An n! Lower Bound on Formula Size

MICAH ADLER and NEIL IMMERMAN
UMass

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, $\text{Occur}_n \equiv \mathbf{E}[\mathbf{F}p_1 \wedge \mathbf{F}p_2 \wedge \cdots \wedge \mathbf{F}p_n]$, which says that there is a path along which the predicates p_1 through p_n all occur, 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 [Wilke 1999].

We also use our games to prove an optimal $\Omega(n)$ lower bound on the number of boolean variables needed for forward reachability logic ($\mathcal{RL}^f$) to polynomially embed the language CTL$^+$. The number of booleans needed for full reachability logic $\mathcal{RL}$ and the transitive closure logic FO^2(TC) remain open [Immerman and Vardi 1997; Alechina and Immerman 2000].

Categories and Subject Descriptors: F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes—*relations among complexity classes*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical logic—*temporal logic*

General Terms: Theory

Additional Key Words and Phrases: Descriptive complexity, temporal logic, lower bounds

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,

$$\text{Occur}_n \quad \equiv \quad \mathbf{E}[\mathbf{F}p_1 \wedge \mathbf{F}p_2 \wedge \cdots \wedge \mathbf{F}p_n] \tag{1.1}$$

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.

The research of M. Adler was supported by National Science Foundation (NSF) Faulty Early Career Development Award CCR-0133664.
The research of N. Immerman was supported by National Science Foundation (NSF) grants CCR-9877078 and CCR-0207373.
Authors' address: Computer Science Department, Room 374, 140 Governor's Drive, University of Massachusetts, Amherst, MA 01003-9264; emails: {Adler,Immerman}@cs.mass.edu, URLs: http:www.cs.umass.edu/{~micah,~immerman}.

(**E** is the existential path quantifier: there exists a maximal path starting from the current point. **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^+ [Wilke 1999]. 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 $\{1, 2, \ldots, n\}$.

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

The main contributions of this article are the introduction of the new formula-size games, and 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, that is, 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 [Immerman 1999].

The formula-size games introduced here generalize standard EF games. They are also related to the communication complexity games that Karchmer and Wigderson used to prove lower bounds on the depth of monotone circuits [Karchmer 1989].[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 article takes a step in correcting.

[1]Karchmer 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.

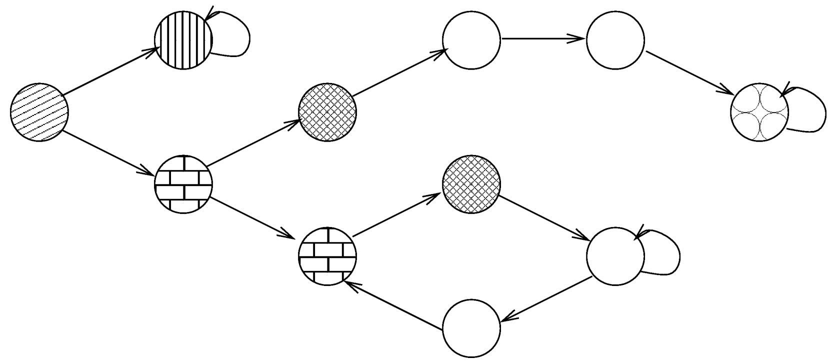

Fig. 1. A Kripke structure.

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 the optimal result that $\Omega(n)$ Booleans are needed to translate an LTL formula of size n to a polynomial-size formula of the reachability logic, $\mathcal{RL}^f$.

This article is organized as follows: In Section 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 Section 3, we review Ehrenfeucht–Fraïssé games and present the new formula-size games for FO(TC). In Section 4, we present the formula-size game for CTL. In Section 5, we define the graphs G_n over which we prove our lower bound. In Section 6, we prove our main result, the optimal $n!$ lower bound on the succinctness of CTL$^+$ with respect to CTL. In Section 7, we prove an $n - O(\log n)$ lower bound on the number of Boolean variables needed for $\mathcal{RL}^f$ to express Occur_n in polynomial size.

2. BACKGROUND

In this section we review some basic definitions concerning finite model theory and transitive closure logic [Immerman 1999].

