On the Expressive Power of Hybrid Branching-Time Logics

Daniel Kernberger
School of Electrical Engineering and Computer Science
University of Kassel, Germany
daniel.kernberger@uni-kassel.de

Martin Lange
School of Electrical Engineering and Computer Science
University of Kassel, Germany
martin.lange@uni-kassel.de

Abstract

Hybrid branching-time logics are a powerful extension of branching-time logics like CTL, CTL* or even the modal μ-calculus through the addition of binders, jumps and variable tests. Their expressiveness is not restricted by bisimulation-invariance anymore. Hence, they do not retain the tree model property, and the finite model property is equally lost. Their satisfiability problems are typically undecidable, their model checking problems (on finite models) are decidable with complexities ranging from polynomial to non-elementary time. In this paper we study the expressive power of such hybrid branching-time logics beyond some earlier results about their invariance under hybrid bisimulations. In particular, we aim to extend the hierarchy of non-hybrid branching-time logics CTL, CTL$^+$, CTL* and the modal μ-calculus to their hybrid extensions. We show that most separation results can be transferred to the hybrid world, even though the required techniques become a bit more involved. We also present some collapse results for restricted classes of models that are especially worth investigating, namely linear, tree-shaped and finite models.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics

Keywords and phrases branching-time, μ-calculus, hybrid logics, expressiveness

Digital Object Identifier 10.4230/LIPIcs.TIME.2018.16

1 Introduction

Temporal logics like LTL [14], CTL [7] and CTL* [9] are important specification formalisms for the behaviour of programs because they extend modal logic with the ability to reason about properties of unbounded or infinite computations. Their satisfiability and model checking problems are decidable, ranging from polynomial [7] to doubly exponential time [10]. This is somewhat remarkable given that typical temporal properties like "something happens infinitely often along some path" are not definable in First-Order Logic (FO). The key to decidability is bisimulation-invariance which is rooted in the modal nature of their logical operators. This, however, also limits the expressive power accordingly, for instance by not being able to distinguish a graph from its tree unfolding.

Hybrid logic [2] is the name known for a framework of extensions of modal logics with limited features of FO, aiming at increasing the expressiveness of modal logics whilst hopefully retaining most of its desirable computational properties [1]. Hybrid logics thus feature first-order variables and limited ways of manipulating them in the context of a modal or temporal logic whose evaluation in a Kripke structure can intuitively be understood as a search through the graph. It then becomes possible to bind the current state of evaluation to a variable, to test for re-occurrence of such a state and to continue the evaluation at one such previously marked state. It is not hard to see that such features break bisimulation-invariance. Whilst this can be seen as undesirable for pure program specification purposes, hybrid logics have found some prominence in related fields like knowledge representation [3] etc.

The paper at hand presents some first results on a study of the expressive power of hybrid logics that results from an extension of well-known branching-time temporal logics – mostly CTL* and its fragments like CTL and CTL$^+$– with the aforementioned first-order features. It is part of a larger program to develop a model theory of hybrid branching-time logics. Previous work has investigated their model checking problems [12] (shown to range between polynomial space and non-elementary time/space) and introduced a small syntactical hierarchy of hybridisations of CTL* and its fragments. Thus, there is not just one hybrid CTL* but – depending on how one allows hybrid features to interact with state and path formulas – three versions of hybrid CTL*, distinguished also by computational complexity.

We briefly recall the construction of hybrid branching-time temporal logics in Sect. 2. In Sect. 3 we develop Ehrenfeucht-Fraïssé games for hybrid CTL as a standard tool for bounding the expressive power of a logic from above. These games can be seen as an extension of the games defined in [11] for hybrid CTL interpreted solely on trees. Sect. 4 then shows that some results known for branching-time logics can be lifted to their hybrid variants, at the expense of more involved constructions, though. Sect. 5 then compares hybrid branching-time logics to the extension of the modal μ-calculus with hybrid operators [13, 15]. It gives yet another argument for the distinction of the three versions of hybrid CTL* mentioned above: it is known already that – unlike the case of the non-hybrid logics – two of them cannot be translated into the hybrid μ-calculus. Here we show that the weakest of them can indeed. The paper then concludes in Sect. 6 with an overview of what is known now about the hierarchy of expressiveness amongst hybrid branching-time logics and a discussion on further work in this area.

2 The Full Hybrid Branching-Time Logic

Syntax. Let *Prop* $= \{p, q, \ldots\}$ be a finite set of atomic propositions and *Var* $= \{x, y, \ldots\}$ be a countable set of first-order *variables*. Formulas of the full hybrid branching time logic HCTL$^*_{\mathsf{pp}}$ are given by the grammar

$$\varphi := p \mid x \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}\psi \mid {\downarrow} x.\varphi \mid @_x\, \varphi$$
$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \mid {\downarrow} x.\psi \mid @_x\, \psi,$$

where $p \in Prop$ and $x \in Var$. The formulas derived by φ are called *state* formulas. Those generated by ψ are called *path* formulas. They can only occur as genuine subformulas in an HCTL$^*_{\mathsf{pp}}$ formula. We additionaly require the syntactic sanity restriction that formulas $@_x\, \psi$, where ψ is a genuine path formula, can only occur if there is no path quantifier E or A between $@_x\, \psi$ and the smallest ${\downarrow} x.\psi'$ in the syntax tree above $@_x\, \psi$.

The hybrid operators ${\downarrow} x$ and $@_x$ are called *binder* and *jump*. The atomic formula x is sometimes referred to as *variable test*. We are making use of the usual propositional abbreviations for $\mathtt{tt}$, $\mathtt{ff}$, $\wedge$, as well as the temporal ones $\mathsf{F}\psi := \mathtt{tt}\mathsf{U}\psi$, $\mathsf{G}\psi := \neg\mathsf{F}\neg\psi$, $\mathsf{A}\psi := \neg\mathsf{E}\neg\psi$, $\psi_1\mathsf{R}\psi_2 := \neg(\neg\psi_1\mathsf{U}\neg\psi_2)$ etc.

Semantics. Formulas of HCTL$^*_{\mathsf{pp}}$ are interpreted with respect to Kripke structures. A *Kripke structure* is a tuple $\mathcal{K} = \langle S, \to, L\rangle$ where S is a set of states, $\to \subseteq S \times S$ is a transition relation such that for every $s \in S$ there is a $t \in S$ with $s \to t$ and $L : S \to 2^{AP}$ is a labeling function.

A *path* π in $\mathcal{K}$ is an infinite sequence of pairs of states and the propositions that hold at them: $(s_0, L(s_0)), (s_1, L(s_1)), (s_2, L(s_2)), \ldots \in (S \times 2^{Prop})^\omega$ such that $s_i \to s_{i+1}$ for every $i = 0, 1, \ldots$. For a path π we write π^i to denote the i-th state of the path.

Hybrid branching-time formulas are interpreted over Kripke structures. State formulas are interpreted with respect to a state s of a Kripke structure $K = \langle S, \to, L\rangle$ and a variable mapping $\rho : \mathit{Var} \to S$ in order to give meaning to free variables in the inductive definition of the semantics as follows.

$$\begin{array}{ll}
K, s, \rho \models p & \text{iff } p \in L(s) \\
K, s, \rho \models \neg\varphi & \text{iff } K, s, \rho \not\models \varphi \\
K, s, \rho \models \varphi_1 \vee \varphi_2 & \text{iff } K, s, \rho \models \varphi_1 \text{ or } K, s, \rho \models \varphi_2 \\
K, s, \rho \models \mathsf{E}\psi & \text{iff there exists a path } \pi \text{ with } \pi^0 = s \text{ and } K, \pi, \rho \models \psi \\
K, s, \rho \models x & \text{iff } \rho(x) = s \\
K, s, \rho \models {\downarrow} x.\varphi & \text{iff } K, s, \rho[x \mapsto s] \models \varphi \\
K, s, \rho \models @_x\, \varphi & \text{iff } K, \rho(x), \rho \models \varphi.
\end{array}$$

Path formulas are interpreted on a path. To give meaning to them properly we need a function $\sigma : \mathit{Var} \to \mathbb{N}$ that stores the position of a bound variable on the current path under evaluation which helps to give meaning to the jump operator.[1] Thus, path formulas here are interpreted over a path π in K, a moment k on the path, a variable mapping ρ and a function σ storing the position on the path to which a variable is bound:

$$\begin{array}{ll}
K, \pi, k, \rho, \sigma \models \varphi & \text{iff } K, \pi^k, \rho \models \varphi \\
K, \pi, k, \rho, \sigma \models \psi_1 \vee \psi_2 & \text{iff } K, \pi, k, \rho, \sigma \models \psi_1 \text{ or } K, \pi, k, \rho, \sigma \models \psi_2 \\
K, \pi, k, \rho, \sigma \models \mathsf{X}\psi & \text{iff } K, \pi, k+1, \rho, \sigma \models \psi \\
K, \pi, k, \rho, \sigma \models \psi_1 \mathsf{U} \psi_2 & \text{iff there exists } j \in \mathbb{N} \text{ with } j \geq k \text{ such that } K, \pi, j, \rho, \sigma \models \psi_2 \\
 & \quad \text{and for all } k \leq i < j\text{: } K, \pi, i, \rho, \sigma \models \psi_1 \\
K, \pi, k, \rho, \sigma \models {\downarrow} x.\psi & \text{iff } K, \pi, k, \rho[x \to \pi^k], \sigma[x \to k] \models \psi \\
K, \pi, k, \rho, \sigma \models @_x\, \psi & \text{iff } K, \pi, \sigma(x), \rho, \sigma \models \psi.
\end{array}$$

Fragments. The index $\cdot_{\mathsf{pp}}$ in the logic's name stands for path−path and indicates that binders as well as jumps can range over path formulas (and therefore also state formulas). We obtain (syntactically) weaker fragments imposing stronger restrictions here.

