The Fully Hybrid μ-Calculus

Daniel Kernberger[1] and Martin Lange[2]

1 **School of Electrical Engineering and Computer Science, University of Kassel, Kassel, Germany**
2 **School of Electrical Engineering and Computer Science, University of Kassel, Kassel, Germany**

Abstract

We consider the hybridisation of the μ-calculus through the addition of nominals, binder and jump. Especially the use of the binder differentiates our approach from earlier hybridisations of the μ-calculus and also results in a more involved formal semantics. We then investigate the model checking problem and obtain EXPTIME-completeness for the full logic and the same complexity as the modal μ-calculus for a fixed number of variables. We also show that this logic is invariant under hybrid bisimulation and use this result to show that – contrary to the non-hybrid case – the hybrid extension of the full branching time logic CTL* is not a fragment of the fully hybrid μ-calculus.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases μ-calculus, hybrid logics, model checking, bisimulation invariance

Digital Object Identifier 10.4230/LIPIcs.TIME.2017.17

1 Introduction

Hybrid Extensions of Modal Logic

Hybrid logic [19, 4] has emerged from modal logic as an attempt to extend a well-behaved but relatively weak (in terms of expressive power) fragment of first-order logic with additional features whilst retaining good properties like decidability etc. This is achieved by extending the syntax of modal logic with first-order variables and some very restricted form of first-order quantification over these variables.

The availability of first-order variables in the language gives the logic the power to express properties that are inherently non-modal. For instance, it is possible to express that a state of a Kripke structure has an edge to itself; it is – even without the definition of a formal semantics – not hard to guess that the formula $x \wedge \Diamond x$ should be true at exactly the states of that kind. The two other typical operators are the *binder* and the *jump*. Intuitively, $\downarrow x.\varphi$ binds x to the current state for the evaluation of φ. It is sometimes also known as the *freeze* modality [1]. The jump operator – also called the *satisfaction* operator [4] – is written $@_x\,\varphi$ and, intuitively, continues the evaluation of φ at the state that is bound to x. For both operators it is important to remember that modal formulas, as opposed to (e.g. closed) first-order formulas are interpreted at states of a Kripke structure, not the structure as a whole.

Sometimes, when defining hybrid logics, one distinguishes two kinds of variables, depending on whether they can be bound or not, and calls those that do not get bound *nominals*.

The additional power induced by these hybrid features and operators comes at a cost when compared to modal logic. Clearly, hybrid logics do not possess the tree model property anymore, as the little example $x \wedge \Diamond x$ above shows. Strongly related to that is the loss of

24th International Symposium on Temporal Representation and Reasoning (TIME 2017).
Editors: Sven Schewe, Thomas Schneider, and Jef Wijsen; Article No. 17; pp. 17:1–17:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

bisimulation-invariance, an inherent feature of modal logic [5]. The notion of bisimulation has been refined accordingly to *hybrid bisimulation* which relates two states when they can mimick each others transitions locally in the presence of a fixed number of named states. It has been shown that the hybrd extension of modal logic with binder and jump is invariant under this equivalence relation [3].

The term *hybrid logic* suggests speaking about one particular logic but in fact denotes a family of logics; its members are obtained by extending *some* modal logic with *some* of the features mentioned above. This is of course not restricted to modal logics alone, any logic that is not subsumed by first-order logic is a natural candidate for the basis of a hybrid logic. For instance, temporal and dynamic logics have been extended in this way, namely the EF-fragment of CTL with past operators [9], CTL and CTL$^+$ [11], as well as CTL* [12]. In [11] the hybrid extensions of CTL and CTL$^+$ are only interpreted over computation trees to retain decidability of the satisfiability problem. However, the semantics naturally extends to Kripke structures. Hybrid CTL and CTL$^+$ are then – as in the non-hybrid case – subsumed by the hybrid extensions of CTL* in [12].

Hybrid Extensions of the μ-Calculus

In this paper we consider the extension of the well-known modal μ-calculus L_μ [13] with hybrid operators. This is not the first attempt at doing so; Sattler and Vardi [16] have considered a hybrid μ-calculus which extends L_μ with nominals only, i.e. with additional first-order variables but no mechanism to change them during the course of the evaluation of a formula. This is in some sense a smallest hybrid extension of L_μ, even though they use the term *hybrid full μ-calculus* for this logic. “Full” in that context seems to refer to the addition of converse modalities. They are, however, thrown away then in favour of a universal modality with which one can jump to any state in an underlying Kripke structure. It is easy to see that this subsumes the specialised jumps $@_x$ conventionally used in hybrid logics.

Here we consider a different logic, namely the extension of the modal μ-calculus with *all* hybrid features, in particular including binders. To subtly distinguish these logics, we refer to the one used here as the *fully hybrid μ-calculus*. This then clearly subsumes full hybrid modal logic for which satisfiability is undecidable [4]. The context of Sattler and Vardi's hybrid μ-calculus of course forbids this, as their primary interest is description logics for which decidability of satisfiability is a must. The motivation for the extension of L_μ with all hybrid features here is not driven by concrete applications; we study this logic in order to understand the effect that extending temporal logics in various ways has on their logical and computational properties.