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, that is, the number of occurrences of logical connectives, quantifiers, operators, and atomic symbols.

For our purposes, a Kripke structure (Figure 1) is a finite labeled graph:

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

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. We assume that there is at least one edge from each vertex. (This is achieved by adding a self-loop to each vertex that has no other outgoing edges.)

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

"There is a path from where we are (x) to a vertex where p_{17} holds." (2.2)

For this reason we add a transitive closure operator to first-order logic to allow us to express reachability [Immerman 1987].

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

3. EHRENFEUCHT–FRAÏSSÉ GAMES

We assume that the reader is somewhat familiar with classical Ehrenfeucht–Fraïssé (EF) games [Ehrenfeucht 1961; Fraisse 1954; Immerman 1999]. 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 FO^2(TC)—first-order logic with two variables and the transitive closure operator. We chose this logic because it is simple, expressive, and quite general. The language FO(TC) captures the complexity class NSPACE[$\log n$]. It is discussed in detail in Immerman [1999]. In Immerman and Vardi [1997], a linear-time algorithm is given that translates any formula from CTL into an equivalent expression in FO^2(TC).

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 FO^2(TC) game to a less general language, CTL, where we will prove our main results.

Throughout this article, we consider formula-size games as two-person games. As we will point out, these games all have an obvious (although somewhat wasteful) optimal strategy for one of the players (Delilah). Nonetheless, we feel that it is more intuitive to discuss and to play these games as two-player games, rather than as solitaire.

Definition 3.1 (FO^2(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. There is some fixed subset $S \subseteq \{x, y\}$ for which each structure in $\mathcal{A}_0$ and B_0 interprets exactly the variables in S.

(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 ($B_0 \models \neg\Phi$). Φ may include free variables from S.)

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.

—*∃ 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 $\mathcal{A}$: $v^{\mathcal{A}} = a_0, a_1, a_2, \ldots, a_t = v'^{\mathcal{A}}$. Delilah then answers by choosing for every structure $\mathcal{B} \in B$ a similar sequence, $v^{\mathcal{B}} = b_0, b_1, b_2, \ldots, a_{t'} = v'^{\mathcal{B}}$. 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 $\mathrm{TC}_{x,y}(\delta)(v, v')$ 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, that is, before any TC move or ∃ move. Delilah is not involved in the other moves.

In the formula-size game, Delilah has an obvious optimal strategy, 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'.

The reason that Delilah is allowed to make copies is easiest to understand for an existential move. 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,

$$\exists x(p_1(x) \ \wedge \ p_2(x) \ \wedge \ \cdots \ \wedge \ p_n(x)),$$

that is, $\mathcal{A}$ has a point satisfying all n predicate symbols, but $\mathcal{B}$ does not. If Delilah could not make duplicates, then Samson would just choose the relevant a in $\mathcal{A}$ and Delilah would have to answer with a single b from $\mathcal{B}$. Then, it must be the case that for some i, $\mathcal{A}, a/x \models p_i(x)$ and $\mathcal{B}, b/x \models \neg p_i(x)$. Thus, Samson could close a game tree of size 2. However, if Delilah could make copies, she would make n copies of $\mathcal{B}$, $\mathcal{B}_i$, $i = 1, \ldots, n$. She would choose b_i from $\mathcal{B}_i$ that satisfies all predicates except p_i. Then, Samson would have to split B into n pieces in order the close the game tree. A game tree with n leaves is thus unavoidable.

Similar examples can be constructed for a TC move. The above intuitive idea is formalized in the discussion of the "$\exists$" move and the "TC" move in the proof of the fundamental theorem of the $\mathrm{FO}^2(\mathrm{TC})$ formula-size game:

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* $\varphi \in FO^2(TC)$ *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 $B' \models \neg\psi$.
— $\varphi = \mathrm{TC}_{x,y}(\delta)(v, v')$. 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 with a tree T of size s and that Delilah has played optimally. It follows that T is also the parse tree of a 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, that is, 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", ∃ 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), and ψ_i is the formula encoded by that subtree.

