CTL+ Is Exponentially More Succinct Than CTL

Thomas Wilke*

Aachener Informatik-Berichte 99-7†

Herausgeber:
Fachgruppe Informatik der RWTH Aachen
52056 Aachen, Germany

*Author's ddress: Lehrstuhl für Informatik VII, RWTH Aachen, 52056 Aachen, Germany, phone: +49 241 807708, fax: +49 241 8888215, email: wilke@informatik.rwth-aachen.de.

†This electronic version of report 99-7 only differs from its printed version in the layout.

Abstract

It is proved that CTL^+ is exponentially more succinct than CTL. More precisely, it is shown that every CTL formula (and every modal μ-calculus formula) equivalent to the CTL^+ formula $\mathsf{E}(\mathsf{F}p_0 \wedge \cdots \wedge \mathsf{F}p_{n-1})$ is of length at least $\binom{n}{\lceil n/2 \rceil}$, which is $\Omega(2^n/\sqrt{n})$. This matches almost the upper bound provided by Emerson and Halpern, which says that for every CTL^+ formula of length n there exists an equivalent CTL formula of length at most $2^{n \log n}$.

It follows that the exponential blow-up as incurred in known conversions of nondeterministic Büchi word automata into alternation-free μ-calculus formulas is unavoidable. This answers a question posed by Kupferman and Vardi.

The proof of the above lower bound exploits the fact that for every CTL (μ-calculus) formula there exists an equivalent alternating tree automaton of linear size. The core of the proof is an involved cut-and-paste argument for alternating tree automata.

1 Introduction

Expressiveness and *succinctness* are two important aspects to consider when one investigates a (specification) logic. When studying the expressiveness of a logic one is interested in characterizing *what* properties can be expressed, whereas when studying the succinctness one is interested in *how short* a formula can be found to express a given property. Succinctness is especially of importance in a situation where one has characterized the expressive power of a logic by a different but equally expressive logic. In such a situation, succinctness is the foremost quantitative measure to distinguish the logics. For instance, linear-time temporal logic (LTL) is known to be exactly as expressive as first-order logic (FO), [9], but FO is much more succinct than LTL: from work by Stockmeyer's, [11], it follows that there exists a sequence of FO formulas of linear length such that the length of shortest equivalent LTL formulas cannot be bounded by an elementary recursive function.

In this paper, the succinctness of computation tree logic (CTL) is compared to the succinctness of CTL^+, an extension of CTL, which is known to have exactly the same expressive power as CTL, [4, 5]. I present a sequence of CTL^+ formulas of length $\mathcal{O}(n)$ such that the length of shortest equivalent CTL formulas is $\Omega(2^n/\sqrt{n})$. More precisely, I prove that every CTL formula equivalent to the CTL^+ formula

$$\mathsf{E}(\mathsf{F}p_0 \wedge \cdots \wedge \mathsf{F}p_{n-1})$$

is of length at least $\binom{n}{\lceil n/2 \rceil}$, which shows that CTL^+ is exponentially more succinct than CTL. This lower bound is almost tight, because a result by Emerson and Halpern's, [4, 5], says that for every CTL^+ formula of length n there exists an equivalent CTL formula of length at most $2^{n \log n}$.

It is important to note that this exponential lower bound is not based on any complexity-theoretic assumption, and it does not follow from the fact that model checking for CTL is known to be P-complete whereas model checking for CTL^+ is NP- and co-NP-hard (and in Δ_2^p), [3, 4, 5].

The proof of the lower bound presented in this paper makes use of automata-theoretic arguments, following other approaches to similar questions. The main idea is based on the following fact. For every CTL formula (and for every μ-calculus formula) φ there exists an alternating tree automaton $\boldsymbol{A}_\varphi$ of size linear in the length of φ that accepts exactly the models of φ, [6, 1, 2]. So in order to obtain a lower bound on the length of the CTL (or μ-calculus) formulas defining a given class of Kripke structures,[1] it is enough to establish a lower bound on the number of states of the alternating tree automata recognizing the given class of structures.

As mentioned above, automata-theoretic arguments have been used in this way in different places, for instance by Etessami, Vardi, and myself in [8] or Kupferman and Vardi in [10]. The difference, however, is that in this paper the automaton model (alternating automata on trees) is rather intricate compared to the automaton models used in [8] and [10] (nondeterministic automata on words and nondeterministic automata on trees, respectively).

The more elaborate argument that is needed here also answers a question raised in the paper by Kupferman and Vardi. A particular problem the authors consider is constructing for a given nondeterministic Büchi word automaton an alternation free μ-calculus (AFMC) formula that denotes in every Kripke structure the set of all worlds where all infinite paths originating in this world are accepted by the automaton. They show that if such a formula exists, then there is a formula of size at most exponential in the number of states of the given Büchi automaton, but they cannot give a matching lower bound. This is what is provided in this paper.

Outline. In Section 2, the syntax and semantics of CTL and CTL^+ are reviewed and the main result of the paper is presented. In Section 3, alternating tree automata are reviewed and subsequently, in Section 4, the succinctness problem is reduced to an automata-theoretic problem. Section 5 describes this problem in a more general setting, and in Section 6 the solution of this more general problem is presented.

Acknowledgment. I would like to thank Kousha Etessami, Martin Grohe, Neil Immerman, Christof Löding, Philippe Schnoebelen, and Moshe Y. Vardi for having discussed with me the problem addressed in this paper.

Trees and tree arithmetic. In this paper, a tree is a triple (V, E, λ) where (V, E) is a directed tree in the graph-theoretic sense and λ is a labeling function with domain V. By convention, when $\boldsymbol{T}$ denotes a tree, then V, E, and λ always denote the set of nodes, set of edges, and labeling function of $\boldsymbol{T}$. The same applies to decorations such as $\boldsymbol{T}'$, $\boldsymbol{T}^*$, $\boldsymbol{T}_i$, etc.

Let $\boldsymbol{T}$ be an arbitrary tree. A node $v' \in V$ is a *successor* of a node $v \in V$ in $\boldsymbol{T}$ if $(v, v') \in E$. The set of all successors of a node v in $\boldsymbol{T}$ is denoted by $\mathit{Scs}(\boldsymbol{T}, v)$.

[1] Strictly speaking, a CTL formula defines a class of pointed Kripke structures, see Section 2.

The set of *leaves* of a tree $\boldsymbol{T}$, that is, the set of nodes without successors, is denoted by $\mathit{Lvs}(\boldsymbol{T})$. The set of *inner nodes* is denoted by $\mathit{Inn}(\boldsymbol{T})$.

Given a tree $\boldsymbol{T}$ and a vertex v of $\boldsymbol{T}$, the *ancestors path,* denoted $\boldsymbol{T}{\uparrow}v$, is the unique path from the root of $\boldsymbol{T}$ to v (inclusively). The *descendants tree,* denoted $\boldsymbol{T}{\downarrow}v$, is the subgraph of $\boldsymbol{T}$ induced by all nodes reachable from v (v itself included).

I will use two kinds of concatenations for trees. When $\boldsymbol{T}$ and $\boldsymbol{T'}$ are trees and v is a node of $\boldsymbol{T}$, then $\boldsymbol{T} \cdot (v, \boldsymbol{T'})$ denotes the tree that results from $\boldsymbol{T}$ by first making an isomorphic copy of $\boldsymbol{T'}$ whose node set is disjoint from the set of nodes of $\boldsymbol{T}$ and then adding an edge from v to the root of $\boldsymbol{T'}$. Similarly, $\boldsymbol{T} \odot (v, \boldsymbol{T'})$ denotes the tree that results from $\boldsymbol{T}$ by first making an isomorphic copy of $\boldsymbol{T'}$ whose node set is disjoint from the set of nodes of $\boldsymbol{T}$ and then identifying the root of the isomorphic copy of $\boldsymbol{T'}$ with v. By convention, the node v is retained in the resulting tree (rather than the root of $\boldsymbol{T'}$) and the label of v is kept.— These two concatenation operations are extended in a straightforward way: when $\boldsymbol{T}$ is a tree and M a set of pairs $(v, \boldsymbol{T'})$, with $v \in V$ and $\boldsymbol{T'}$ an arbitrary tree, I might write $\boldsymbol{T} \cdot M$ and $\boldsymbol{T} \odot M$ to denote the result of concatenating (in the respective way) all trees from M to $\boldsymbol{T}$.

For ease in notation, when π is a finite path (a finite tree with exactly one leaf) with last node v and $\boldsymbol{T}$ is a tree, I simply write $\pi \cdot \boldsymbol{T}$ for the tree $\pi \cdot (v, \boldsymbol{T})$ as defined above. To make things even simpler, strings and finite paths are identified. So when u is a string and $\boldsymbol{T}$ a tree, I might write $u \cdot \boldsymbol{T}$ to denote the tree which is obtained by viewing u as a path and concatenating $\boldsymbol{T}$ to it.