Contribution and Organisation

The paper is organised as follows. Section 2 defines the fully hybrid μ-calculus formally. We discuss that the semantics of hybrid temporal logics is inadequate for a logic with fixpoint quantifiers as under this semantics the implicit recursion mechanism does not obey the meaning one would intuitively expect the binder to have. This is fixed by letting second-order variables stand for sets of pairs of states and first-order variable bindings, rather than sets of states only.

Section 3 examines the complexity of model checking the fully hybrid μ-calculus. Using a reduction to L_μ model checking we obtain (1) an EXPTIME upper bound in general, (2) that for formulas with a fixed number of variables the complexity is only polynomially worse than that for L_μ model checking, and (3) a game-theoretic characterisation similar to the one for L_μ [17] which can be used to understand the properties expressed by formulas of the fully hybrid μ-calculus. We also show that the EXPTIME upper bound is tight by giving a matching lower bound.

In Section 4 we investigate questions of the logic's expressiveness. We prove that, not surprisingly, invariance under hybrid bisimulations carries over from hybrid modal logic to the fully hybrid μ-calculus. To ease argumentation in this context, we also develop a game-theoretic characterisation of hybrid bisimilarity similar to the well-known bisimulation games [18]. We then use these games to show indistinguishability between two different Kripke structures and deduce that, perhaps surprisingly, the fully hybrid μ-calculus does not subsume the hybrid extensions of CTL*, namely that it cannot express the property "there is a path on which no state occurs twice", even though this is easily possible in hybrid CTL*.

We conclude the paper with a discussion on further work in this area.

2 Preliminaries

2.1 Syntax

Let $k \in \mathbb{N}$, $\mathcal{V} = \{x_1, x_2, \ldots, x_k\}$ be a finite set of first-order variables, $\mathit{Prop} = \{p, q, \ldots\}$ be a countable set of atomic propositions, $\mathcal{V}_2 = \{X, Y, \ldots\}$ be a countable set of second-order variables and $\mathit{Nom} = \{m, n, \ldots\}$ be a countable set of first-order constants referred to as nominals. All sets are assumed to be pairwise disjoint. Formulas of the k-variable fragment of the fully hybrid μ-calculus H_μ^k 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 $p \in \mathit{Prop}$, $x \in \mathcal{V} \cup \mathit{Nom}$ and $X \in \mathcal{V}_2$. The fully hybrid μ-calculus H_μ is the union of all H_μ^k for $k \geq 1$. The modal μ-calculus L_μ is obtained as H_μ in the special case when $\mathcal{V} = \mathit{Nom} = \emptyset$.

We are making use of $\mathtt{tt}$, $\mathtt{ff}$, $\wedge$, $\Diamond$, $\nu X.\varphi$ as abbreviations in the usual way. By $\mathsf{Sub}(\varphi)$ we denote the set of all subformulas of φ. Further, we say that a formula $\varphi \in H_\mu$ is in *negation normal form* if and only if negation only occurs directly in front of atomic formulas.

We assume the following standard sanity condition on formulas: every $X \in \mathcal{V}_2$ is bound at most once by a fixpoint quantifier μ or ν and can only occur under an even number of negations. The function mapping each $X \in \mathcal{V}_2$ to its unique binding formula is called fp_φ. We say that a second-order variable X is of type μ or ν if its defining fixpoint formula $\mathsf{fp}_\varphi(X)$ is a least, resp. greatest fixpoint formula. Formulas with no free first-order variables will be referred to as sentences.

All results easily extend to a multi-modal version of H_μ; for the sake of simplicity we only work with a uni-modal version here.

2.2 Considerations on the Semantics in the Presence of Fixpoints

A Kripke structure is a tuple $K = \langle S, \rightarrow, L\rangle$ where S is a set of states, $\rightarrow \subseteq S \times S$ is a transition relation and $L : \mathit{Prop} \rightarrow 2^S$ labels the states with the sets of propositions that hold true in them.

Formulas of L_μ are usually interpreted over Kripke structures via a mapping $[\![\cdot]\!]_\rho^K$, which maps a formula together with a Kripke structure K as above and an assignment $\rho : \mathcal{V}_2 \rightarrow 2^S$ to the states that satisfy this formula. A formula $\varphi(X)$ with a free second-order variable X

thus induces a monotonic operator $V \mapsto [\![\varphi(X)]\!]^K_{\rho[X\mapsto V]}$, mapping a set V of states to the set of states that satisfy $\varphi(X)$ under the assumption that X holds on the states in V.

The hybrid μ-calculus considered by Sattler and Vardi [16] can be given a semantics in the same way, in particular with an interpretation of type $\mathcal{V}_2 \to 2^S$ of the second-order variables. First-order variables are nominals in the absence of a binder; hence, their interpretation can be fixed in the Kripke structure by extending the labelling function to the type $L : \mathit{Prop} \cup \mathit{Nom} \to 2^S$ with the requirement that $L(m)$ is a singleton set for all $m \in \mathit{Nom}$.

