper descendant of v_k , contradicting our assumption.

Case 2: $\langle t_{\pi^j}, f_{\pi^j} \rangle$ occurs at v_k for $j > k$. Samson must step from t_{σ^k} to $t_{\sigma^{k+1}}$ and either leave t_{π^j} fixed, or step from t_{π^j} to $t_{\pi^{j+1}}$. Let d be the not-necessarily-proper descendant of t_{π^j} that Samson steps to. Delilah answers with the path from f_{σ^k} to d . Since we have assumed that progress on σ is made at this node, Samson cannot challenge f_{σ^k} . Thus he must challenge d and the pair $\langle d, d \rangle$ occurs at the left child of v_k . This contradicts our assumption that \mathcal{T} is closed.

Thus we have proved that the branches of π and σ cannot remain together after the second one has moved past level k . \square

7 Lower Bound on Booleans in Reachability Logic

In this section we give an interesting application of formula-size games to characterize the number of boolean variables needed in a reachability logic. In [IV97] it is shown that CTL* is linearly embeddable in the transitive closure logic FO²(TC). Furthermore in [AI00] a sublanguage of FO²(TC) called reachability logic (\mathcal{RL}) is described.

CTL* remains linearly embeddable in \mathcal{RL} . The complexity of checking whether a Kripke structure, \mathcal{K} , satisfies an \mathcal{RL} formula, φ , is $O(|\mathcal{K}||\varphi|2^{n_b})$ where n_b is the number of boolean variables occurring in \mathcal{RL} . Both \mathcal{RL} and $\text{FO}^2(\text{TC})$ may contain boolean-valued variables in addition to their two domain variables. Since the time to model check is linear in the size of the formula and the size of the structure, but exponential in the number of booleans, information about how many booleans are needed is important.

The boolean variables are not needed to embed CTL; however in the linear embeddings of CTL* in \mathcal{RL} and $\text{FO}^2(\text{TC})$ at most a linear number of boolean variables may be used. It was left open in [IV97] whether any such booleans are actually needed. It was shown in [AI00] that at least one boolean is needed to embed CTL* at all in $\text{FO}^2(\text{TC})$ or \mathcal{RL} . Whether more than one such boolean variable is needed remains open.

In this section we use a size game for a weakened version of \mathcal{RL} which we call \mathcal{RL}^w . The main result of this section is that for the formulas Occur_n to be translated to polynomial-size formulas in \mathcal{RL}^w , $\Theta(n)$ boolean variables are needed. The main weakness of \mathcal{RL}^w is that we do not allow new unary relations to be defined. We also require weak adjacency formulas to imply $R(x, y)$ as opposed to $R(x, y) \vee R(y, x) \vee x = y$, but this is just for convenience. It can be shown that $\text{LTL} \subseteq \mathcal{RL}^w$ but $\text{CTL} \not\subseteq \mathcal{RL}^w$. Due to lack of space we do not give a full explanation of \mathcal{RL} , directing the reader instead to [AI00]. (We do provide the definition of \mathcal{RL} and a few examples in Appendix B.)

Our original motivation in trying to prove lower bounds on the formula Occur_n was to characterize how many boolean variables are needed in the translations of CTL* to $\text{FO}^2(\text{TC})$ and \mathcal{RL} . In this section we are only able to prove a good lower bound for translations to the weaker language \mathcal{RL}^w . We believe that even this partial result is of interest, and we suspect this approach will lead to a similar lower bound for the full \mathcal{RL} .

