The Complexity of CTL* + Linear Past

Laura Bozzelli

[1] Università di Napoli Federico II, Via Cintia, 80126 - Napoli, Italy

Abstract. We investigate the complexity of satisfiability and finite-state model-checking problems for the branching-time logic CTL^*_{lp}, an extension of CTL* with past-time operators, where past is linear, finite, and cumulative. It is well-known that CTL^*_{lp} has the same expressiveness as standard CTL*, but the translation of CTL^*_{lp} into CTL* is of non-elementary complexity, and no elementary upper bounds are known for its satisfiability and finite-state model checking problems. In this paper, we provide an elegant and uniform framework to solve these problems, which non-trivially extends the standard automata-theoretic approach to CTL* model-checking. In particular, we show that the satisfiability problem for CTL^*_{lp} is 2EXPTIME-complete, which is the same complexity as that of CTL*, but for the existential fragment of CTL^*_{lp}, the problem is EXPSPACE-complete, hence exponentially harder than that of the existential fragment of CTL*. For the model-checking, the problem is already EXPSPACE-complete for the existential and universal fragments of CTL^*_{lp}. For full CTL^*_{lp}, the proposed algorithm runs in time polynomial in the size of the Kripke structure and doubly exponential in the size of the formula. Thus, the exact complexity of model-checking full CTL^*_{lp} remains open: it lies somewhere between EXPSPACE and 2EXPTIME.

1 Introduction

Temporal logics provide a fundamental framework for the description of dynamic behavior of reactive systems [Pnu77]. Usually, in standard temporal logics such as CTL* [EH86], CTL [CE81] and LTL [Pnu77], the modalities only refer to the future of the current time. On the other hand, it is well-known that temporal logics combining past and future modalities make some specifications easier to write and more natural, and for standard linear-time temporal logics, these extensions do not increase the complexity of basic decision problems [Var88]. For the branching-time setting, there are essentially two possible views regarding the nature of the past. In the first view, past is branching and each moment in time may have several possible futures and several possible pasts. In the second view, *past is linear* and each moment in time may have several possible futures and a unique past. Usually, the past is assumed to be finite (since program computations have a definite starting time) and cumulative (i.e., the history of the current situation increases with time and is never forgotten). However, the linear past (rather that branching-past) approach is more suited to the specification of dynamic behavior because it considers states in a computation tree, while

R. Amadio (Ed.): FOSSACS 2008, LNCS 4962, pp. 186–200, 2008.

the branching-past approach consider machine states (where past is not very meaningful to specify behavioral constraints) [LS95].

For the future (regular) branching-time temporal logic CTL*, the most simple linear-past extension is the logic PCTL* [HT87], obtained by adding the past counterparts of the standard linear-time modalities 'next' and 'until'. However, since the semantics of the path quantifiers in PCTL* is the same as for CTL* (i.e., path quantification ranges over paths that starts in the current node of the computation tree), the usage of past-time modalities is very limited. In other terms, past cannot go beyond the present. It is not surprising then, that PCTL* has the same expressivity and complexity as CTL*. A more interesting and meaningful linear past extension of CTL* is the logic CTL^*_{lp} [KP95]. CTL^*_{lp} has the same syntax as PCTL*. However, path quantification is 'memoryful', i.e., it ranges over paths that start at the root and visits the current node. CTL^*_{lp} is as expressive as CTL*, but the translation of CTL^*_{lp} into CTL* is of non-elementary complexity, and no elementary upper bounds are known for its satisfiability and finite-state model checking problems [KP95, LS95, LS00, KV06]. More recently, Kupferman and Vardi [KV06] introduce a memoryful variant of CTL*, called mCTL*, which unifies CTL* and the Pistore-Vardi logic [PV03]. This new logic has the same syntax as CTL*. The unique difference is the adding of a special proposition `present` which is needed to emulate the ability of CTL* to talk about the 'present'. By letting path quantification to range over paths that start at the root, an mCTL* formula can refer to events that happen in the past. However, since mCTL* do not contain explicit past-time operators, the ability to refer to the past is limited. The logic mCTL* is as expressive as CTL*, but while satisfiability for mCTL* is 2Exptime-complete, not harder than that of CTL*, its model checking problem is Expspace-complete, exponentially harder than that of CTL* (and this last result holds also for the fragment mCTL^*_- obtained by disallowing the special atom `present`). Moreover, mCTL^*_- can be linearly translated into CTL^*_{lp} and mCTL* can be linearly translated into the extension of CTL^*_{lp} with the special atom `present`.

Our contribution. In this paper, we study the complexity of the satisfiability and (finite-state) model checking problems for CTL^*_{lp} and its existential and universal fragments ECTL^*_{lp} and ACTL^*_{lp}. The existential (resp., universal) fragment consists of formulas where the only allowed path quantifier is the existential (resp., universal) one, assuming that formulas are written in positive normal form. We also consider the extension $\text{CTL}^*_{lp}+$ of CTL^*_{lp} obtained by adding the special atom `present` and its existential and universal fragments $\text{ECTL}^*_{lp}+$ and $\text{ACTL}^*_{lp}+$. Our results are summarized in Figure 1 in which we also recall the well-known results about the complexity of the considered problems for CTL* and its existential and universal fragments (see, e.g., [KV00]). For the satisfiability problem, the complexity for CTL^*_{lp} and $\text{CTL}^*_{lp}+$ is the same as that of CTL*, i.e. 2Exptime-complete. However, for the universal and existential fragments of the considered logics, the situation is quite different. While the complexity of ACTL^*_{lp} is the same as that of ACTL* (i.e., Pspace-complete), for the fragments ECTL^*_{lp}, $\text{ECTL}^*_{lp}+$, and $\text{ACTL}^*_{lp}+$, the problem is

significantly harder being EXPSPACE-complete. For the model checking problem, the complexity is already EXPSPACE-complete for the existential and universal fragments of CTL^*_{lp}. For full CTL^*_{lp} and $\mathrm{CTL}^*_{lp}+$, our algorithm runs in time polynomial in the size of the Kripke structure and doubly exponential in the size of the formula. Thus, the exact complexity of model-checking full CTL^*_{lp} and $\mathrm{CTL}^*_{lp}+$ remains open: it lies somewhere between EXPSPACE and 2EXPTIME.

The upper bounds of the considered problems are established by a uniform automata-theoretic framework which non-trivially generalizes the standard one for CTL^* [KVW00], and is based on the translation of $\mathrm{CTL}^*_{lp}+$ formulas, with a single exponential blow-up, into a two-way extension of the *symmetric* version of *hesitant alternating* (*finite-state*) *tree automata* (HAA, for short) [KVW00]. Two-way symmetric alternating tree automata (two-way SAA, for short), and in particular, two-way (symmetric) HAA, operate on *arbitrary* (also infinite-branching) Σ-labelled trees for a given alphabet Σ. However, SAA cannot distinguish between the different children of a node, and send copies to the children of the current input node in either a universal or an existential manner. Moreover, *two-way* SAA can send copies to the parent (if any) of the current node.

The key aspects of our approach which enable us to solve *partially* the open problems regarding the complexity of CTL^*_{lp} are the following:

- the complementation result for tree languages accepted by alternating tree automata based on the construction of the dual automaton [MS87], holds also for pointed tree languages (i.e., languages consisting of pairs (T, x) where T is a labelled tree and x is a T-node) accepted by two-way SAA. This is a consequence of determinacy of (finitely-coloured) parity games, that holds also when a vertex in the underlying graph has *infinite* successors [Zie98].
- We show that parity two-way SAA can be linearly translated in the full modal μ-calculus (with both backward and forward modalities), which satisfies the bounded-degree tree-model property [Var98]. As a consequence nonemptiness of parity two-way SAA can be linearly reduced to nonemptiness of standard parity two-way alternating tree automata operating on complete n-ary trees [Var98], where n is the size of the given two-way SAA.
- The ability of combining both forward and backward moves in two-way HAA is restricted in such a way in every run each (infinite) path has a suffix which is fully *downward*. This allows us to solve the model checking problem for this class of automata by a direct construction. In particular, for the existential fragments of the considered logics, the model checking for the corresponding two-way HAA can be reduced to nonemptiness of 1-letter (one–way) HAA (over infinite words) [KVW00]. For full CTL^*_{lp} instead, we obtain an extended version of 1-letter HAA in which the 'universal requirement' for universal components of the automaton is relaxed. This explains our difficulty in obtaining membership in EXPSPACE for model checking of full CTL^*_{lp}. However, the extended 1-letter HAA obtained in the construction have a special structure, but actually we do not know if this is sufficient to solve nonemptiness with the same complexity as for 1-letter HAA.