This, however, is not enough in the presence of the binder modality as it should change the mapping of first-order variables to states dynamically during the evaluation of a formula. The naïve approach is to extend the assignment ρ of all second-order variables to a function $\mathcal{V} \cup \mathcal{V}_2 \to 2^S$ such that $\rho(x)$ is a singleton set for each $x \in \mathcal{V}$. This is essentially incorporating their treatment in hybrid temporal logics, c.f. [12]. We could then just extend the usual semantics for L_μ to H_μ via

$$[\![x]\!]^K_\rho = \rho(x)$$

$$[\![@_x\,\varphi]\!]^K_\rho = \begin{cases} S & \text{if } \rho(x) \in [\![\varphi]\!]^K_\rho, \\ \emptyset & \text{otherwise} \end{cases}$$

as it was done in [16] and

$$[\![\downarrow x.\varphi]\!]^K_\rho = \{s \in S \mid s \in [\![\varphi]\!]^K_{\rho[x\to s]}\}.$$

for the binder modality. However, this does not capture the intuition one would have about the interaction between binders and fixpoint recursion; namely that bindings made in one iteration have an effect on the following iterations.

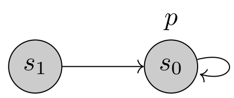
► **Example 1.** Consider the formula $(p \wedge \neg x) \vee \downarrow x.\Diamond X$. Obviously the value of x is supposed to change throughout the evaluation of this formula: the second disjunct is satisfied by a tuple (K, s, ρ) if $(K, s, \rho[x \to s])$ satisfies $\Diamond X$. However, the update on ρ does not have any impact on the valuation of $\Diamond X$ because under the standard μ-calculus semantics extended as stated above, $\Diamond X$ is evaluated without involving x at all and thus $(p\wedge\neg x)\vee\downarrow x.\Diamond X \equiv (p\wedge\neg x)\vee\Diamond X$.

Now consider the least fixpoint of the transformation defined by this formula, $\psi := \mu X.(p \wedge \neg x) \vee \downarrow x.\Diamond X$. This change in $\rho(x)$ *should* have some impact on the fixpoint in the sense that $\Diamond X$ should be calculated relative to the new valuation of x. Moreover, the unfolding principle for fixpoints postulates that X should just be a placeholder for ψ, but the evaluation of ψ surely depends on the value of x. Nonetheless, we have

$$\begin{aligned} \mu X.(p \wedge \neg x) \vee \downarrow x.\Diamond X &\equiv \mu X.(p \wedge \neg x) \vee \Diamond X \\ &\not\equiv (p \wedge \neg x) \vee \downarrow x.\Diamond(\mu X.(p \wedge \neg x) \vee \downarrow x.\Diamond X)\ . \end{aligned} \tag{1}$$

The equivalence is a simple consequence of the fact that – under the semantics proposed above – we have $\downarrow x.\varphi \equiv \varphi$ whenever x is not free in φ. The inequivalence is also easy to grasp: the left-hand side is evaluated independently of the update of x. The right-hand side, however, updates x once before the evaluation of the fixpoint formula is started, then with the new value of x.

To illustrate this, we evaluate both formulas on the following simple Kripke structure:



It is quite obvious that $s_1, \{x \mapsto s_0\} \not\models \mu X.(p \wedge \neg x) \vee \Diamond X$ because under this assignment for the variable x no state satisfies $p \wedge \neg x$ and thus the fixpoint is just the empty set.

However, $s_1, \{x \mapsto s_0\} \models \downarrow x.\Diamond(\mu X.(p \wedge \neg x) \vee \downarrow x.\Diamond X)$ because changing the assignment of x to s_1 and then calculating the fixpoint results in $\{s_1, s_0\}$ and thus $s_1, \{x \mapsto s_0\} \models (p \wedge \neg x) \vee \downarrow x.\Diamond(\mu X.(p \wedge \neg x) \vee \downarrow x.\Diamond X)$. This is because after unfolding the fixpoint once, it is calculated relative to the updated valuation of x rather than the old.

On the other hand, the last formula in (1) should be equivalent to the first one in there because of the desirable equivalence $\mu X.\varphi(X) \equiv \varphi(\mu X.\varphi)$ – the aforementioned unfolding principle.

This example shows that the proposed semantics is inadequate for the fully hybrid μ-calculus including the binder modality. Interestingly, the fixpoint principle still holds semantically but not syntactically, i.e. in general we have

$$[\![\mu X.\varphi(X)]\!]^K_\rho \quad = \quad [\![\varphi(X)]\!]^K_{\rho[X \mapsto [\![\mu X.\varphi(X)]\!]^K_\rho]}$$

for any K, X, φ and ρ, but

$$[\![\mu X.\varphi(X)]\!]^K_\rho \quad \neq \quad [\![\varphi(\mu X.\varphi(X))]\!]^K_\rho$$

as the evaluation of the outer φ may change the variable assignment ρ that is used for the evaluation of the inner fixpoint formula. In other words, this semantics is not compositional, i.e. in general we have

$$[\![\varphi[\psi/X]]\!]^K_\rho \quad \neq \quad [\![\varphi]\!]^K_{\rho[X \mapsto [\![\psi]\!]^K_\rho]} \;.$$

2.3 A Compositional Semantics

To account for such phenomena we propose a new semantics for the fully hybrid μ-calculus. Formulas are still interpreted over Kripke structures $K = \langle S, \rightarrow, L\rangle$. However, the meaning of a formula is now a set pairs consisting of a state and an assignment for the first-order variables. Consequently the variable assignment has to map second-order variables to the same type; it becomes an assignment $\rho : \mathcal{V}_2 \rightarrow 2^{S \times (\mathcal{V} \rightarrow S)}$.