—*"not" move.* Inductively, we know that $A_1 \models \psi_1$ and $B_1 \models \neg\psi_1$, where ψ_1 has size s_1. The "not" move simply switches A and B. Thus, $A \models \neg\psi_1$ and $B \models \psi_1$, and thus A and B differ on the formula "$\neg\psi_1$" of size $s_1 + 1$.

—*"or" move.* Inductively, we know that $A_1 \models \psi_1$, $A_2 \models \psi_2$, and $A = A_1 \cup A_2$. Thus, $A \models \psi_1 \vee \psi_2$. Furthermore, $B = B_1 = B_2$ and $B \models \neg\psi_1 \wedge \neg\psi_2$. Thus, A and B differ on "$\psi_1 \vee \psi_2$" which has size $s_1 + s_2 + 1$.

—∃ *move.* Inductively, we know that $A_1 \models \psi_1$ and $B_1 \models \neg\psi_1$. Thus, $A \models \exists v(\psi_1)$ where $v \in \{x, y\}$ is the variable that this ∃ move was played on. Note that since Delilah plays optimally, if it were the case that some $\mathcal{B} \in B$ satisfies $(\exists v)\psi_1$, then Delilah would have chosen the appropriate witness for this $\mathcal{B}$ and it would not have been the case that $B_1 \models \neg\psi_1$. Thus, as desired, $A \models (\exists v)\psi_1$, $B \models \neg(\exists v)\psi_1$, and thus they differ on a formula of size $s_1 + 1$.

—TC *move.* Inductively, we know that $A_1 \models \psi_1$ and $B_1 \models \neg\psi_1$. By the definition of the TC move, since $A_1 \models \psi_1$, we know that for every $\mathcal{A} \in A$, there is a ψ_1-path from $v^{\mathcal{A}}$ to $v'^{\mathcal{A}}$. Furthermore, if there were a ψ_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 ψ_1. Therefore, $A \models \mathrm{TC}_{x,y}(\psi_1)(v, v')$ and $B \models \neg\mathrm{TC}_{x,y}(\psi_1)(v, v')$. Thus, they differ on a formula of size $s_1 + 1$.

Thus, A_0 and B_0 differ on the formula of size s that is expressed by the game tree, T. □

4. DEFINITION OF THE CTL GAME

We assume that the reader is somewhat familiar with the languages CTL and CTL⋆ [Emerson 1991; Clarke et al. 1999]. Recall that 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**.[2] The importance of CTL is that unlike CTL⋆ it admits linear-time model checking [Clarke and Emerson 1981]. 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 [Wilke 1999]. Our main result shows exactly how succinct.

We now define the CTL formula-size game.[3] Since CTL is embedable in FO^2(TC) [Immerman and Vardi 1997], it is natural to define the CTL game as a modification of the FO^2(TC) formula-size game (Definition 3) as follows.

—There is only one variable: x. This corresponds to the current state.

—The "not" and "or" moves are unchanged.

[2]We do not need **AX** because it is equivalent to ¬**EX**¬

[3]It is easy to generalize this also to the CTL⋆ formula-size game, but we leave this to the reader.

—The atomic move is unchanged except that it is played only using atomic formulas p_i.

—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: $x^{\mathcal{A}} = a_0, a_1, \ldots, a_r$. Delilah answers as above with a path $x^{\mathcal{B}} = b_0, b_1, \ldots, b_s$ for each copy she makes of each $\mathcal{B} \in B$. Samson is trying to assert that $(A, x) \models \mathbf{E}(\alpha\mathbf{U}\beta)$, that is, 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 nonfinal 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 having a nonfinal point that does not satisfy α, Samson chooses this point and puts the structure into B_1. At the end of the move we have that $A_1 \models \alpha$, $B_1 \models \neg\alpha$, $A_2 \models \beta$, and $B_2 \models \neg\beta$. 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 is not surprising that:

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 $\varphi \in$ CTL of size at most s such that every structure in A_0 satisfies φ and no structure in B_0 does.*

PROOF. This is very similar to the proof of Theorem 3.2. Suppose that $\varphi \in$ CTL 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 $B \models \neg\varphi$. We just consider the cases new to the CTL game:

—$\varphi = \mathbf{EX}\psi$. Samson plays the **EX** move. For each $\mathcal{A} \in A$, he assigns the new x to a child of the old x that satisfies ψ. Since $\mathcal{B} \models \neg\varphi$, whatever Delilah answers, we have that $A' \models \psi$ and $B' \models \neg\psi$.

