On the Hybrid Extension of CTL and CTL$^+$

Ahmet Kara[1], Volker Weber[1], Martin Lange[2], and Thomas Schwentick[1]

[1] Technische Universität Dortmund
[2] Ludwig-Maximilians-Universität München

Abstract. The paper studies the expressivity, relative succinctness and complexity of satisfiability for hybrid extensions of the branching-time logics CTL and CTL$^+$ by variables. Previous complexity results show that only fragments with *one variable* do have elementary complexity. It is shown that H^{1}CTL$^+$ and H^{1}CTL, the hybrid extensions with one variable of CTL$^+$ and CTL, respectively, are expressively equivalent but H^{1}CTL$^+$ is exponentially more succinct than H^{1}CTL. On the other hand, HCTL$^+$, the hybrid extension of CTL with arbitrarily many variables does not capture CTL*, as it even cannot express the simple CTL* property EGFp. The satisfiability problem for H^{1}CTL$^+$ is complete for triply exponential time, this remains true for quite weak fragments and quite strong extensions of the logic.

1 Introduction

Reasoning about trees is at the heart of many fields in computer science . A wealth of sometimes quite different frameworks has been proposed for this purpose, according to the needs of the respective application. For reasoning about computation trees as they occur in verification, branching-time logics like CTL and tree automata are two such frameworks. In some settings, the ability to mark a node in a tree and to refer to this node turned out to be useful. As neither classical branching-time logics nor tree automata provide this feature, many different variations have been considered, including tree automata with pebbles [8,22,25], memoryful CTL* [15], branching-time logics with forgettable past [17,18], and logics with the "freeze" operator [12]. It is an obvious question how this feature can be incorporated into branching-time logics *without losing their desirable properties* which made them prevailing in verification [23].

This question leads into the field of hybrid logics, where such extensions of temporal logics are studied [3]. In particular, a hybrid extension of CTL has been introduced in [25]. As usual for branching-time logics, formulas of their hybrid extensions are evaluated at nodes of a computation tree, but it is possible to bind a variable to the current node, to evaluate formulas relative to the root and to check whether the current node is bound to a variable. As an example, the HCTL-formula $\downarrow x @_{\mathrm{root}} \mathrm{EF}(p \wedge \mathrm{EF}x)$ intuitively says "I can place x at the current node, jump back to the root, go to a node where p holds and follow some (downward) path to reach x again. Or, equivalently: "there was a node fulfilling p in the past of the current node".

R. Královič and D. Niwiński (Eds.): MFCS 2009, LNCS 5734, pp. 427–438, 2009.

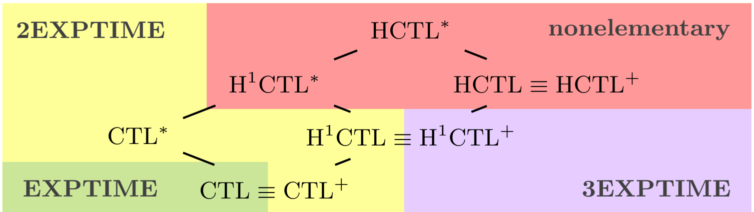

Fig. 1. Expressivity and complexity of satisfiability for hybrid branching-time logics. The lines indicate strict inclusion, unrelated logics are incomparable.

In this paper we continue the investigation of hybrid extensions of classical branching-time logics started in [25]. The main questions considered are (1) expressivity, (2) complexity of the satisfiability problem, and (3) succinctness. Figure 1 shows our results in their context.

Classical branching-time logics are CTL (with polynomial time model checking and exponential time satisfiability) and CTL* (with polynomial space model checking and doubly exponential time satisfiability test). As CTL is sometimes not expressive enough[1] and CTL* is considered too expensive for some applications, there has been an intense investigation of intermediate logics. We take up two of them here: CTL$^+$, where a path formula is a Boolean combination of basic path formulas[2] and ECTL, where fairness properties can be stated explicitly.

Whereas (even simpler) hybrid logics are undecidable over arbitrary transition systems [1], their restriction to trees is decidable via a simple translation to Monadic Second Order logic. However, the complexity of the satisfiability problem is high even for simple hybrid temporal logics over the frame of natural numbers: nonelementary [9] , even if only two variables are allowed [21,25]. The one variable extension of CTL, H^{1}CTL, behaves considerably better, its satisfiability problem can be solved in **2EXPTIME** [25]. This is the reason why this paper concentrates on natural extensions of this complexity-wise relatively modest logic. Even H^{1}CTL can express properties that are not bisimulation-invariant (e.g., that a certain configuration can be reached along two distinct computation paths) and is thus not captured by CTL*. In fact, [25] shows that H^{1}CTL captures and is strictly stronger than CTL with past, another extension of CTL studied in previous work [14]. One of our main results is that H^{1}CTL (and actually even HCTL$^+$) does not capture ECTL (and therefore not CTL*) as it cannot express simple fairness properties like EGFp. To this end, we introduce a simple Ehrenfeucht-style game (in the spirit of [2]). We show that existence of a winning strategy for the second player in the game for a property P implies that P cannot be expressed in HCTL$^+$.