Formally the semantics for H^k_μ for all k with respect to a Kripke structure $K = \langle S, \rightarrow, L\rangle$ over *Prop* and *Nom* and an assignment $\rho : \mathcal{V}_2 \rightarrow 2^{S \times (\mathcal{V} \rightarrow S)}$ is the following:

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

with $p \in \mathit{Prop} \cup \mathit{Nom}$, $x \in \mathcal{V}$ and $X \in \mathcal{V}_2$.

We will 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. Furthermore, we will also sometimes write $(s_1, \ldots, s_k)$ to indicate the function $\sigma : \mathcal{V} \to S$ with $\sigma(x_i) = s_i$ when an order on $\mathcal{V}$ is implictly given, for instance when $\mathcal{V} = \{x_1, \ldots, x_k\}$. To shorten notation even further we sometimes write $K, s \models \varphi$ to express that $K, s, (s, \ldots, s) \models \varphi$.

▶ **Example 2.** Reconsider the formula given in Example 1, now with the new semantics proposed above. We claim

$$\mu X.(p \wedge \neg x) \vee {\downarrow} x.\Diamond X \quad \equiv \quad (p \wedge \neg x) \vee {\downarrow} x.\Diamond(\mu X.(p \wedge \neg x) \vee {\downarrow} x.\Diamond X)$$

holds. We do not prove this formally here; see Proposition 3 below for a general statement. Instead we just give a hint that now this equivalence holds by re-evaluating both formulas on the Kripke structure given in Example 1.

Having two states and one variable, the domain for the semantics is the set of subsets of

$$\{(s_0, x \mapsto s_0), (s_0, x \mapsto s_1), (s_1, x \mapsto s_0), (s_1, x \mapsto s_1)\} \; .$$

Clearly, $p \wedge \neg x$ holds only at $(s_0, x \mapsto s_1)$. Moreover, the least fixpoint $\mu X.(p \wedge \neg x) \vee {\downarrow} x.\Diamond X$ evaluates to the set $M := \{(s_0, x \mapsto s_1), (s_1, x \mapsto s_0), (s_1, x \mapsto s_1)\}$. The first element is included because it satisfies $(p \wedge \neg x)$ so every prefixpoint must contain it. The other two elements then also have to be elements of all prefixpoints because for every prefixpoint T with $T \supseteq \{(s_0, x \mapsto s_1)\}$ we have $[\![{\downarrow} x.\Diamond X]\!]^K_{\rho[X \to T]} \supseteq \{(s_1, x \mapsto s_0), (s_1, x \mapsto s_1)\}$. Finally one can easily check that M a fixpoint.

On the other hand, $M \subseteq [\![(p \wedge \neg x) \vee {\downarrow} x.\Diamond(\mu X.(p \wedge \neg x) \vee {\downarrow} x.\Diamond X)$ because the first element of M satisfies $p \wedge \neg x$ and the other two elements satisfy this formula because clearly when placing x at s_1 we get to $(s_1, x \mapsto s_1)$ and then we can make a transition to $(s_0, x \mapsto s_1)$ which is already part of the least fixpoint. Lastly, $(s_0, x \mapsto s_0) \not\models (p \wedge \neg x) \vee {\downarrow} x.\Diamond(\mu X.(p \wedge \neg x) \vee {\downarrow} x.\Diamond X)$ because it does not satisfy the first disjunct $p \wedge \neg x$ as seen and placing the x at s_0 still leaves us with $(s_0, x \mapsto s_0)$ from where we can only get back to itself with any transition available, and $(s_0, x \mapsto s_0)$ is not an element of the least fixpoint M as seen. So it does not satisfy the second disjunct either.

The semantics proposed here is indeed compositional, as one can routinely check by induction over the formula structure.

▶ **Proposition 3.** *Let $\varphi(X), \psi \in H_\mu$, K be any Kripke structure and ρ assign values of the variables in φ, ψ w.r.t. K. Let $\varphi[\psi/X]$ denote the formula that is obtained from φ by replacing every free occurrence of X with ψ. We have $[\![\varphi[\psi/X]]\!]^K_\rho = [\![\varphi(X)]\!]^K_{\rho[X \mapsto V]}$ where $V = [\![\psi]\!]^K_\rho$.*

This means in particular, that we can use the fixpoint unfolding principle syntactically in H_μ.

Our analysis will mostly focus on formulas in negation normal form. This is not a restriction as the following Lemma shows.