Definition 7.1 A *weak adjacency formula* $\delta(x, \bar{b}, y, \bar{b}')$ is the conjunction of $R(x, y)$ with a boolean combination of the booleans \bar{b}, \bar{b}' and the unary relations $p_i(x), p_i(y)$. Define \mathcal{RL}^w to be the smallest fragment of $\text{FO}^2(\text{TC})$ that satisfies the following:

1. If p is a unary relation symbol then $p \in \mathcal{RL}^w$.
2. If $\varphi, \psi \in \mathcal{RL}^w$, then $\neg\varphi \in \mathcal{RL}^w$ and $\varphi \wedge \psi \in \mathcal{RL}^w$.
3. If $\varphi \in \mathcal{RL}^w$ and $\delta(x, \bar{b}, y, \bar{b}')$ is a weak adjacency formula then the following formulas are in \mathcal{RL}^w :

- (a) $\text{REACH}(\delta)\varphi$
- (b) $\text{CYCLE}(\delta)$

Semantics of \mathcal{RL}^w :

$$\begin{aligned} p &\equiv p(x) \\ \text{REACH}(\delta)\varphi &\equiv \exists y(\text{TC } \delta)(x, \bar{0}, y, \bar{1}) \wedge \varphi[y/x] \\ \text{CYCLE}(\delta) &\equiv (\text{TC } \delta)(x, \bar{0}, x, \bar{1}) \end{aligned}$$

□

As an example, we translate Occur_n to \mathcal{RL}^w as follows: $\text{Occur}_n \equiv \text{REACH}(\delta_n)\mathbf{true}$ where $\delta_n(x, \bar{b}, y, \bar{b}') \equiv R(x, y) \wedge \bigwedge_{i=1}^n (b'_i \rightarrow (b_i \vee p_i(x)))$.

The idea is that boolean variable b_i keeps track of whether predicate p_i has ever been satisfied in the current path. We can reach a point where all the booleans are one iff Occur_n holds.

The \mathcal{RL}^w formula-size game is very similar to the CTL formula-size game. In the Reach move, Samson asserts that $\text{REACH}(\delta)\varphi$ holds for all the vertices $v_0 \in A$. For each such v_0 he produces a path:

$$(v_0, \bar{b}^0), (v_1, \bar{b}^1), \dots, (v_r, \bar{b}^r)$$

where $\bar{b}^0 = \bar{0}, \bar{b}^r = \bar{1}$, and $R(v_i, v_{i+1})$ holds for all $i < r$. Delilah answers with a similar path,

$$(w_0, \bar{0}), (w_1, \bar{c}^1), \dots, (w_s, \bar{1}),$$

for as many copies as she wishes of each $w_0 \in B$. For each of Delilah's paths, Samson either challenges the final point, \underline{w}_s , and puts it in B_2 , or he challenges some pair $\langle (w_i, \bar{c}^i), (w_{i+1}, \bar{c}^{i+1}) \rangle$ and puts it in B_1 . Then Delilah lets A_2 contain all the v_r 's and A_1 contains all pairs, $\langle (v_i, \bar{b}^i), (v_{i+1}, \bar{b}^{i+1}) \rangle$. If originally A and B differed on $\text{REACH}(\delta)\varphi$ then after the move, A_1 and B_1 differ on δ and A_2 and B_2 differ on φ . Note that δ is quantifier free and only concerns the booleans together with the unary predicates true at the two points of each pair. In the game we consider below Delilah will only play pairs that correspond to pairs played by Samson, so Samson will never challenge a pair, but rather the endpoint of each of Delilah's paths.

The Cycle move is similar to the Reach move. Since the graphs we will play on below are acyclic, it will not be useful for Samson to play the Cycle move. Let the \mathcal{RL}_k^w game be the \mathcal{RL}^w game in which the tuples of booleans are of size at most k .

We next define the graph H_n on which we will play the \mathcal{RL}^w game. These are simpler than the G_n from Section 5 because we only need an exponential lower bound, not an $n!$ lower bound. Thus we only need consider all subsets of the n propositional variables, not all possible paths through them.

Let X_n be the set of all proper subsets of the n predicates. For any element e of X_n , let $S(e)$ be a path that visits every predicate of e exactly once, and then visits a blank vertex. Let $F(e)$ be a path that visits every predicate

of e exactly once. The order of the predicates in $F(e)$ and $S(e)$ does not matter.

H_n contains $2^n - 1$ “true” vertices, t_e , one for each $e \in X_n$. Node t_e starts with the path $S(e)$, and then from the last (blank) vertex — call it b_e — there is an edge to each first vertex of $F(f)$, for any $f \in X_n$ such that $e \cup f \neq [n]$ and also to $F(\bar{e})$ where $\bar{e} = [n] - e$.