—$\varphi = \mathbf{E}(\alpha\mathbf{U}\beta)$. Samson plays the **EU** move. For each $\mathcal{A} \in A$, he chooses a path $x^{\mathcal{A}} = a_0, a_1, \ldots, a_r$, such that $\mathcal{A}, a_r \models \beta$ and $\mathcal{A}, a_i \models \alpha$ for all $i < r$. For any path chosen by Delilah, $x^{\mathcal{B}} = b_0, b_1, \ldots, b_s$, it must be that either $\mathcal{B}, b_s \models \neg\beta$, or $\mathcal{B}, b_i \models \neg\alpha$ for some $i < s$. In the former case, Samson places $\mathcal{B}, b_s$ in B_2. In the latter case, Samson places $\mathcal{B}, b_i$ in B_1. Thus, as desired, $A_1 \models \alpha$; $B_1 \models \neg\alpha$; $A_2 \models \beta$; $\mathcal{B}_2 \models \neg\beta$.

—$\varphi = \mathbf{A}(\alpha\mathbf{U}\beta)$. Samson plays the **AU** move. In the first phase of the move, for each $\mathcal{B} \in B$, he chooses a maximal path, $p_{\mathcal{B}}$ such that $\mathcal{B}, p_{\mathcal{B}} \models \neg(\alpha\mathbf{U}\beta)$. For all the paths $p_{\mathcal{A}}$ chosen by Delilah, $\mathcal{A}, p_{\mathcal{A}} \models \alpha\mathbf{U}\beta$. Then, the move continues as above, that is, for each $\mathcal{A}, p_{\mathcal{A}}$, Samson chooses a finite path, $x^{\mathcal{A}} = a_0, a_1, \ldots, a_r$, along $p_{\mathcal{A}}$ such that $\mathcal{A}, a_r \models \beta$ and $\mathcal{A}, a_i \models \alpha$ for all $i < r$. Whatever finite paths Delilah chooses along $p_{\mathcal{B}}$ will not have this property.

Conversely, suppose that Samson has succeeded in closing the game with a tree T of size s and that Delilah has played optimally. It follows that T is also the parse tree of a 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. We only consider the cases in the CTL game not considered in the $\mathrm{FO}^2(\mathrm{TC})$ game. 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 the "**EX**" moves and $i = 1, 2$ for "**EU**" and "**AU**" moves. Here, s_i is the size of the subtree rooted at (A_i, B_i), and ψ_i is the formula encoded by that subtree.

—**EX** *move.* Inductively, we know that $A_1 \models \psi_1$ and $B_1 \models \neg\psi_1$. Thus $A \models \mathbf{EX}\psi_1$. Since Delilah played optimally, if it were the case that some $\mathcal{B} \in B$ satisfies $\mathbf{EX}\psi_1$, then Delilah would have chosen the appropriate child of $x^{\mathcal{B}}$ for this $\mathcal{B}$ and it would not have been the case that $B_1 \models \neg\psi_1$. Thus, $B \models \neg\mathbf{EX}\psi_1$, and A and B differ on a formula of size $s_1 + 1$.

—**EU** *move.* Inductively, we know that $A_i \models \psi_i$ and $B_i \models \neg\psi_i$, for $i = 1, 2$. Recall that, in the **EU** move, Samson chose a path for every structure $\mathcal{A} \in A$, and for each such path Delilah placed all the nonfinal points in A_1, and the final points in A_2. Thus, $A \models \mathbf{E}(\psi_1\mathbf{U}\psi_2)$, Since Delilah plays optimally, if it were the case that some $\mathcal{B} \in B$ satisfies $\mathbf{E}(\psi_1\mathbf{U}\psi_2)$, then Delilah would have chosen the appropriate path from $x^{\mathcal{B}}$ and thus Samson must have placed $\mathcal{B}, b_i \models \psi_1$ in B_1 or $\mathcal{B}, b_s \models \psi_2$ in B_2. Thus, $B \models \neg\mathbf{E}(\psi_1\mathbf{U}\psi_2)$ and so A and B differ on a formula of size $s_1 + s_2 + 1$.