▶ **Lemma 4.** *For every formula $\varphi \in H^k_\mu$ there is an equivalent formula $\varphi' \in H^k_\mu$ in negation normal form and φ' is only polynomially larger.*

Proof. The proof is fairly standard. We simply push negation inwards with de Morgan's laws, the usual equivalences for fixpoints, like $\mu X.\varphi \equiv \neg\nu X.\neg\varphi[\neg X/X]$ and the following equivalences for hybrid operators: $\neg {\downarrow} x.\varphi \equiv {\downarrow} x.\neg\varphi$, $\neg @_x \varphi \equiv @_x \neg\varphi$. ◀

H_μ clearly subsumes hybrid modal logic which is known to be undecidable [4]. Thus, we immediately get the following result concerning H_μ's satisfiability problem.

▶ **Theorem 5.** *Satisfiability for H_μ is undecidable.*

3 Model Checking

In this section we investigate the model checking problem for H_μ. We provide a reduction to L_μ model checking and derive upper complexity bounds for this, prove a matching lower bound for the general case, and finally define model checking games for H_μ based on this reduction and the well-known games for L_μ [17]. They can then be used to aid the understanding of properties expressed by formula of H_μ.

3.1 A Reduction to L_μ Model Checking

Let $\varphi \in H_\mu^k$ for some $k \in \mathbb{N}$ and $K = \langle S, \to, L\rangle$ be a Kripke structure over *Prop*. From these, we construct a Kripke structure $\widehat{K}$ and a formula $\widehat{\varphi}$ of the (multi-modal) μ-calculus over the set of actions $\mathcal{A} = \{\bullet\} \cup \{@_x \mid x \in \mathcal{V}\} \cup \{\downarrow x \mid x \in \mathcal{V}\}$ and atomic propositions from $Prop \cup Nom \cup \mathcal{V}$ as follows.

Let $\varphi \mapsto \widehat{\varphi}$ be the homomorphism such that $\widehat{\Diamond\psi} = \langle\bullet\rangle\widehat{\psi}$, $\widehat{@_x\,\psi} = \langle @_x\rangle\widehat{\psi}$ and $\widehat{\downarrow x.\psi} = \langle\downarrow x\rangle\widehat{\psi}$. Moreover, $\widehat{K} = \langle S \times (\mathcal{V} \to S), \Delta, \widehat{L}\rangle$ where the labeling is defined as $\widehat{L}(p) = \{(s,\sigma) \mid s \in L(p)\}$ for every $p \in Prop \cup Nom$ and $\widehat{L}(x) = \{(s,\sigma) \mid s = \sigma(x)\}$ for every $x \in \mathcal{V}$.

The transition relation Δ is defined as follows.

- $(s,\sigma) \xrightarrow{a} (t,\sigma)$ iff $s \to t$ in K,
- $(s,\sigma) \xrightarrow{@_x} (\sigma(x),\sigma)$ for every $x \in \mathcal{V}$, and
- $(s,\sigma) \xrightarrow{\downarrow x} (s,\sigma[x \mapsto s])$ for every $x \in \mathcal{V}$.

The following can be proved by a straightforward induction over φ.

▶ **Lemma 6.** *For all Kripke structures $K = \langle S, \to, L\rangle$, $s \in S$ and $\sigma : \mathcal{V} \to S$ we have $K, s, \sigma \models \varphi$ iff $\widehat{K}, (s,\sigma) \models \widehat{\varphi}$.*

This realises a reduction from H_μ model checking to L_μ model checking which is polynomial for every fixed k. From this we can derive the following upper complexity bound on the former in the general case.

▶ **Theorem 7.** *The model checking problem for H_μ is in* EXPTIME.

Proof. It is known that L_μ model checking on a Kripke structure K' and a formula ψ can be done in time $\mathcal{O}((|K'| \cdot |\psi|)^{\mathsf{ad}(\psi)})$ [8] where $\mathsf{ad}(\psi)$ denotes the depth of fixpoint alternation in φ. Moreover, $|\psi|$ denotes the size of ψ as measured by the number of its distinct subformulas, and $|K'|$ is the sum of the number of states and edges in K'.

Now take an H_μ^k formula φ and a Kripke structure K and consider $\widehat{K}$ and $\widehat{\varphi}$ as defined above. It is not hard to see that $|\widehat{\varphi}| = \mathcal{O}(|\varphi|)$ and $|\widehat{K}| = \mathcal{O}(|K|^{k+1})$. Hence, Lemma 6 facilitates an exponential reduction to L_μ model checking. Since this is not known to be solvable in polynomial time, the EXPTIME upper bound does not follow directly but requires a slightly more detailed analysis: the reduction produces a Kripke structure $\widehat{K}$ and a formula $\widehat{\varphi}$ such that $\mathsf{ad}(\widehat{\varphi}) = \mathsf{ad}(\varphi)$ and, hence, model checking on these can be performed in time $\mathcal{O}((|K|^{k+1} \cdot |\varphi|)^{\mathsf{ad}(\varphi)})$, i.e. in exponential time. ◀

Clearly, the number of first-order variables is the only source of exponentiation in this reduction. Hence, if this number is fixed, we obtain a better bound.