H_n also contains $2^n - 1$ “false” vertices, f_e , one for each $e \in X_n$. Node f_e starts with the path $S(e)$, and then from the last (blank) vertex — call it b'_e — there is an edge to each first vertex of $F(f)$, for any $f \in X_n$ such that $e \cup f \neq [n]$.

Let $T_n = \{t_e \mid e \in X_n\}$; $F_n = \{f_e \mid e \in X_n\}$. Clearly $T_n \models \text{Occur}_n$ and $F_n \models \neg \text{Occur}_n$.

Lemma 7.2 *Samson cannot close the \mathcal{RL}_k^w game on (T_n, F_n) in a game tree with fewer than $2^n/2^k$ nodes.*

Proof: Note that the paths from t_e and f_e are identical through the blank vertices b_e, b'_e at the bottom of their starting paths, $S(e)$, and the only difference after that is that b_e has an edge to $F(\bar{e})$. Thus, to close the game tree, Samson must play a series of Reach moves from t_e to b_e , and then into $F(\bar{e})$ for each $e \in X_n$.

The key observation is that while we are standing on b_e , all that we know is what node of the game tree we are in, plus the current values of our k booleans. Indeed, we prove that Samson cannot play a REACH move that includes a path in which (b_e, \bar{c}) is an intermediate node, and also includes a path in which (b_g, \bar{c}) is an intermediate node, for distinct subsets $e \neq g$ and the same k -tuple of booleans \bar{c} . It follows that Samson can move through at most 2^k different b_e 's at the same time. Our lower bound will then follow.

Suppose for the sake of a contradiction that for distinct subsets $e, g \in X_n$, Samson plays a Reach move that includes a step from b_e and from b_g at the same node of the game tree and that the booleans associated with b_e and b_g are identical.

Since $e \neq g$ we may assume that $e \cup \bar{g} \neq [n]$, otherwise interchange e and g . Delilah answers with a Reach path from f_e to b'_e that first copies the booleans on Samson's path from t_e to b_e . Delilah continues this path to $F(\bar{g})$ copying Samson's path from b_g to $F(\bar{g})$. Since each step in Delilah's spliced path is identical to a step in one of Samson's paths, Samson cannot challenge any of the steps. Thus, Samson must challenge the bottom of Delilah's path. However this is identical to the bottom of Samson's path from t_g .

Thus our assumption was false, so at most 2^k t_e 's can move from their blank vertices, b_e , at the same node of the game tree. Thus there must be at least $(2^n - 1)/2^k$ intermediate nodes of the game tree. Since there are at least n leaves, the total number of nodes is at least $2^n/2^k$ as claimed. \square

Corollary 7.3 $\Omega(n)$ booleans are required to express the CTL⁺ and LTL formula Occur_n as a polynomial-size formula of \mathcal{RL}^w .

8 Conclusions and Future Directions

In this paper we have introduced Ehrenfeucht-Fraïssé games on the size of formulas rather than their operator depth. We have used these games to prove a new, optimal bound which exactly characterizes the succinctness of CTL⁺ with respect to CTL. We have also used these games to prove an $\Omega(n)$ lower bound on the number of booleans needed to translate LTL to \mathcal{RL}^w .

The formula-size games introduced here offer promise in settling many conjectures in descriptive complexity. In particular, questions about true complexity involve languages where an ordering relation on the universe is present. In the presence of ordering, we can express complex properties using low operator depth, with huge disjunctions over all possible input structures of a given size. Thus bounds on operator depth are not helpful here. Bounds on size would be extremely helpful. The formulas involved must be large, assuming well-believed complexity-theoretic conjectures. Although the size game is combinatorially complex, we expect that the methods introduced in this paper will help make progress towards lower bounds for languages with ordering.

We expect that the lower bounds from Section 7 can be extended to the full reachability logic, \mathcal{RL} . Another open problem was suggested by one of the referees: Wilke showed his exponential lower bound for the alternation-free μ -calculus which properly contains CTL [Wil99]. Can our Theorem 6.1 be similarly extended to the alternation-free μ -calculus?