First, if we restrict path formulas to

$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \mid {\downarrow} x.\psi,$$

i.e. disallow jumps on path formulas or, equivalently, require them to operate on genuine state formulas only, then we get the path−state fragment $\text{HCTL}^*_{\mathsf{ps}}$. If we also disallow ${\downarrow} x.\psi$ on path formulas we obtain the state−state fragment $\text{HCTL}^*_{\mathsf{ss}}$ in which hybrid-operators can only occur as state-formulas.

[1] The syntactic restriction of $\text{HCTL}^*_{\mathsf{pp}}$ formulas about the non-occurrence of path quantifiers between binders and corresponding jumps ensures that variables which are referenced while evaluating a path formula are actually bound on the same path. This makes the semantics be well-defined.

We can also consider hybrid variants of weaker branching-time temporal logics as they are known in the literature: if we restrict the temporal operators further to not allow nesting of path formulas with the exception of fairness constraints, i.e. path formulas are generated by the grammar

$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\varphi \mid \varphi\mathsf{U}\varphi \mid \mathsf{GF}\varphi$$

we obtain the logic HFCTL$^+$. Restricting the grammar even further to

$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\varphi \mid \varphi\mathsf{U}\varphi$$

we get HCTL$^+$. And finally, also disallowing boolean combinations of path formulas,

$$\psi := \mathsf{X}\varphi \mid \varphi\mathsf{U}\varphi$$

we get HCTL.

Lastly, we also need linear temporal logic for some technical details. We obtain hybrid LTL by restricting the grammar of HCTL$^*_{\mathsf{pp}}$ to

$$\varphi := \mathsf{E}\psi$$
$$\psi := p \mid x \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\psi \mid \psi\mathsf{U}\psi \mid {\downarrow}x.\psi \mid @_x\,\psi.$$

Thus, hybrid LTL formulas basically consist only of a single path formula. Furthermore, hybrid LTL formulas are only interpreted over linear Kripke structures, i.e. Kripke structures $\mathcal{K} = \langle S, \to, L\rangle$ with $S = \mathbb{N}$ and $s \to t$ if and only if $t = s + 1$.

The usual non-hybrid temporal logics CTL, CTL$^+$, FCTL$^+$, CTL* and LTL are obtained by completely restricting the use of hybrid operators in their respective hybrid-variants. Note that the possibility to nest path formulas arbitrarily is vital for the distinction between the three fragments of path−path etc. formulas. Hence, for hybrid variants of branching-time logics "smaller" than CTL* we do not make these distinctions anymore.

3 Ehrenfeucht-Fraïssé Game for HCTL

We start by defining Ehrenfeucht-Fraïssé (EF) games in order to capture the expressive power of HCTL. Such games often prove to be useful when comparing the expressive power of two logics because they condense all possibilities of how to distinguish two structures via a logical formula into a single framework of finding a winning strategy for one player. In this paper we will only use such games for HCTL. However, the general framework of these games can also be extended to similar games for the other hybrid logics above HCTL and may prove to be useful for future research into this topic.

Hybrid logics extend branching-time logics with certain first-order aspects. Thus, it is not surprising that these expressiveness games combine features from branching-time logics with aspects of EF games which are known from FO [6].

▶ **Definition 1.** Let $K_0 = \langle S_0, \to_0, L_0\rangle$ and $K_1 = \langle S_1, \to_1, L_1\rangle$ be two Kripke structures. The game $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, K_1, s_1)$ is played between two players – Spoiler and Duplicator on K_0 and K_1.

The game is played for m rounds. At the beginning we only have one pebble in each structure placed at s_0 resp. s_1. In each round a new pair of pebbles gets placed on K_0 and K_1 according to the following rules. Spoiler first chooses one of the structures K_i, $i \in \{0, 1\}$, and a previously placed pebble p_i in this structure and then chooses one of the following moves:

- Choose a successor of p_i and place a new pebble on this successor. Duplicator then responds by choosing a pebble in K_{1-i} and also places a new pebble on one of its successors.
- Spoiler chooses a path π_i starting at p_i and a position l on this path. Then Duplicator chooses a path π_{1-i} on K_{1-i} starting at some previously placed pebble and some position l' on this path. Now Spoiler has two options. He can place a new pebble on π_i^l, forcing Duplicator to place a new pebble on $\pi_{1-i}^{l'}$. Or he can choose some $k' < l'$ and place a new pebble at $\pi_{1-i}^{k'}$ on K_{1-i}. Duplicator then has the possibility to choose some $k < l$ and place a new pebble at π_i^k on K_i.

It is Duplicator's task to maintain the following conditions after each round:

- For all pairs of pebbles placed in some round it holds that the states marked by those pebbles agree on all $p \in \mathit{Prop}$.
- For all pairs of pebbles (p_i, p_i') and (p_j, p_j') it holds that $p_i = p_j$ iff $p_i' = p_j'$

Spoiler wins if Duplicator cannot maintain these conditions after some round. Duplicator wins if Spoiler has not won after m rounds.

The following theorem can be proven by a standard induction on the number of rounds in this game; it is carried out in the appendix. Observe that the two types of moves cover the until- and next-operators. Hybrid operators are covered by placing pebbles and starting from some pebble in each round.

▶ **Theorem 2.** *If Duplicator wins $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, K_1, s_1)$ then it holds for all formulas $\varphi \in$ HCTL with temporal nesting depth at most m that $K_0, s_0 \models \varphi$ if and only if $K_1, s_1 \models \varphi$.*

These games are essentially a combination of Ehrenfeucht-Fraissé games for first-order logic and expressiveness games for branching-time logics. The moves closely resemble the temporal operators in branching-time logic. However, in each step we also remember the new state similar to Ehrenfeucht-Fraissé games. As a consequence winning strategies also combine elements of both types of games.

4 The Expressive Power of Hybrid Branching-Time Logics

We begin by studying the connection between HCTL and HCTL$^+$ and will then continue to work our way up to the logics above HCTL$^+$.

4.1 HCTL$^+$ and HCTL

We first show that HCTL$^+$ $\equiv$ HCTL. Thus, as in the non-hybrid case adding boolean connectives to the path formulas does not increase the expressive power. The proof is very similar to the translation in the non-hybrid case [8] and has already been extended to HCTL$^+$ over tree-structures [11]. However, we will build upon the translation in Theorem 5. So we briefly review the key elements for the translation in the hybrid case here.

▶ **Theorem 3.** *For each* HCTL$^+$ *formula φ there is an* HCTL *formula φ' such that for all Kripke structures $K = \langle S, \to, L\rangle$, states $s \in S$ and variable assignments $\sigma : \mathit{Var} \to S$ it holds that $K, s, \sigma \models \varphi$ if and only if $K, s, \sigma \models \varphi'$.*

Proof sketch. Inspecting the grammar for HCTL$^+$, we see that the hybrid operators only occur as state formulas. Thus, path formulas are essentially non-hybrid in the sense that they are evaluated with respect to a fixed variable interpretation. Fixed variables however can simply be regarded as atomic propositions that happen to hold at exactly one state.

For this reason, we can simply utilise the techniques that are used in the non-hybrid case and follow [8]: we first transform path formulas into a normal form such that every path formula is a conjunction of Next-, Until- and Generally-operators and then guess the order in which all Until-formulas will be satisfied making sure that the Next- and Generally-formulas are also satisfied. ◀

A detailed proof of Theorem 3 can be found in the appendix.

4.2 HFCTL$^+$ and HCTL$^+$

In the non-hybrid case we know that CTL$^+$ is less expressive than FCTL$^+$. For example it is shown in [9] that the formula $\mathsf{EGF}p$, which states that there is a path along which p holds infinitely often, cannot be expressed by CTL$^+$.

A similar result was already shown for HCTL$^+$ interpreted over computation trees in [11]. This of course also gives us a separation result over general Kripke structures.

▶ **Theorem 4.** *There is no formula in* HCTL$^+$ *that is equivalent to the* HFCTL$^+$ *formula* $\mathsf{EGF}p$*.*

However, if we only consider finite structures the picture is different. Next we will see that Theorem 4 no longer holds on finite structures. This is because on finite structures we can characterise an infinite occurrence of p on a finite structure by a state that satisfies p and that lies on some (finite) loop in the structure. Thus, on finite structures we get that the HFCTL$^+$ formula $\mathsf{EGF}p$ is equivalent to the HCTL formula $\mathsf{EF}{\downarrow}x.\mathsf{EF}(p \wedge \mathsf{EXEF}x)$.