2 CTL, CTL+, and Main Result

I start with recalling the syntax and the semantics of CTL and CTL+. For technical reasons, I only define formulas in positive normal form. This is not an essential restriction, because every CTL formula is equivalent to a CTL formula in positive normal form of the same length, and the same applies to CTL+.

Syntax of CTL. Let Prop $= \{p_0, p_1, p_2, \dots\}$ be an infinite supply of distinct propositional variables. The set of all *CTL formulas* is the smallest set satisfzing the following conditions.

1. 0 and 1 are CTL formulas.
2. For $p \in$ Prop, p and $\neg p$ are CTL formulas.
3. If φ and ψ are CTL formulas, then so are $\varphi \vee \psi$ and $\varphi \wedge \psi$.

4. If φ is a CTL formula, then so are $\mathsf{EX}\varphi$ and $\mathsf{AX}\varphi$.
5. If φ and ψ are CTL formulas, then so are $\mathsf{EU}(\varphi, \psi)$, $\mathsf{AU}(\varphi, \psi)$, $\mathsf{ER}(\varphi, \psi)$, and $\mathsf{AR}(\varphi, \psi)$.

It is important that every path quantifier (E and A) is immediately followed by a temporal modality such as X, U, or R. In this respect, CTL^+ is less restrictive; it allows to throw in boolean connectives between path quantifiers and temporal modalities.

Syntax of CTL^+. The formal definition of the syntax of CTL^+ requires a definition of the notion of a path formula. The set of all *CTL^+ formulas* and the set of all *path formulas* are the smallest sets satisfying the following conditions.

1. 0 and 1 are CTL^+ formulas.
2. For $p \in \text{Prop}$, p and $\neg p$ are CTL^+ formulas.
3. If φ and ψ are CTL^+ formulas, then so are $\varphi \vee \psi$ and $\varphi \wedge \psi$.
4. Every CTL^+ formula is a path formula.
5. If φ and ψ are CTL^+ formulas, then $\mathsf{X}\varphi$, $\mathsf{U}(\varphi, \psi)$ and $\mathsf{R}(\varphi, \psi)$ are path formulas.
6. If φ and ψ are path formulas, then so are $\varphi \vee \psi$ and $\varphi \wedge \psi$.
7. If φ is a path formula, then $\mathsf{E}\varphi$ and $\mathsf{A}\varphi$ are CTL^+ formulas.

Obviously, every CTL formula is a CTL^+ formula.

I will use the standard shorthand $\mathsf{F}\varphi$ for $\mathsf{U}(1, \varphi)$ when φ is an arbitrary CTL^+ formula.

Kripke structures. CTL and CTL^+ formulas are interpreted in Kripke structures, which are directed graphs with specific labeling functions for their nodes. Formally, a *Kripke structure* is a tuple

$$\boldsymbol{K} = (W, R, \alpha) \tag{1}$$

where

- W is a set of *worlds,*
- $R \subseteq W \times W$ is an *accessibility relation,* and
- $\alpha \colon W \to 2^{\text{Prop}}$ is a *labeling function,* which assigns to each world the set of propositional variables that hold true in it.

Just as with trees I will follow the convention that whenever a Kripke structure is denoted by $\boldsymbol{K}$, then its components are denoted W, R, and α, and the same applies to decorations such as $\boldsymbol{K'}$, $\boldsymbol{K^*}$, and so on.

Given a world w of a Kripke structure $\boldsymbol{K}$ as above, a world w' is called a *successor* of w in $\boldsymbol{K}$ if $(w, w') \in R$. Just as with trees, the set of all successors of a world w is denoted by $\mathit{Scs}(\boldsymbol{K}, w)$. A *path* through a Kripke structure $\boldsymbol{K}$ as above is a nonempty sequence $w_0, w_1, \dots$ such that $(w_0, w_1) \in R$, $(w_1, w_2) \in R$, $\dots$ A *maximal path* is a path that is either infinite or finite and ends in a world without successor.

A *pointed Kripke structure* is a pair $(\boldsymbol{K}, w)$ of a Kripke structure and a *distinguished* world of it. A *path* through a pointed Kripke structure is a path through this structure starting in the distinguished world. A *path-equipped Kripke structure* is a pair $(\boldsymbol{K}, \pi)$ of a Kripke structure and a maximal path through it.

Semantics of CTL and CTL⁺. As usual, I define what it means for a CTL⁺ formula to hold in a pointed Kripke structure, denoted $(\boldsymbol{K}, w) \models \varphi$, and simultaneously, what it means for a path formula to hold in a path-equipped Kripke structure, denoted $(\boldsymbol{K}, \pi) \models \varphi$.

The boolean constants 0 and 1, the boolean connectives $\vee$ and $\wedge$, and the propositional variables and its negations are dealt with in the usual way.

When $(\boldsymbol{K}, \pi)$ is a path-equipped Kripke structure with $\pi = w_0, w_1, \dots$ and φ is a CTL⁺ formula, then

- $(\boldsymbol{K}, \pi) \models \mathsf{X}\varphi$ if π has length at least 2 and $(\boldsymbol{K}, w_1) \models \varphi$.

Similarly, when φ and ψ are CTL⁺ formulas, then

- $(\boldsymbol{K}, \pi) \models \mathsf{U}(\varphi, \psi)$ if there exists $i \geq 0$ such that
  - $(\boldsymbol{K}, w_j) \models \varphi$ for every $j < i$, and
  - $(\boldsymbol{K}, w_i) \models \psi$,
- $(\boldsymbol{K}, \pi) \models \mathsf{R}(\varphi, \psi)$ if for every $i \geq 0$,
  - $(\boldsymbol{K}, w_i) \models \psi$ or
  - there exists $j < i$ such that $(\boldsymbol{K}, w_j) \models \varphi$.

When φ is a path formula, then

- $(\boldsymbol{K}, w) \models \mathsf{E}\varphi$ if there exists a maximal path π through $(\boldsymbol{K}, w)$ such that $(\boldsymbol{K}, \pi) \models \varphi$, and
- $(\boldsymbol{K}, w) \models \mathsf{A}\varphi$ if $(\boldsymbol{K}, \pi) \models \varphi$ for all maximal paths π through $(\boldsymbol{K}, w)$.

Given a CTL⁺ formula φ, I write $\mathrm{Mod}(\varphi)$ for the class of all pointed Kripke structures that are models of φ, i. e., $\mathrm{Mod}(\varphi) = \{(\boldsymbol{K}, w) \mid (\boldsymbol{K}, w) \models \varphi\}$. CTL⁺ formulas φ and ψ are *equivalent* if they have the same models, i. e., if $\mathrm{Mod}(\varphi) = \mathrm{Mod}(\psi)$.

Main Result. The main result of this paper is:

Theorem 1 *For every $n > 0$, let φ_n be the CTL^+ formula defined by*

$$\varphi_n = \mathsf{E}(\mathsf{F}p_0 \wedge \cdots \wedge \mathsf{F}p_{n-1}) \ . \tag{2}$$

Every CTL formula equivalent to φ_n has length at least $\binom{n}{\lceil n/2 \rceil}$, which clearly is $\Omega(2^n/\sqrt{n})$.

In other words, CTL^+ is exponentially more succinct than CTL.

It is easy to come up with a formula of length $\mathcal{O}(n!)$ equivalent to φ_n:

$$\bigvee_{\beta} \mathsf{EF}(p_{\beta(0)} \wedge \mathsf{EF}(p_{\beta(1)} \wedge \cdots \wedge \mathsf{EF}p_{\beta(n-1)}) \dots) \tag{3}$$

where β ranges over all permutations on $\{0, \dots, n-1\}$.

3 Alternating Tree Automata

As indicated in the abstract and the introduction, I will use an automata-theoretic argument to prove Theorem 1. In this section, the automaton model I work with is introduced. It differs from other models used in the literature in several respects. First, it can handle trees with arbitrary degree of branching in a simple way. Second, the class of objects accepted by an automaton as used here is a class of pointed Kripke structures rather than just a set of trees. Both facts make it much easier to phrase theorems such as Theorem 2 below and also simplify the presentation of a combinatorial (lower-bound) argument like the one given in Section 6.

Format. An *alternating tree automaton (ATA)* is a tuple

$$\boldsymbol{A} = (Q, P, q_I, \delta, \Omega) \tag{4}$$

where

- Q is a finite set of *states,*
- P is a finite subset of Prop,
- $q_I \in Q$ is an *initial state,*
- δ is a *transition function* as specified below, and
- Ω is an *acceptance condition* for ω-automata such as a Büchi or Muller condition.

The transition function δ is a function $Q \times 2^P \to \mathrm{TC}(Q)$, where $\mathrm{TC}(Q)$ is the set of *transition conditions* over Q, which is defined to be the smallest set satisfying the following conditions.