—**AU** *move.* Inductively, we know that $A_i \models \psi_i$ and $B_i \models \neg\psi_i$, for $i = 1, 2$. Since Delilah played optimally, if any $\mathcal{A} \in A$ had a maximal path not satisfying $\psi_1\mathbf{U}\psi_2$, then she would have chosen it. Similarly if some $\mathcal{B}$ satisfied $\mathbf{A}(\psi_1\mathbf{U}\psi_2)$, then no matter what maximal path $p_{\mathcal{B}}$ Samson chose, Delilah would have chosen a path witnessing $\psi_1\mathbf{U}\psi_2$, and thus it would not be the case that both $B_1 \models \neg\psi_1$ and $B_2 \models \neg\psi_2$. Thus, $A \models \mathbf{A}(\psi_1 \mathbf{U} \psi_2)$ and $B \models \neg\mathbf{A}(\psi_1 \mathbf{U} \psi_2)$, so they differ on a formula of size $s_1 + s_2 + 1$.

Thus, A_0 and B_0 differ on the formula of size s that is expressed by the game tree, T. □

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 build two sets of graphs A_0, B_0 such that $A_0 \models \mathrm{Occur}_n$ and $B_0 \models \neg\mathrm{Occur}_n$. 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, $V_n = T_n \cup F_n$,

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

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

$$\pi : [|\mathrm{rng}(\pi)|] \to \mathrm{rng}(\pi) \subseteq [n] \;.$$

For any such permutation π on at least two elements, define its tail, $\mathrm{tail}(\pi) : [|\mathrm{rng}(\pi)| - 1] \to \mathrm{rng}(\pi) - \{\pi(1)\}$ where $\mathrm{tail}(\pi)(i) = \pi(i+1)$. For ease of notation, let $\pi^2 = \mathrm{tail}(\pi)$, and in general, $\pi^{k+1} = \mathrm{tail}^k(\pi)$, that is, 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 $\mathrm{rng}(\sigma) \subseteq \mathrm{rng}(\pi) - \{j\}$, for some $j \in \mathrm{rng}(\pi)$, $j \neq \pi(1)$
— $f_\sigma \in F_n$ where $\mathrm{rng}(\sigma) \subseteq \mathrm{rng}(\pi) - \{j\}$, for some $j \in \mathrm{rng}(\pi)$

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,

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

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

LEMMA 5.1. *For any $\pi \in \Pi_n$,*

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

6. PLAYING THE CTL GAME

In this section, we prove the following,

THEOREM 6.1. *The formula* $Occur_n$ (*Eq.* (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* $n!$.

Emerson and Halpern [1985] proved that any CTL^+ formula of size n may be translated to an equivalent, size-$\Theta(n)!$ CTL formula. Theorem 6.1 shows that the Emerson and Halpern bound is optimal.

COROLLARY 6.2. CTL^+ *has succinctness exactly* $\Theta(n)!$ *with respect to CTL.*

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 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 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 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):*

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

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 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 (A) or on the right (B). 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. Thus, Samson must play one of the following moves: **EX**, **EU**, **AU**.

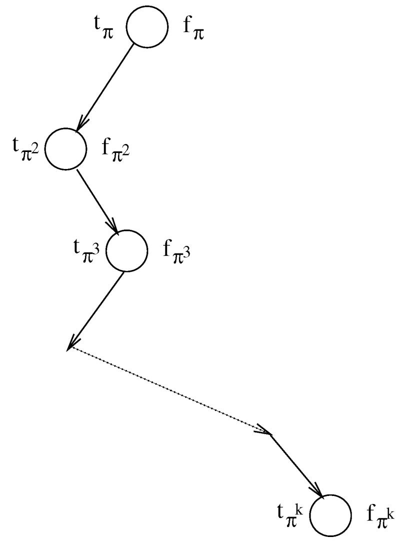

Fig. 2. Proof of Lemma 6.5.