▶ **Corollary 8.** *For any fixed $k \in \mathbb{N}$ we have that the model checking problem for H_μ^k is at most polynomially worse than that of L_μ.*

This implies membership in NP∩coNP [8], UP∩coUP [10], PLS [20], etc. for model checking each H_μ^k.

$$\frac{s,\sigma \vdash \psi_1 \wedge \psi_2}{s,\sigma \vdash \psi_1 \quad s,\sigma \vdash \psi_2}\ (1) \qquad \frac{s,\sigma \vdash @_x\, \varphi}{\sigma(x),\sigma \vdash \varphi} \qquad (0)\ \frac{s,\sigma \vdash \psi_1 \vee \psi_2}{s,\sigma \vdash \psi_1 \quad s,\sigma \vdash \psi_2}$$

$$\frac{s,\sigma \vdash \Box\psi}{t,\sigma \vdash \psi}\ (1 : s \to t) \qquad \frac{s,\sigma \vdash {\downarrow} x.\varphi}{s,\sigma[x \mapsto s] \vdash \varphi} \qquad (0 : s \to t)\ \frac{s,\sigma \vdash \Diamond\psi}{t,\sigma \vdash \psi}$$

$$\frac{s,\sigma \vdash \nu X.\psi(X)}{s,\sigma \vdash \psi(X)}\ (1) \qquad \frac{s,\sigma \vdash X}{s,\sigma \vdash \mathsf{fp}_\varphi(X)} \qquad (0)\ \frac{s,\sigma \vdash \mu X.\psi(X)}{s,\sigma \vdash \psi(X)}$$

Figure 1 The game rules for H_μ model checking.

3.2 Model Checking Games

Next we give a game-theoretic characterisation of H_μ's model checking problem. Such games are particularly useful for reasoning about the (un-)satisfaction of a formula and therefore to understand the properties expressed by H_μ formulas, for instance in the proof of the lower bound in the next section.

▶ **Definition 9.** Let $\varphi \in H_\mu$ be in negation normal form, and $K = \langle S, \to, L\rangle$ be a Kripke structure. The model checking game $\mathcal{G}(K, \varphi)$ is played by 2 players – called 0 and 1. It is Player 0's task to show that the formula holds while Player 1 tries to refute this. The game's positions are $S \times (\mathcal{V} \to S) \times \mathsf{Sub}(\varphi)$. We usually write such a position as $s, \sigma \vdash \psi$.

The game can evolve using the rules in Figure 1. Those that are annotated with player i induce a choice for this player. For example in a configuration $(s, \sigma) \vdash \Diamond\psi$ it is player 0's task to choose a successor t of s in K and then the play continues in the position $t, \sigma \vdash \psi$.

A player wins a play if their opponent is stuck, i.e. cannot perform a prescribed choice anymore. Furthermore, player 0 wins if she can reach a position $s, \sigma \vdash p$ with $s \in L(p)$ for some $p \in \mathit{Prop} \cup \mathit{Nom}$ or $s, \sigma \vdash x$ with $\sigma(x) = s$ for some $x \in \mathcal{V}$. On the other hand, if $s \notin L(p)$ resp. $\sigma(x) \neq s$ player 1 wins. Likewise, player 0 wins in a position $s, \sigma \vdash \neg p$ if $s \notin L(p)$, and a position $s, \sigma \vdash \neg x$ if $\sigma(x) \neq s$.

Finally, let $>_\varphi$ be the smallest relation such that $X >_\varphi Y$ if X has a free occurrence in $\mathsf{fp}_\varphi(Y)$ that is closed under transitivity. The winner of an infinite play is determined by the type of the unique largest (with respect to $>_\varphi$) fixpoint variable that occurs infinitely often. Player 0 wins if its type is ν and player 1 wins if its type is μ.

Next we need to show that these games characterise the model checking problem for H_μ. This is particularly easy with Lemma 6 at hand, which lets us lift the correctness property of the L_μ model checking games – player 0 wins iff the formula holds – to the H_μ games.

Let $K = \langle S, \to, L\rangle$ and $\varphi \in H^k_\mu$ for some k be given, and let $\widehat{\mathcal{G}}(\widehat{K}, \widehat{\varphi})$ be the model checking game in the multi-modal L_μ for $\widehat{K}$ and $\widehat{\varphi}$. Remember that $L_\mu = H^0_\mu$ and note that model checking games for H^0_μ can ignore the variable assignment in their positions.

▶ **Lemma 10.** *Player* 0 *wins a position* $s, \sigma \vdash \varphi$ *in* $\mathcal{G}(K, \varphi)$ *if and only if Player* 0 *wins a position* $(s, \sigma) \vdash \widehat{\varphi}$ *in* $\widehat{\mathcal{G}}(\widehat{K}, \widehat{\varphi})$.

Proof. "⇐" Suppose player 0 has a winning strategy χ for $\widehat{\mathcal{G}}(\widehat{K}, \widehat{\varphi})$. Because of positional determinacy for L_μ model checking games [17] we can assume χ to prescribe a choice to player 0 in each configuration that contains a disjunction or a diamond formula (regardless of the play's history). This strategy can easily be transferred into a positional strategy χ' for player 0 in $\mathcal{G}(K, \varphi)$ via $\chi'((s, \sigma) \vdash \psi) = \chi(s, \sigma \vdash \widehat{\psi})$. Note that states in K are of the form (s, σ) and, as said above, positions in the L_μ, resp. H^0_μ model checking games are pairs of states and subformulas only. Nominally, player 0 has more choices with χ than with χ'

because binder und jump modalities in φ have become diamond modalities in $\widehat{\varphi}$. However, the underlying edge relations in $\widehat{K}$ are deterministic which means that player 0's only choice in such positions is to do what the semantics of binders and jumps require.