The full version of this paper can be asked to the author by e-mail.

			Satisfiability	Model checking
CTL*			2EXPTIME-complete	PSPACE-complete
ACTL*	ECTL*		PSPACE-complete	PSPACE-complete
$CTL^*_{lp}+$	CTL^*_{lp}		2EXPTIME-complete	$\in$2EXPTIME
$ACTL^*_{lp}+$	$ECTL^*_{lp}+$	$ECTL^*_{lp}$	EXPSPACE-complete	EXPSPACE-complete
$ACTL^*_{lp}$			PSPACE-complete	EXPSPACE-complete

Fig. 1. Summary of known and new results

2 Linear-Past Branching-Time Temporal Logic

In this section we recall syntax and semantics of the linear-past branching-time temporal logic CTL^*_{lp} [KP95] and its extension, denoted $CTL^*_{lp}+$, obtained by adding the special atomic proposition `present` [KV06], which intuitively allows to refer to the 'present'. We also define the problems addressed in this paper.

Let $\mathbb{N}$ be the set of natural numbers. A *tree* T is a prefix closed subset of $\mathbb{N}^*$. The elements of T are called *nodes* and the empty word ε is the *root* of T. For $x \in T$, the set of *children* of x (in T) is $\mathsf{children}(x,T) = \{x \cdot i \in T \mid i \in \mathbb{N}\}$, and the *branching degree* of x is the cardinality (possibly infinite) of $\mathsf{children}(x,T)$. The tree T is *infinite* if each its node has at least a child. A *path* of T is an infinite sequence $\pi = x_0 x_1 \ldots$ of T-nodes such that $x_{i+1} \in \mathsf{children}(x_i, T)$ for each $i \geq 0$. Let $\pi(i)$ be the i^{th} node of π. For $x \in T$, an x-path is a path starting from x. For an alphabet Σ, a Σ-labelled tree is a pair $\langle T, V\rangle$ where T is a tree and $V : T \to \Sigma$. For $x \in T$, the pair $(\langle T, V\rangle, x)$ is called *pointed* Σ-labelled tree.

The logic $CTL^*_{lp}+$ combines both branching-time and linear-time operators. A path quantifier, E ("for some path") or A ("for all paths"), can be followed by an arbitrary linear-time formula over the usual future linear temporal operators X^+ ("forward next"), U^+ ("forward until"), and G^+ ("forward always"), and their past counterparts X^-, U^-, and G^-. As in standard CTL*, for a given finite set of atomic propositions AP, there are two types of formulas in $CTL^*_{lp}+$: *state formulas* φ, whose satisfaction is related to a specific node of a 2^{AP}-labelled tree, and *path formulas* ξ, whose satisfaction is related to a specific path. Their syntax (in *positive normal form*) is defined as follows:

$$\varphi := \top \mid prop \mid \neg prop \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathsf{E}\,\xi \mid \mathsf{A}\,\xi$$
$$\xi := \varphi \mid \mathtt{present} \mid \neg\mathtt{present} \mid \xi \wedge \xi \mid \xi \vee \xi \mid \mathsf{X}^{dir}\xi \mid \neg\mathsf{X}^{-}\top \mid \xi\,\mathsf{U}^{dir}\xi \mid \mathsf{G}^{dir}\xi$$

where $\top$ denotes `true`, $prop \in AP$, $\mathtt{present} \notin AP$ and $dir \in \{+,-\}$. We also use the classical shortcut $\mathsf{F}^{dir}\xi$ for $dir \in \{+,-\}$ ("backward and forward eventually") which stands for $\top\,\mathsf{U}^{dir}\xi$. The set of state formulas φ forms the language $CTL^*_{lp}+$.[1] CTL^*_{lp} is the fragment of $CTL^*_{lp}+$ obtained by disallowing the atom `present`. We also study the existential fragment $ECTL^*_{lp}$ (resp., $ECTL^*_{lp}+$) and

[1] Note that the given syntax is complete since the dual $\widetilde{\mathsf{U}}^{dir}$ of the until operator U^{dir} can be expressed as follows: $\xi_1\,\widetilde{\mathsf{U}}^{dir}\,\xi_2 \equiv \mathsf{G}^{dir}\xi_2 \vee (\xi_2\,\mathsf{U}^{dir}(\xi_1 \wedge \xi_2))$.

the universal fragment ACTL^*_{lp} (resp., $\mathrm{ACTL}^*_{lp}+$) of CTL^*_{lp} (resp., $\mathrm{CTL}^*_{lp}+$) obtained by disallowing the path quantifier A and E, respectively.

$\mathrm{CTL}^*_{lp}+$ formulas are interpreted over 2^{AP}-labelled infinite trees. Fix a 2^{AP}-labelled infinite tree $\langle T, V\rangle$ and let π be an ε-path of T, $x \in T$, and $k, k_0 \in \mathbb{N}$. For a state formula φ, we write $x \models \varphi$ to mean that φ holds at node x. Similarly, for a path formula ξ, we write $(\pi, k, k_0) \models \xi$ to indicate that ξ holds at position k along the ε-path π of $\langle T, V\rangle$, where k_0 is the reference position (intuitively, the 'present'). Formally, we have the following (we omit the rules for atoms in AP and boolean connectives, which are standard):

$x \models \mathsf{E}\xi$	`iff`	there is an ε-path $\pi = x_0 x_1 \ldots$ and $k \geq 0$ such that $x_k = x$ and $(\pi, k, k) \models \xi$
$x \models \mathsf{A}\xi$	`iff`	for each ε-path $\pi = x_0 x_1 \ldots$ such that $x_k = x$ for some $k \geq 0$, we have $(\pi, k, k) \models \xi$
$(\pi, k, k_0) \models \varphi$	`iff`	$\pi(k) \models \varphi$
$(\pi, k, k_0) \models$ `present`	`iff`	$k = k_0$
$(\pi, k, k_0) \models \mathsf{X}^+\xi$	`iff`	$(\pi, k+1, k_0) \models \xi$
$(\pi, k, k_0) \models \mathsf{X}^-\xi$	`iff`	$k > 0$ and $(\pi, k-1, k_0) \models \xi$
$(\pi, k, k_0) \models \xi_1 \,\mathsf{U}^+ \xi_2$	`iff`	$\exists\, n \geq k.\, (\pi, n, k_0) \models \xi_2$ and $\forall\, k \leq i < n.\, (\pi, i, k_0) \models \xi_1$
$(\pi, k, k_0) \models \xi_1 \,\mathsf{U}^- \xi_2$	`iff`	$\exists\, n \leq k.\, (\pi, n, k_0) \models \xi_2$ and $\forall\, n < i \leq k.\, (\pi, i, k_0) \models \xi_1$
$(\pi, k, k_0) \models \mathsf{G}^+\xi$	`iff`	$\forall\, n \geq k.\, (\pi, n, k_0) \models \xi$
$(\pi, k, k_0) \models \mathsf{G}^-\xi$	`iff`	$\forall\, n \leq k.\, (\pi, n, k_0) \models \xi$

For a $\mathrm{CTL}^*_{lp}+$ formula φ, we denote by $\mathcal{L}_p(\varphi)$ the set of pointed 2^{AP}-labelled infinite trees $(\langle T, V\rangle, x)$ such that $x \models \varphi$. Note that while in standard CTL^*, path quantification ranges over paths that start in the current node, in $\mathrm{CTL}^*_{lp}+$ path quantification ranges over paths that start at the root and visit the current node. For example, $\mathsf{AG}^+\mathsf{EF}^-(\xi \wedge \neg\mathsf{X}^-\top)$, when viewed as a formula of CTL^* extended with backward modalities, is unsatisfiable. When viewed as a $\mathrm{CTL}^*_{lp}+$ formula, it holds iff for each node x of the given tree, the partial path from the root to x can be extended to a path (initially) satisfying ξ.