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. This contradicts the assumption that v_k is the lowest node at which $\langle t_{\pi^k}, f_{\pi^k}\rangle$ occurs. 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. (See Figure 2.) □

Lemma 6.5 guarantees that the path of permutation π occurs along at least one branch of $\mathcal{T}$, but it 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, that is, 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, that is, $\pi(i) = \sigma(i)$ 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, that is, $\langle t_{\pi^i}, f_{\pi^i}\rangle$ occurs at the same node as $\langle t_{\sigma^i}, f_{\sigma^i}\rangle$ or one may be ahead of the other, for example, $\langle t_{\pi^{i+1}}, f_{\pi^{i+1}}\rangle$ 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 played **EX**, then Delilah will answer with f_{σ^k} stepping to $t_{\pi^{k+1}}$. Thus, $t_{\pi^{k+1}}$ would be on both sides and Delilah would win.

Thus, Samson must have stepped to $t_{\pi^{k+1}}$ and $t_{\sigma^{k+1}}$ using an **EU** move, and Delilah will step from f_{σ^k} to $t_{\pi^{k+1}}$ and from f_{π^k} to $t_{\sigma^{k+1}}$ in response.

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 descendent 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 application of formula-size games to characterize the number of Boolean variables needed in a reachability logic. In Immerman and Vardi [1997], it is shown that CTL* is linearly embedable in the transitive closure logic $\mathrm{FO}^2(\mathrm{TC})$. Furthermore, in Alechina and Immerman [2000], a sublanguage of $\mathrm{FO}^2(\mathrm{TC})$ called reachability logic ($\mathcal{RL}$) is described. CTL* remains linearly embedable 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 φ. Both $\mathcal{RL}$ and $\mathrm{FO}^2(TC)$ 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 $\mathrm{FO}^2(\mathrm{TC})$ at most a linear number of Boolean variables are used. It was left open in Immerman and Vardi [1997] whether any such Booleans are actually needed. It was shown in Alechina and Immerman [2000] that at least one Boolean is needed to embed CTL* at all in $\mathrm{FO}^2(\mathrm{TC})$ or $\mathcal{RL}$. Whether more than one such Boolean variable is needed remains open.

In this section, we use a size game for a slightly weakened version of $\mathcal{RL}$, which we call $\mathcal{RL}^f$. The main result of this section is that for the formulas Occur_n to be translated to polynomial-size formulas in $\mathcal{RL}^f$, $n - O(\log n)$ Boolean variables are needed. The weakness of $\mathcal{RL}^f$ is that we require adjacency formulas to imply $R(x, y)$ as opposed to $R(x, y) \vee R(y, x) \vee x = y$. Thus, $\mathcal{RL}^f$ can only discuss forward paths. It still holds that $\mathrm{CTL}^\star \subseteq \mathcal{RL}^f$.

7.1 Background on $\mathcal{RL}$

Here we give the definition of Reachability Logic ($\mathcal{RL}$) [Alechina and Immerman 2000].

Definition 7.1. 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.

Definition 7.2. $\mathcal{RL}$ is the smallest fragment of $\mathrm{FO}^2(\mathrm{TC})$ that satisfies the following:

(1) If p is a unary relation symbol then $p \in \mathcal{RL}$; also $\top, \bot \in \mathcal{RL}$.
(2) If $\varphi, \psi \in \mathcal{RL}$, then $\neg\varphi \in \mathcal{RL}$ and $\varphi \vee \psi \in \mathcal{RL}$.
(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 $(\mathbf{let}\, q = \varphi \ \mathbf{in}\ \psi)$ is in $\mathcal{RL}$.
(5) If $\varphi \in \mathcal{RL}$ and $\delta(x, \overline{b}, y, \overline{b'})$ is an adjacency formula (a binary relation between two n-tuples $\langle x, b_1, \ldots, b_{n-1}\rangle$ and $\langle y, b'_1, \ldots, b'_{n-1}\rangle$), then the following formulas are in $\mathcal{RL}$:
    (a) $\mathrm{REACH}(\delta)\varphi$
    (b) $\mathrm{CYCLE}(\delta)$