It is not hard to see that χ' is winning if χ is because $\widehat{\varphi}$ has essentially the same structure as φ. Hence, if player 0 can use χ to enforce a play in which the outermost fixpoint variable occurring infinitely often is of type ν then so can she using χ'.

"⇒" This can be proved in the same way by transforming a winning strategy in $\mathcal{G}(K,\varphi)$ into one in $\widehat{\mathcal{G}}(\widehat{K},\widehat{\varphi})$ now adding the deterministic choices for player 0 at additional diamond subformulas. ◀

Putting Lemmas 6 and 10 together we obtain correctness of the H_μ model checking games.

▶ **Theorem 11.** *Player 0 has a winning strategy in a position $s, \sigma \vdash \varphi$ in the model checking game $\mathcal{G}(K,\varphi)$ if and only if $K, s, \sigma \models \varphi$.*

3.3 A Lower Bound

It remains to be seen that the exponential time upper bound for H_μ is tight. For this we reduce the n-corridor tiling game problem [6] to the model checking problem of H_μ.

A tiling system is a tuple $\mathcal{T} = \langle T, H, V, t_1\rangle$ consisting of a set of tiles $T = \{t_1, \dots, t_m\}$, a horizontal matching relation $H \subseteq T \times T$, a vertical matching relation $V \subseteq T \times T$ and an initial tile t_1.

Let $n \geq 1$. The n-corridor tiling game is played between two players Adam and Eve on such a $\mathcal{T}$ and the $(\mathbb{N} \times \{0, \dots, n-1\})$-corridor as follows. At the beginning, the initial tile t_1 is being placed at position $(0,0)$. Whenever the first tile of a row has been placed, Eve needs to complete this row with tiles respecting the vertical and horizontal matching relations. Whenever a row is finished, Adam's places a tile onto the first position of the next row such that this tile matches the one below w.r.t. V.

A play is won by a player if their opponent is unable to place a tile without violating the matching relations. Additionally, Eve wins any play that goes on forever.

The n-corridor tiling game problem is the following: given a tiling system $\mathcal{T}$ and an $n \in \mathbb{N}$ in unary encoding, decide whether Eve has a winning strategy for the n-corridor tiling game on $\mathcal{T}$. This problem is known to be ExpTime-hard [6].

▶ **Theorem 12.** *The model checking problem for H_μ is* ExpTime*-hard.*

Proof. Let $\mathcal{T} = \langle T, H, V, t_1\rangle$ with $T = \{t_1, \dots, t_m\}$ and $n \in \mathbb{N}$ be given. To help with notation define $H_{t_i} := \{t \mid (t_i, t) \in H\}$ as the possible horizontal successors of t_i and $V_{t_i} := \{t \mid (t_i, t) \in V\}$ as the possible vertical successors of t_i for each $i \in \{1, \dots, m\}$.

We build a Kripke structure $K^n_{\mathcal{T}}$ over $Prop = T$ with a designated state s_0 and an H^n_μ formula $\varphi^n_{\mathcal{T}}$ such that $K^n_{\mathcal{T}}, s_0, \sigma \models \varphi_{\mathcal{T}}$ for any σ if and only if Eve wins the n-corridor tiling game on $\mathcal{T}$.

Intuitively, a path through $K^n_{\mathcal{T}}$ corresponds to a particular play in the n-corridor tiling game on $\mathcal{T}$. Each state is labeled with exactly one atomic proposition from T representing the tile placed at a particular position in the n-corridor. It is encoded row-wise as an infinite path, i.e. the i-th state on this path represents the position $(\lfloor i \div n\rfloor, i \mod n)$ in the n-corridor. It is possible to let $K^n_{\mathcal{T}}$ consist of the full clique of m states only – one for each tile. However, it is more convenient to encode the horizontal matching relation into the structure such that an edge from (a state labeled) t in some column to t' in the next column only exists if $(t, t') \in H$ and both represent positions in a common row in the n-corridor.

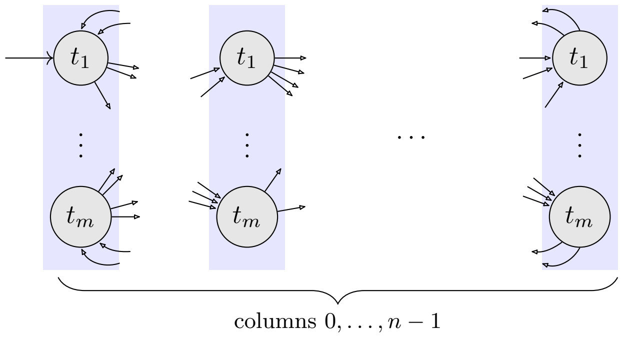


Figure 2 The Kripke structure $K^n_{\mathcal{T}}$ used in the reduction from the n-corridor tiling game problem.