Acknowledgments: Thanks to Natasha Alechina and Thomas Wilke for many helpful comments and suggestions.

References

- [AI00] N. Alechina and N. Immerman, “Reachability Logic: An Efficient Fragment of Transitive Closure Logic,” *Logic Journal of the IGPL* 8(3) (2000), 325-338.
- [CE81] E.M. Clarke and E.A. Emerson, “Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic,” in *Proc. Workshop on Logic of Programs*, LNCS 131, 1981, Springer-Verlag, 52-71.
- [CGP99] E. Clarke, O. Grumberg and D. Peled, *Model Checking*, 1999, M.I.T. press,
- [Ehr61] A. Ehrenfeucht, “An Application of Games to the Completeness Problem for Formalized Theories,” *Fund. Math.* 49 (1961), 129-141.

- [EH85] E.A. Emerson and J.Y. Halpern, “Decision Procedures and Expressiveness in the Temporal Logic of Branching Time,” *J. Comput. Sys. Sci.* 30(1) (1985), 1–24.
- [EI95] K. Etessami and N. Immerman, “Tree Canonization and Transitive Closure,” to appear in *Information and Computation*. A preliminary version appeared in *IEEE Symp. Logic In Comput. Sci.* (1995), 331-341.
- [EW96] K. Etessami and T. Wilke, “An Until Hierarchy for Temporal Logic,” *IEEE Symp. Logic In Comput. Sci.* (1996).
- [Fra54] R. Fraïssé, “Sur les Classifications des Systems de Relations,” *Publ. Sci. Univ. Alger I* (1954).
- [Imm99] N. Immerman, *Descriptive Complexity*, 1999, Springer Graduate Texts in Computer Science, New York.
- [Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16(4):760–778, 1987.
- [IV97] N. Immerman and M.Y. Vardi. Model Checking and Transitive Closure Logic. *Proc. 9th Int’l Conf. on Computer-Aided Verification (CAV’97)*, Lecture Notes in Computer Science, Springer-Verlag 291 - 302, 1997.
- [Kar89] M. Karchmer, *Communication Complexity: A New Approach to Circuit Depth*, 1989, M.I.T. Press.
- [SC85] A. P. Sistla and E. M. Clarke. The Complexity of Propositional Linear Temporal Logics. *JACM*, 32(3), 733-749, 1985.
- [Wil99] T. Wilke, “CTL⁺ is Exponentially More Succinct than CTL”, *Foundations of Software Technology and Theoretical Computer Science: 19th Conference*, (1999), 110–121.

A Background on CTL

A popular and quite expressive language for Model Checking is computation tree logic CTL* [CGP99]. Here we briefly describe CTL* together with some of its sublanguages: $\text{CTL} \subseteq \text{CTL}^+ \subset \text{CTL}^*$ and $\text{LTL} \subset \text{CTL}^*$. CTL and CTL⁺ express the same set of formulas, but CTL⁺ is more succinct. CTL and LTL are incomparable.

CTL* has two kinds of formulas: *state formulas*, which are true or false at each state, and *path formulas*, which are true or false with respect to a maximal path through some Kripke structure, \mathcal{K} . The following is an inductive definition of the state and path formulas of CTL*.

Definition A.1 (Syntax of CTL*) State formulas \mathcal{S} and path formulas \mathcal{P} of CTL* are the smallest sets of formulas satisfying the following:

State Formulas, \mathcal{S} :

the boolean constants **true** and **false** are elements of \mathcal{S} ;

for $i \in \mathbf{N}$, $p_i \in \mathcal{S}$;

if $\varphi \in \mathcal{P}$, then $\mathbf{E}\varphi \in \mathcal{S}$.

Intuitively, $\mathbf{E}\varphi$ means that there exists a maximal path starting at the current state and satisfying φ .

Path Formulas, \mathcal{P} :

if $\alpha \in \mathcal{S}$ then $\alpha \in \mathcal{P}$;

if $\varphi, \psi \in \mathcal{P}$, then $\neg\varphi, \varphi \wedge \psi, \mathbf{X}\varphi$, and $\varphi\mathbf{U}\psi$ are in \mathcal{P} .