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

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

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

—REACH$(\delta)p$ where $\delta(x, b_1, b_2, y, b'_1, b'_2)$ is $(R_a(x, y) \wedge b_1b_2 = 00 \wedge b'_1b'_2 = 01) \vee (R_b(x, y) \wedge b_1b_2 = 01 \wedge b'_1b'_2 = 11)$ (this is $\langle a; b\rangle p$ of PDL).
—$\varphi_1 =$ REACH$(R)p$ ($\mathbf{EF}p$ of CTL*);
—$\varphi_2 =$ REACH(δ)CYCLE(δ), where δ is $R(x, y) \wedge q(x)$ ($\mathbf{EG}q$ of CTL*);
—($\mathbf{let}\, q = \varphi_1$ *in* φ_2) ($\mathbf{EGEF}p$ of CTL*).

$\mathcal{RL}$ is a logical language and it is a fragment of FO^2(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 FO^2(TC) formula. This allows the linear size embedding of CTL*, which presumably does not hold for FO^2(TC) (without a circuit representation or an extra domain variable, cf. Immerman and Vardi [1997]).

Definition 7.3 ($\mathcal{RL}^f$). A *forward adjacency formula* $\delta(x, \overline{b}, y, \overline{b'})$ is the conjunction of $R(x, y)$ with a Boolean combination of the Booleans $\overline{b}$, $\overline{b'}$ and the unary relations $p_i(x)$, $p_i(y)$. Define $\mathcal{RL}^f$ to be the sublanguage of $\mathcal{RL}$ all of whose adjacency formulas are forward adjacency formulas. That is, $\mathcal{RL}^f$ is that part of $\mathcal{RL}$ that does not look back. Let $\mathcal{RL}^f_k$ be the sublanguage of $\mathcal{RL}^f$ that has at most k pairs of Booleans: $b_1, \ldots, b_k, b'_1, \ldots, b'_k$.

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

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.

Definition 7.4 (*$\mathcal{RL}^f_k$ formula-size game*). The $\mathcal{RL}^f_k$ formula-size game is very similar to the CTL formula-size game. The main differences are that in addition to the position variable x, each structure may assign at most k Boolean variables: $b_1, \ldots, b_k$. Instead of the **EX**, **EU**, and **AU** moves in the CTL game, we have the following:

—*REACH move.* Intuitively, Samson wants to assert that REACH$(\delta)\varphi$ holds for all $\mathcal{A} \in A$. For each $\mathcal{A}$, he produces a path:

$$(x^{\mathcal{A}} = v_0, \overline{b^0}), (v_1, \overline{b^1}), \ldots, (v_r, \overline{b^r})$$

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

$$(x^{\mathcal{B}} = w_0, \overline{0}), (w_1, \overline{c^1}), \ldots, (w_s, \overline{1}),$$

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, w_s, and puts it in B_2, or he challenges some pair $\langle(w_i, \overline{c^i}), (w_{i+1}, \overline{c^{i+1}})\rangle$ and puts it in B_1. Then, Delilah lets A_2 contain all the v_r's and A_1 contains all pairs, $\langle(v_i, \overline{b^i}), (v_{i+1}, \overline{b^{i+1}})\rangle$. If originally A and B differed on 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 a forward adjacency formula. On pair nodes such as A_2, B_2, Samson may only play "or" and "not" moves. He may also *choose* the first half of all the pairs, or the second half of all the pairs, that is, x is assigned to that point in the pair and the corresponding Booleans are assigned. This is the way that a pair node may become a full-fledged node again.

(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.)