Extending the well-known translation from HCTL$^+$ to HCTL a bit we get an even stronger result:

▶ **Theorem 5.** *On finite structures, every* HFCTL$^+$ *formula is logically equivalent to an* HCTL *formula.*

Proof. We start with the same equivalences used in the proof for Theorem 3 and transform each path formula into one of the form

$$\mathsf{E}(\mathsf{X}\Lambda_1 \wedge \bigwedge_{i\in I_1} \varphi_i \mathsf{U} \psi_i \wedge \mathsf{G}\Lambda_2 \wedge (\bigwedge_{i\in I_2} \mathsf{GF}\chi_i)). \tag{1}$$

To transform this formula into an HCTL formula we first guess the order in which all Until-formulas are satisfied – ignoring the fairness constraints for the initial part – and then we guess a point from which there are cyclic paths along which all χ_i formulas will be satisfied. Thus, we get the following translation:

$$\Lambda_2 \wedge \bigvee_{J\subseteq I_1} (\bigwedge_{j\notin J} \psi_j) \wedge (\bigwedge_{j\notin J} \varphi_j) \wedge \mathsf{EX}\Big(\Lambda_1 \wedge \bigvee_{\pi\in\mathsf{Perm}(J)} \mathsf{E}((\Lambda_2 \wedge \bigwedge_{j\in J} \varphi_{\pi(j)})\mathsf{U}\Big(\psi_{\pi(1)} \wedge$$

$$\vdots$$

$$\mathsf{E}((\Lambda_2 \wedge \varphi_{\pi(|J|)})\mathsf{U}(\psi_{\pi(|J|)} \wedge \xi)\Big) \dots\Big)$$

where $\mathsf{Perm}(J)$ denotes the set of all permutations over J and

$$\xi := \mathsf{E}(\Lambda_2 \mathsf{U}(\Lambda_2 \wedge {\downarrow}x. \bigwedge_{i\in I_2} \mathsf{E}(\Lambda_2 \mathsf{U}(\Lambda_2 \wedge \chi_i \wedge \mathsf{E}(\Lambda_2 \mathsf{U} x))))).$$

Suppose some state satisfies (1) in some finite structure. Then there is a path π satisfying all conjuncts of (1). We only argue correctness of the translation for the infinite part of the path after all Until-formulas were satisfied and also ignore that Λ_2 is satisfied on every state of the path. Correctness for the initial part follows along the same lines as the translation from HCTL$^+$ to HCTL in Theorem 3.

Since the structure is finite there has to be some point x occurring infinitely often along π. Furthermore, since every χ_i is satisfied infinitely often there has to be a part of the path such that χ_i is satisfied on some state between two occurrences of x. Thus, for every i there is a cycle starting at x in the structure along which χ_i is satisfied. Hence, ξ is satisfied. The converse direction follows by a piecewise reconstruction of the whole path with infinitely many occurrences of each cycle satisfying some χ_i. ◀

4.3 HCTL$^*_{\mathsf{ss}}$ and HFCTL$^+$

In the following we will prove that HCTL$^*_{\mathsf{ss}}$ is still more expressive than HFCTL$^+$. We will first prove that there are two classes of *finite structures* distinguishable by HCTL$^*_{\mathsf{ss}}$ such that no HCTL formula can distinguish these classes. Together with Theorem 5 this will also prove that HCTL$^*_{\mathsf{ss}}$ is more expressive than HFCTL$^+$.

To see this, we define the structure $\mathcal{A}$ as depicted in Figure 1a. Note that $\mathcal{A}$, despite being infinite as a whole, is essentially finite from each state because every path simply traverses the structure downwards ending either in t_1 or in t'_1. In the following we refer to the index of a state's name as the level of the structure and the letter of its name as the type of the state. Also note that each path that goes from level i to level $i-1$ goes either through s_{i-1} or s'_{i-1}.

▶ **Theorem 6.** *Let $n \in \mathbb{N}$ and $m = 2^{n+1} + n$. Duplicator wins $\mathcal{G}^n_{\mathrm{HCTL}}(\mathcal{A}, s_m, \mathcal{A}, s'_m)$.*

Proof. We describe a winning strategy for Duplicator. Suppose that i rounds have been played already. To win, Duplicator maintains the following invariant throughout the game:

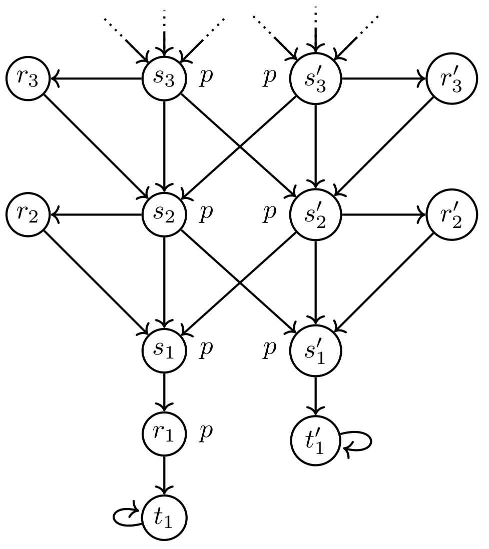
- The pebbles placed by Duplicator are always on the same level and of the same type as Spoiler's pebble in the same round.
- There is some $k \geq 2^{n-i}$ such that each pair of pebbles placed by Spoiler and Duplicator in the same round on level k or smaller mark exactly the same state. Furthermore, pairs of pebbles placed above level k are on opposing sides in the structure.
- The first pair of pebbles placed above level k is at least on level $k + 2^{n-i} + (n-i)$.

It is obvious that if Duplicator can maintain this invariant for n rounds then she wins. Furthermore, at the beginning of the game the invariant holds with $k = 2^n$.

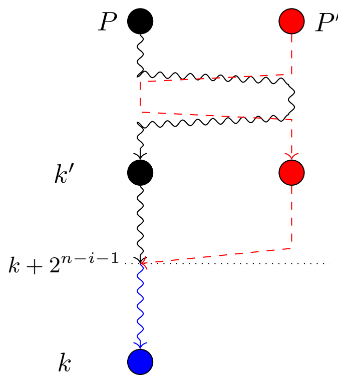
Suppose now that i rounds have been played in the game and Spoiler decides to move from some pebble P. Duplicator will always answer with the pebble P' on the same level. Spoiler has two types of moves available. First, he can simply choose a successor of P. If P, P' mark the same state then Duplicator simply copies Spoiler's pick, if not then P' is on the opposing side of the structure compared to P and Duplicator chooses the same type of successor on the other side of the structure. In both cases the invariant is maintained.

Suppose Spoiler chooses some path through the structure and some point on this path. It is obvious that to maintain the first part of the invariant Duplicator has to choose his endpoint of the path on the same level and at the same type of node as Spoiler. There are two possibilities for Spoiler's path:

First, P is at or below level k. In this case P, P' mark the same state. Thus, Duplicator simply copies Spoiler's path as well as his choice for the next pebble.



(a) Structure $\mathcal{A}$.



(b) Sketch of Duplicator's path choice. The black path is Spoiler's choice. Duplicator's path is depicted in red. The blue part means that both paths have joined.

Figure 1 Structure $\mathcal{A}$ and a sketch of Duplicator's path choice on $\mathcal{A}$.

Second, P is above level k. So P and P' are on opposing sides of the structure. Let k' be the level of the lowest pebbles above k. Then $k' \geq k + 2^{n-i} + (n - i)$. Suppose for the moment that Spoiler has chosen his endpoint of the path anywhere else than at the state $r_{k+2^{n-i-1}+1}$. Then Duplicator chooses his path as follows:

- Up to level $k + 2^{n-i-1} + 1$ he mimics Spoiler's path but on the opposing side of the structure. For example if Spoiler's path goes through s_j then Duplicator's will go through s'_j etc.
- At $s_{k+2^{n-i-1}+1}$ or $s'_{k+2^{n-i-1}+1}$ Duplicator's path changes sides to meet up with Spoiler's path at $s_{k+2^{n-i-1}}$ or $s'_{k+2^{n-i-1}}$ – even if Spoiler's path goes through $r_{k+2^{n-i-1}+1}$ or $r'_{k+2^{n-i-1}+1}$.
- From $s_{k+2^{n-i-1}}$ or $s'_{k+2^{n-i-1}}$ he simply copies Spoiler's path.

A rough illustration of Duplicator's path is depicted in Figure 1b. Spoiler now has only two options (the case that Spoiler chooses to play the endpoint of his path gets subsumed here since Duplicator will always play the same node on the same level). Either he places a new pebble above level $k + 2^{n-i-1}$ on Duplicator's path. Then Duplicator can answer with the same type of node on the same level in Spoiler's path on the opposing side of the structure. Or Spoiler places a pebble at or below level $k + 2^{n-i-1}$ on Duplicator's path. In this case Duplicator simply answers with the same state since both paths are the same there.

In both cases one can check that the invariant is maintained, possibly with a bigger k if Spoiler's choice is somewhere between level k and $k + 2^{n-i-1}$. In any case, the gap to k' is large enough to also maintain the third part of the invariant. This strategy only works if Spoiler cannot explicitly choose to go to $r_{k+2^{n-i-1}+1}$ or $r'_{k+2^{n-i-1}+1}$, i.e. if he does not choose either of those points as the endpoints of his path, since Duplicator's path changes the sides of the structure on this level and thus cannot go through the opposing r-node.