Intuitively, $\mathbf{X}\varphi$ means that φ holds at the next time and $\varphi\mathbf{U}\psi$ means that at some time now or in the future, ψ holds, and from now until then, φ holds. \square

Next, we formally define the semantics of the above operators. In this paper all structures will be finite and acyclic except perhaps for self-loops. Thus all paths will be finite, except perhaps for an infinite loop on the final point. A maximal path $\rho = \rho_1, \rho_2, \dots, \rho_\ell$ is a mapping from $[\ell]$ to states in \mathcal{K} such that for all $i < \ell$, $\mathcal{K} \models R(\rho_i, \rho_{i+1})$ and such that ρ_ℓ either has no successors or it has a self-loop. We use the notation ρ^i for the tail of ρ , with states $\rho_1, \rho_2, \dots, \rho_{i-1}$ removed.

Definition A.2 (Semantics of CTL*) The following are inductive definitions of the meaning of CTL* formulas:

State Formulas:

$$(\mathcal{K}, s) \models p_i \text{ iff } \mathcal{K} \models p_i(s)$$

$$(\mathcal{K}, s) \models \mathbf{E}\varphi \text{ iff } (\exists \text{ path } \rho \text{ s.t. } \rho_0 = s)(\mathcal{K}, \rho) \models \varphi$$

Path Formulas:

$$(\mathcal{K}, \rho) \models \alpha \text{ iff } (\mathcal{K}, \rho_0) \models \alpha; \text{ for } \alpha \in \mathcal{S}$$

$$(\mathcal{K}, \rho) \models \varphi \wedge \psi \text{ iff } (\mathcal{K}, \rho) \models \varphi \text{ and } (\mathcal{K}, \rho) \models \psi$$

$$(\mathcal{K}, \rho) \models \neg\varphi \text{ iff } (\mathcal{K}, \rho) \not\models \varphi$$

$$(\mathcal{K}, \rho) \models \mathbf{X}\varphi \text{ iff } (\mathcal{K}, \rho^1) \models \varphi$$

$$(\mathcal{K}, \rho) \models \varphi\mathbf{U}\psi \text{ iff } (\exists i)(\mathcal{K}, \rho^i) \models \psi \wedge (\forall j < i)(\mathcal{K}, \rho^j) \models \varphi$$

\square

It is convenient to introduce a few other operators commonly used in CTL* all of which may be defined from the above:

$$\mathbf{A}\varphi \equiv \neg\mathbf{E}\neg\varphi \text{ for All paths}$$

$$\mathbf{F}\varphi \equiv \mathbf{trueU}\varphi \text{ some time in the Future}$$

$$\mathbf{G}\varphi \equiv \neg\mathbf{F}\neg\varphi \text{ Globally, i.e., for all times in the future}$$

The language CTL is the restriction of CTL* so that path quantifiers (**E**, **A**) and temporal operators (**X**, **U**) are always

paired. That is, the allowable operators are **EU**, **AU**, **EX**⁴. The importance of CTL is that unlike CTL* it admits linear-time model checking [CE81]. The language CTL⁺ allows boolean combinations of the temporal operators to be paired with the path quantifiers. CTL⁺ is no more expressive than CTL but it is more succinct [Wil99]. Our main result shows exactly how succinct. The language LTL (linear temporal logic) consists of CTL* formulas that have exactly one path quantifier **E** or **A** and that begin with this path quantifier.

B Background on \mathcal{RL}

Here we give the definition of Reachability Logic (\mathcal{RL}). See [AI00] for proofs of the theorems and much more motivation and discussion.