In [25] it is also shown that the satisfiability problem for H^{1}CTL* has nonelementary complexity. We show here that the huge complexity gap between

[1] Some things cannot be expressed at all, some only in a very verbose way.
[2] Precise definitions can be found in Section 2.

H^1CTL and H^1CTL^* does not yet occur between H^1CTL and H^1CTL^+: we prove that there is only an exponential complexity gap between H^1CTL and H^1CTL^+, even when H^1CTL^+ is extended by past modalities and fairness operators. We pinpoint the exact complexity by proving the problem complete for **3EXPTIME**.

The exponential gap between the complexities for satisfiability of H^1CTL and H^1CTL^+ already suggests that H^1CTL^+ might be exponentially more succinct than H^1CTL. In fact, we show an exponential succinctness gap between the two logics by a proof based on the height of finite models. It should be noted that an $\mathcal{O}(n)!$-succinctness gap between CTL and H^1CTL was established in [25]. We mention that there are other papers on hybrid logics and hybrid tree logics that do not study expressiveness or complexity issue, e.g., [10,20].

The paper is organized as follows. Definitions of the logics we use are in Section 2. Expressivity results are presented in Section 3. The complexity results can be found in Section 4, the succinctness results in Section 5. Proofs omitted due to space constraints can be found in the full version of this paper [13].

Note. We mourn the loss of Volker Weber, who died suddenly and unexpectedly on the 7th of April 2009. He was 30 years old. Volker contributed a lot to the present paper which we prepared and submitted after his death.

2 Definitions

Tree logics. We first review the definition of CTL and CTL^* [5]. Formulas of CTL^* are composed from *state formulas* φ and *path formulas* ψ. They have the following abstract syntax.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathrm{E}\psi \mid \mathrm{A}\psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathrm{X}\psi \mid \psi\mathrm{U}\psi$$

We use the customary abbreviations $\mathrm{F}\psi$ for $\top\mathrm{U}\psi$ and $\mathrm{G}\psi$ for $\neg\mathrm{F}\neg\psi$. The semantics of formulas is defined inductively. The semantics of path formulas is defined relative to a tree[3] $\mathcal{T}$, a path π of $\mathcal{T}$ and a position $i \geq 0$ of this path. E.g., $\mathcal{T}, \pi, i \models \psi_1\mathrm{U}\psi_2$ if there is some $j \geq i$ such that $\mathcal{T}, \pi, j \models \psi_2$ and, for each $l, i \leq l < j$, $\mathcal{T}, \pi, l \models \psi_1$. The semantics of state formulas is defined relative to a tree $\mathcal{T}$ and a node v of $\mathcal{T}$. E.g., $\mathcal{T}, v \models \mathrm{E}\psi$ if there is a path π in $\mathcal{T}$, starting from v such that $\mathcal{T}, \pi, 0 \models \psi$. A state formula φ holds in a tree $\mathcal{T}$ if it holds in its root. Thus, sets of trees can be defined by CTL^* state formulas.

CTL is a strict sub-logic of CTL^*. It allows only path formulas of the forms $\mathrm{X}\varphi$ and $\varphi_1\mathrm{U}\varphi_2$ where $\varphi, \varphi_1, \varphi_2$ are state formulas. CTL^+ is the sub-logic of CTL^* where path formulas are Boolean combinations of formulas of the forms $\mathrm{X}\varphi$ and $\varphi_1\mathrm{U}\varphi_2$ and $\varphi, \varphi_1, \varphi_2$ are state formulas.

[3] In general, we consider finite and infinite trees and, correspondingly, finite and infinite paths in trees. It should always be clear from the context whether we restrict attention to finite or infinite trees.

Hybrid logics. In hybrid logics, a limited use of variables is allowed. For a general introduction to hybrid logics we refer to [3]. As mentioned in the introduction, we concentrate in this paper on hybrid logic formulas with *one* variable x. However, as we also discuss logics with more variables, we define hybrid logics $\mathrm{H^kCTL^*}$ with k variables. For each $k \geq 1$, the syntax of $\mathrm{H^kCTL^*}$ is defined by extending $\mathrm{CTL^*}$ with the following rules for state formulas.

$$\varphi ::= \downarrow x_i\, \varphi \mid x_i \mid @_{x_i}\, \varphi \mid \mathrm{root} \mid @_{\mathrm{root}}\, \varphi$$

where $i \in \{1, \ldots, k\}$. The semantics is now relative to a vector $\boldsymbol{u} = (u_1, \ldots, u_k)$ of nodes of $\mathcal{T}$ representing an assignment $x_i \mapsto u_i$. For a node v and $i \leq k$ we write $\boldsymbol{u}[i/v]$ to denote $(u_1, \ldots, u_{i-1}, v, u_{i+1}, \ldots, u_k)$. For a tree $\mathcal{T}$ a node v and a vector $\boldsymbol{u}$, the semantics of the new state formulas is defined as follows.