1. 0 and 1 are transition conditions over Q.
2. For every $q \in Q$, q is a transition condition over Q.
3. For every $q \in Q$, $\Box q$ and $\Diamond q$ are transition conditions over Q.
4. If φ and ψ are transition conditions over Q, then $\varphi \wedge \psi$ and $\varphi \vee \psi$ are transition conditions over Q.

A transition condition is said to be *ϵ-free* if 2. is not needed to build the condition. An ATA is said to be ϵ-free if every condition $\delta(q, a)$ for $q \in Q$ and $a \in 2^P$ is ϵ-free; it is said to be in *normal form* if it is ϵ-free and every condition $\delta(q, a)$ is in disjunctive normal form.

I will use a notational convention with ATA's analogous to the one used with trees and Kripke structures.

Behavior. ATA's work on pointed Kripke structures. Their computational behavior is explained using the notion of a run.

Assume $\boldsymbol{A}$ is an ATA as above and $(\boldsymbol{K}, w_I)$ a pointed Kripke structure as above. A *run* of $\boldsymbol{A}$ on $(\boldsymbol{K}, w_I)$ is a $(W \times Q)$-labeled tree

$$\boldsymbol{R} = (V, E, \lambda) \tag{5}$$

satisfying the conditions described further below.

To explain these conditions, we need some more definitions. For simplicity in notation, I will write $w_{\boldsymbol{R}}(v)$ and $q_{\boldsymbol{R}}(v)$ for the first and second component of $\lambda(v)$, respectively.

For every node v of $\boldsymbol{R}$, I define what it means for a transition condition τ over Q to hold in v, denoted $\boldsymbol{K}, \boldsymbol{R}, v \models \tau$. This definition is by induction on the structure of τ, where the boolean constants 0 and 1 and the boolean connectives are dealt with in the usual way. Further:

– $\boldsymbol{K}, \boldsymbol{R}, v \models q$ if there exists $v' \in \mathit{Scs}(\boldsymbol{R}, v)$ such that $\lambda(v') = (w_{\boldsymbol{R}}(v), q)$,
– $\boldsymbol{K}, \boldsymbol{R}, v \models \Diamond q$ if there exists $v' \in \mathit{Scs}(\boldsymbol{R}, v)$ such that $q_{\boldsymbol{R}}(v') = q$ and $w_{\boldsymbol{R}}(v') \in \mathit{Scs}(\boldsymbol{K}, w_{\boldsymbol{R}}(v))$, and
– $\boldsymbol{K}, \boldsymbol{R}, v \models \Box q$ if for every $w \in \mathit{Scs}(\boldsymbol{K}, w_{\boldsymbol{R}}(v))$ there exists $v' \in \mathit{Scs}(\boldsymbol{R}, v)$ such that $\lambda(v') = (w, q)$.

I can now state the two additional conditions that are required of a run.

1. *Initial condition.* Let v_0 be the root of (V, E). Then $\lambda(v_0) = (w_I, q_I)$.

2. *Local consistency.* For every $v \in V$,

$$\boldsymbol{K}, \boldsymbol{R}, v \models \delta(q_{\boldsymbol{R}}(v), L(w_{\boldsymbol{R}}(v)) \cap P) \ . \tag{6}$$

Note that the intersection with P allows us to deal easily with the fact that in our definition of Kripke structure an infinite number of propositional variables is always present.

A run $\boldsymbol{R}$ is said to be *accepting* if the state labeling of every infinite path through $\boldsymbol{R}$ satisfies the given *acceptance condition* Ω. For instance, if $\Omega \subseteq 2^Q$ is a Muller condition, then every infinite path $v_0, v_1, \dots$ through $\boldsymbol{R}$ must have the property that the set formed by the states occurring infinitely often in $q_{\boldsymbol{R}}(v_0), q_{\boldsymbol{R}}(v_1), \dots$ is a member of Ω.

A pointed Kripke structure is *accepted* by $\boldsymbol{A}$ if there exists an accepting run of $\boldsymbol{A}$ on the Kripke structure. The class of pointed Kripke structures accepted by $\boldsymbol{A}$ is denoted by $\mathcal{K}(\boldsymbol{A})$; it is said to be the class of pointed Kripke structures that is *recognized* by $\boldsymbol{A}$.

4 Reduction to Automata-Theoretic Problem

In order to reduce the lower bound claim for the translation from CTL^+ to CTL to a claim on alternating automata, I describe the models of a CTL formula by an alternating tree automaton, following the ideas of Kupferman, Vardi, and Wolper, [2], but using the automaton model introduced in the previous section.

Let φ be an arbitrary CTL formula and P the set of propositional variables occurring in φ. The ATA $\boldsymbol{A}_\varphi$ is defined by

$$\boldsymbol{A}_\varphi = (Q, P, \varphi, \delta, \Omega) \tag{7}$$

where

- Q is the set of all CTL subformulas of φ including φ itself,
- Ω is the Muller acceptance condition that contains all sets of subformulas of φ that do not contain formulas starting with EU, or AU, and
- δ is defined by induction as follows. The induction base and the rules for $\wedge$,

$\vee$, and X are:

$$
\begin{aligned}
\delta(0, a) &= 0\,, & \delta(1, a) &= 1\,, \\
\delta(p, a) &= \begin{cases} 1\,, & \text{if } p \in a, \\ 0\,, & \text{else,} \end{cases} & \delta(\neg p, a) &= \begin{cases} 1\,, & \text{if } p \notin a, \\ 0\,, & \text{else,} \end{cases} \\
\delta(\psi \wedge \chi, a) &= \psi \wedge \chi\,, & \delta(\psi \vee \chi, a) &= \psi \vee \chi\,, \\
\delta(\mathsf{EX}\psi, a) &= \Diamond\psi\,, & \delta(\mathsf{AX}\psi, a) &= \Box\psi\,.
\end{aligned}
$$

The rules for U are:

$$\delta(\mathsf{EU}(\psi, \chi), a) = \chi \vee (\psi \wedge \Diamond\mathsf{EU}(\psi, \chi))\,, \tag{8}$$

$$\delta(\mathsf{AU}(\psi, \chi), a) = \chi \vee (\psi \wedge \Diamond\mathsf{EU}(\psi, \chi) \wedge \Box\mathsf{AU}(\psi, \chi))\,. \tag{9}$$

The rules for R are perfectly dual to the rules for U:

$$\delta(\mathsf{ER}(\psi, \chi), a) = \chi \wedge (\psi \vee \Box\mathsf{ER}(\psi, \chi) \vee \Diamond\mathsf{ER}(\psi, \chi))\,, \tag{10}$$

$$\delta(\mathsf{AR}(\psi, \chi), a) = \chi \wedge (\psi \vee \Box\mathsf{ER}(\psi, \chi))\,. \tag{11}$$

Note that on the right-hand sides of the above equations the boolean connectives $\vee$ and $\wedge$ are part of the syntax of transition conditions, whereas on the left-hand sides they are part of CTL formulas. I should also mention that the additional conjunct $\Diamond\mathsf{EU}(\psi, \chi)$ in the rule for $\mathsf{AU}(\psi, \chi)$ and the additional disjunct $\Box\mathsf{ER}(\psi, \chi)$ take care of worlds without successors (dead ends). To simplify this, one could as well introduce an always successful state q_S (with $\delta(q_S, a) = 1$ for every $a \in 2^P$) and an always failing state q_F (with $\delta(q_F, a) = 0$ for every $a \in 2^P$) and replace the additional conjunct by $\Diamond q_S$ ("there exists a succesor") and the additional disjunct by $\Box q_F$ ("there exists no successor").

Similar to [2], we prove:

Theorem 2 *Let φ be an arbitrary CTL formula of length l. Then $\boldsymbol{A}_\varphi$ is an ATA with at most l states such that Mod$(\varphi) = \mathcal{K}(\boldsymbol{A}_\varphi)$.*

Proof. The proof goes by induction on the length of φ, the base cases being trivial. Of the two boolean connectives, $\vee$ and $\wedge$, I treat only one, namely conjunction; disjunction is dealt with in the same way. Similarly, I treat only EX of the two operators involving X; the universal version is dealt with in the same way.

In the following let $(\boldsymbol{K}, w)$ be an arbitrary pointed Kripke structure.