—*CYCLE Move.* This is very similar to the Reach move. The only difference is that all paths chosen by Samson and Delilah must be cycles, that is, start and end at the same vertex. Since the graphs we will play on below have no cycles except self-loops, it will not be useful for Samson to play the Cycle move.
—**let** *move.* Samson may take any two open leaves of the game tree labeled A, B and A', B', and merge them into a single node labeled $(A \cup A'), (B \cup B')$. In this way, we are replacing game trees with game DAG's.
—$\exists$ *Boolean move.* Samson may assign Boolean b_i in each $\mathcal{A} \in A$. Delilah answers by making as many copies of any $\mathcal{B} \in B$ as she chooses and assigning the Boolean b_i in $\mathcal{B}$.
—*Atomic move.* Samson chooses an atomic formula α. (α may be any of $p_i(x)$, $b_i = b_j$, $b_1 = 0$, or $b_i = 1$) Samson can only make this move if every structure in A satisfies α and no structure in B does. In this case, the current leaf is *closed*.

Very similar to the proofs of Theorems 3.2 and 4.1, we have the following:

THEOREM 7.5. *Samson can close the $\mathcal{RL}_k^f$ formula-size game started at A_0, B_0 in a DAG of size s iff there is a formula $\varphi \in \mathcal{RL}$ of size at most s, having at most k pairs of boolean variables, and such that every structure in A_0 satisfies φ and no structure in B_0 does.*

We next define the graph H_n on which we will play the $\mathcal{RL}^f$ game. This is simpler than 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.

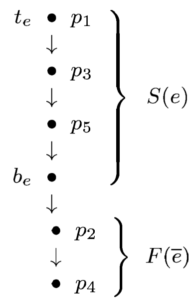

Fig. 3. Paths $S(e)$ and $F(\overline{e})$ for $e = \{1, 3, 5\}$ and $n = 5$.

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 $F(\overline{e})$ where $\overline{e} = [n] - e$ (See Figure 3).

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 7.6. *Samson cannot close the $\mathcal{RL}_k^f$ 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 $F(\overline{e})$. Thus, to close the game tree, Samson must play a series of Reach moves from t_e to b_e, and then into $F(\overline{e})$ 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 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 $e \cup \overline{g} \neq [n]$, 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 $F(\overline{g})$ copying Samson's path from b_g to $F(\overline{g})$. 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. □

COROLLARY 7.7. *$n - O(\log n)$ Booleans are required to express the CTL$^+$ formula $Occur_n$ as a polynomial-size formula of $\mathcal{RL}^f$.*

8. CONCLUSIONS AND FUTURE DIRECTIONS

In this article, 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 CTL$^+$ to $\mathcal{RL}^f$.

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 article 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 [Wilke 1999]. 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

ALECHINA, N. AND IMMERMAN, N. 2000. Reachability logic: An efficient fragment of transitive closure logic. *Logic J. IGPL 8*, 3, 325–338.

CLARKE, E. M. AND EMERSON, E. A. 1981. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proceedings of the Workshop on Logic of Programs*. 52–71.

CLARKE, E. M., GRUMBERG, O., AND PELED, D. 1999. *Model Checking*. M.I.T. Press, Cambridge, Mass.

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

EMERSON, E. A. 1991. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed. M.I.T. Press, Cambridge, Mass., 995–1072.

EMERSON, E. A. AND HALPERN, J. Y. 1985. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. Syst. Sci. 30*, 1, 1–24.

FRAISSE, R. 1954. Sur les classifications des systems de relations. *Publ. Sci. Univ. Alger I*.

IMMERMAN, N. 1987. Languages that capture complexity classes. *SIAM J. Comput. 16*, 4, 760–778.
IMMERMAN, N. 1999. *Descriptive Complexity*. Springer-Verlag, New York.
IMMERMAN, N. AND VARDI, M. Y. 1997. Model checking and transitive closure logic. In *Proceedings of the 9th International Conference on Computer-Aided Verification*. Springer-Verlag, 291–302.
KARCHMER, M. 1989. *Communication Complexity: A New Approach to Circuit Depth*. M.I.T. Press, Cambridge, Mass.
WILKE, T. 1999. CTL^+ is exponentially more succinct than CTL. In *Foundations of Software Technology and Theoretical Computer Science: 19th Conference*. 110–121.

Received March 2002; revised July 2002; accepted August 2002