$$\begin{array}{lll}
\mathcal{T}, v, \boldsymbol{u} \models \downarrow x_i\, \varphi & \text{if} & \mathcal{T}, v, \boldsymbol{u}[i/v] \models \varphi \\
\mathcal{T}, v, \boldsymbol{u} \models x_i & \text{if} & v = u_i \\
\mathcal{T}, v, \boldsymbol{u} \models @_{x_i}\, \varphi & \text{if} & \mathcal{T}, u_i, \boldsymbol{u} \models \varphi \\
\mathcal{T}, v, \boldsymbol{u} \models \mathrm{root} & \text{if} & v \text{ is the root of } \mathcal{T} \\
\mathcal{T}, v, \boldsymbol{u} \models @_{\mathrm{root}}\, \varphi & \text{if} & \mathcal{T}, r, \boldsymbol{u} \models \varphi, \text{ where } r \text{ is the root of } \mathcal{T}
\end{array}$$

Similarly, the semantics of path formulas is defined relative to a tree $\mathcal{T}$, a path π of $\mathcal{T}$, a position $i \geq 0$ of π and a vector $\boldsymbol{u}$. Intuitively, to evaluate a formula $\downarrow x_i\, \varphi$ one puts a pebble x_i on the current node v and evaluates φ. During the evaluation, x_i refers to v (unless it is bound again by another $\downarrow x_i$-quantifier).

The hybrid logics $\mathrm{H^kCTL^+}$ and $\mathrm{H^kCTL}$ are obtained by restricting $\mathrm{H^kCTL^*}$ in the same fashion as for $\mathrm{CTL^+}$ and CTL, respectively. The logic HCTL is the union of all logics $\mathrm{H^kCTL}$, likewise $\mathrm{HCTL^+}$ and $\mathrm{HCTL^*}$.

(Finite) satisfiability of formulas, the notion of a model and equivalence of two (path and state) formulas ψ and ψ' (denoted $\psi \equiv \psi'$) are defined in the obvious way. We say that a logic $\mathcal{L}'$ is at least as expressive as $\mathcal{L}$ (denoted as $\mathcal{L} \leq \mathcal{L}'$) if for every $\varphi \in \mathcal{L}$ there is a $\varphi' \in \mathcal{L}'$ such that $\varphi \equiv \varphi'$. $\mathcal{L}$ and $\mathcal{L}'$ have the *same expressive power* if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \leq \mathcal{L}$. $\mathcal{L}'$ is *strict more expressive* than $\mathcal{L}$ if $\mathcal{L} \leq \mathcal{L}'$ but not $\mathcal{L}' \leq \mathcal{L}$.

Size, depth and succinctness. For each formula φ, we define its *size* $|\varphi|$ as usual and its *depth* $d(\varphi)$ as the nesting depth with respect to path quantifiers.

The formal notion of *succinctness* is a bit delicate. We follow the approach of [11] and refer to the discussion there. We say that a logic $\mathcal{L}$ is *h-succinct in* a logic $\mathcal{L}'$, for a function $h : \mathbb{N} \to \mathbb{R}$, if for every formula φ in $\mathcal{L}$ there is an equivalent formula φ' in $\mathcal{L}'$ such that $|\varphi'| \leq h(|\varphi|)$. $\mathcal{L}$ is *$\mathcal{F}$-succinct in $\mathcal{L}'$* if $\mathcal{L}$ is h-succinct in $\mathcal{L}'$, for some h in function class $\mathcal{F}$. We say that $\mathcal{L}$ is *exponentially more succinct* than $\mathcal{L}'$ if $\mathcal{L}$ is *not* h-succinct in $\mathcal{L}'$, for any function $h \in 2^{o(n)}$.

Normal forms. We say that a $\mathrm{H^kCTL}$ formula is in E-*normal form*, if it does not use the path quantifier A at all. A formula is in U-*normal form* if it only uses the combinations EX, EU and AU (but not, e.g., EG and AX).

Proposition 1. *Let $k \geq 1$. For each* $\mathrm{H^kCTL}$ *formula φ there is an equivalent* $\mathrm{H^kCTL}$*-formula of linear size in* U*-normal form and an equivalent* $\mathrm{H^kCTL}$*-formula in* E*-normal form.*

3 Expressivity of HCTL and HCTL$^+$

3.1 The Expressive Power of HCTL$^+$ Compared to HCTL

Syntactically CTL$^+$ extends CTL by allowing Boolean combinations of path formulas in the scope of a path quantifier A or E. Semantically this gives CTL$^+$ the ability to fix a path and test its properties by *several* path formulas. However in [6] it is shown that every CTL$^+$-formula can be translated to an equivalent CTL-formula. The techniques used there are applicable to the hybrid versions of these logics.

Theorem 2. *For every* $k \geq 1$, $\mathrm{H^kCTL}$ *has the same expressive power as* $\mathrm{H^kCTL^+}$.