Assume $\varphi = \psi \wedge \chi$ and that $\boldsymbol{A}_\psi$ and $\boldsymbol{A}_\chi$ work correctly. Further, assume $(\boldsymbol{K}, w) \models \varphi$. Then, by definition of the semantics of CTL, $(\boldsymbol{K}, w)$ is a model of ψ and of χ, which means there are accepting runs $\boldsymbol{R_0}$ and $\boldsymbol{R_1}$ of $\boldsymbol{A}_\varphi$ and $\boldsymbol{A}_\psi$ on $(\boldsymbol{K}, w)$. Let $\boldsymbol{T}$ be a one-node tree with root label (w, φ) and suppose v is the root of $\boldsymbol{T}$. Clearly, $\boldsymbol{T} \cdot \{(v, \boldsymbol{R_0}), (v, \boldsymbol{R_1})\}$ is an accepting run of $\boldsymbol{A}_\varphi$ on $(\boldsymbol{K}, w)$. Conversely, if $\boldsymbol{R}$ is an accepting run of $\boldsymbol{A}_\varphi$ on a pointed Kripke struckture $(\boldsymbol{K}, w)$, then, by definition of the transition function and because of local consistency, the root of $\boldsymbol{R}$ must have two successors v_0 and v_1 labeled (w, ψ) and (w, χ), respectively. The subtrees $\boldsymbol{R}{\downarrow}v_0$ and $\boldsymbol{R}{\downarrow}v_1$ are accepting runs of $\boldsymbol{A}_\psi$ and $\boldsymbol{A}_\chi$ on $(\boldsymbol{K}, w)$. By induction hypothesis, this means $(\boldsymbol{K}, w) \models \psi$ and $(\boldsymbol{K}, w) \models \chi$, which implies $(\boldsymbol{K}, w) \models \varphi$.

Next, let $\varphi = \mathsf{EX}\psi$. First, assume $(\boldsymbol{K}, w) \models \varphi$. Then, by definition of the semantics of CTL, there exists $w' \in \mathit{Scs}(\boldsymbol{K}, w)$ such that $(\boldsymbol{K}, w') \models \psi$. By induction hypothesis, there is an accepting run $\boldsymbol{R}$ of $\boldsymbol{A}_\psi$ on $(\boldsymbol{K}, w)$. As above, this run can be easily modified so as to become an accepting run of $\boldsymbol{A}_\varphi$ on $(\boldsymbol{K}, w)$: let $\boldsymbol{T}$ be a one-node tree with root v labeled (w, φ); then $\boldsymbol{T} \cdot (v, \boldsymbol{R})$ is such a run. Conversely, assume $\boldsymbol{R}$ is an accepting run of $\boldsymbol{A}_\varphi$ on $(\boldsymbol{K}, w)$. Because of local consistency, the root of $\boldsymbol{R}$ has a successor v labeled (w', ψ) where w' is a successor of w in $\boldsymbol{K}$. As $\boldsymbol{R}$ is an accepting run of $\boldsymbol{A}_\varphi$, $\boldsymbol{R}{\downarrow}w'$ is an accepting run of $\boldsymbol{A}_\psi$. Thus, by induction hypothesis, $(\boldsymbol{K}, w') \models \psi$, hence $(\boldsymbol{K}, w) \models \varphi$.

Now, let $\varphi = \mathsf{EU}(\psi, \chi)$. First, assume $(\boldsymbol{K}, w) \models \varphi$. Then there exists a maximal path $\pi = w_0, w_1, w_2, \dots$ and $i \geq 0$ such that

- $(\boldsymbol{K}, w_j) \models \varphi$ for every $j < i$, and
- $(\boldsymbol{K}, w_i) \models \psi$.

The inductive assumption implies:

- there exists an accepting run $\boldsymbol{R_j}$ of $\boldsymbol{A}_\psi$ on $(\boldsymbol{K}, w_j)$ for every $j < i$, and
- there exists an accepting run $\boldsymbol{R_i}$ of $\boldsymbol{A}_\chi$ on $(\boldsymbol{K}, w_i)$.

Let $\boldsymbol{T}$ be an arbitrary path of length i with nodes $v_0, \dots, v_i$ such that every node v_j is labeled (w_j, φ). Consider the tree $\boldsymbol{R}$ defined by

$$\boldsymbol{R} = \boldsymbol{T} \cdot \{(v_j, \boldsymbol{R_j}) \mid j \leq i\} \ .$$

I claim this tree is an accepting run of $\boldsymbol{A}_\varphi$ on $(\boldsymbol{K}, w)$. The only thing to verify is local consistency for the nodes $v_0, \dots, v_i$. So let $j \leq i$. There are two cases to distinguish. First, assume $j < i$. Remember that v_j is labeled (w_j, φ) and recall (8). Clearly, the second disjunct holds true in v_j: the root of $\boldsymbol{R_j}$, which is a successor of v_j, takes care of the first conjunct, and v_{j+1} takes care of the second conjunct. Second, consider v_i. Remember that by construction v_i has a successor labeled (w_i, χ). So the first disjunct of (8) holds in v_i.

For the converse, assume there is an accepting run $\boldsymbol{R}$ of $\boldsymbol{A}_\varphi$ on $(\boldsymbol{K}, w)$. Because of local consistency, (8) guarantees one of the following.

1. There exists an infinite path $v_0, v_1, \ldots$ through $\boldsymbol{R}$ starting in the root of $\boldsymbol{R}$ such that $\lambda(v_0) = (w, \varphi)$ and for every j the following holds.
   (a) $q_{\boldsymbol{R}}(v_j) = \varphi$ and $w_{\boldsymbol{R}}(v_{j+1}) \in \mathit{Scs}(\boldsymbol{K}, w_{\boldsymbol{R}}(v_j))$.
   (b) There exists $v'_j \in \mathit{Scs}(\boldsymbol{R}, v_j)$ with $q_{\boldsymbol{R}}(v'_j) = \psi$ and $w_{\boldsymbol{R}}(v'_j) = w_{\boldsymbol{R}}(v_j)$.
2. There exists a finite path $v_0, \ldots, v_i$ through $\boldsymbol{R}$ starting in the root of $\boldsymbol{R}$ such that for every $j < i$, the aforementioned conditions 1.(a) and 1.(b) hold. Further, 1.(a) holds for $j = i$ and there exists $v'_i \in \mathit{Scs}(\boldsymbol{R}, v_i)$ with $q_{\boldsymbol{R}}(v'_i) = \chi$ and $w_{\boldsymbol{R}}(v'_i) \in \mathit{Scs}(\boldsymbol{K}, w_{\boldsymbol{R}}(v_i))$.

The first option is impossible: it implies that $\varphi = \mathsf{EU}(\psi, \chi)$ occurs infinitely often on the path $v_0, v_1, \ldots$, contradicting the acceptance condition. So 2. holds true. Let's rewrite 2. using the induction hypothesis: there exists a finite path $w_0, \ldots, w_i$ starting with w_0 such that for every $j < i$, $(\boldsymbol{K}, w_j) \models \psi$ and $(\boldsymbol{K}, w_i) \models \chi$. Extending $w_0, \ldots, w_n$ to a maximal path shows $(\boldsymbol{K}, w) \models \varphi$.

The argument for $\varphi = \mathsf{AU}(\psi, \chi)$ is similar to the argument for $\mathsf{EU}(\psi, \chi)$. In the first part, where it needs to be shown that if a pointed Kripke structure satisfies φ then it is accepted by $\boldsymbol{A}_\varphi$, one uses that if $(\boldsymbol{K}, w) \models \varphi$ then there exist set $W_0, W_1 \subseteq W$ such that

- ψ holds in every world of W_0,
- χ holds in every world of W_1,
- on every maximal path through $(\boldsymbol{K}, w)$ worlds from W_0 appear until a world from W_1 appears.

Using this and the induction hypothesis, one constructs an accepting run of $\boldsymbol{A}_\varphi$ on $(\boldsymbol{K}, w)$ similar to above. In the second part of the proof, where it needs to be shown that if $\boldsymbol{A}_\varphi$ accepts then $(\boldsymbol{K}, w) \models \varphi$, the additional conjunct $\Diamond\mathsf{EU}(\psi, \chi)$ guarantees that in 1. and 2. above the path is infinite respectively ends in a node that satisfies the additional condition mentioned under 2.

The proofs for $\varphi = \mathsf{ER}(\psi, \chi)$ and $\varphi = \psi\chi$ are similar to the proofs for $\varphi = \mathsf{EU}(\varphi, \chi)$ and $\varphi = \mathsf{EU}(\varphi, \psi)$, respectively.[2] $\square$

It should be noted that in the above definition of $\boldsymbol{A}_\varphi$ one could use a mode of acceptance simpler (and weaker) than Muller's mode, but this is not relevant in this context. See also [2].

So in order to prove Theorem 1 we only need to show:

Theorem 3 *Every ATA recognizing $\mathit{Mod}(\varphi_n)$ has at least $\binom{n}{\lceil n/2 \rceil}$ states.*

[2]As $\mathsf{ER}(\psi, \chi)$ is the negation of $\mathsf{AU}(\neg\psi, \neg\chi)$ and $\mathsf{AR}(\psi, \chi)$ is the negation of $\mathsf{EU}(\neg\psi, \neg\chi)$ we could also use the fact that if one exchanges $\vee$ with $\wedge$ and 0 with 1 in an ATA then the resulting ATA recognizes the complement class. But this we haven't proved here. Of course, it follows from determinacy theorems such as the one given in [12].

This is what we are going to show in the two subsequent sections.