In the following we also consider the linear temporal logic PLTL+ (LTL + Past + `present`) corresponding to $\mathrm{CTL}^*_{lp}+$ formulas which do not contain occurrences of A and E. PLTL+ is interpreted on *pointed* (*infinite*) *words* over 2^{AP}, i.e. pairs (w, k) such that $w \in (2^{AP})^\omega$ and $k \in \mathbb{N}$. The satisfaction relation $(w, k, k_0) \models \xi$, meaning that ξ holds at position k of w w.r.t. the reference position k_0, is defined similarly to the relation $(\pi, k, k_0) \models \xi'$ for path formulas ξ' of $\mathrm{CTL}^*_{lp}+$. Let $\mathcal{L}_p(\xi)$ be the set of pointed words (w, k) such that $(w, k, k) \models \xi$.

A *Kripke structure* over AP is a tuple $\mathcal{K} = \langle S, s_0, \Delta, L\rangle$, where S is a *finite* set of states, $s_0 \in S$ is an initial state, $\Delta \subseteq S \times S$ is a transition relation that must be total, and $L : S \to 2^{AP}$ maps each state s to the set of atomic propositions true in s. The Kripke structure $\mathcal{K}$ induces a 2^{AP}-labelled tree, denoted by $CT_{\mathcal{K}}$, which corresponds to the unwinding of $\mathcal{K}$ from s_0 (defined in the usual way).

We address the following problems for $\mathrm{CTL}^*_{lp}+$ (and its mentioned fragments):

- the *satisfiability* problem is to decide, given a $\mathrm{CTL}^*_{lp}+$ formula φ over AP, whether $(\langle T, V\rangle, \varepsilon) \in \mathcal{L}_p(\varphi)$ for some 2^{AP}-labelled infinite tree $\langle T, V\rangle$;

– the (*finite-state*) *model checking* problem is to decide, given a Kripke structure $\mathcal{K}$ and a CTL$^*_{lp}$+ formula φ over AP, whether $(CT_{\mathcal{K}}, \varepsilon) \in \mathcal{L}_p(\varphi)$.

3 Alternating Finite–State Automata for Linear Past

In order to solve satisfiability and model-checking for CTL$^*_{lp}$+ and its fragments, we propose an extension of the automata-theoretic approach to branching-time model checking [KVW00]. In particular, we consider *two-way symmetric alternating* (*finite-state*) *tree automata* (two-way SAA), and more specifically we focus on a subclass of such automata. One–way SAA were first introduced in [Wil99] and operate on *arbitrary* Σ-labeled infinite trees (whose nodes can have infinite branching degrees) for a given alphabet Σ. SAA cannot distinguish between the different children of a node, and send copies to the children of the current input node in either a universal or an existential manner. Moreover, *two-way* SAA can send copies to the parent (if any) of the current node. In order to formally define such a class of automata, we need additional notation.

For a set X, $\mathcal{B}_+(X)$ denotes the set of *positive* boolean formulas over X, built from elements in X using $\vee$ and $\wedge$ (we also allow the formulas `true` and `false`). A subset Y of X *satisfies* $\theta \in \mathcal{B}_+(X)$ iff the truth assignment that assigns `true` to the elements in Y and `false` to the elements of $X \setminus Y$ satisfies θ; Y *exactly satisfies* θ if Y satisfies θ and every proper subset of Y does not satisfy θ.

A two-way SAA is a tuple $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, Acc \rangle$, where Σ is the input alphabet, Q is a finite set of states, $q_0 \in Q$ is an initial state, $\delta : Q \times \Sigma \to \mathcal{B}_+\big((\{\Box, \Diamond\} \times Q) \cup (\{\uparrow\} \times Q \times \{\texttt{true}, \texttt{false}\})\big)$ is the transition function, and Acc is an acceptance condition. Intuitively, a target of a move of $\mathcal{A}$ is encoded by an element in $(\{\Box, \Diamond\} \times Q) \cup (\{\uparrow\} \times Q \times \{\texttt{true}, \texttt{false}\})$. An atom $(\Diamond, q)$ means that a copy of $\mathcal{A}$ in state q moves to some child of the current node, while an atom $(\Box, q)$ means that for each child x of the current node, a copy of $\mathcal{A}$ in state q is sent to node x. Finally, an atom $(\uparrow, q, b)$ can be chosen iff either the current node is not the root or $b = \texttt{true}$. In the first case, a copy of $\mathcal{A}$ in state q is sent to the parent of the current node. A *one-way* SAA is a two-way SAA whose transition function satisfies $\delta(q, a) \in \{\Box, \Diamond\} \times Q$ for each $(q, \sigma) \in Q \times \Sigma$.

For a pointed Σ-labelled infinite tree $(\langle T, V \rangle, x_0)$, a *run* of $\mathcal{A}$ over $(\langle T, V \rangle, x_0)$ is a $Q \times T$-labelled tree $r = \langle T_r, V_r \rangle$, where each node of T_r labelled by (q, x) describes a copy of $\mathcal{A}$ that is in state q and reads the node x of T. Moreover, we require that $r(\varepsilon) = (q_0, x_0)$ (initially, $\mathcal{A}$ is in state q_0 reading node x_0), and for each $y \in T_r$ with $r(y) = (q, x)$, there is a (possibly empty) set $H \subseteq (\{\Box, \Diamond\} \times Q) \cup (\{\uparrow\} \times Q \times \{\texttt{true}, \texttt{false}\})$ exactly satisfying $\delta(q, V(x))$ such that H does not contain atoms $(\uparrow, q', \texttt{false})$ if $x = \varepsilon$, and $\mathsf{children}(y, T_r)$ satisfies the following for each $at \in H$:

- if $at = (\Diamond, q')$, then $\exists x' \in \mathsf{children}(x, T)$, $\exists y' \in \mathsf{children}(y, T_r)$. $r(y') = (q', x')$;
- if $at = (\Box, q')$, then $\forall x' \in \mathsf{children}(x, T)$, $\exists y' \in \mathsf{children}(y, T_r)$. $r(y') = (q', x')$;
- if $at = (\uparrow, q', b)$ and $x = x' \cdot i$, then $\exists y' \in \mathsf{children}(y, T_r)$. $r(y') = (q', x')$.

For a path $\pi = y_0 y_1 \ldots$ of the run $r = \langle T_r, V_r \rangle$, let $inf(\pi)$ be the set of states in Q that appear in $V_r(y_0)V_r(y_1)\ldots$ infinitely often. We say that π is accepting

iff $inf(\pi)$ *satisfies* the acceptance condition Acc of $\mathcal{A}$. The run $r = \langle T_r, V_r \rangle$ is *accepting* iff each its path is accepting. Here, we consider *parity acceptance conditions*, specified by mappings[2] $\Omega : Q \to \mathbb{N}$ assigning to each state $q \in Q$ an integer (called *priority*). A path π trough a run satisfies Ω if the smallest priority of the states in $inf(\pi)$ is *even*. The *index* of a parity two-way SAA with parity acceptance condition Ω is the cardinality of the set $\{\Omega(q) \mid q \in Q\}$.

For a two-way SAA over Σ, the *pointed language* $\mathcal{L}_p(\mathcal{A})$ of $\mathcal{A}$ is the set of pointed Σ-labeled infinite trees PT such that $\mathcal{A}$ has an accepting run over PT. The *language* $\mathcal{L}(\mathcal{A})$ is the set of Σ-labelled trees $\langle T, V \rangle$ s.t. $(\langle T, V \rangle, \varepsilon) \in \mathcal{L}_p(\mathcal{A})$.

Given a parity two-way SAA $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \Omega \rangle$, the *dual automaton of* $\mathcal{A}$ is the parity two-way SAA $\widetilde{\mathcal{A}} = \langle \Sigma, Q, q_0, \widetilde{\delta}, \widetilde{\Omega} \rangle$, where for each $q \in Q$, $\widetilde{\Omega}(q) = \Omega(q) + 1$, and for each (q, σ), $\widetilde{\delta}(q, \sigma)$ is obtained from $\delta(q, \sigma)$ by switching $\Box$ and $\Diamond$, switching $\vee$ and $\wedge$, and switching `true` and `false`. If, for example, $\delta(q, \sigma) = (\Box, p) \vee (\uparrow, q, \texttt{true})$, then $\widetilde{\delta}(q, \sigma) = (\Diamond, p) \wedge (\uparrow, q, \texttt{false})$.