Proof (Sketch). For a given $k \geq 1$ it is clear that every $\mathrm{H^kCTL}$-formula is also a $\mathrm{H^kCTL^+}$-formula. It remains to show that every $\mathrm{H^kCTL^+}$-formula can be transformed into an equivalent $\mathrm{H^kCTL}$-formula. In [6] , rules for the transformation of a CTL$^+$ formula into an equivalent CTL formula are given. Here, we have to consider the additional case in which a subformula in the scope of the $\downarrow x$-operator is transformed. However, it is not hard to see that the transformation extends to this case as any assignment to a variable x can be viewed as a proposition that only holds in one node. It should be noted that for a $\mathrm{H^kCTL^+}$-formula φ the whole transformation constructs a $\mathrm{H^kCTL}$-formula of size $2^{\mathcal{O}(|\varphi| \log |\varphi|)}$. □

The transformation algorithm in Theorem 2 also yields an upper bound for the succinctness between $\mathrm{H^1CTL^+}$ and $\mathrm{H^1CTL}$.

Corollary 3. $\mathrm{H^1CTL^+}$ *is* $2^{\mathcal{O}(n \log n)}$*-succinct in* $\mathrm{H^1CTL}$.

3.2 Fairness Is Not Expressible in HCTL$^+$

In this subsection, we show the following result.

Theorem 4. *There is no formula in* HCTL$^+$ *which is logically equivalent to* $\mathrm{E}\overset{\infty}{\mathrm{F}}p$.

Here, $\mathcal{T}, v, \boldsymbol{u} \models \mathrm{E}\overset{\infty}{\mathrm{F}}\varphi$ if there is a path π starting from v that has infinitely many nodes v' with $\mathcal{T}, v', \boldsymbol{u} \models \varphi$. As an immediate consequence of this theorem, HCTL$^+$ does not capture CTL*.

In order to prove Theorem 4, we define an Ehrenfeucht-style game that corresponds to the expressive power of HCTL. A game for a different hybrid logic was studied in [2]. We show that if a set L of trees can be characterized by a HCTL-formula, the spoiler has a winning strategy in the game for L. We expect the converse to be true as well but do not attempt to prove it as it is not needed for our purposes here.

Let L be a set of (finite or infinite) trees. The HCTL-*game* for L is played by two players, the *spoiler* and the *duplicator*. First, the spoiler picks a number k which will be the number of rounds in the core game. Afterwards, the duplicator chooses two trees, $\mathcal{T} \in L$ and $\mathcal{T}' \notin L$. The goal of the spoiler is to make use of the difference between $\mathcal{T}$ and $\mathcal{T}'$ in the core game.

The *core game* consists of k rounds of moves, where in each round i a node from $\mathcal{T}$ and a node from $\mathcal{T}'$ are selected according to the following rules. The spoiler can choose whether she starts her move in $\mathcal{T}$ or in $\mathcal{T}'$ and whether she plays a node move or a path move.

In a *node move* she simply picks a node from $\mathcal{T}$ (or $\mathcal{T}'$) and the duplicator picks a node in the other tree. We refer to these two nodes by a_i (in $\mathcal{T}$) and a'_i (in $\mathcal{T}'$), respectively, where i is the number of the round.

In a *path move*, the spoiler first chooses one of the trees. Let us assume she chooses $\mathcal{T}$, the case of $\mathcal{T}'$ is completely analogous. She picks an already selected node a_j of $\mathcal{T}$, for some $j < i$ and a path π starting in a_j. However, a node a_j can only be selected if there is no other node a_l, $l < i$ below a_j. The duplicator answers by selecting a path π' from a'_j. Then, the spoiler selects some node a'_i from π' and the duplicator selects a node a_i from π.

The duplicator wins the game if at the end the following conditions hold, for every $i, j \leq k$:

- a_i is the root iff a'_i is the root;
- $a_i = a_j$ iff $a'_i = a'_j$;
- for every proposition p, p holds in a_i iff it holds in a'_i;
- there is a (downward) path from a_i to a_j iff there is a path from a'_i to a'_j;
- a_j is a child of a_i iff a'_j is a child of a'_i.

Theorem 5. *If a set L of (finite and infinite) trees can be characterized by a* HCTL*-formula, the spoiler has a winning strategy on the* HCTL*-game for L.*

The proof of Thm. 5 is by induction on the structure of the HCTL-formula [13].

Now we turn to the proof of Thm. 4. It makes use of the following lemma which is easy to prove using standard techniques (see, e.g., [19]). The lemma will be used to show that the duplicator has certain move options on paths starting from the root. The parameter S_k given by the lemma will be used below for the construction of the structures $\mathcal{B}_k$.

For a string $s \in \Sigma^*$ and a symbol $a \in \Sigma$ let $|s|$ denote the length of s and $|s|_a$ the number of occurrences of a in s.