The following proposition states that without loss of generality we can restrict our considerations to ATA's in normal form.

Proposition 1 *For every ATA there exists an equivalent ATA in normal form with the same number of states.*

Proof. Clearly, it is enough to show that for every ATA there exists an equivalent ϵ-free ATA with the same number of states, and this is what I prove.

The construction that is used in the proof of this proposition is similar to the class-room construction that is used to convert NFA's into ϵ-free NFA's by building "ϵ-closures." The situation here is only a little more involved; one has to define what the ϵ-closure of a transition condition is.

The formal proof goes as follows. Assume $\boldsymbol{A}$ is an arbitrary ATA. Let $\boldsymbol{A}^*$ be the ATA that results from $\boldsymbol{Q}$ as follows.

– The state set is replaced by $Q \cup (\{\Diamond, \Box\} \times Q)$.
– In every transition condition $\delta(q, a)$ subformulas of the form $\Diamond q$ and $\Box q$ are replaced by $(\Diamond, q)$ and $(\Box, q)$, respectively.
– For every state $q \in Q$, $\delta^*(\Diamond q, a)$ and $\delta^*(\Box q, a)$ are set to 1.

For every state $q \in Q$, denote by $\boldsymbol{A}_q$ the ATA that results from $\boldsymbol{A}^*$ by changing its initial state to q. Observe that in the values of the transition function of the $\boldsymbol{A}_q$'s neither $\Box$ nor $\Diamond$ occurs.

Let $q \in Q$ and $a \in 2^P$. A set $M \subseteq \{\Diamond, \Box\} \times Q$ is an *ϵ-option* of (q, a) if there exists an accepting run $\boldsymbol{R}$ of $\boldsymbol{A}_q$ on the pointed one-node Kripke structure a such that M is the of all states from $\{\Diamond, \Box\} \times Q$ occurring in $\boldsymbol{R}$. The ϵ-closure of (q, a), denoted $\epsilon(q, a)$, contains all ϵ-options for (q, a).

For every set $M \subseteq \{\Diamond, \Box\} \times Q$ let τ_M be a conjunction which for every element (X, q) contains the formula Xq as a conjunct but no other conjuncts.

A *choice function* for (q, a) is a function mapping every subformula q' of $\delta(q, a)$ to an element of $\epsilon(q', a)$. Given a choice function β for (q, a), the formula $\beta[a, q]$ is obtained from $\delta(q, a)$ by replacing every occurrence of a subformula q' by $\tau_{\beta(q')}$.

To obtain an ϵ-free ATA equivalent to $\boldsymbol{A}$, one only needs to modify the transition function as follows. For every $(q, a) \in Q \times 2^P$, replace $\delta(q, a)$ by the disjunction $\bigvee_\beta [q, a]$ where β ranges over all choice function for (q, a). So, in particular, if there is no choice function β for (q, a), the disjunction is empty, which, as usual, is interpreted as false. Formally, the resulting transition condition is 0.

It should be clear that the resulting ATA is an ϵ-free ATA equivalent to $\boldsymbol{A}$. $\Box$

In other words, all we need to show is:

Theorem 3* *Every ATA in normal form recognizing Mod*(φ_n) *has at least* $\binom{n}{\lceil n/2 \rceil}$ *states.*

5 The General Setting

The method I use to prove Theorem 3* (a cut-and-paste argument) does not only apply to the specific properties defined by the φ_n's but to a large class of "path properties." As with many other situations, the method is best understood when presented in its full generality. In this section, I explain the general setting and present the extended version of Theorem 3*, namely Theorem 4.

In the following, *word* stands for nonempty string or ω-word. The set of all words over a given alphabet A is denoted by A^∞. A *language* is a subset of $(2^P)^\infty$ where P is some finite subset of Prop. Given a language L over some alphabet 2^P, $\mathsf{E}L$ denotes the class of pointed Kripke structures $(\boldsymbol{K}, w)$ where there exists a maximal path through $(\boldsymbol{K}, w)$ whose labeling (restricted to the propositional variables in P) is an element of L. (Remember that a path through a pointed Kripke structure always starts in its distinguished world.)

Observe that for every n, we clearly have $\text{Mod}(\varphi_n) = \mathsf{E}L_n$ where

$$L_n = \{a_0 a_1 \cdots \in P_n^\infty \mid \forall i (i < n \rightarrow \exists j (p_i \in a_j))\}$$

and $P_n = \{p_0, \dots, p_{n-1}\}$.

Let L be a regular language. We say a family $\{(u_i, u_i')\}_{i<m}$ is a *discriminating family* for L if $u_i u_i' \in L$ and $u_i u_j' \notin L$ for all $i < m$ and all $j < m$ with $j \neq i$. Obviously, the number of classes of the Nerode congruence[3] associated with L is an upper bound for m. The maximum number m such that there exists a discriminating family of that size for L is denoted $\iota(L)$.

The generalized version of Theorem 3* now reads:

Theorem 4 *Let L be a regular language. Then every ATA recognizing $\mathsf{E}L$ has at least $\iota(L)$ states.*

Before we turn to the proof of this theorem in the next section, let's apply it to the languages L_n (as defined above) to obtain the desired lower bounds.

[3]The Nerode congruence of a language L is the congruence that considers strings u and v equivalent if for every word x (including the empty word), $ux \in L$ iff $vx \in L$.

Fix an arbitrary positive natural number $n > 1$ and let $m = \lceil n/2 \rceil$ and $t = \binom{n}{m}$. Write N for the set $\{0, \ldots, n-1\}$ and $\bar{\ }$ for set-theoretic complementation with respect to N. For every $M \subseteq N$, let $u(M)$ be a string over 2^{P_n} of length $|M|$ such that for every $p_i \in M$, the letter $\{p_i\}$ occurs in $u(M)$. (In other words, $u(M)$ should be a sequence of singletons where for each $i \in M$ the singleton $\{p_i\}$ occurs exactly once and no other singleton occurs.) Let $M_0, \ldots, M_{t-1}$ be an enumeration of all m-subsets of N and let $u_i = u(M_i)$ and $u'_i = u(\bar{M}_i)$. Then $\{(u_i, u'_i)\}_{i<t}$ is a discriminating family for L_n, which means $\iota(L_n) \geq \binom{n}{\lceil n/2 \rceil}$. So from Theorem 4, we can conclude that Theorem 3* is true, which means Theorem 1 is true as well. (Observe that for $n = 1$ the claim is trivial.)

The only thing left is the proof of Theorem 4. This is what we will do in the next section.

6 Saturation

In this section, where our objective is to prove Theorem 4, we will see trees in two different roles. On the one hand, we will look at runs of ATA's, and runs of ATA's are trees by definition. On the other hand, we will consider Kripke structures that are trees. In order to not get confused, I will strictly follow the notational conventions introduced earlier, for instance, that the labeling function of a run $\boldsymbol{R'}$ is referred to by λ'. As we will only work with Kripke structures that are trees, I will use the term *Kripke tree.* A Kripke tree will also be viewed as a pointed Kripke structure where the root of the tree is the distinguished node.

For the rest of this section, fix a language L over some alphabet 2^P, and an arbitrary ATA $\boldsymbol{A}$. For each state q, write $\boldsymbol{A}_q$ for the ATA that results from $\boldsymbol{A}$ by changing its initial state to q and $\mathcal{K}_q$ for the class $\mathcal{K}(\boldsymbol{A}_q)$, the class of pointed Kripke structures recognized by $\boldsymbol{A}_q$.

As stated above, the main argument in the proof of Theorem 4 will be a cut-and-paste argument for runs of alternating tree automata. Clearly, transition conditions of the form $\Box q$ make runs complicated, because they require to consider all successors of a world (rather than just one). So we will try to "avoid" as many $\Box q$ conditions as possible. Formally, we use the following definition.

Let u be a string, $\boldsymbol{K}$ a Kripke tree, and q a state of $\boldsymbol{A}$. The Kripke tree $\boldsymbol{K}$ *avoids* q in u if $u \cdot \boldsymbol{K} \notin \mathsf{E}L$ and $\boldsymbol{K} \notin \mathcal{K}_q$. If there exists a Kripke tree avoiding q in u, then q is *avoidable* in u. The set of all states avoidable in u is denoted by *avd*(u). For every state $q \in$ *avd*(u) we pick once and for all a Kripke tree avoiding q in u and denote it by $\boldsymbol{K}^u_q$.

The important observation here is that if $\boldsymbol{K}$ is a Kripke tree, $w \in W$, and $q \in \mathit{avd}(\boldsymbol{K}{\uparrow}w)$, then $\boldsymbol{K'}$ defined by $\boldsymbol{K'} = \boldsymbol{K} \cdot (w, \boldsymbol{K}^u_q)$ with $u = \boldsymbol{K}{\uparrow}w$ has the following two properties. First, if $\boldsymbol{K} \notin \mathsf{E}L$, then $\boldsymbol{K'} \notin \mathsf{E}L$. Second, there exists no accepting run $\boldsymbol{R}$ of $\boldsymbol{A}$ on $\boldsymbol{K'}$ that has a node v with $w_{\boldsymbol{R}}(v) = w$ and $\boldsymbol{K'}, \boldsymbol{R}, v \models \Box q$. In a certain sense, by adding $\boldsymbol{K}^q_u$, the condition $\Box q$ is "avoided" in w.