So, suppose Spoiler has chosen a path through $r_{k+2^{n-i-1}+1}$ or $r'_{k+2^{n-i-1}+1}$ and has chosen this point as his endpoint of the path. In this case Duplicator chooses almost the same path as before, however he switches sides to Spoiler's path one level earlier such that if Spoiler decides to go to $r_{k+2^{n-i-1}+1}$ resp. $r'_{k+2^{n-i-1}+1}$ Duplicator can move to the same state. As one can check the invariant here is also maintained with a new k between the previous k and $k+2^{n-i-1}+1$ – at most one bigger than before.

Thus, by maintaining this invariant Duplicator wins $\mathcal{G}^n_{\text{HCTL}}(\mathcal{A}, s_{2^{n+1}+n}, \mathcal{A}, s'_{2^{n+1}+n})$. ◀

▶ **Theorem 7.** *There is no* HFCTL$^+$ *formula that is logically equivalent to the* CTL* *formula* $\mathsf{AF}(p \wedge \mathsf{X}p)$*.*

Proof. Suppose there was such a formula. By Theorem 5 this formula would be equivalent to an HCTL formula on finite structures. Let n be the operator depth of this HCTL formula. Then by Theorem 2 and 6 this formula cannot distinguish between the states $s_{2^{n+1}+n}$ and $s'_{2^{n+1}+n}$ in $\mathcal{A}$. However, $\mathcal{A}, s_j \models \mathsf{AF}(p \wedge \mathsf{X}p)$ while $\mathcal{A}, s'_j \not\models \mathsf{AF}(p \wedge \mathsf{X}p)$ for all j. ◀

Thus, we get that CTL* and HFCTL$^+$ are incomparable and since HCTL$^*_{\mathsf{ss}}$ is an extension of CTL* and HFCTL$^+$:

▶ **Corollary 8.** *Already on finite structures* HCTL$^*_{\mathsf{ss}}$ *is more expressive than* HFCTL$^+$.

This result transfers to general Kripke structures. However, it is quite interesting to see that this proof does not simply carry over to the class of computation trees.

There, we can exploit that the path to some state s is unique. Using this, we can for example get that on tree structures the formula $\mathsf{AF}(p \wedge \mathsf{X}p)$ is equivalent to the HCTL formula $\downarrow s.\mathsf{AF}(p \wedge \downarrow x. @_s\, \mathsf{EF}(\mathsf{EX}x \wedge p))$. The latter formula states that on all paths we can finally find some state y satisfying p such that if we jump back to the root of the tree we can find a state that has a successor y and also satisfies p. Since predecessors on trees are unique this equivalence holds on all trees.

Since the latter formula is already in HCTL the proof showing that CTL* is not subsumed by HCTL does not carry over.

5 Hybrid Temporal Logics and the Hybrid μ-Calculus

The hybrid μ-calculus H$_\mu$ is an extension of the modal μ-calculus L$_\mu$ with hybrid operators. However, despite increasing the expressive power of the modal μ-calculus substantially it is known that H$_\mu$ – contrary to the non-hybrid case – does not subsume all hybrid extensions of CTL*. In [13] it was shown that formulas of H$_\mu$ using at most k first-order variables are invariant under hybrid k-bisimulations – a bisimulation notion that links $(k+1)$-tuples of states such that the i-th states of these tuples have matching atomic propositions, matching (hybrid) accessibility relations which includes jumping to the other k states and rebinding them and also match in regard to the other k states of the tuple. It is not too hard, though, to see that HCTL$^*_{\mathsf{ps}}$ contains formulas which are not invariant under hybrid k-bisimulations for any k. An example is $\mathsf{EG}(\downarrow x.\mathsf{XG}\neg x)$ stating that there is a loop-less infinite path. As an immediate consequence one obtains that HCTL$^*_{\mathsf{ps}}$ cannot be embedded into H$_\mu$.

We continue the study of the connection between hybrid extensions of CTL* and H$_\mu$ in this section. First, we briefly recall H$_\mu$. Secondly, we show that at least HCTL$^*_{\mathsf{ss}}$ is subsumed by H$_\mu$. As a byproduct of this translation we get that HCTL$^*_{\mathsf{ps}}$ is strictly more expressive than HCTL$^*_{\mathsf{ss}}$ and that the latter is also invariant under hybrid k-bisimulations. And finally, we lift the standard proof that the μ-calculus is more expressive than CTL* to the hybrid world, showing that HCTL$^*_{\mathsf{pp}}$ and H$_\mu$ are incomparable in terms of their expressive power.

5.1 The Hybrid μ-Calculus

Let $\mathit{Var}_2 = \{X, Y, \ldots\}$ be a countable set of second-order variables that is disjoint from Var and Prop. Formulas of the fully hybrid μ-calculus H_μ are given by the grammar

$$\varphi := p \mid x \mid X \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid @_x\,\varphi \mid {\downarrow} x.\varphi \mid \mu X.\varphi(X)$$

where $p \in \mathit{Prop}$, $x \in \mathit{Var}$ and $X \in \mathit{Var}_2$. The modal μ-calculus L_μ is obtained by disallowing the occurrence of first-order variables. We make use of $\mathtt{tt}$, $\mathtt{ff}$, $\wedge$, $\Diamond$, $\nu X.\varphi$ as abbreviations in the usual way, and we assume the following standard sanity condition on formulas: every $X \in \mathit{Var}_2$ is bound at most once by a fixpoint quantifier μ or ν and can only occur under an even number of negations within its binding formula.

Formulas of H_μ are interpreted over Kripke structures $K = \langle S, \to, L\rangle$. Formally the semantics for H_μ with respect to a Kripke structure $K = \langle S, \to, L\rangle$ over Prop and an assignment $\rho : \mathit{Var}_2 \to 2^{S \times (\mathit{Var} \to S)}$ is the following:

$$\begin{aligned}
[\![p]\!]^K_\rho &= \{(s,\sigma) \mid p \in L(s)\},\\
[\![X]\!]^K_\rho &= \rho(X),\\
[\![x]\!]^K_\rho &= \{(s,\sigma) \mid s = \sigma(x)\},\\
[\![\neg\varphi]\!]^K_\rho &= \{(s,\sigma) \mid (s,\sigma) \notin [\![\varphi]\!]^K_\rho\},\\
[\![\varphi_1 \vee \varphi_2]\!]^K_\rho &= [\![\varphi_1]\!]^K_\rho \cup [\![\varphi_2]\!]^K_\rho,\\
[\![\Box\varphi]\!]^K_\rho &= \{(s,\sigma) \mid \forall t \in S\text{: if } s \to t, \text{ then } (t,\sigma) \in [\![\varphi]\!]^K_\rho\},\\
[\![@_x\,\varphi]\!]^K_\rho &= \{(s,\sigma) \mid (\sigma(x),\sigma) \in [\![\varphi]\!]^K_\rho\},\\
[\![{\downarrow} x.\varphi]\!]^K_\rho &= \{(s,\sigma) \mid (s,\sigma[x \mapsto s]) \in [\![\varphi]\!]^K_\rho\},\\
[\![\mu X.\varphi(X)]\!]^K_\rho &= \bigcap\{T \subseteq S \times (\mathit{Var} \to S) \mid [\![\varphi]\!]^K_{\rho[X \mapsto T]} \subseteq T\}
\end{aligned}$$

with $p \in \mathit{Prop}$, $x \in \mathit{Var}$ and $X \in \mathit{Var}_2$. We write $K, s, \sigma, \rho \models \varphi$ if $(s,\sigma) \in [\![\varphi]\!]^K_\rho$. If there are no free second-order variables we also may drop ρ.

5.2 $\mathrm{HCTL}^*_{\mathsf{ss}}$ and H_μ

Our aim is to translate $\mathrm{HCTL}^*_{\mathsf{ss}}$ into H_μ. However, already for CTL^* the translation into the μ-calculus is nontrivial. A key part in the non-hybrid translation is that path formulas can be regarded as simple LTL formulas with embedded CTL^* state-formulas. Ignoring the state formulas for a moment, we can translate LTL-formulas into suitable Büchi automata on ω-words which accept a path if and only if it satisfies this formula. These automata can then be translated into a μ-calculus formula that basically simulates the automaton along *some* path. The embedded state formulas can be handled by a decomposition method as usual in CTL^*.

Something similar can be done in the hybrid case. Inspecting the grammar of $\mathrm{HCTL}^*_{\mathsf{ss}}$ again, we see that path formulas are basically also only simple LTL formulas with embedded state formulas. In particular path formulas are evaluated with respect to a fixed variable interpretation. The only difference is that these path formulas are not simply over the atomic propositions as vocabulary but they also encompass some variables that can be used. For example the path formula $\mathsf{F}x$ in ${\downarrow} x.\mathsf{EF}x$ features the variable x and thus a Büchi automaton that checks that somewhere along the path x holds, also needs to take care of these variables.