We can give a game-theoretic interpretation of acceptance in two-way SAA $\mathcal{A}$ by (finitely coloured) parity games. Since the determinacy result for such a class of games holds also when the number of successors of a vertex in the underlying graph is infinite [Zie98], by a readaptation of the proof given in [MS87], it follows that the *dual automaton* of $\mathcal{A}$ accepts the complement of $\mathcal{L}_p(\mathcal{A})$, i.e., the set of pointed Σ-labeled infinite trees $PT \notin \mathcal{L}_p(\mathcal{A})$.

Proposition 1. *The dual automaton of a parity two-way* SAA $\mathcal{A}$ *accepts the complement of* $\mathcal{L}_p(\mathcal{A})$.

As we will see in order to capture $\text{CTL}^*_{lp}+$ formulas, it suffices to consider a subclass of parity two-way SAA of index 3 corresponding to a two-way extension of *hesitant alternating tree automata* (HAA) introduced in [KVW00] as an optimal automata-theoretic framework for CTL*. Formally, a *two-way* HAA is a two-way SAA $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, Acc \rangle$ satisfying the following conditions. As in weak alternating automata, there is a partition of Q into disjoint sets $Q_1, \ldots, Q_m$ (called *components* of $\mathcal{A}$) and a partial order $\leq$ on these sets such that transitions from a state in Q_i lead to states in either the same Q_i or components Q_j such that $Q_j < Q_i$ (*partial order requirement*). Moreover, each component Q_i is classified either as *transient*, *existential*, or *universal*, and the following holds:

1. for each transient set Q_i and $q \in Q_i$, $\delta(q, \sigma)$ contains no states of Q_i;
2. for each existential component Q_i and $q \in Q_i$, if $\delta(q, a)$ is rewritten in disjunctive normal form, then there is at most one (forward) atom (c, q') with $q' \in Q_i$ in each disjunct. Moreover, $c = \Diamond$ (*existential requirement*);
3. for each universal component Q_i and $q \in Q_i$, if $\delta(q, a)$ is rewritten in conjunctive normal form, then there is at most one (forward) atom (c, q') with $q' \in Q_i$ in each conjunct. Moreover, $c = \Diamond$ (*universal requirement*);
4. Acc consists of a pair $\langle G, B \rangle$ of sets of states interpreted as the parity condition $\Omega_{\langle G,B \rangle}$ of index 3 assigning 0 to the states in $Q_\exists \cap G$, assigning 1 to the states in $(Q_\exists \setminus G) \cup (Q_\forall \cap B)$, and assigning 2 to the remaining states, where $Q_\exists$ (resp., $Q_\forall$) is the set of existential (resp., universal) states.

[2] We use the symbol Ω instead of Acc to specify such acceptance conditions.

5. in addition for a *two-way* HAA $\mathcal{A}$, we require that Q is also partitioned into a set Q_+ of *positive* states and a set Q_- of *negative* states such that: (i) $q_0 \in Q_+$, (ii) for each atom $(\uparrow, q, b)$ (backward choice) occurring in δ, $q \in Q_-$, and for each atom (c, q) (forward choice) occurring in δ, $q \in Q_+$, and (iii) for each component Q_i and negative state $q_- \in Q_i$, $\delta(q_-, \sigma)$ does not contain atoms of the form $(\Diamond, q)$ or $(\Box, q)$ with $q \in Q_i$.

The partial order requirement and Condition 1 ensure that every path π of a run of $\mathcal{A}$ gets trapped within some existential or universal component Q_i. Then, by Condition 4, the path satisfies the acceptance condition $Acc = \langle G, B\rangle$ iff either Q_i is an existential set and $inf(\pi) \cap G \neq \emptyset$ (Büchi condition), or Q_i is a universal set and $inf(\pi) \cap B = \emptyset$ (co-Büchi condition). Condition 5 ensures that every path π get trapped in Q_+, and in particular from a certain point on, π becomes fully *downward*, i.e. there is a suffix of π such that each node along this suffix is obtained from the previous by applying a forward choice (corresponding to an atom of the form (c, q) with $c \in \{\Box, \Diamond\}$). The *depth* of $\mathcal{A}$ is the number of components of $\mathcal{A}$. The two-way HAA $\mathcal{A}$ is *existential* if each its component is not universal, and is *strictly existential* if its transition function does not contain atoms of the form $(\Box, q)$. Note that the dual automaton $\widetilde{\mathcal{A}} = \langle \Sigma, Q, q_0, \widetilde{\delta}, \widetilde{\Omega_{\langle G,B\rangle}}\rangle$ of $\mathcal{A}$ is still a two-way HAA. Indeed, the components of $\widetilde{\mathcal{A}}$ are the same as $\mathcal{A}$ (with the same partial order) with the difference that a component that is existential in $\mathcal{A}$ is universal in $\widetilde{\mathcal{A}}$, and vice versa. Moreover, Condition 5 continue to hold (the sets of positive states and negative states of $\widetilde{\mathcal{A}}$ are the same as $\mathcal{A}$). Finally, it is easy to show that the parity condition $\widetilde{\Omega_{\langle G,B\rangle}}$, when interpreted on runs of $\widetilde{A}$, is equivalent to the parity condition $\Omega_{\langle B,G\rangle}$ associated with $\langle B, G\rangle$. Thus, by Proposition 1 we obtain the following result.

Proposition 2. *The dual automaton of a two-way* HAA $\mathcal{A}$ *is a two-way* HAA *accepting the complement of* $\mathcal{L}_p(\mathcal{A})$.

We address the following problems for the class of two-way HAA:

- the *nonemptiness* problem is to decide, for a two-way HAA, whether $\mathcal{L}(\mathcal{A}) \neq \emptyset$;
- the (*finite-state*) *model checking* problem is to decide, given a Kripke structure $\mathcal{K}$ over AP and a two-way HAA $\mathcal{A}$ over 2^{AP}, whether $CT_{\mathcal{K}} \in \mathcal{L}(\mathcal{A})$.

In the following, we also consider (one-way) HAA on *infinite words* over a 1-letter alphabet (1-letter HAA). Note for such a class of automata, choices represented by atoms $(\Diamond, q)$ and $(\Box, q)$ are equivalent, and thus the transition function can be given as a mapping $\delta : Q \to \mathcal{B}_+(Q)$, where Q is the set of states.

3.1 Decision Procedures for Two-Way HAA

Model checking. We reduce the model checking problem for two-way HAA to the nonemptiness problem of *extended* 1-letter HAA corresponding to 1-letter HAA in which the universal requirement for universal components (see Condition 3 in the definition of HAA) is relaxed.

Theorem 1. *For a Kripke structure $\mathcal{K}$ over AP of size n and a* two-way HAA *$\mathcal{A}$ over 2^{AP} with depth d and size m, one can build an extended* 1*-letter* HAA *$\mathcal{A}_{\mathcal{K}}$ with depth d and size $O(n \cdot m \cdot 2^{O(m)})$ s.t. $\mathcal{L}(\mathcal{A}_{\mathcal{K}}) \neq \emptyset$ iff $CT_{\mathcal{K}} \in \mathcal{L}(\mathcal{A})$. Moreover, if $\mathcal{A}$ is an existential two-way* HAA*, then $\mathcal{A}_{\mathcal{K}}$ is an existential* 1*-letter* HAA*.*

Proof. Let $\mathcal{K} = \langle S, s_0, \Delta, L\rangle$ and $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \langle G, B\rangle\rangle$ with $\Sigma = 2^{AP}$. Essentially, the 1-letter extended HAA $\mathcal{A}_{\mathcal{K}}$ guesses a run of $\mathcal{A}$ over $(CT_{\mathcal{K}}, \varepsilon)$ and checks that it is accepting. At a given node x of a run of $\mathcal{A}_{\mathcal{K}}$, $\mathcal{A}_{\mathcal{K}}$ keeps track by its finite control of the following information: (i) the positive state q_+ associated with the current 'positive' copy of $\mathcal{A}$ in the guessed run, (ii) the state s of $\mathcal{K}$ associated with the node x_{curr} of $CT_{\mathcal{K}}$ which is read by the current copy of $\mathcal{A}$, and (iii) the guessed set P_- of negative states of $\mathcal{A}$ associated with the copies of $\mathcal{A}$ which read x_{curr} and belong to the subrun starting from the current copy of $\mathcal{A}$. Note that since $\mathcal{A}$ is a two-way HAA, P_- cannot contain states in components Q_i of $\mathcal{A}$ that are upper in the partial order than the q_+-component. The paths of a run of $\mathcal{A}_{\mathcal{K}}$ correspond to the *downward* paths of the simulated run of $\mathcal{A}$ over $(CT_{\mathcal{K}}, \varepsilon)$. Since $\mathcal{A}$ is a two-way HAA, each infinite path of a run of $\mathcal{A}$ has a suffix which is downward. Thus, a run of $\mathcal{A}_{\mathcal{K}}$ keeps track of all meaningful information associated with the corresponding simulated run of $\mathcal{A}$ over $(CT_{\mathcal{K}}, \varepsilon)$.