Lemma 6. *For each $k \geq 0$ there is a number $S_k \geq 0$ such that, for each $s \in \{0,1\}^*$ there is an $s' \in \{0,1\}^*$ such that $|s'| \leq S_k$ and $s \equiv_k s'$.*

Here, $\equiv_k$ is equivalence with respect to the k-round Ehrenfeucht game on strings (or equivalently with respect to first-order sentences of quantifier depth k). It should be noted that, if $k \geq 3$ and $s \equiv_k s'$, then the following conditions hold.

- $s \in \{0\}^*$ implies $s' \in \{0\}^*$.
- If the first symbol of s is 1 the same holds for s'.
- If s does not have consecutive 1's, s' does not either.

We fix some S_k, for each k.

The proof of Thm. 4 uses the HCTL-game defined above. Remember that the spoiler opens the game with the choice of a $k \in \mathbb{N}$ and the duplicator responds with two trees $\mathcal{T} \in L$ and $\mathcal{T}' \notin L$. We want to show that the duplicator has a winning strategy so we need to construct such trees, and then need to show that the duplicator has a winning strategy for the k-round core game on $\mathcal{T}$ and $\mathcal{T}'$.

We will use transition systems in order to finitely represent infinite trees. A transition system is a $\mathcal{K} = (V, E, v_0, \ell)$ where (V, E) is a directed graph, $v_0 \in V$, and ℓ labels each state $v \in V$ with a finite set of propositions. The *unraveling* $T(\mathcal{K})$ is a tree with node set V^+ and root v_0. A node $v_0 \ldots v_{n-1}v_n$ is a child of $v_0 \ldots v_{n-1}$ iff $(v_{n-1}, v_n) \in E$. Finally, the label of a node $v_0 \ldots v_n$ is $\ell(v_n)$.

Inspired by [7] we define transition systems $\mathcal{A}_i$, for each $i \geq 0$, as depicted in Fig. 2 (a). Nodes in which p holds are depicted black, the others are white (and we subsequently refer to them as black and white nodes, respectively).

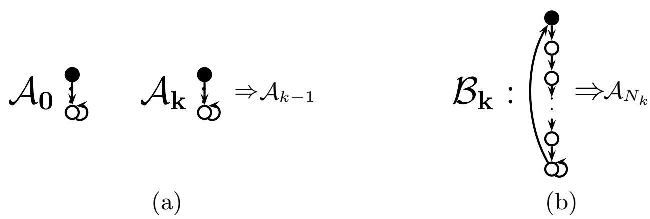

Fig. 2. Illustration of the definition of (a) $\mathcal{A}_k$ and (b) $\mathcal{B}_k$. The path of white nodes in $\mathcal{B}_k$ consists of S_k nodes. The double arrow $\Rightarrow$ indicates that every white node on the left is connected to every black node on the right.

Thus, $\mathcal{A}_0$ has a black (root) node and a white node with a cycle. $\mathcal{A}_i$ has a black (root) node, a white node with a cycle and a copy of $\mathcal{A}_{i-1}$. Furthermore, there is an edge from the white node below the root of $\mathcal{A}_i$ to each black node in the copy of $\mathcal{A}_{i-1}$ (as indicated by $\Rightarrow$). Let $\mathcal{T}_i := T(\mathcal{A}_i)$. We first introduce some notation and state some simple observations concerning the tree $\mathcal{T}_i$.

(1) For a node v in $\mathcal{T}_i$ we denote the maximum number of black nodes on a path starting in v (and not counting v itself) the *height* $h(v)$ of v. Then the root of $\mathcal{T}_i$ has height i.
(2) If u and v are black nodes of some $\mathcal{T}_i$ with $h(u) = h(v)$ then the subtrees $T(u)$ and $T(v)$ induced by u and v are isomorphic.
(3) The height of a tree is defined as the height of its root.
(4) A white node v of height i has one white child (of height i) and i black children of heights $0, \ldots, i-1$. A black node has exactly one white son.
(5) Each finite path π of $\mathcal{T}_i$ induces a string $s(\pi) \in \{0,1\}^*$ in a natural way: $s(\pi)$ has one position, for each node of π, carrying a 1 iff the corresponding node is black.
(6) The root of $\mathcal{T}_i$ has only one child. We call the subtree induced by this (white!) child $\mathcal{U}_i$. If v is a white node of height i then $T(v)$ is isomorphic to $\mathcal{U}_i$.

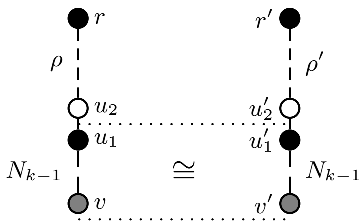

Fig. 3. Illustration of the case where $h(v) \leq N_{k-1}$. The colors of v and v' are not known a priori.

Next we define numbers N_k inductively as follows: $N_0 := 0$ and $N_k := N_{k-1} + \max(S_3, S_k) + 1$.

The following lemma shows that the duplicator has a winning strategy in two structures of the same kind, provided they both have sufficient depth.