So adding avoiding trees as subtrees avoids conditions of the form $\Box q$. On the other hand, the added subtrees can potentially satisfy conditions of the form $\Diamond q$ which were not satisfied before. This leads to the following definition.

Let q, $\boldsymbol{K}$, and u as above. Further, let q' be another state of $\boldsymbol{A}$. The state q *makes* q' *successful* in u if $\boldsymbol{K}^u_q \in \mathcal{K}_{q'}$. If there exists a state q making q' successful in u, then q' is *successful* in u. The set of all states successful in u is denoted by $\mathit{scf}(u)$. For every state $q \in \mathit{scf}(u)$, we pick once and for all a state making q successful in u and denote it by q^u.

A world w in a Kripke tree $\boldsymbol{K}$ is *saturated* if for every state $q \in \mathit{avd}(\boldsymbol{K}{\uparrow}w)$, that is, for every state q avoidable in $\boldsymbol{K}{\uparrow}w$, there exists $w' \in \mathit{Scs}(\boldsymbol{K}, w)$ such that $\boldsymbol{K}{\downarrow}w' = \boldsymbol{K}^u_q$ with $u = \boldsymbol{K}{\uparrow}w$.

Let $\boldsymbol{K}$ be an arbitrary Kripke tree. The *completion* of $\boldsymbol{K}$ is the Kripke tree $\boldsymbol{K}^{\mathbf{c}}$ defined by

$$\boldsymbol{K}^{\mathbf{c}} = \boldsymbol{K} \cdot \{(w, \boldsymbol{K}^u_q) \mid w \in \mathit{Inn}(\boldsymbol{K}), u = \boldsymbol{K}{\uparrow}w, \text{and } q \in \mathit{avd}(\boldsymbol{K}{\uparrow}w)\} \quad , \tag{12}$$

that is, in $\boldsymbol{K}^{\mathbf{c}}$, every inner node from $\boldsymbol{K}$ is saturated.

Remark 1 *Let $\boldsymbol{K}$ be an arbitrary Kripke tree. If $\boldsymbol{K} \in \mathsf{E}L$, then $\boldsymbol{K}^{\mathbf{c}} \in \mathsf{E}L$.*

This is because every maximal path through $\boldsymbol{K}$ is also present in $\boldsymbol{K}^{\mathbf{c}}$; no successors are added to leaves.

In the next two paragraphs, I will introduce the notion of a "partial run," which combines avoidable and successful states in one definition and connects runs on Kripke trees with their completions.

Let τ be an arbitrary transition condition over Q and $X, Y \subseteq Q$. The X-Y-*reduct* of τ, denoted $\tau^{X,Y}$, is obtained from τ by replacing

- every atomic subformula of the form $\Box q$ with $q \in X$ by 0,
- every atomic subformula of the form $\Box q$ with $q \in Q \setminus X$ by 1, and
- every atomic subformula of the form $\Diamond q$ with $q \in Y$ by 1.

Let $\boldsymbol{K}$ be an arbitrary Kripke tree. A *partial run* of $\boldsymbol{A}$ on $\boldsymbol{K}$ is defined just as an ordinary accepting run with the following modification of local consistency.

For every $v \in V$ such that $w_{\boldsymbol{R}}(v) \in \mathit{Inn}(\boldsymbol{K})$, it is required that

$$\boldsymbol{K}, \boldsymbol{R}, v \models \tau_v^{X_v, Y_v} \tag{13}$$

where

$$\tau_v = \delta(q_{\boldsymbol{R}}(v), L(w_{\boldsymbol{R}}(v)) \cap P) \ , \tag{14}$$

and

$$X_v = \mathit{avd}(\boldsymbol{K}{\uparrow}w_{\boldsymbol{R}}(v)) \ , \qquad Y_v = \mathit{scf}(\boldsymbol{K}{\uparrow}w_{\boldsymbol{R}}(v)) \ . \tag{15}$$

For ease in notation, I will write *IntNds*$(\boldsymbol{R})$ for the set of all $v \in V$ with $w_{\boldsymbol{R}}(v) \in$ *Inn*$(\boldsymbol{K})$, the set of all *interior nodes* of $\boldsymbol{R}$. Accordingly, I will write *FrtNds*$(\boldsymbol{R})$ for the set of all $v \in V$ with $w_{\boldsymbol{R}}(v) \in$ *Lvs*$(\boldsymbol{K})$, the set of all *frontier nodes* of $\boldsymbol{R}$. (Observe that both definitions also refer to $\boldsymbol{K}$, which is, for ease in writing, not mentioned in the notation.)

Note that in general neither τ implies $\tau^{X,Y}$ nor $\tau^{X,Y}$ implies τ. So there is no apriori relation between the existence of runs and partial runs. But we have the following.

Lemma 1 *Let $\boldsymbol{K}$ be an arbitrary Kripke tree. Assume $\mathcal{K}(\boldsymbol{A}) = \mathsf{E}L$ and $\boldsymbol{K} \in \mathcal{K}(\boldsymbol{A})$. Then there exists a partial run of $\boldsymbol{A}$ on $\boldsymbol{K}$.*

Proof. Since we assume $\mathcal{K}(\boldsymbol{A}) = \mathsf{E}L$ and $\boldsymbol{K} \in \mathcal{K}(\boldsymbol{A})$, we know, by Remark 1, there exists an accepting run $\boldsymbol{R}'$ of $\boldsymbol{A}$ on $\boldsymbol{K}^{\mathbf{c}}$. An appropriate "restriction" of this run is a partial run of $\boldsymbol{A}$ on $\boldsymbol{K}$.

Formally, restrictions of runs are defined as follows. Let $\boldsymbol{R}$ be a run of $\boldsymbol{A}$ on a Kripke tree $\boldsymbol{K}$ and assume W_0 is a subset of W containing the root of $\boldsymbol{K}$. The *restriction* of $\boldsymbol{R}$ to W_0 is the maximal subgraph of $\boldsymbol{R}$ which is a tree and contains the root of $\boldsymbol{R}$ and only nodes v with $w_{\boldsymbol{R}}(v) \in W_0$.

I claim that the restriction $\boldsymbol{R}$ of $\boldsymbol{R}'$ on W is a partial run of $\boldsymbol{A}$ on $\boldsymbol{K}$. Obviously, the initial condition and the acceptance condition are satisfied. We only have to check for local consistency in the sense of (13).

Let $v \in$ *IntNds*$(\boldsymbol{R})$ be arbitrary, and write q for $q_{\boldsymbol{R}}(v)$, w for $w_{\boldsymbol{R}}(v)$, and u for $\boldsymbol{K}{\uparrow}w$. We have

$$\boldsymbol{K}^{\mathbf{c}}, \boldsymbol{R}', v \models \Delta \tag{16}$$

where Δ is some disjunct of τ_v (as defined in (14)). Without loss of generality, we can assume Δ is of the form

$$\bigwedge_{q' \in Q_0} \Box q' \wedge \bigwedge_{q' \in Q_1} \Box q' \bigwedge_{q' \in Q_2} \Diamond q' \wedge \bigwedge_{q' \in Q_3} \Diamond q' \tag{17}$$

where $Q_0 \subseteq \mathit{avd}(u)$, $Q_1 \cap \mathit{avd}(u) = \emptyset$, $Q_2 \subseteq \mathit{scf}(u)$, and $Q_3 \cap \mathit{scf}(u) = \emptyset$. By construction, w is saturated, which means Q_0 must be the empty set. (Recall that for every $q' \in \mathit{avd}(u)$ there is no accepting run of $\boldsymbol{A}$ on $\boldsymbol{K}_u^{q'}$, which is a subtree of $\boldsymbol{K}^{\mathbf{c}}$ rooted at some successor of w.) Therefore, Δ' defined by

$$\Delta' = \bigwedge_{q' \in Q_1} \Box q' \wedge \bigwedge_{q' \in Q_3} \Diamond q' \tag{18}$$

is (equivalent to) a disjunct of $\tau_v^{X_v, Y_v}$ (with X_v and Y_v as defined in (15)). I claim

$$\boldsymbol{K}, \boldsymbol{R}, v \models \Delta' \quad , \tag{19}$$

which is obviously enough. As $\boldsymbol{R}$ results from $\boldsymbol{R}'$ by a restriction to the worlds of $\boldsymbol{K}$, (16) implies

$$\boldsymbol{K}, \boldsymbol{R}, v \models \bigwedge_{q' \in Q_1} \Box q' \quad , \tag{20}$$

which means the first big conjunct of Δ' is satisfied.