The idea now is the same as in the non-hybrid case: for path formulas we will construct a Büchi-automaton over an extended vocabulary that treats variables simply like atomic propositions. This automaton, of-course, accepts more paths than in hybrid-logic intended. For example the restriction that a variable only holds at exactly one state will not be checked by this automaton.

However, since these automata are only an intermediate state in the translation this will not be a problem because we will then translate these automata into suitable H_μ formulas which will – by the semantics of H_μ– check that variables only occur at a single state.

To get started we need some definitions and notation:

▶ **Definition 9.** A path formula ψ is called *pure* if there are no occurrences of path quantifiers E, A or hybrid operators $\downarrow, @$ in ψ.

▶ **Definition 10.** Let $K = \langle S, \rightarrow, L\rangle$ be a Kripke structure over *Prop* and $\sigma : \mathit{Var} \rightarrow S$ a variable assignment. We define $K_\sigma = \langle S, \rightarrow, L'\rangle$ with $L'(s) := L(s) \cup \{x \in \mathit{Var} \mid \sigma(x) = s\}$.

Thus, K_σ extends K with new propositions for each variable. There is a 1-1 connection between paths from K and paths from K_σ. Thus, let $\mathsf{pr} : \mathsf{Paths}(K) \rightarrow \mathsf{Paths}(K_\sigma)$ be the unique function that maps a path from K to its copy in K_σ. Thus, for every path $\pi \in (S \times 2^{\mathit{Prop}})^\omega$ we get $\mathsf{pr}(\pi) \in (S \times 2^{\mathit{Prop} \cup \mathit{Var}})^\omega$.

▶ **Definition 11.** Let $K = \langle S, \rightarrow, L\rangle$ be a Kripke structure and $\sigma : \mathit{Var} \rightarrow S$ a variable assignment. A path $\pi \in (S \times 2^{\mathit{Prop} \cup \mathit{Var}})^\omega$ is called *consistent with* σ if for all positions $\pi^i = (s, M)$ on π it holds that $s = \sigma(x)$ if and only if $x \in M$ for all $x \in \mathit{Var}$.

The following lemma about consistent paths is easy to see:

▶ **Lemma 12.** *Let $K = \langle S, \rightarrow, L\rangle$ be a Kripke structure over Prop, $\sigma : \mathit{Var} \rightarrow S$ a variable assignment and π be a path in K. Then $\mathsf{pr}(\pi)$ is consistent with σ.*

The next lemma yields a connection between $\mathrm{HCTL}^*_{\mathsf{ss}}$ path formulas and LTL formulas. We use LTL as an index for the satisfaction relation to indicate that the (pure) path formula is interpreted as an LTL formula, i.e. $\pi \models_{\mathrm{LTL}} \psi$ means that the path π satisfies ψ interpreted as an LTL formula.

▶ **Lemma 13.** *Let $K = \langle S, \rightarrow, L\rangle$ be a Kripke structure, π a path in K and $\sigma : \mathit{Var} \rightarrow S$ a variable assignment. For every pure $\mathrm{HCTL}^*_{\mathsf{ss}}$ path formula ψ over atomic propositions Prop and variables $\{x_1, \ldots, x_k\}$ it holds that $K, \pi, \sigma \models \psi$ if and only if $K_\sigma, \mathsf{pr}(\pi) \models_{\mathrm{LTL}} \psi$.*

Proof. We will prove this by induction on ψ. Suppose first that $\psi = \varphi$ for some state-formula φ and suppose $K, \pi, \sigma \models \psi$. Since ψ is pure, there are no path quantifiers $\mathsf{E}, \mathsf{A}, \downarrow x.\varphi'$ nor $@_x \varphi'$ in ψ. Thus, φ is a boolean combination of propositions and variables and because $\mathsf{pr}(\pi)$ is consistent with σ and all states in K_σ agree with their respective states in K on atomic propositions we also get that $K_\sigma, \mathsf{pr}(\pi) \models_{\mathrm{LTL}} \psi$.

So, suppose $\psi = \mathsf{X}\psi'$. Then $K, \pi, 1, \sigma \models \psi$. With the induction hypothesis we get that $\mathsf{pr}(\pi), 1 \models_{\mathrm{LTL}} \psi'$ and also that $\mathsf{pr}(\pi) \models_{\mathrm{LTL}} \psi$.

For the last case, suppose that $\psi = \psi_1 \mathsf{U} \psi_2$ and $K, \pi, \sigma \models \psi$. Then there is some j such that $K, \pi, j, \sigma \models \psi_2$ and for all $i \leq j$ it holds that $K, \pi, i, \sigma \models \psi_1$. By the induction hypothesis we get that $\mathsf{pr}(\pi), j \models_{\mathrm{LTL}} \psi_2$ and $\mathsf{pr}(\pi), i \models_{\mathrm{LTL}} \psi_1$ for all $i \leq j$. Thus, we also get $\mathsf{pr}(\pi) \models_{\mathrm{LTL}} \psi$. This finishes the proof. ◀

The following theorems extend well-known results for LTL, Büchi automata and the μ-calculus to deal with the hybrid world.

▶ **Theorem 14.** *For each pure* $\mathrm{HCTL}^*_{\mathsf{ss}}$ *path formula* ψ *of size* n *over atomic propositions* $Prop$ *and variables* $\{x_1, \ldots, x_k\}$ *there is a Büchi automaton* $\mathcal{A}_\psi$ *of size* $\mathcal{O}(n \cdot 2^n)$ *such that*

1. *a path* $\pi \in (S \times 2^{Prop \cup Var})^\omega$ *is accepted by* $\mathcal{A}_\psi$ *if and only if* $\pi \models_{\mathrm{LTL}} \psi$ *and*
2. *for all paths* $\pi' \in (S \times 2^{Prop})^\omega$, $\mathsf{pr}(\pi')$ *is accepted by* $\mathcal{A}_\psi$*, if and only if* $\pi', \sigma \models \psi$.

Proof. To construct $\mathcal{A}_\psi$ we first observe that ψ is a pure LTL formula with possibly added tests for variables. We treat variable test for the moment like usual atomic propositions. Furthermore, we know that for each LTL formula ψ there is a Büchi automaton $\mathcal{A}_\psi$ that accepts a path π iff this path satisfies ψ [16]. This immediately yields the first part of the theorem as well as the size estimation on the automaton. The second part follows from the first one in combination with Lemma 13. ◀

Observe that the constructed Büchi automaton only checks the sequence of propositions and thus, $\mathcal{A}_\psi$ accepts more paths than intended, for example paths in which the "proposition" x can occur on more than one state and thus cannot truly be an encoding of a hybrid variable.

To utilise these Büchi automata in the hybrid μ-calculus we also need to bridge the gap between L_μ and H_μ in some sense. Similar to the LTL case we write $K, s \models_{\mathrm{L}_\mu} \varphi$ to indicate that φ is interpreted as a purely modal μ-calculus formula.

▶ **Lemma 15.** *For each* H_μ*-formula* φ *without any occurrence of* $\downarrow x.\psi$ *or* $@_x\, \psi$ *in it, it holds that* $K, s, \sigma \models \varphi$ *if and only if* $K_\sigma, s \models_{L_\mu} \varphi$.

Proof sketch. To prove this by induction on φ we need to strengthen the hypothesis in order to deal with free second-order variables. Let $\varphi(X_1, \ldots, X_m)$ be a formula with free second-order variables $X_1, \ldots, X_m$ and $\rho : \{X_1, \ldots, X_m\} \to 2^{S \times (Var \to S)}$ be an interpretation for them. We define $\rho' : \{X_1, \ldots, X_m\} \to 2^S$ to be $\rho'(X) := \{s \mid (s, \sigma) \in \rho(x)\}$. We now show by induction on φ that $K, s, \sigma, \rho \models \varphi(X_1, \ldots, X_m)$ if and only if $K_\sigma, s, \rho' \models_{\mathrm{L}_\mu} \varphi(X_1, \ldots, X_m)$.

Suppose $\varphi = p$. Then the statement holds because K and K_σ agree everywhere on atomic propositions. The case $\varphi = x$ is by construction of K_σ because x as an atomic proposition in K_σ holds exactly at $\sigma(x)$. The case $\varphi = X$ follows by the construction of ρ'. The boolean cases as well as box and diamond operators follow by simple semantic arguments.

For the last case, suppose that $\varphi = \mu X.\psi(X)$. To show this case we can use the characterisation of a least fixpoint as the union of its approximations. We can then show that the statement holds for each approximation by a separate induction (and thus for the union of all of them). ◀

▶ **Theorem 16.** *For each Büchi automaton* $\mathcal{A}$ *over* $Prop \cup \{x_1, \ldots, x_k\}$ *of size* m *there is an* H_μ *formula* $\varphi_\mathcal{A}$ *of size at most* $O(m \cdot 2^m)$ *such that* $K, s, \sigma \models \varphi_\mathcal{A}$ *if and only if there is a path* π *in* K *starting at* s *such that* $\mathcal{A}$ *accepts* $\mathsf{pr}(\pi)$.