Lemma 7. *Let i, j, k be numbers such that $i, j \geq N_k$. Then the duplicator has a winning strategy in the k-round core game on (a) $\mathcal{T}_i$ and $\mathcal{T}_j$, and (b) $\mathcal{U}_i$ and $\mathcal{U}_j$.*

Proof (Sketch). In both cases, the proof is by induction on k, the case $k = 0$ being trivial. We consider (a) first. Let $k > 0$ and let us assume that the spoiler chooses $v \in \mathcal{T}_i$ in her first node move. We distinguish two cases based on the height of v.

$h(v) > N_{k-1}$: Let π denote the path from r to v. By Lemma 6 there is a string s' with $|s'| \leq S_l$ such that $s(\pi) \equiv_l s'$, where $l = \max(k, 3)$. Here, $l \geq 3$ guarantees in particular that s' does not have consecutive 1's. As $j \geq N_k = N_{k-1} + S_l + 1$, there is a node v' of height $\geq N_{k-1}$ in $\mathcal{T}_j$ such that the path π' from r' to v' satifies $s(\pi') = s'$. The duplicator chooses v' as her answer in this round. By a compositional argument, involving the induction hypothesis, it can be shown that the duplicator has a winning strategy for the remaining $k - 1$ rounds.

$h(v) \leq N_{k-1}$: Let π be the path from r to v, and u_1 be the highest black node on π with $h(u_1) \leq N_{k-1}$. Then we must have $h(u_1) = N_{k-1}$ because π contains black nodes of height up to $i \geq N_k$. Hence, u_1 has a white parent u_2 s.t. $h(u_2) > N_{k-1}$. We determine a node u_2' in $\mathcal{T}'$ in the same way we picked v' for v in the first case. In particular, $h(u_2') \geq N_{k-1}$ and for the paths ρ leading from r to u_2 and ρ' leading from r' to u_2' we have $s(\rho) \equiv_k s(\rho')$.

Let u_1' be the black child of u_2' of height $h(u_1)$. As $h(u_1) = h(u_1')$ there is an isomorphism σ between $T(u_1)$ and $T(u_2)$ and we choose $v' := \sigma(v)$. An illustration is given in Figure 3.

The winning strategy of the duplicator for the remaining $k - 1$ rounds follows σ on $T(u_1)$ and $T(u_2)$ and is analogous to the first case in the rest of the trees. The case of path moves is very similar, see [13].

□

We are now prepared to prove Thm. 4.

Proof (of Thm. 4). By Thm. 2 it is sufficient to show that no formula equivalent to $\mathrm{E}\overset{\infty}{\mathrm{F}}p$ exists in HCTL. To this end, we prove that the duplicator has a winning strategy in the HCTL-game for the set of trees fulfilling $\mathrm{E}\overset{\infty}{\mathrm{F}}p$.

We define transition systems $\mathcal{B}_k$, for $k \geq 0$. As illustrated in Figure 2 (b), $\mathcal{B}_k$ has a black root from which a path of length S_k of white nodes starts. The last of these white nodes has a self-loop and an edge back to the root. Furthermore, $\mathcal{B}_k$ has a copy of $\mathcal{A}_{N_k}$ and there is an edge from each white node of the initial path to each black node of the copy of $\mathcal{A}_{N_k}$. Clearly, for each k, $T(\mathcal{B}_k) \models \mathrm{E}\overset{\infty}{\mathrm{F}}p$ and $T(\mathcal{A}_k) \not\models \mathrm{E}\overset{\infty}{\mathrm{F}}p$.

It can be shown that, for each k, the duplicator has a winning strategy in the k-round core game on $\mathcal{T} = T(\mathcal{B}_k)$ and $\mathcal{T}' = T(\mathcal{A}_{N_k})$ [13]. □

4 Satisfiability of H^{1}CTL$^+$

Theorem 8. *Satisfiability of* H^{1}CTL$^+$ *is hard for* **3EXPTIME***.*

Proof. The proof is by reduction from a tiling game (with **3EXPTIME** complexity) to the satisfiability problem of H^{1}CTL$^+$. Actually we show that the lower bound even holds for the fragment of H^{1}CTL$^+$ without the U-operator (but with the F-operator instead).

An instance $I = (T, H, V, F, L, n)$ of the *2EXP-corridor tiling game* consists of a finite set T of *tile types*, two relations $H, V \subseteq T \times T$ which constitute the *horizontal and vertical constraints*, respectively, two sets $F, L \subseteq T$ which describe the starting and end conditions, respectively, and a number n given in unary. The game is played by two players, E and A, on a board consisting of 2^{2^n} columns and (potentially) infinitely many rows. Starting with player E and following the constraints H, V and F the players put tiles to the board consecutively from left to right and row by row. The constraints prescribe the following conditions:

- A tile t' can only be placed immediately to the right of a tile t if $(t, t') \in H$.
- A tile t' can only be placed immediately above a tile t if $(t, t') \in V$.
- The types of all tiles in the first row belong to the set F.