The extended 1-letter HAA $\mathcal{A}_{\mathcal{K}} = \langle \{a\}, Q_{\mathcal{K}}, q^0_{\mathcal{K}}, \delta_{\mathcal{K}}, \langle G_{\mathcal{K}}, B_{\mathcal{K}}\rangle\rangle$ is formally defined as follows. Let $Q_1, \ldots, Q_d$ be a fixed total ordering of the components of $\mathcal{A}$ extending the partial order $\leq$ of $\mathcal{A}$, and let Q_+ (resp., Q_-) be the set of positive states (resp., negative states) of $\mathcal{A}$. For a state $q \in Q_i$, let $index(q) := i$, and for $q \in Q$, let $\Pi(q) = \{P_- \subseteq Q_- \mid \forall q_- \in P_-, index(q_-) \leq index(q)\}$.

A state of $\mathcal{A}_{\mathcal{K}}$ is either of the form $(q_+, s, P_{up}, root) \in Q_+ \times S \times 2^{Q_-} \times \{0,1\}$ or of the form $(q_+, s, P_{up}, P_{curr}, root) \in Q_+ \times S \times 2^{Q_-} \times 2^{Q_-} \times \{0,1\}$ such that $P_{curr} \in \Pi(q_+)$, where: (i) q_+ represents the state associated with the current 'positive' copy of $\mathcal{A}$ which reads a node x of the computation tree of $\mathcal{K}$ labelled by state s, (ii) P_{curr} represents the guessed set of negative states of $\mathcal{A}$ associated with the copies of $\mathcal{A}$ which read x and belong to the 'subrun' of $\mathcal{A}$ starting from the current copy, (iii) P_{up} represents the set of negative states of $\mathcal{A}$ associated with the copies of $\mathcal{A}$ which read the parent node y of x and belong to the 'subrun' of $\mathcal{A}$ associated with a positive copy (reading y) which has generated (in one step or many steps) the current copy, (iv) *root* is a flag which is 1 iff the current node x of $CT_{\mathcal{K}}$ is the root. The initial state is $q^0_{\mathcal{K}} = (q_0, s_0, \emptyset, 1)$. The components of $\mathcal{A}_{\mathcal{K}}$ are $Q'_1, \ldots, Q'_d$ with $Q'_i \leq Q'_j$ iff $i \leq j$, where Q'_i is the set of states $(q_+, s, P_{up}, root)$ or $(q_+, s, P_{up}, P_{curr}, root)$ such that $q_+ \in Q_i$. Moreover, Q'_i is existential if Q_i is either existential or transient, and is universal otherwise.

The transition function $\delta_{\mathcal{K}}$ is defined as follows:

1. $\delta_{\mathcal{K}}(q_+, s, P_{up}, root) = \bigvee_{P_{curr} \in \Pi(q_+)} (q_+, s, P_{up}, P_{curr}, root)$;
2. $\delta_{\mathcal{K}}(q_+, s, P_{up}, P_{curr}, root) = \theta(q_+, s, P_{up}, P_{curr}) \wedge \bigwedge_{q_- \in P_{curr}} \theta(q_-, s, P_{up}, P_{curr})$, where $\theta(q, s, P_{up}, P_{curr})$ is obtained from $\delta(q, L(s))$ as follows:
   - each atom $(\Box, p_+)$ (resp., $(\Diamond, p_+)$) occurring in $\delta(q, L(s))$ is replaced with $\bigwedge_{s' \in succ_{\mathcal{K}}(s)} (p_+, s', P_{curr}, 0)$ (resp., $\bigvee_{s' \in succ_{\mathcal{K}}(s)} (p_+, s', P_{curr}, 0)$);
   - each atom $(\uparrow, p_-, b)$ in $\delta(q, L(s))$ is replaced with `true` if either $root = 0$ and $p_- \in P_{up}$ or $root = 1$ and $b =$ `true`, and with `false` otherwise.

where $succ_{\mathcal{K}}(s)$ denotes the set of successors of s in $\mathcal{K}$.

The acceptance condition $\langle G_{\mathcal{K}}, B_{\mathcal{K}} \rangle$ is defined as follows:

- $G_{\mathcal{K}} = \{(q_+, s, P_{up}, root), (q_+, s, P_{up}, P_{curr}, root) \mid q_+ \in G\}$;
- $B_{\mathcal{K}} = \{(q_+, s, P_{up}, root), (q_+, s, P_{up}, P_{curr}, root) \mid q_+ \in B\}$.

Note that $\delta_{\mathcal{K}}$ satisfies the partial order requirement. Moreover, a path of a run of $\mathcal{A}_{\mathcal{K}}$ cannot be trapped within an existential component Q'_i corresponding to a transient component of $\mathcal{A}$. Also, Condition 5 in def. of two-way HAA ensures that an existential component of $\mathcal{A}_{\mathcal{K}}$ satisfies the existential requirement corresponding to condition 2 in def. of HAA. However, if $\mathcal{A}$ contains universal components, then the universal components of $\mathcal{A}_{\mathcal{K}}$ do not satisfy the universal requirement due to the nondeterministic choice of the set P_{curr} in the transitions from states of the form $(q_+, s, P_{up}, root)$. Thus, if $\mathcal{A}$ is existential, then $\mathcal{A}_{\mathcal{K}}$ is an existential 1-letter HAA. Otherwise, $\mathcal{A}_{\mathcal{K}}$ is an *extended* 1-letter HAA. □

In [KVW00] it is shown that nonemptiness of 1-letter HAA (hence, also 1-letter existential HAA) of depth d and size n can be solved in space $O(d \log^2 n)$. Since extended 1-letter HAA are also 1-letter parity alternating automata over infinite words of index 3, and for such class of automata, nonemptiness can be solved in cubic time [KV98], by Theorem 1, we obtain the following upper bounds for the model checking of two-way HAA and two-way existential HAA.

Theorem 2. *Given a Kripke structure $\mathcal{K}$ over AP of size n and a* two-way HAA *$\mathcal{A}$ over 2^{AP} with depth d and size m, the model checking problem for $\mathcal{K}$ and $\mathcal{A}$ can be solved in time $O(n^3 \cdot m^3 \cdot 2^{O(m)})$. Moreover, if $\mathcal{A}$ is* existential*, then the same problem can be solved in space $O(d \cdot \log^2(n \cdot m \cdot 2^{O(m)}))$.*

Nonemptiness problem. For the nonemptiness problem of two-way HAA and, more in general, parity two-way SAA, we obtain the following result.

Theorem 3. *The nonemptiness problem of parity two-way* SAA (*hence, also two-way* HAA) *is in* Exptime.