Proof. It is known that for each Büchi automaton $\mathcal{A}$ there is an L_μ formula $\varphi_\mathcal{A}$ such that $K_\sigma, s \models_{\mathrm{L}_\mu} \varphi_\mathcal{A}$ if and only if there exists a path π' starting at s such that $\mathcal{A}$ accepts π', c.f. [4, Chp. 10]. Since π' is a path in K_σ, there is a path π in K such that $\mathsf{pr}(\pi) = \pi'$. By Lemma 15 and the fact that $\varphi_\mathcal{A}$ does not have any occurrences of $\downarrow x$ or $@_x$ (in fact, $\varphi_\mathcal{A}$ is a pure modal μ-calculus formula extended by variable tests) we get that $K_\sigma, s \models_{\mathrm{L}_\mu} \varphi_\mathcal{A}$ if and only if $K, s, \sigma \models \varphi_\mathcal{A}$. ◀

We will use these theorems to show the following:

▶ **Theorem 17.** *For each formula* $\varphi \in \mathrm{HCTL}^*_{\mathsf{ss}}$ *there is a formula* $\varphi' \in H_\mu$ *such that* $K, s, \sigma \models \varphi$ *if and only if* $K, s, \sigma \models \varphi'$ *for all Kripke structures* K*, states* s *in* K *and variable assignments* σ.

Proof. We will first give a translation for $\mathrm{HCTL}^*_{\mathsf{ss}}$ formulas and then argue correctness of the translation. The cases up to path formulas are straightforward:

$$\tau(p) := p \qquad \tau(\varphi \vee \chi) := \tau(\varphi) \vee \tau(\chi)$$
$$\tau(x) := x \qquad \tau(\downarrow x.\varphi) := \downarrow x.\tau(\varphi)$$
$$\tau(\neg\varphi) := \neg\tau(\varphi) \qquad \tau(@_x\, \varphi) := @_x\, \tau(\varphi)$$

For the case of $\varphi = \mathsf{E}\psi$, let $\{\varphi_1, \ldots, \varphi_m\}$ be the maximal state-subformulas in ψ, i.e. subformulas that start with $\mathsf{E}, \mathsf{A}, \downarrow x., @_x$. We first replace those by fresh atomic propositions p_{φ_i}. The resulting formula is a pure $\mathrm{HCTL}^*_{\mathsf{ss}}$ path formula over the propositions $\mathit{Prop} \cup \{p_{\varphi_1}, \ldots, p_{\varphi_m}\}$ and variables $\{x_1, \ldots, x_k\}$. According to Theorem 14 we construct a Büchi-automaton $\mathcal{A}_\psi$. Furthermore, according to Theorem 16 there is a H_μ formula $\varphi_{\mathcal{A}_\psi}$ that simulates this Büchi-automaton and thus the formula ψ.

Finally, we translate the remaining maximal state subformulas $\{\varphi_1, \ldots, \varphi_m\}$ recursively. Let $\tau(\varphi_1), \ldots, \tau(\varphi_m)$ be their respective translations. We obtain the final translated formula by replacing the atomic propositions p_{φ_i} in $\varphi_{\mathcal{A}_\psi}$ by their respective translations. Thus:

$$\tau(\mathsf{E}\psi) := \varphi_{\mathcal{A}_\psi}\, [\tau(\varphi_1)/p_{\varphi_1}, \ldots, \tau(\varphi_m)/p_{\varphi_m}]\,.$$

It remains to be shown that this translation is correct. For this, let $K = \langle S, \to, L\rangle$ be a Kripke structure, $s \in S$ and $\sigma : \mathit{Var} \to S$ a variable assignment. We prove that $K, s, \sigma \models \varphi$ if and only if $K, s, \sigma \models \tau(\varphi)$ by induction on φ.

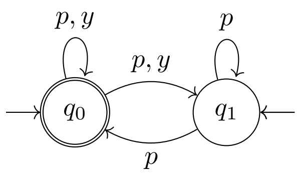
The only interesting case is $\varphi = \mathsf{E}\psi$. Suppose first, that ψ is pure and $K, s, \sigma \models \mathsf{E}\psi$. Then there is a path on K starting at s such that $K, \pi, \sigma \models \psi$. By the second part of Theorem 14 we get that $K, \pi, \sigma \models \psi$ if and only if $\mathsf{pr}(\pi)$ is accepted by $\mathcal{A}_\psi$ and by Theorem 16 we then get that $\mathsf{pr}(\pi)$ is accepted by $\mathcal{A}_\psi$ if and only if $K, s, \sigma \models \varphi_{\mathcal{A}_\psi}$.

For the case that ψ is not pure, we additionally need the fact that $K, s, \sigma \models \varphi\,[\chi/p] \Leftrightarrow K', s, \sigma \models \varphi$ where K' extends K with an atomic proposition p such that $p \in L(s) \Leftrightarrow K, s, \sigma \models \chi$. This can be shown by a straightforward induction on φ, both for $\mathrm{HCTL}^*_{\mathsf{ss}}$ and H_μ. ◀

To illustrate the translation we will give a short example:

▶ **Example 18.** Consider the formula $\chi := \downarrow y.\mathsf{EG}\,(\mathsf{F}y \wedge \downarrow x.\mathsf{EXF}x)$. The formula states that there is a path whose starting point is seen infinitely often along the path and at every point of the path there is another path that loops back to the current point.

We first begin by extracting the maximal state-subformula $\downarrow x.\mathsf{EXF}x$ leaving us with the formula $\downarrow y.\mathsf{EG}\,(\mathsf{F}y \wedge p)$. We first construct a Büchi automaton $\mathcal{A}$ for the LTL-formula $\mathsf{G}\,(\mathsf{F}y \wedge p)$ (ignoring that y is a variable test):



This Büchi-automaton can be translated into the following μ-calculus formula which is satisfied by a state s iff there is a path emerging from s that is accepted by $\mathcal{A}$.

$$\begin{aligned}\varphi := {} & [\nu Y.(p \wedge y \wedge \Diamond Y) \vee (p \wedge y \wedge \Diamond\mu Z.(p \wedge \Diamond Z) \vee (p \wedge \Diamond Y))] \vee \\ & \mu Z.(p \wedge \Diamond Z) \vee (p \wedge \Diamond\nu Y.(p \wedge y \wedge \Diamond Y) \vee (p \wedge y \wedge \Diamond\mu Z.(p \wedge \Diamond Z) \vee (p \wedge \Diamond Y))\end{aligned}$$

Here the first line describes an accepting run starting at q_0 and the second line a run starting at q_1. The automaton accepts a path π if p holds everywhere and y is seen infinitely often along π. For the sake of presentation we will use that

$$\varphi \equiv \nu Y.\mu Z.p \wedge ((y \wedge \Diamond Y) \vee \Diamond Z)$$

and continue to use this shorter formula. Doing the same recursively we get that $\tau(\mathsf{EXF}x) = \Diamond\mu X.x \vee \Diamond X$. Putting them together we get $\tau(\chi) := {\downarrow} y.\nu Y.\mu Z.({\downarrow} x.\Diamond\mu X.x \vee \Diamond X) \wedge ((y \wedge \Diamond Y) \vee \Diamond Z)$.

Knowing that $\mathrm{HCTL}^*_{\mathsf{ss}}$ is a fragment of H_μ also helps us to understand the connection between $\mathrm{HCTL}^*_{\mathsf{ss}}$ and $\mathrm{HCTL}^*_{\mathsf{ps}}$ given that the latter cannot be embedded into H_μ [13].

▶ **Corollary 19.** $\mathrm{HCTL}^*_{\mathsf{ps}}$ *is strictly more expressive than* $\mathrm{HCTL}^*_{\mathsf{ss}}$.

In particular, no $\mathrm{HCTL}^*_{\mathsf{ss}}$ formula is equivalent to the $\mathrm{HCTL}^*_{\mathsf{ps}}$ formula $\mathsf{EG}{\downarrow} x.\mathsf{XG}\neg x$.

Let $\mathrm{H}^k\mathrm{CTL}^*_{\mathsf{ss}}$ and H^k_μ be the fragments of $\mathrm{HCTL}^*_{\mathsf{ss}}$ and H_μ that use at most k first-order variables. We know from [13] that H^k_μ is hybrid k-bisimulation-invariant. Thus, since the translation from $\mathrm{HCTL}^*_{\mathsf{ss}}$ to H_μ basically leaves variables untouched we also get:

▶ **Corollary 20.** $\mathrm{H}^k\mathrm{CTL}^*_{\mathsf{ss}}$ *is hybrid k-bisimulation-invariant.*

5.3 $\mathrm{HCTL}^*_{\mathsf{ps}}/\mathrm{HCTL}^*_{\mathsf{pp}}$ and H_μ are incomparable

Finally, we are also interested in the connection between $\mathrm{HCTL}^*_{\mathsf{ps}}$ resp. $\mathrm{HCTL}^*_{\mathsf{pp}}$ and H_μ. We already know that one cannot be embedded into the other; in the remainder of this section we will show that this is also true of the other way: there are formulas in H_μ which cannot be expressed in $\mathrm{HCTL}^*_{\mathsf{pp}}$ and, hence, not on $\mathrm{HCTL}^*_{\mathsf{ps}}$ either. Hence, we will show that $\mathrm{HCTL}^*_{\mathsf{pp}}$ (resp. $\mathrm{HCTL}^*_{\mathsf{ps}}$) and H_μ are incomparable.