Player E wins the game if a row is completed containing only tiles from L or if A makes a move that violates the constraints. On the other hand, player A wins if E makes a forbidden move or the game goes on ad infinitum.

A winning strategy for E has to yield a countermove for all possible moves of A in all possible reachable situations. Furthermore, the starting condition and the horizontal and vertical constraints have to be respected. Finally, the winning strategy must guarantee that either player A comes into a situation where he can no longer make an allowed move or a row with tiles from L is completed.

The problem to decide for an instance I whether player E has a winning strategy on I is complete for **3EXPTIME**. This follows by a straightforward extension of [4].

□

We can obtain, by simple instantiation, a consequence of this lower complexity bound which will be useful later on in proving the exponential succinctness of H^1CTL^+ in H^1CTL.

Corollary 9. *There are finitely satisfiable* H^1CTL^+ *formulas* φ_n, $n \in \mathbb{N}$, *of size* $\mathcal{O}(n)$ *s.t. every tree model* $\mathcal{T}_n$ *of* φ_n *has height at least* $2^{2^{2^n}}$.

Proof. It is not difficult to construct instances I_n, $n \in \mathbb{N}$, of the 2EXP-tiling game with $|I| = \mathcal{O}(n)$ over a set T of tiles with $|T| = \mathcal{O}(1)$ such that player E has a winning strategy and any successful tiling of the 2^{2^n}-corridor requires $2^{2^{2^n}}$ rows. In order to achieve this, one encodes bits using tiles and forms the constraints in a way that enforces the first row to encode the number 0 in binary of length 2^{2^n}, and each other row to encode the successor in the natural number of the preceding row, while winning requires the number $2^{2^{2^n}}$ to be reached. The construction in the proof of Thm. 8 then maps each such I_n to a formula φ_n of size $\mathcal{O}(n)$ that is finitely satisfiable such that every finite model $\mathcal{T}_n$ of φ_n encodes a winning strategy for player E in the 2^{2^n}-tiling game. Such a strategy will yield a successful tiling of the 2^{2^n}-corridor for any counterstrategy of player A, and any such tiling is encoded on a path of $\mathcal{T}_n$ which contains each row of length 2^{2^n} as a segment of which there are $2^{2^{2^n}}$ many. Thus, $\mathcal{T}_n$ has to have height at least $2^{2^n} \cdot 2^{2^{2^n}}$. □

Using the ideas of the transformation mentioned in Theorem 2 we can show that the lower bound for H^1CTL^+ is optimal. Even for strictly more expressive logics than H^1CTL^+ the satisfiability problem remains in **3EXPTIME**.

Theorem 10. *The satisfiability problem for* H^1CTL^+ *is* **3EXPTIME**-*complete.*

Proof. The lower bound follows from Thm. 8. The upper bound of **3EXPTIME** also holds when H^1CTL^+ is extended by the fairness operators $\overset{\infty}{F}$ and $\overset{\infty}{G}$ and the operators Y (previous) and S (since) [14] which are the past counterparts of X and U. The proof is by an exponential reduction to the satisfiability problem of H^1CTL extended by Y and S which is **2EXPTIME**-complete [24]. It should be noted that because of Thm. 4 the extension of H^1CTL^+ by $\overset{\infty}{F}$ yields a strictly more expressive logic. □

5 The Succinctness of H^1CTL^+ w.r.t. H^1CTL

In Corollary 3 an upper bound of $2^{\mathcal{O}(n \log n)}$ for the succinctness of H^1CTL^+ in H^1CTL is given. In this section we establish the lower bound for the succinctness between the two logics. Actually we show that H^1CTL^+ is exponentially more succinct than H^1CTL. The model-theoretic approach we use in the proof is inspired by [16]. We first establish a kind of small model property for H^1CTL.

Theorem 11. *Every finitely satisfiable* H^1CTL*-formula* φ *with* $|\varphi| = n$ *has a model of depth* $2^{2^{\mathcal{O}(n)}}$.

Proof. In [24] it was shown that for every H^{1}CTL-formula φ, an equivalent nondeterministic Büchi tree automaton A_φ with $2^{2^{\mathcal{O}(|\varphi|)}}$ states can be constructed. It is easy to see by a pumping argument that if A_φ accepts some finite tree at all, it accepts one of depth $2^{2^{\mathcal{O}(|\varphi|)}}$. It should be noted that the construction in [24] only constructs an automaton that is equivalent to φ with respect to satisfiability. However, the only non-equivalent transformation step is from φ to a formula φ' without nested occurrences of the $\downarrow$-operator (Lemma 4.3 in [24]). It is easy to see that this step only affects the propositions of models but not their shape let alone depth. □

Corollary 9 and Theorem 11 together immediately yield the following.

Corollary 12. H^{1}CTL$^+$ *is exponentially more succinct than* H^{1}CTL.