Proof. We can show that for a parity two-way SAA $\mathcal{A}$ over Σ of size n and index h, it is possible to build a formula $\varphi_{\mathcal{A}}$ of the *full modal μ-calculus* (with both forward and backward modalities) over Σ [Var98] such that $\varphi_{\mathcal{A}}$ has size bounded by $2n$ and for each Σ-labelled tree LT, $LT \in \mathcal{L}(\mathcal{A})$ iff LT is a tree-model of $\varphi_{\mathcal{A}}$. Since a formula φ of the full μ-calculus is satisfiable iff there is a tree-model of φ whose branching degrees are bounded by the size of φ [Var98], it follows that for the given parity two-way SAA $\mathcal{A}$ over Σ of size n and index h, $\mathcal{L}(\mathcal{A}) \neq \emptyset$ iff $\mathcal{L}(\mathcal{A}) \cap \Upsilon_{2n}(\Sigma) \neq \emptyset$, where $\Upsilon_{2n}(\Sigma)$ denotes the set of Σ-labelled trees whose branching degrees are bounded by $2n$. Now, let $\perp$ be a symbol non in Σ. We can encode a labelled tree LT in $\Upsilon_{2n}(\Sigma)$ as a $\Sigma \cup \{\perp\}$-labelled complete $2n$-ary tree $\langle \{1, \ldots, 2n\}^*, V \rangle$ as follows: first, for each node x of LT with i children (note that $i \leq 2n$), we add $2n - i$ new children and label these new nodes with $\perp$; finally, for each node x labelled by $\perp$, we add recursively $2n$ children labelled by $\perp$. Then, starting from $\mathcal{A}$, it is easy to construct a standard parity two-way alternating

tree automaton [Var98] (parity two-way ATA) $\mathcal{A}'$ operating on $\Sigma \cup \{\bot\}$-labelled complete $2n$-ary trees having the same index as $\mathcal{A}$ and size linear in the size of $\mathcal{A}$, and such that $\mathcal{A}'$ accepts all and only the trees encoding Σ-labelled trees in $\mathcal{L}(\mathcal{A}) \cap \Upsilon_{2n}(\Sigma)$. Hence, $\mathcal{L}(\mathcal{A}') \neq \emptyset$ iff $\mathcal{L}(\mathcal{A}) \neq \emptyset$. Since nonemptiness of parity two-way ATA of size n, index h over k-ary trees can be solved in time $2^{O(n^2 \cdot k \cdot h)}$ [Var98], Theorem 3 follows. □

However, for nonemptiness of strictly existential two-way HAA $\mathcal{A}$, we can do better. Indeed, the transition function of $\mathcal{A}$ does not contain atoms of the form $(\Box, q)$ corresponding to universal choices. Thus, the only forward choices are existential (nondeterminism). Moreover, the automaton cannot distinguish between the different children of the current input node. Thus, if $\mathcal{A}$ is one-way, then it actually corresponds to a nondeterministic tree automaton, hence nonemptiness can be trivially reduced to nonemptiness of 1-letter HAA. If instead $\mathcal{A}$ is two-way, then we need to keep track only of the downward paths of a run of $\mathcal{A}$. These observations enable us to solve nonemptiness for $\mathcal{A}$ by a direct reduction to nonemptiness of an (existential) 1-letter HAA $\mathcal{A}_W$ having the same depth as $\mathcal{A}$ and exponential size. Thus, we obtain the following result.

Theorem 4. *Nonemptiness of* strictly existential *two-way* HAA *is in* PSPACE.

4 Decision Procedures for $\mathrm{CTL}^*_{lp}+$ and Its Fragments

In this section we describe an automata-theoretic approach to solve satisfiability and model-checking for $\mathrm{CTL}^*_{lp}+$ based on the translation (with a single exponential blow-up) of $\mathrm{CTL}^*_{lp}+$ formulas φ into equivalent two-way HAA $\mathcal{A}_\varphi$ accepting the set of pointed labelled trees satisfying φ. Before illustrating this, we need a preliminary result concerning the translation of the linear temporal logic PLTL+ into a simple variant of two-way Büchi word automata [Var88].

A *simple two-way Büchi* (*nondeterministic*) *word automaton* (Büchi SNWA, for short) is a tuple $\mathcal{A} = \langle \Sigma, Q, Q_0, \rho, F_-, F_+ \rangle$, where Σ is the input alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, $\rho : Q \times \Sigma \times \{+, -\} \to 2^Q$ is a transition function, and F_- and F_+ are sets of accepting states. A run of $\mathcal{A}$ over a pointed word (w, i), where $w = w(0)w(1)\ldots$, is a pair $r = (r_-, r_+)$ such that $r_+ = q_i^+, q_{i+1}^+ \ldots$ is an infinite sequence of states, $r_- = q_i^-, q_{i-1}^- \ldots q_0^- q_{-1}^-$ is a finite sequence of states, and: (i) $q_i^+ = q_i^- \in Q_0$; (ii) for each $h \geq i$, $q_{h+1}^+ \in \rho(q_h^+, w(h), +)$; and (iii) for each $0 \leq h \leq i$, $q_{h-1}^- \in \rho(q_h^-, w(h), -)$.

Thus, starting from the initial position i in the input pointed word (w, i), the automaton splits in two copies: the first one moves forwardly along the suffix of w starting from position i and the second one moves backwardly along the prefix $w(0) \ldots w(i)$. The run $r = (r_-, r_+)$ is *accepting* if $q_{-1}^- \in F_-$ and r_+ visits infinitely often some state in F_+. A pointed word (w, i) is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ over (w, i). By a readaptation of the standard translation of LTL into Büchi word automata [VW94], we obtain the following.

Proposition 3. *Given a* PLTL+ *formula* ξ *over* AP*, one can construct a Büchi* SNWA *over* 2^{AP} *of size* $2^{O(|\xi|)}$ *accepting the set of pointed words satisfying* ξ.

Theorem 5. *For a* $\mathrm{CTL}^*_{lp}+$ *formula* ψ *over* AP*, one can build a two-way* HAA $\mathcal{A}_\psi$ *over* 2^{AP} *of size* $2^{O(|\psi|)}$ *and depth* $O(|\psi|)$ *such that* $\mathcal{L}_p(\mathcal{A}_\psi) = \mathcal{L}_p(\psi)$*. Also, if* ψ *is a* $\mathrm{ECTL}^*_{lp}+$ *formula, then* $\mathcal{A}_\psi$ *is a* strictly existential *two-way* HAA.

Proof. We need some definitions. A $\mathrm{CTL}^*_{lp}+$ formula φ is *trivial* if either $\varphi = p$ or $\varphi = \neg p$, where $p \in AP$. For two $\mathrm{CTL}^*_{lp}+$ formulas φ_{sub} and φ, φ_{sub} is *maximal* in φ if φ_{sub} is a strict state subformula of φ and there is an occurrence of φ_{sub} in φ s.t. there is no occurrence of a strict state subformula of φ which strictly contains the considered occurrence of φ_{sub}. We denote by $max(\varphi)$ the set of all formulas maximal in φ. For example, $max(\mathsf{A}\big((\mathsf{X}^+\neg p)\,\mathsf{U}^+(\mathsf{EX}^-\neg p)\big)) = \{\neg p, \mathsf{EX}^-\neg p\}$.

As in the case of the one-way HAA for CTL* [KVW00], we construct the two-way HAA $\mathcal{A}_\psi$ by induction on the structure of ψ. With each state subformula φ of ψ, we associate a two-way HAA $\mathcal{A}_\varphi$ over $\Sigma = 2^{AP}$ of size $2^{O(|\varphi|)}$ and depth $O(|\varphi|)$ such that $\mathcal{L}_p(\mathcal{A}_\varphi) = \mathcal{L}_p(\varphi)$. For the base of the induction, either $\varphi = p$ or $\varphi = \neg p$, where $p \in AP$. In both cases, $\mathcal{A}_\varphi$ has one state q_0, which is positive and transient, acceptance condition $\langle \emptyset, \emptyset \rangle$, and transition function δ defined as follows. In the first case ($\varphi = p$), $\delta(q_0, \sigma) = \mathtt{true}$ if $p \in \sigma$, and $\delta(q_0, \sigma) = \mathtt{false}$ if $p \notin \sigma$. In the second case ($\varphi = \neg p$), $\delta(q_0, \sigma) = \mathtt{true}$ if $p \notin \sigma$, and $\delta(q_0, \sigma) = \mathtt{false}$ if $p \in \sigma$. Now, assume that φ is a non-trivial state subformula of ψ (induction step). Let $max(\varphi) = \{\varphi_1, \ldots, \varphi_n\}$. By the induction hypothesis for each $1 \leq i \leq n$, we can construct a two-way HAA $\mathcal{A}_{\varphi_i} = \langle \Sigma, Q^i, q_0^i, \delta^i, \langle G^i, B^i \rangle \rangle$ of size $2^{O(|\varphi_i|)}$ and depth $O(|\varphi_i|)$ such that $\mathcal{L}_p(\mathcal{A}_{\varphi_i}) = \mathcal{L}_p(\varphi_i)$. We assume that the state sets of the two-way HAA $\mathcal{A}_{\varphi_1}, \ldots, \mathcal{A}_{\varphi_n}$ are disjoint (otherwise, we rename the states). We construct a two-way HAA $\mathcal{A}_\varphi$ composed from $\mathcal{A}_{\varphi_1}, \ldots, \mathcal{A}_{\varphi_n}$ as follows. Since φ is in positive normal form, there are only the following cases:

- $\varphi = \varphi_1 \wedge \varphi_2$ ($n = 2$). We define $\mathcal{A}_\varphi = \langle \Sigma, Q^1 \cup Q^2 \cup \{q_0\}, q_0, \delta, \langle G^1 \cup G^2, B^1 \cup B^2 \rangle \rangle$, where q_0 is a new (positive) state and δ is defined as follows. For states in Q^1 and Q^2, δ agrees with δ^1 and δ^2, respectively (recall that Q^1 and Q^2 are disjoint). For the state q_0 and for each $\sigma \in \Sigma = 2^{AP}$, $\delta(q_0, \sigma) = \delta^1(q_0^1, \sigma) \wedge \delta^2(q_0^2, \sigma)$. Thus, from the initial node of the pointed input tree and in the initial state q_0, $\mathcal{A}_\varphi$ sends all the copies sent (initially) by both $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_{\varphi_2}$. The singleton $\{q_0\}$ constitutes a transient component, with the ordering $\{q_0\} > Q'$ for all components Q' of $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_{\varphi_2}$. Evidently, $\mathcal{L}_p(\mathcal{A}_\varphi) = \mathcal{L}_p(\mathcal{A}_{\varphi_1}) \cap \mathcal{L}_p(\mathcal{A}_{\varphi_2})$. Moreover, by the induction hypothesis, we have that $\mathcal{L}_p(\mathcal{A}_\varphi) = \mathcal{L}_p(\varphi)$, and $\mathcal{A}_\varphi$ has size $2^{O(|\varphi|)}$ and depth $O(|\varphi|)$.
- $\varphi = \varphi_1 \vee \varphi_2$. The construction of $\mathcal{A}_\varphi$ is similar to the previous case with the difference that now $\delta(q_0, \sigma) = \delta^1(q_0^1, \sigma) \vee \delta^2(q_0^2, \sigma)$.
- $\varphi = \mathsf{E}\xi$. Let $max(\varphi) = \{\varphi_1, \ldots, \varphi_n\}$ and let $\widehat{AP} = \{p_1, \ldots, p_n\}$ be a set of fresh propositions. Moreover,let $\widehat{\xi}$ be the PLTL+ formula over $\widehat{AP}$ obtained from the path formula ξ by replacing each occurrence of φ_i in φ which is maximal in φ with proposition p_i. Since $\widehat{\xi}$ and ξ are in positive normal form and $\widehat{\xi}$ does not contain subformulas of the form $\neg p_i$ for each $p_i \in \widehat{AP}$, it easily follows that for each pointed 2^{AP}-labelled tree $(\langle T, V \rangle, x)$,

Claim 1: $(\langle T, V\rangle, x) \models \mathsf{E}\xi$ *if and only if* there is a ε-path of T $\pi = x_0 x_1 \dots x_k \dots$ with $x_k = x$ and an infinite word w over $2^{\widehat{AP}}$ such that $(w, k, k) \models \widehat{\xi}$ and for each $i \geq 0$ and $p_h \in w(i)$, $(\langle T, V\rangle, x_i) \models \varphi_h$.

Claim 1 suggests the following construction. First, we build a two-way HAA $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ over $\widehat{\Sigma} = 2^{\widehat{AP}}$ accepting $\mathcal{L}_p(\mathsf{E}\widehat{\xi})$ as follows. Let $\mathcal{A}'_{\widehat{\xi}} = \langle \widehat{\Sigma}, Q, Q_0, \rho, F_-, F_+\rangle$ be the Büchi SNWA accepting the set of infinite pointed words over $\widehat{\Sigma}$ satisfying $\widehat{\xi}$ (whose existence is guaranteed by Proposition 3). Then, $\mathcal{A}_{\mathsf{E}\widehat{\xi}} = \langle \widehat{\Sigma}, \widehat{Q}, \widehat{q}_0, \widehat{\delta}, \langle \widehat{F}, \emptyset\rangle\rangle$ extends $\mathcal{A}'_{\widehat{\xi}}$ to trees by simulating it along a single path. Formally, $\widehat{Q} = \{\widehat{q}_0\} \cup (Q \times \{+, -\})$, $\widehat{F} = F_+ \times \{+\}$, where $\{\widehat{q}_0\} \cup (Q \times \{+\})$ is the set of positive states and $Q \times \{-\}$ is the set of negative states. The transition function $\widehat{\delta}$ is defined as follows, where for each $p \in Q$, b_p denotes `true` if $p \in F_-$, and b_p denotes `false` otherwise:

- $\widehat{\delta}((q, +), \widehat{\sigma}) = \bigvee_{p \in \rho(q, \widehat{\sigma}, +)} (\Diamond, (p, +))$;
- $\widehat{\delta}((q, -), \widehat{\sigma}) = \bigvee_{p \in \rho(q, \widehat{\sigma}, -)} (\uparrow, (p, -), b_p)$;
- $\widehat{\delta}(\widehat{q}_0, \widehat{\sigma}) = \bigvee_{q_0 \in Q_0} \bigvee_{q \in \rho(q_0, \widehat{\sigma}, +)} \bigvee_{p \in \rho(q_0, \widehat{\sigma}, -)} [(\Diamond, (q, +)) \wedge (\uparrow, (p, -), b_p)]$.

Note that $\widehat{Q}$ constitutes a single existential component (in particular, $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ is an existential two-way HAA). Intuitively, starting from the initial node x of the input tree, $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ guesses an ε-path $\pi = x_0 x_1 \dots x_k \dots$ such that $x_k = x$ and simulates an infinite run (r_-, r_+) of $\mathcal{A}'_{\widehat{\xi}}$ over the pointed word $(V(x_0)V(x_1)\dots, k)$ by simulating r_- by backward moves along the prefix $x_0 x_1 \dots x_k$ of π and by simulating r_+ by existential moves along the suffix $x_k x_{k+1} \dots$ of π. Thus, $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ accepts all the pointed $\widehat{\Sigma}$-labelled trees satisfying $\mathsf{E}\widehat{\xi}$. Now, we define $\mathcal{A}_\varphi$ as follows. Intuitively, $\mathcal{A}_\varphi$ simulates $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ and starts additional copies of the HAA $\mathcal{A}_{\varphi_i}$. According to Claim 1 these copies guarantee that whenever $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ assumes that proposition p_i labels the current node along the guessed path, then formula φ_i holds at this node.

Formally, $\mathcal{A}_\varphi = \langle \Sigma, \widehat{Q} \cup \bigcup_{i=1}^n Q^i, \widehat{q}_0, \delta, \langle \widehat{F} \cup \bigcup_{i=1}^n G^i, \bigcup_{i=1}^n B^i\rangle\rangle$, where for states in $\bigcup_{i=1}^n Q^i$, the transition function δ agrees with the corresponding δ^i. For $q \in \widehat{Q}$ and $\sigma \in \Sigma$, $\delta(q, \sigma) = \bigvee_{\widehat{\sigma} \in \widehat{\Sigma}} \big(\widehat{\delta}(q, \widehat{\sigma}) \wedge \bigwedge_{p_i \in \widehat{\sigma}} \delta^i(q_0^i, \sigma)\big)$.

Each conjunction in $\delta(q, \sigma)$ corresponds to a label $\widehat{\sigma} \in \widehat{\Sigma}$. Some copies of $\mathcal{A}_\varphi$ (those originated from $\widehat{\delta}(q, \widehat{\sigma})$) proceed as $\mathcal{A}_{\mathsf{E}\widehat{\xi}}$ when it reads $\widehat{\sigma}$. The other copies guarantee that for each $p_i \in \widehat{\sigma}$, φ_i holds at the current node. The set $\widehat{Q}$ constitutes an existential component, with the ordering $\widehat{Q} > Q'$ for each component Q' of $\mathcal{A}_{\varphi_i}$ $(i = 1, \dots, n)$. Correctness of the construction follows from Claim 1 and the induction hypothesis. Since the SNWA $\mathcal{A}'_{\widehat{\xi}}$ has size $2^{O(|\widehat{\xi}|)}$, $\mathcal{A}_{\varphi_i}$ has size $2^{O(|\varphi_i|)}$ and depth $O(|\varphi_i|)$, and the size of $\widehat{\Sigma}$ is $2^{O(|max(\varphi)|)}$, it holds that $\mathcal{A}_\varphi$ has size $2^{O(|\varphi|)}$ and depth $O(|\varphi|)$.