Definition B.1 An *adjacency formula* (with booleans) is a disjunction of conjunctions where each conjunct contains at least one of $x = y$, $R_a(x, y)$ or $R_a(y, x)$ for some edge label a ; in addition, the conjuncts may contain expressions of the form $(\neg)(b_1 = b_2)$, $(b_1 = 0)$, $(b_1 = 1)$ and $p(x)$, where b_1 and b_2 are boolean variables. \square

Definition B.2 \mathcal{RL} is the smallest fragment of $\text{FO}^2(\text{TC})$ that satisfies the following:

1. If p is a unary relation symbol then $p \in \mathcal{RL}$; also $\top, \perp \in \mathcal{RL}$.
2. If $\varphi, \psi \in \mathcal{RL}$, then $\neg\varphi \in \mathcal{RL}$ and $\varphi \wedge \psi \in \mathcal{RL}$.
3. If $\varphi \in \mathcal{RL}$ and b is a boolean variable, then $\exists b\varphi \in \mathcal{RL}$.
4. If $\varphi, \psi \in \mathcal{RL}$ and q is a new unary predicate symbol, then **(let $q = \varphi$ in ψ)** is in \mathcal{RL} .
5. If $\varphi \in \mathcal{RL}$ and $\delta(x, \bar{b}, y, \bar{b}')$ is an adjacency formula (a binary relation between two n -tuples $\langle x, b_1, \dots, b_{n-1} \rangle$ and $\langle y, b'_1, \dots, b'_{n-1} \rangle$), then the following formulas are in \mathcal{RL} :

- (a) $\text{REACH}(\delta)\varphi$
- (b) $\text{CYCLE}(\delta)$

Semantics of \mathcal{RL} : The semantics of \mathcal{RL} is defined as follows. In each case below assume that $\delta(x, \bar{b}, y, \bar{b}')$ is an adjacency formula.

$$\begin{aligned}
p &\equiv p(x) \\
(\text{let } q = \varphi \text{ in } \psi) &\equiv \psi[\varphi/q] \\
\text{REACH}(\delta)\varphi &\equiv \exists y(\text{TC } \delta)(x, \bar{0}, y, \bar{1}) \wedge \varphi[y/x] \\
\text{CYCLE}(\delta) &\equiv (\text{TC}^s \delta)(x, \bar{0}, x, \bar{1})
\end{aligned}$$

⁴We do not need **AX** because it is equivalent to $\neg\text{EX}\neg$

\square

Here are some examples of formulas in \mathcal{RL} :

- $\text{REACH}(\delta)p$ where $\delta(x, b_1, b_2, y, b'_1, b'_2)$ is $(R_a(x, y) \wedge b_1 b_2 = 00 \wedge b'_1 b'_2 = 01) \vee (R_b(x, y) \wedge b_1 b_2 = 01 \wedge b'_1 b'_2 = 11)$ (this is $\langle a; b \rangle p$ of PDL).
- $\varphi_1 = \text{REACH}(R)p$ (**EF** p of CTL*);
- $\varphi_2 = \text{REACH}(\delta)\text{CYCLE}(\delta)$, where δ is $R(x, y) \wedge q(x)$ (**EG** q of CTL*);
- **(let $q = \varphi_1$ in φ_2)** (**EGEF** p of CTL*).

\mathcal{RL} is a logical language and it is a fragment of $\text{FO}^2(\text{TC})$. However, because of the ‘let’ construct, when we talk about size in the representation of \mathcal{RL} , we are really talking about circuits. Thus the size of an \mathcal{RL} -circuit may be logarithmic in the size of the smallest equivalent $\text{FO}^2(\text{TC})$ formula. This allows the linear size embedding of CTL* which presumably does not hold for $\text{FO}^2(\text{TC})$ (without a circuit representation or an extra domain variable cf. [IV97]).

Boolean variables however add extra complexity, which is not surprising since model checking CTL* is PSPACE complete [SC85].

Theorem B.3 *There is an algorithm that given a graph G and a formula $\varphi(x) \in \mathcal{RL}$ marks the vertices in G that satisfy φ . This algorithm runs in time $O(|G||\varphi|2^{n_b})$ where n_b is the number of boolean variables occurring in φ .*

Theorem B.4 *There is a linear-time computable function g that maps any CTL* formula φ to an equivalent formula $g(\varphi) \in \mathcal{RL}$. While $g(\varphi)$ has only two domain variables, it may have a linear number of boolean variables.*