To see that the other big conjunct is satisfied, assume $q' \in Q_3$. Then there exists $v' \in \mathit{Scs}(\boldsymbol{R}', v)$ such that $q_{\boldsymbol{R}'}(v') = q'$ because of the local consistency of $\boldsymbol{R}'$. We only need to show that $w_{\boldsymbol{R}'}(v') \in W$, because if this is true, then v' belongs to $\boldsymbol{R}$ and is a successor of v in $\boldsymbol{R}$. For contradiction, assume $w_{\boldsymbol{R}'}(v')$ does not belong to W. Then $\boldsymbol{K}^{\mathbf{c}}{\downarrow}w_{\boldsymbol{R}'}(v') = \boldsymbol{K}_u^{q''}$ for some state $q'' \in \mathit{avd}(u)$. But this would mean $q' \in \mathit{scf}(u)$—a contradiction. $\Box$

Let m be a natural number. A Kripke tree $\boldsymbol{K}$ is *m-branching* if for every world $w \in W$ the following is true. For every successor w_0 of w there exist at least $m-1$ other successors $w_1, \ldots, w_{m-1}$ of w such that all subtrees $\boldsymbol{K}{\downarrow}w_0, \ldots, \boldsymbol{K}{\downarrow}w_{m-1}$ are isomorphic.

Let $\boldsymbol{R}$ be a partial run of $\boldsymbol{A}$ on a Kripke tree $\boldsymbol{K}$. The run $\boldsymbol{R}$ is *distributed* if for every $w \in W$ there exists at most one $v \in V$ with $w_{\boldsymbol{R}}(v) = w$. The set of *frontier pairs* of $\boldsymbol{R}$, denoted $\mathit{FrtPrs}(\boldsymbol{R})$, is defined by $\mathit{FrtPrs}(\boldsymbol{R}) = \{\lambda(v) \mid v \in \mathit{FrtNds}(\boldsymbol{R})\}$. Similarly, the set of *frontier states* of $\boldsymbol{R}$, denoted $\mathit{FrtSts}(\boldsymbol{R})$, is defined by $\mathit{FrtSts}(\boldsymbol{R}) = \{q_{\boldsymbol{R}}(v) \mid v \in \mathit{FrtNds}(\boldsymbol{R})\}$. (Observe that this definition refers to $\boldsymbol{K}$, but, for ease in writing, it is not mentioned in the notation.)

Lemma 2 *Let $\boldsymbol{K}$ be a $|Q|$-branching Kripke tree and $\boldsymbol{R}$ a partial run of $\boldsymbol{A}$ on $\boldsymbol{K}$. Then there exists a distributed partial run $\boldsymbol{R'}$ of $\boldsymbol{A}$ on $\boldsymbol{K}$ such that FrtSts$(\boldsymbol{R}') \subseteq$ FrtSts$(\boldsymbol{R})$.*

Proof. This is a straightforward inductive argument. Remember that in a partial run no constraint of the form $\Box q$ is important. $\Box$

The crucial lemma connecting saturated trees and partial runs is as follows.

Lemma 3 *Let $\boldsymbol{K}$ be a Kripke tree and $\boldsymbol{R}$ a distributed partial run of $\boldsymbol{A}$ on $\boldsymbol{K}$. Assume that for every $q \in$ FrtSts$(\boldsymbol{R})$, there exists a Kripke tree $\boldsymbol{K}_q \in \mathcal{K}_q$ such that the tree $\boldsymbol{K}^*$ defined by*

$$\boldsymbol{K}^* = \boldsymbol{K} \odot \{(w, \boldsymbol{K}_q) \mid (q, w) \in \mathit{FrtPrs}(\boldsymbol{R})\} \tag{21}$$

does not belong to $\mathsf{E}L$.

Then there exists an accepting run of $\boldsymbol{A}$ on the Kripke tree $\boldsymbol{K}^{**}$ defined by

$$\boldsymbol{K}^{**} = \boldsymbol{K}^c \odot \{(w, \boldsymbol{K}_q) \mid (q, w) \in \mathit{FrtPrs}(\boldsymbol{R})\} \ , \tag{22}$$

which does not belong to $\mathsf{E}L$.

Note that because $\boldsymbol{R}$ is supposed to be distributed, the trees $\boldsymbol{K}^*$ and $\boldsymbol{K}^{**}$ are obtained from $\boldsymbol{K}$ and $\boldsymbol{K}^{\mathbf{c}}$, respectively, by adding to each leaf at most one of the trees $\boldsymbol{K}_q$.

Proof. I first construct a run $\boldsymbol{R}^{**}$ of $\boldsymbol{A}$ on $\boldsymbol{K}^{**}$ and then argue it is in fact accepting.

By assumption, $\boldsymbol{K}_q \in \mathcal{K}_q$ for every $q \in \mathit{FrtSts}(\boldsymbol{R})$. So we can pick an accepting run $\boldsymbol{R}_q$ of $\boldsymbol{A}_q$ on $\boldsymbol{K}_q$ for every $q \in \mathit{FrtSts}(\boldsymbol{R})$.

Remember that for every $q \in \mathit{scf}(u)$, the state q' defined by $q' = q^u$ is a state in $\mathit{avd}(u)$ such that there exists an accepting run of $\boldsymbol{A}_q$ on $\boldsymbol{K}^u_{q'}$. So for every $v \in V$ and $q \in \mathit{scf}(\boldsymbol{K}{\uparrow}w_{\boldsymbol{R}}(v))$, we can pick an accepting run $\bar{\boldsymbol{R}}^v_q$ of $\boldsymbol{A}_q$ on $\boldsymbol{K}^u_{q'}$ where $u = \boldsymbol{K}{\uparrow}w_{\boldsymbol{K}}(v)$ and $q' = q^u$.

Also, remember that for every $q \in Q \setminus \mathit{avd}(u)$ and every Kripke tree $\boldsymbol{K'}$ such that $u \cdot \boldsymbol{K'} \notin \mathsf{E}L$, there exists an accepting run of $\boldsymbol{A}_q$ on $\boldsymbol{K'}$. So for every $v \in V$, $q \notin \mathit{avd}(\boldsymbol{K}{\uparrow}w_{\boldsymbol{R}}(v))$, and $w \in \mathit{Scs}(\boldsymbol{K}^{**}, w_{\boldsymbol{R}}(v))$, we can pick an accepting run $\hat{\boldsymbol{R}}^w_q$ of $\boldsymbol{A}_q$ on $\boldsymbol{K}^{**}{\downarrow}w$.

Consider the tree $\boldsymbol{R}^{**}$ defined by

$$\boldsymbol{R}^{**} = \boldsymbol{R} \cdot \{(v, \bar{\boldsymbol{R}}^v_q) \mid v \in \mathit{IntNds}(\boldsymbol{R}) \text{ and } q \in \mathit{scf}(\boldsymbol{R}{\uparrow}v)\} \tag{23}$$

$$\cdot \{(v, \hat{\boldsymbol{R}}^w_v) \mid v \in \mathit{IntNds}(\boldsymbol{R}), q \in Q \setminus \mathit{avd}(\boldsymbol{K}{\uparrow}v), \tag{24}$$

$$\text{and } w \in \mathit{Scs}(\boldsymbol{K}^{**}, w_{\boldsymbol{R}}(v))\} \tag{25}$$

$$\odot \{(v, \boldsymbol{R}_q) \mid v \in \mathit{FrtNds}(\boldsymbol{R}) \text{ and } q = q_{\boldsymbol{R}}(v)\} \ . \tag{26}$$

I claim that $\boldsymbol{R}^{**}$ is an accepting run of $\boldsymbol{A}$ on $\boldsymbol{K}^{**}$. First of all, the initial condition is satisfied. Clearly, the acceptance condition is satisfied as well. And as all runs $\boldsymbol{R}^w_q$, $\bar{\boldsymbol{R}}^v_q$, and $\hat{\boldsymbol{R}}^w_v$ are accepting, we only need to show

$$\boldsymbol{K}^{**}, \boldsymbol{R}^{**}, v \models \tau_v \tag{27}$$

for every $v \in V$.

So let $v \in V$ be arbitrary, and write w for $w_{\boldsymbol{R}}(v)$, q for $q_{\boldsymbol{R}}(q)$, and u for $\boldsymbol{K}{\uparrow}w$. Note that $w_{\boldsymbol{R}}(v) = w_{\boldsymbol{R}^{**}}(v)$ and $q_{\boldsymbol{R}}(q) = q_{\boldsymbol{R}^{**}}(v)$. We distinguish two cases. In the first case, when $v \in \mathit{FrtNds}(\boldsymbol{R})$, there is nothing to show because of the trees we added to $\boldsymbol{R}$ in (26). (Remember that $\boldsymbol{R}$ is assumed to be distributed.) The other case, where $v \in \mathit{IntNds}(\boldsymbol{R})$, is more complicated.