▶ **Theorem 21.** *Let K be a linear structure and $\varphi \in \mathrm{HCTL}^*_{\mathsf{pp}}$. Let φ' be the formula that is constructed by simply removing all path quantifiers in φ. Then $K, s, \sigma \models \varphi$ if and only if $K, s, \sigma \models \mathsf{E}\varphi'$.*

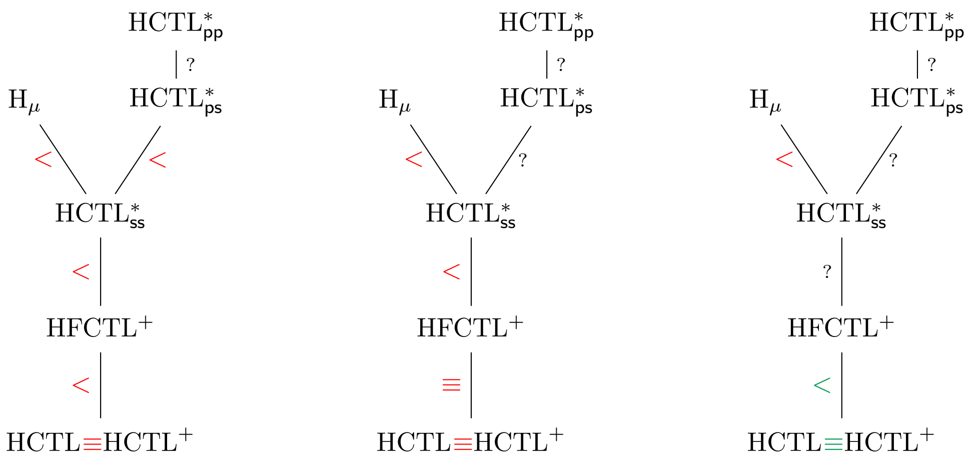
Proof. To prove this, we see that on linear structures there is no difference between E and A path quantifiers since from every point in the structure there is exactly one path. Consequently we can just drop the path quantifiers altogether. ◀

Thus, Theorem 21 essentially states that on linear structures $\mathrm{HCTL}^*_{\mathsf{pp}}$ (and $\mathrm{HCTL}^*_{\mathsf{ps}}$) is as expressive as hybrid LTL.

▶ **Theorem 22.** *There is no* $\mathrm{HCTL}^*_{\mathsf{pp}}$ *formula that can express the H_μ property $\mu X.\, p \vee \Diamond\Diamond X$.*

Proof. Suppose for the sake of contradiction that such a formula φ exists. Then by Theorem 21 we get that there is a hybrid LTL formula that characterises reachability in an even number of steps on word structures. However, hybrid LTL can be translated into first-order logic on word structures which, in turn, cannot express this property, c.f. [5]. Thus, such a formula cannot exist. ◀

▶ **Corollary 23.** $\mathrm{HCTL}^*_{\mathsf{ps}}/\mathrm{HCTL}^*_{\mathsf{pp}}$ *and H_μ are incomparable.*



$\text{HCTL}^*_{\mathsf{pp}}$
?
H_μ
$\text{HCTL}^*_{\mathsf{ps}}$
$<$
$<$
?
$\text{HCTL}^*_{\mathsf{ss}}$
$<$
?
HFCTL^+
$<$
$\equiv$
$\text{HCTL}\equiv\text{HCTL}^+$

(a) general structures. **(b)** finite structures. **(c)** tree structures.

Figure 2 The branching-time hierarchy on different classes of Kripke structures. Results colored in red are newly obtained in this paper while results colored in green have been previously shown. Still open questions are marked with a "?".

6 Conclusion & Further Work

To conclude, we have studied the expressive power of hybrid branching-time logics. The results are summarised in Figure 2.

For general structures the results, depicted in Figure 2a, are similar to their non-hybrid counterparts – at least for the part below $\text{HCTL}^*_{\mathsf{ss}}$. We have proven that HCTL^+ and HCTL have the same expressive power. HFCTL^+ then can express more properties than HCTL^+ and $\text{HCTL}^*_{\mathsf{ss}}$ can express even more. However, above $\text{HCTL}^*_{\mathsf{ss}}$ the picture is a bit different: only $\text{HCTL}^*_{\mathsf{ss}}$ can be translated into H_μ. Allowing dynamic naming of states as part of a path formula in CTL^* then drastically increases the expressive power in a way that is not even captured by H_μ. We do not yet know if allowing jumps as part of path formulas increases the expressive power of the resulting logic even further.

It is also interesting to see that these expressiveness results change depending on the classes of structures in reference. For general Kripke structures we have obtained an (almost) complete picture. However, if we restrict the attention to finite structures or tree structures we only have an incomplete and possibly quite different picture. For example, on finite structures we have shown that HFCTL^+ is equi-expressive to HCTL which helped us prove the expressiveness gap between HFCTL^+ and $\text{HCTL}^*_{\mathsf{ss}}$. However, it is still unclear if this expressiveness gap also holds for trees. We have also summarised what we know about finite structures and trees in Figures 2b and 2c.

Future work will focus on studying the expressiveness over interesting classes of structures such as finite structures or trees and will aim to close the gaps in their respective hierarchies. We will also continue to study the expressiveness games defined in Section 3. Here, we have only proven one direction: if Duplicator wins, then there is no formula that can distinguish the structures in question. A proof for the reverse direction would in itself strengthen this framework but has turned out to be rather challenging.

References

1 C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic J. of the IGPL*, 8(5):653–679, 2000.

2 C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*, pages 821–868. Elsevier, 2006.

3 P. Blackburn. Representation, reasoning, and relational structures: A hybrid logic manifesto. *Logic J. of the IGPL*, 8(3):339–365, 2000.

4 S. Demri, V. Goranko, and M. Lange. *Temporal Logics in Computer Science*, volume I – Finite State Systems of *Cambridge Tracts in Theor. Comp. Sc.* Cambridge Univ. Press, 2016.

5 H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Perspectives in Math. Logic. Springer, Berlin, 1995.

6 A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

7 E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.

8 E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. of Comp. and Sys. Sc.*, 30:1–24, 1985.

9 E. A. Emerson and J. Y. Halpern. “Sometimes” and “not never” revisited: On branching versus linear time temporal logic. *J. of the ACM*, 33(1):151–178, 1986.

10 E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. on Computing*, 29(1):132–158, 2000.

11 A. Kara, V. Weber, M. Lange, and T. Schwentick. On the hybrid extension of CTL and CTL$^+$. In *Proc. 34th Int. Symp. on Mathematical Foundations of Computer Science, MFCS’09*, volume 5734 of *LNCS*, pages 427–438. Springer, 2009.

12 D. Kernberger and M. Lange. Model checking for the full hybrid computation tree logic. In *Proc. 23rd Int. Symp. on Temporal Representation and Reasoning, TIME’16*, pages 31–40. IEEE Computer Society, 2016.

13 D. Kernberger and M. Lange. The fully hybrid mu-calculus. In *Proc. 24th Int. Symp. on Temporal Representation and Reasoning, TIME’17*, volume 90 of *LIPIcs*, pages 17:1–17:16. Dagstuhl-Leibniz-Zentrum, 2017. URL: `http://www.dagstuhl.de/dagpub/978-3-95977-052-1`.

14 A. Pnueli. The temporal logic of programs. In *Proc. 18th Symp. on Foundations of Computer Science, FOCS’77*, pages 46–57, Providence, RI, USA, 1977. IEEE.

15 U. Sattler and M. Y. Vardi. The hybrid μ-calculus. In *Proc. 1st Int. Joint Conf. on Automated Reasoning, IJCAR’01*, volume 2083 of *LNCS*, pages 76–91. Springer, 2001.

16 M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *Proc. 1st Symp. on Logic in Computer Science, LICS’86*, pages 332–344. IEEE, Washington, DC, 1986.

A Appendix

We need a technical definition for the proof of Theorem 2.

▶ **Definition 24.** For any $\varphi \in \mathrm{HCTL}$ we define the *temporal nesting depth* of φ ($\mathsf{nd}(\varphi)$) as follows:

$$\begin{aligned}
\mathsf{nd}(p) &:= 0, & \mathsf{nd}(x) &:= 0,\\
\mathsf{nd}(\neg\varphi) &:= \mathsf{nd}(\varphi), & \mathsf{nd}(\varphi_1 \vee \varphi_2) &:= \max\{\mathsf{nd}(\varphi_1), \mathsf{nd}(\varphi_2)\},\\
\mathsf{nd}(\downarrow x.\varphi) &:= \mathsf{nd}(\varphi), & \mathsf{nd}(@_x\, \varphi) &:= \mathsf{nd}(\varphi),\\
\mathsf{nd}(\mathsf{EX}\varphi) &:= \mathsf{nd}(\varphi) + 1, & \mathsf{nd}(\mathsf{E}\varphi_1\mathsf{U}\varphi_2) &:= \max\{\mathsf{nd}(\varphi_1), \mathsf{nd}(\varphi_2)\} + 1.
\end{aligned}$$

▶ **Theorem 2** (restated). *If Duplicator wins $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, K_1, t_0)$ then it holds for all formulas $\varphi \in \mathrm{HCTL}$ with $\mathsf{nd}(\varphi) \le m$ that $K_0, s_0 \models \varphi$ if and only if $K_1, t_0 \models \varphi$.*

Proof. We prove an extended statement for formulas with unbounded occurrences of variables. For this, let $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, \dots, s_{n-1}, K_1, t_0, \dots, t_{n-1})$ be the game with n pebbles already placed. We will show the following statement:

We show that if Duplicator wins $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, \dots, s_{n-1}, K_1, t_0, \dots, t_{n-1})$ then it holds for all formulas $\varphi \in \mathrm{HCTL}$ with $\mathsf{nd}(\varphi) \le m$ and all variable assignments $\sigma : \mathit{Var} \to S_0$, $\sigma' : \mathit{Var} \to S_1$ such that $\sigma(x) = s_i$ if and only if $\sigma'(x) = t_i$ that $K_0, s_i, \sigma \models \varphi$ if and only if $K_1, t_i, \sigma' \models \varphi$.