– $\varphi = \mathsf{A}\xi$. We have that $\mathsf{A}\xi \equiv \neg\mathsf{E}\neg\xi$. Let $\mathsf{E}\xi'$ be the positive normal form of $\mathsf{E}\neg\xi$ (note that $|\mathsf{E}\xi'| = O(|\mathsf{E}\neg\xi|)$). By ind. hyp. we can construct a two-way HAA $\mathcal{A}_{\mathsf{E}\xi'}$ over Σ of size $2^{O(|\mathsf{E}\xi'|)} = 2^{O(|\varphi|)}$ and depth $O(|\mathsf{E}\xi'|) = O(|\varphi|)$ such that $\mathcal{L}_p(\mathcal{A}_{\mathsf{E}\xi'})$ is the complement of $\mathcal{L}_p(\varphi)$. By Proposition 2, the dual of $\mathcal{A}_{\mathsf{E}\xi'}$ is a two-way HAA accepting $\mathcal{L}_p(\varphi)$ of size $2^{O(|\varphi|)}$ and depth $O(|\varphi|)$.

Note that if ψ is a ECTL$^*_{lp}$+ formula, then there is no state subformula of ψ of the form $\mathsf{A}\xi$, and the construction given for the cases $\varphi = \varphi_1 \vee \varphi_2$, $\varphi = \varphi_1 \wedge \varphi_2$, and $\varphi = \mathsf{E}\xi$ ensures that $\mathcal{A}_\psi$ is a strictly existential two-way HAA. □

Now, we can prove the main results of this paper.

Theorem 6 (Model-checking). *Model-checking of* CTL$^*_{lp}$+ *can be solved in time polynomial in the size of the structure and doubly exponential in the size of the formula. Moreover, for the existential and universal fragments of* CTL$^*_{lp}$+ *and* CTL$^*_{lp}$*, the problem is* Expspace*-complete and can be solved in space logarithmic in the size of the structure and exponential in the size of the formula.*

Proof. The first part follows from Theorems 2 and 5. For the second part, since ACTL$^*_{lp}$+ (resp., ACTL$^*_{lp}$) is the dual of ECTL$^*_{lp}$+ (resp., ECTL$^*_{lp}$) and co-Expspace = Expspace, it suffices to prove the result for ECTL$^*_{lp}$+ and ECTL$^*_{lp}$. The upper bounds follows from Theorems 2 and 5. The lower bound for ECTL$^*_{lp}$, (hence, the lower bound for ECTL$^*_{lp}$+ follows) can be proved by a reduction from the word problem for Expspace-bounded Turing machines. □

Theorem 7 (Satisfiability). *Satisfiability of* CTL$^*_{lp}$+ *and* CTL$^*_{lp}$ *is* 2Exptime*-complete. Moreover, for the fragment* ACTL$^*_{lp}$*, the problem is* Pspace*-complete, and for the fragments* ACTL$^*_{lp}$+*,* ECTL$^*_{lp}$*,* ECTL$^*_{lp}$+*, it is* Expspace*-complete.*

Proof. 2Exptime-completeness for satisfiability of CTL$^*_{lp}$+ and CTL$^*_{lp}$ directly follows from Theorems 3 and 5 and 2Exptime-hardness of satisfiability of standard CTL* [VS85] (note that CTL* can be trivially linearly translated into CTL$^*_{lp}$). For the logic ACTL$^*_{lp}$, it suffices to observe that a ACTL$^*_{lp}$ formula φ is satisfiable iff the PLTL formula $[\varphi]$ is (initially) satisfiable, where $[\varphi]$ is obtained from φ by omitting all its (universal) path quantifiers. Thus, satisfiability of ACTL$^*_{lp}$ is linearly reducible to satisfiability of PLTL. Since the converse also holds, and satisfiability of PLTL is PSPACE-complete [Var88], the result for ACTL$^*_{lp}$ follows. For the fragments ACTL$^*_{lp}$+, ECTL$^*_{lp}$, ECTL$^*_{lp}$+, the lower bounds are proved by a reduction from the word problem for Expspace-bounded Turing machines. For the upper bounds, membership in Expspace for ECTL$^*_{lp}$ and ECTL$^*_{lp}$+ follows from Theorems 4 and 5. Finally, it remains to prove membership in Expspace for the universal fragment ACTL$^*_{lp}$+ Ḟirst, we extend the linear temporal logic PLTL+ by a new unary modality R (which reads as 'reset'), whose semantics is defined as follows: for a given word w, $(w, i, k) \models \mathsf{R}\xi$ iff $(w, i, i) \models \xi$. Intuitively, the modality R emulates the ability of the path quantifiers of CTL$^*_{lp}$+ to 'reset' the present. We denote by RLTL the extension of PLTL+ with R. Fix an ACTL$^*_{lp}$+ formula φ and let $[\varphi]$ be the RLTL formula obtained from φ by replacing each occurrence of the path quantifier A with R. Evidently, φ is satisfiable iff $[\varphi]$ is satisfiable. Thus, it suffices to show that satisfiability of RLTL is in Expspace. This is proved by a translation, with a single exponential blow-up, of RLTL into Büchi two-way alternating *word* automata (with ε-moves). By [Var98], these automata can be converted,

with a single exponential blow-up, into parity nondeterministic *word* automata whose nonemptiness problem is in NLOGSPACE. Hence, the result follows. □

References

[CE81] Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)

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

[HT87] Hafer, T., Thomas, W.: Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In: Ottmann, T. (ed.) ICALP 1987. LNCS, vol. 267, pp. 269–279. Springer, Heidelberg (1987)

[KP95] Kupferman, O., Pnueli, A.: Once and For All. In: Proc. 10th LICS, pp. 25–35. IEEE Comp. Soc. Press, Los Alamitos (1995)

[KV98] Kupferman, O., Vardi, M.Y.: Weak alternating automata and tree automata emptiness. In: Proc. 30th STOC, pp. 224–233. ACM, New York (1998)

[KV00] Kupferman, O., Vardi, M.Y.: An automata-theoretic approach to modular model checking. ACM Trans. Program. Lang. Syst. 22(1), 87–128 (2000)

[KV06] Kupferman, O., Vardi, M.Y.: Memoryful branching-time logic. In: Proc. 21th LICS, pp. 265–274. IEEE Comp. Soc. Press, Los Alamitos (2006)

[KVW00] Kupferman, O., Vardi, M.Y., Wolper, P.: An Automata-Theoretic Approach to Branching-Time Model Checking. J. ACM 47(2), 312–360 (2000)

[LS95] Laroussinie, F., Schnoebelen, P.: A hierarchy of temporal logics with past. Theoretical Computer Science 148(2), 303–324 (1995)

[LS00] Laroussinie, F., Schnoebelen, P.: Specification in CTL+past for verification in CTL. Information and Computation 156(1–2), 236–263 (2000)

[MS87] Muller, D.E., Schupp, P.E.: Alternating Automata on Infinite Trees. Theoretical Computer Science 54, 267–276 (1987)

[Pnu77] Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, pp. 46–57 (1977)

[PV03] Pistore, M., Vardi, M.Y.: The planning spectrum - one, two, three, infinity. In: Proc. 18th LICS, pp. 234–243. IEEE Comp. Soc. Press, Los Alamitos (2003)

[Var88] Vardi, M.Y.: A temporal fixpoint calculus. In: Proc. 15th Annual POPL, pp. 250–259. ACM, New York (1988)

[Var98] Vardi, M.Y.: Reasoning about the past with two-way automata. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 628–641. Springer, Heidelberg (1998)

[VS85] Vardi, M.Y., Stockmeyer, L.: Improved upper and lower bounds for modal logics of programs. In: Proc. 17th STOC, pp. 240–251 (1985)

[VW94] Vardi, M.Y., Wolper, P.: Reasoning about infinite computations. Information and Computation 115(1), 1–37 (1994)

[Wil99] Wilke, T.: CTL^+ is exponentially more succinct than CTL. In: Pandu Rangan, C., Raman, V., Ramanujam, R. (eds.) FST TCS 1999. LNCS, vol. 1738, pp. 110–121. Springer, Heidelberg (1999)

[Zie98] Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theor. Comput. Sci. 200(1–2), 135–183 (1998)