We know

$$\boldsymbol{K}, \boldsymbol{R}, v \models \Delta' \ , \tag{28}$$

where Δ' is some disjunct of $\tau_v^{X_v, Y_v}$, say

$$\Delta' = \bigwedge_{q' \in Q_0} \Diamond q' \tag{29}$$

with $Q_0 \cap \mathit{scf}(u) = \emptyset$. Let Δ be the corresponding disjunct of τ_v, say

$$\Delta = \bigwedge_{q' \in Q_0} \Diamond q' \wedge \bigwedge_{q' \in Q_1} \Diamond q' \wedge \bigwedge_{q' \in Q_2} \Box q' \ , \tag{30}$$

with $Q_1 \subseteq \mathit{scf}(u)$ and $Q_2 \cap \mathit{avd}(u) = \emptyset$. I will argue that

$$\boldsymbol{K}^{**}, \boldsymbol{R}^{**}, v \models \Delta \ , \tag{31}$$

which obviously is enough. To this end, we consider the three big conjuncts of Δ, one by one.

Because $\boldsymbol{K}^{**}$ results from $\boldsymbol{K}$ by only adding worlds and because of (28), we have

$$\boldsymbol{K}^{**}, \boldsymbol{R}^{**}, v \models \bigwedge_{q \in Q_2} \Diamond q \ . \tag{32}$$

This shows the first big conjunct of Δ holds.

From (23) we can conclude that for every $q' \in Q_1$ there exists $v' \in \mathit{Scs}(\boldsymbol{R}^{**}, v)$ such that $\boldsymbol{R}^{**}{\downarrow}v'$ is isomorphic to $\bar{\boldsymbol{R}}^v_{q'}$, in particular, $w_{\boldsymbol{R}^{**}}(v') = q'$. This shows the second big conjunct of Δ holds.

From (24) we can conclude that for every $q' \in Q_2$ and every $w' \in \mathit{Scs}(\boldsymbol{K}^{**}, w)$ there exists $v' \in \mathit{Scs}(\boldsymbol{R}^{**}, v)$ such that $\boldsymbol{R}^{**}{\downarrow}v'$ is isomorphic to $\hat{\boldsymbol{R}}^{w'}_{q'}$, in particular, $\lambda^{**}(v') = (q', w')$. This shows the third big conjunct of Δ holds. □

We can now prove Theorem 4.

Proof of Theorem 4. Let $\{(u_i, u'_i)\}_{i<m}$ be a discriminating family for L of size $\iota(L)$ and $\boldsymbol{A}$ an ATA with $\mathcal{K}(\boldsymbol{A}) = \mathsf{E}L$. I claim that for every $i < m$, there exists a state q such that $u'_i \in \mathcal{K}_q$, but $u'_j \notin \mathcal{K}_q$ for $j < m$ and $j \neq i$. This clearly implies the claim.

By way of contradiction, assume this is not the case. Then there exists $i < m$ such that for every $q \in Q$ with $u'_i \in \mathcal{K}_q$ there exists $j \neq i$ such that $u'_j \in \mathcal{K}_q$. For every such q let j_q be an appropriate index j.

Let $\boldsymbol{K}$ be a $|Q|$-branching Kripke tree such that every maximal path starting with the root is labeled $u_i a_i$ where a_i is the first letter of u'_i. Consider the Kripke tree $\boldsymbol{K'}$ defined by $\boldsymbol{K'} = \boldsymbol{K} \odot \{(w, u'_i) \mid w \in \mathit{Lvs}(\boldsymbol{K})\}$. Clearly, $\boldsymbol{K'} \in \mathsf{E}L$ (because every maximal path through $\boldsymbol{K'}$ is labeled $u_i u'_i$). Thus, by Lemma 1, there exists a partial run of $\boldsymbol{A}$ on $\boldsymbol{K'}$. By restricting this run to the worlds in $\boldsymbol{K}$, just as in the proof of Lemma 1, we obtain a partial run of $\boldsymbol{A}$ on $\boldsymbol{K}$. This run has the obvious property that for every $q \in \mathit{FrtSts}(\boldsymbol{R})$ there exists an accepting run of $\boldsymbol{A}_q$ on u'_{j_q}. So, by Lemma 2, there also exists a distributed partial run with this property. This run, in turn, together with the u'_{j_q}'s replacing the $\boldsymbol{K}_q$ satisfies the assumptions of Lemma 3. Therefore, the Kripke tree $\boldsymbol{K}^{**}$ as defined in Lemma 3, which does not belong to $\mathsf{E}L$, is accepted by $\boldsymbol{A}$—a contradiction. □

7 Further Consequences

One can show that Theorem 2 also holds for the modal μ-calculus (see, for instance, [2]). As the proof of the two previous sections does not depend on the acceptance conditions used, we also obtain: every modal μ-calculus formula equivalent to the CTL^+ formula

$$\mathsf{E}(\mathsf{F}p_0 \wedge \cdots \wedge \mathsf{F}p_{n-1}) \tag{33}$$

has length at least $\binom{n}{\lceil n/2 \rceil}$. This is interesting because of the following.

As the modal μ-calculus is closed under syntactic negation, the above also says that every modal μ-calculus formula equivalent to the CTL^+ formula

$$\mathsf{A}(\mathsf{G}\neg p_0 \vee \cdots \vee \mathsf{G}\neg p_{n-1}) \tag{34}$$

has length at least $\binom{n}{\lceil n/2 \rceil}$. And, clearly, this property can easily be expressed by an alternation-free μ-calculus (AFMC) formula (according to the definition of alternation-freeness as introduced by Emerson and Lei in [7]), because it can be expressed in CTL. On the other hand, the set of all ω-words over 2^{P_n} satisfying the linear-time temporal property

$$\mathsf{G}\neg p_0 \vee \cdots \vee \mathsf{G}\neg p_{n-1} \tag{35}$$

is recognized by a nondeterministic Büchi word automaton (NBW) with $n + 1$ states. We therefore have:

Corollary 1 *There is an exponential lower bound for the translation NBW $\mapsto$ AFMC in the sense of Kupferman and Vardi, [10].*

This answers a question left open by Kupferman and Vardi in [10].

Further, the translation from CTL formulas to alternating tree automata does not make use of the fact that we are starting from a *formula;* it would also work if we were given a "CTL circuit," that is, a DAG whose nodes are labeled with propositional variables and their negations, $\vee$ and $\wedge$, and the operators EX, AX, . . . , provided with a straightforward semantics. So we can also state:

Corollary 2 *Every CTL circuit equivalent to the CTL^+ formula*

$$\mathsf{E}(\mathsf{F}p_1 \wedge \cdots \wedge \mathsf{F}p_n)\ . \tag{36}$$

has at least $\binom{n}{\lceil n/2 \rceil}$ gates.

References

[1] Orna Bernholtz [Kupferman] and Orna Grumberg. Branching temporal logic and amorphous tree automata. In Eike Best, editor, *CONCUR'93, 4th International Conference on Concurrency Theory*, volume 715 of *Lect. Notes in Comput. Sci.*, pages 262–277, Hildesheim, Germany, 1993.

[2] Orna Bernholtz [Kupferman], Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. In David L. Dill, editor, *Computer Aided Verification, 6th International Conference, CAV '94*, volume 818 of *Lect. Notes in Comput. Sci.*, pages 142–155, Stanford, California, 1994.

[3] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications: a practical approach. In ACM, editor, *Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages*, pages 117–126, Austin, Texas, 1983.

[4] E. Allen Emerson and Joseph Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. In ACM, editor, *Proc. of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 169–181, San Francisco, California, 1982.

[5] E. Allen Emerson and Joseph Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. System Sci.*, 30(1):1–24, 1985.

[6] E. Allen Emerson, C. S. Jutla, and A. Prasad Sistla. On model-checking for fragments of μ-calculus. In Costas Courcoubetis, editor, *Computer Aided Verification: 5th International Conference, CAV '93*, volume 697 of *Lect. Notes in Comput. Sci.*, pages 385–396, Elounda, Greece, 1993.

[7] E. Allen Emerson and Chin-Laung Lei. Efficient model checking in fragments of the propositional mu-calculus (extended abstract). In *Proceedings, Symposium on Logic in Computer Science*, pages 267–278, Cambridge, Massachusetts, 1986.

[8] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. In *Proceedings 12th Annual IEEE Symposium on Logic in Computer Science*, pages 228–235, Warsaw, Poland, 1997.

[9] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, Calif., 1968.

[10] Orna Kupferman and Moshe Y. Vardi. Freedom, weakness, and determinism: From linear-time to branching-time. In *13th Annual IEEE Symposium on Logic in Computer Science*, pages 81–92, Indianaplis, Indiana, 1998.

[11] Larry Joseph Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, Dept. of Electrical Engineering, MIT, Boston, Mass., 1974.

[12] Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1–2):135–183, 1998.