6 Conclusion

The aim of this paper is to contribute to the understanding of one-variable hybrid logics on trees, one of the extensions of temporal logics with reasonable complexity. We showed that H^{1}CTL$^+$ has no additional power over H^{1}CTL but is exponentially more succinct, we settled the complexity of H^{1}CTL$^+$ and showed that hybrid variables do not help in expressing fairness (as HCTL$^+$ cannot express EGFp).

However, we leave a couple of issues for further study, including the following.

- We conjecture that the succinctness gap between H^{1}CTL$^+$ and H^{1}CTL is actually $\theta(n)!$.
- We expect the HCTL-game to capture exactly the expressive power of HCTL. Remember that here we needed and showed only one part of this equivalence.
- The complexity of Model Checking for HCTL has to be explored thoroughly, on trees and on arbitrary transition systems. In this context, two possible semantics should be explored: the one, where variables are bound to nodes of the computation tree and the one which binds nodes to states of the transition system (the latter semantics makes the satisfiability problem undecidable on arbitrary transition systems [2])

References

1. Areces, C., Blackburn, P., Marx, M.: The computational complexity of hybrid temporal logics. Logic Journal of the IGPL 8(5), 653–679 (2000)
2. Areces, C., Blackburn, P., Marx, M.: Hybrid logics: Characterization, interpolation and complexity. J. of Symbolic Logic 66(3), 977–1010 (2001)
3. Areces, C., ten Cate, B.: Hybrid logics. In: Handbook of Modal Logic. Studies in Logic, vol. 3, pp. 821–868. Elsevier, Amsterdam (2007)
4. Chlebus, B.S.: Domino-tiling games. J. Comput. Syst. Sci. 32(3), 374–392 (1986)
5. 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)

6. Emerson, E.A., Halpern, J.Y.: Decision procedures and expressiveness in the temporal logic of branching time. In: STOC 1982, pp. 169–180. ACM, New York (1982)
7. Emerson, E.A., Halpern, J.Y.: “Sometimes” and “not never” revisited: on branching versus linear time temporal logic. J. ACM 33(1), 151–178 (1986)
8. Engelfriet, J., Hoogeboom, H.J.: Tree-walking pebble automata. In: Jewels are Forever, pp. 72–83. Springer, Heidelberg (1999)
9. Franceschet, M., de Rijke, M., Schlingloff, B.-H.: Hybrid logics on linear structures: Expressivity and complexity. In: TIME-ICTL 2003, pp. 192–202. IEEE, Los Alamitos (2003)
10. Goranko, V.: Temporal logics with reference pointers and computation tree logics. Journal of Applied Non-Classical Logics 10(3-4) (2000)
11. Grohe, M., Schweikardt, N.: The succinctness of first-order logic on linear orders. Logical Methods in Computer Science 1(1) (2005)
12. Jurdziński, M., Lazić, R.: Alternation-free mu-calculus for data trees. In: Proc. of the 22th LICS 2007. IEEE, Los Alamitos (2007)
13. Kara, A., Weber, V., Lange, M., Schwentick, T.: On the hybrid extension of CTL and CTL^+, arXiv:0906.2541 [cs.LO]
14. Kupferman, O., Pnueli, A.: Once and for all. In: Proc. of the 10th LICS 1995, pp. 25–35. IEEE, Los Alamitos (1995)
15. Kupferman, O., Vardi, M.Y.: Memoryful branching-time logic. In: Proc. of the 21st LICS 2006, pp. 265–274. IEEE, Los Alamitos (2006)
16. Lange, M.: A purely model-theoretic proof of the exponential succinctness gap between CTL^+ and CTL. Information Processing Letters 108, 308–312 (2008)
17. Laroussinie, F., Schnoebelen, P.: A hierarchy of temporal logics with past. Theor. Comput. Sci. 148(2), 303–324 (1995)
18. Laroussinie, F., Schnoebelen, P.: Specification in CTL+Past for verification in CTL. Inf. Comput. 156(1-2), 236–263 (2000)
19. Libkin, L.: Elements of Finite Model Theory. Springer, Heidelberg (2004)
20. Sattler, U., Vardi, M.Y.: The hybrid μ-calculus. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 76–91. Springer, Heidelberg (2001)
21. Schwentick, T., Weber, V.: Bounded-variable fragments of hybrid logics. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 561–572. Springer, Heidelberg (2007)
22. ten Cate, B., Segoufin, L.: XPath, transitive closure logic, and nested tree walking automata. In: Proc. of the 27th PODS 2008, pp. 251–260. ACM, New York (2008)
23. Vardi, M.Y.: From church and prior to psl. In: Grumberg, O., Veith, H. (eds.) 25 Years of Model Checking. LNCS, vol. 5000, pp. 150–171. Springer, Heidelberg (2008)
24. Weber, V.: Hybrid branching-time logics. CoRR, abs/0708.1723 (2007)
25. Weber, V.: Branching-time logics repeatedly referring to states. Accepted to JoLLI (2009); An extended abstract appeared in the proceedings of HyLo 2007