We show this by an induction on the number of rounds m and only show the cases where Duplicator moves on K_0. The other cases are completely analoguous.

Let $m = 0$ and suppose that Duplicator wins $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, \dots, s_{n-1}, K_1, t_0, \dots, t_{n-1})$. Then it holds for all atomic formulas $\varphi = p$ or $\varphi = x_j$ for some $p \in \mathit{Prop}$ and $0 \le j \le n-1$ that $K_0, s_i, \sigma \models \varphi$ if and only if $K_1, t_i, \sigma' \models \varphi$. This follows directly from the winning conditions of Duplicator. The cases for boolean connectives and hybrid operators can be shown by a straightforward induction.

So, let $m \ge 1$ and suppose that the statement already holds for $m-1$. Suppose again, that Duplicator wins $\mathcal{G}^m_{\mathrm{HCTL}}(K_0, s_0, \dots, s_{n-1}, K_1, t_0, \dots, t_{n-1})$. We show that $K_0, s_i, \sigma \models \varphi$ if and only if $K_1, t_i, \sigma' \models \varphi$ with $\varphi, \sigma, \sigma', i$ as above by an induction over the structure of φ. Atomic formulas and boolean connectives can be shown in the same way as for $m = 0$.

- $\varphi = \mathsf{EX}\psi$. Suppose $K_0, s_i, \sigma \models \varphi$, then there is some successor s'_i of s_i such that $K_0, s'_i, \sigma \models \psi$. Suppose Spoiler chooses s'_i and moves there. Since Duplicator wins, she can choose a successor t'_i of t_i such that she wins $\mathcal{G}^{m-1}_{\mathrm{HCTL}}(K_0, s_0, \dots, s_{n-1}, s'_i, K_1, t_0, \dots, t_{n-1}, t'_i)$. Since $\mathsf{nd}(\psi) < m$ we can use that the statement already holds for $m-1$ and get that $K_1, t'_i, \sigma' \models \psi$. This means also that $K_1, t_i, \sigma' \models \varphi$.
- $\varphi = \mathsf{E}\psi_1\mathsf{U}\psi_2$. Suppose $K_0, s_i, \sigma \models \varphi$. Then there is a path π starting at s_i and some l such that $K_0, \pi^l, \sigma_0 \models \psi_2$ and for all $j < l$, $K_0, \pi^j, \sigma \models \psi_1$. Suppose Spoiler chooses to play π on K_0 with l. Since Duplicator wins the game, she can answer with some path τ on K_1 starting at t_i and some l'. Spoiler now has two possibilities:
  First, he can choose to go to π^l. Duplicator then has to play $\tau^{l'}$ and since she plays a winning strategy wins the game $\mathcal{G}^{m-1}_{\mathrm{HCTL}}(K_0, s_0, \dots, s_{n-1}, \pi^l, K_1, t_0, \dots, t_{n-1}, \tau^{l'})$. Since $K_0, \pi^l, \sigma_0 \models \psi_2$ and $\mathsf{nd}(\psi_2) < m$ we can deduce with the hypothesis for $m-1$, that $K_1, \tau^{l'}, \sigma' \models \psi_2$.
  Secondly, Spoiler can choose any $k' < l'$ and move to $\tau^{k'}$. Since Duplicator is winning, she can answer with some $k < l$, move to π^k and win from there. We know that $K_0, \pi^k, \sigma \models \psi_1$ and $\mathsf{nd}(\psi_1) < m$ so by the induction hypothesis we also get that $K_1, \tau^{k'}, \sigma' \models \psi_1$. This holds for any $k' < l'$ since it was Spoiler's choice.

Thus, we have that there is some l' on τ such that $K_1, \tau^{l'}, \sigma' \models \psi_2$ and for all $k' < l'$ it holds that $K_1, \tau^{k'}, \sigma' \models \psi_1$. Thus, since τ starts at t_i we get that $K_1, t_i, \sigma' \models \varphi$.

- $\varphi = {\downarrow} x.\psi$ for some $x \notin \{x_0, \ldots, x_{n-1}\}$ (otherwise $x_0, \ldots, x_{n-1}$ would not be free). Suppose $K_0, s_i, \sigma \models \varphi$. Then $K_0, s_i, \sigma[x \mapsto s_i] \models \psi$. Since ψ is smaller then φ and $\sigma[x \mapsto s_i]$, resp. $\sigma'[x \mapsto t_i]$ also satisfy the assumption for variable interpretations we can use the induction hypothesis for ψ and get that $K_1, t_i, \sigma'[x \mapsto t_i] \models \psi$ and thus also $K_1, t_i, \sigma' \models \varphi$.
- $\varphi = @_{x_j} \psi$ for some $0 \leq j \leq n-1$. Suppose $K_0, s_i, \sigma \models \varphi$. Then with the definition of σ we get that $K_0, s_j, \sigma \models \psi$. With the induction hypothesis for ψ we get that $K_0, t_j, \sigma' \models \psi$ and thus also $K_0, s_i, \sigma \models \varphi$.

This finishes the proof. ◀

▶ **Theorem 25.** *For each* HCTL$^+$ *formula φ there is an* HCTL *formula φ' such that for all Kripke structures $K = \langle S, \rightarrow, L\rangle$, states $s \in S$ and variable assignments $\sigma : Var \rightarrow S$ it holds that $K, s, \sigma \models \varphi$ if and only if $K, s, \sigma \models \varphi'$.*

Proof. We describe how to transform an HCTL$^+$ formula into an HCTL formula. This can be done by rewriting each path formula into an equivalent HCTL formula.

Using the equivalences $\mathsf{A}\psi \equiv \neg\mathsf{E}\neg\psi$, $\neg\mathsf{X}\psi \equiv \mathsf{X}\neg\psi$, $\neg(\psi_1 \mathsf{U} \psi_2) \equiv \mathsf{G}\neg\psi_2 \vee (\neg\psi_2 \mathsf{U}(\neg\psi_1 \wedge \neg\psi_2))$, $\mathsf{E}\psi_1 \vee \psi_2 \equiv \mathsf{E}\psi_1 \vee \mathsf{E}\psi_2$ we can push negations inward and assume conjunctions as the top-level operator in path formulas. We then use $\mathsf{X}\varphi_1 \wedge \mathsf{X}\varphi_2 \equiv \mathsf{X}\varphi_1 \wedge \varphi_2$ and $\mathsf{G}\varphi_1 \wedge \mathsf{G}\varphi_2 \equiv \mathsf{G}\varphi_1 \wedge \varphi_2$ to move Next- and Generally operators upwards and $\mathsf{E}(\varphi \wedge \psi) \equiv \varphi \wedge \mathsf{E}\psi$ for some state formula φ to remove state-formulas directly under a path quantifier. Thus, we can assume that each path formula has the form

$$\mathsf{E}(\mathsf{X}\Lambda_1 \wedge \bigwedge_{i \in I} \varphi_i \mathsf{U} \chi_i \wedge \mathsf{G}\Lambda_2)$$

with suitable state formulas $\Lambda_i, \varphi_i, \chi_i$.

We then obtain an HCTL formula by guessing the order in which the Until-formulas are satisfied along such a path. This is done by the following formula:

$$\begin{aligned}
\Lambda_2 \wedge \bigvee_{J \subseteq I} (\bigwedge_{j \notin J} \chi_j) \wedge (\bigwedge_{j \notin J} \varphi_j) \wedge \mathsf{EX}\Big(\Lambda_1 \wedge \\
\bigvee_{\pi \in \mathsf{Perm}(J)} \mathsf{E}((\Lambda_2 \wedge \bigwedge_{j \in J} \varphi_{\pi(j)}) \mathsf{U} \Big(\chi_{\pi(1)} \wedge \\
\mathsf{E}((\Lambda_2 \wedge \bigwedge_{j \in J, j \neq 1} \varphi_{\pi(j)}) \mathsf{U} \Big(\chi_{\pi(2)} \wedge \\
\vdots \\
\mathsf{E}((\Lambda_2 \wedge \varphi_{\pi(|J|)}) \mathsf{U} (\chi_{\pi(|J|)} \wedge \mathsf{EG}\Lambda_2))\Big) \ldots \Big)
\end{aligned}$$

where $\mathsf{Perm}(J)$ denotes the set of all permutations over J. ◀