$K^n_{\mathcal{T}}$ is depicted in Figure 2. The initial state is the one labeled with the initial tile t_1 in column 0. There is an edge from a state (identified by its unique label) t in column i to state t' in column $i+1$ iff $(t, t') \in H$ and $i < n-1$. Additionally, there are edges from every state in column $n-1$ to every state in column 0.

The formula $\varphi^n_{\mathcal{T}}$ then needs to describe the evolution of the n-corridor tiling game. The fact that it potentially goes on forever is modeled by a greatest fixpoint recursion. Each iteration corresponds to the construction of a row. For technical convenience, since the initial tile in the n-corridor tiling game is fixed, it actually corresponds to the construction of the $n-1$ last tiles of a row plus the first tile of the next row. In order to check whether the players only choose tiles that match vertically, we use n first-order variables $x_0, \ldots, x_{n-1}$ which are placed during the construction of a row, and then can be used to remember those tiles for the construction of the next row.

As a shorthand we use the formula

$$vm_i(Z) := \Big(\bigvee_{(t,t') \in V} t' \wedge @_{x_i} t\Big) \wedge \downarrow x_i.Z$$

for $i = 0, \ldots, n-1$ which compares the tile at the current position with the tile at the position stored in x_i, and additionally binds the variable x_i to the state that it is currently evaluated in. Then let

$$\varphi^n_{\mathcal{T}} := \downarrow x_0.\Diamond\downarrow x_1.\Diamond\downarrow x_2.\ldots\Diamond\downarrow x_{n-1}.\Box\big(\nu Y.\neg vm_0(\neg\Diamond vm_1(\Diamond vm_2(\ldots\Diamond vm_{n-1}(\Box Y)\ldots)))\big).$$

Using the previously introduced model checking games for H_μ one can check that a winning strategy for Eve in the n-corridor tiling game induces a winning strategy for player 0 in the model checking game on $K^n_{\mathcal{T}}$ and $\varphi^n_{\mathcal{T}}$: her choices of tiles in the tiling game correspond directly to choices she can do at $\Diamond$-formulas. With Theorem 11 we then get correctness of this reduction. Moreover, it is easy to see that both $K^n_{\mathcal{T}}$ and $\varphi^n_{\mathcal{T}}$ can be constructed in time polynomial in $|\mathcal{T}|$ and the value of n. ◀

4 Expressiveness

This section studies the expressive power of H_μ with a particular focus on principle bounds imposed in the sense of bisimulation invariance. It is well known that L_μ is bisimulation-invariant, i.e. formulas of L_μ cannot distinguish bisimilar models, but that hybrid operators break this invariance. For example with just one variable one can distinguish a single self-loop from its tree-unraveling using the formula $x \wedge \Diamond x$.

4.1 A Game-Theoretic Characterisation of k-Bisimulation

In [3] it is shown that hybrid modal logic is invariant under a refined form of bisimulation, called k-bisimulation; its formal definition is recalled below. To better suit our framework of games we present the definition in terms of k-bisimulation games extending the well known (ordinary) bisimulation games [18].

▶ **Definition 13** (k-Bisimulation Game). Given two Kripke structures $K_0 = \langle S_0, \rightarrow_0, L_0\rangle$ and $K_1 = \langle S_1, \rightarrow_1, L_1\rangle$ over a set of atomic propositions *Prop* and a set of nominals *Nom*, the k-bisimulation game $\mathcal{G}^k(K_0, K_1)$ is played between two players – Spoiler and Duplicator – on the configuration space $S_1^{k+1} \times S_2^{k+1}$.

We can imagine that on each structure we have one *active* pebble that gets moved across the structure and k *inactive* pebbles that just mark certain states as well as some *fixed* pebbles that mark the positions of the nominals.

The game is strictly turn-based. First, in a configuration $(s, s_1, \dots, s_k, t, t_1, \dots, t_k)$ Spoiler starts by choosing one of the structures K_i for some $i \in \{0, 1\}$ and then chooses to either

- take a transition $s \rightarrow_0 s'$, resulting in a configuration $(s', s_1, \dots, s_k, t, t_1, \dots, t_k)$, or
- move pebble i from s_i to the current state s, resulting in a configuration $(s, s_1, \dots, s_{i-1}, s, s_{i+1}, \dots, s_k, t, t_1, \dots, t_k)$, or
- jump from the current state s to some pebble s_i, resulting in a configuration $(s_i, s_1, \dots, s_k, t, t_1, \dots, t_k)$ or to some nominal n resulting in a configuration $(n, s_1, \dots, s_k, t, t_1, \dots, t_k)$.

After that Duplicator makes the same kind of move on the other structure K_{1-i}.

Spoiler wins the game if after Duplicator's move the game is in a configuration $(s, s_1, \dots, s_k, t, t_1, \dots, t_k)$ such that the atomic propositions or nominals on s and t do not match or for some $i = 1, \dots, k$, $s = s_i$ but not $t = t_i$ or vice versa. On the other hand Duplicator wins the game if he can always successfully mimic Spoiler's move which means that on s and t the atomic propositions and nominals match and $s = s_i$ if and only if $t = t_i$ for all $i = 1, \dots, k$ and thus the game goes on forever.

We say that $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$ if Duplicator wins $\mathcal{G}^k(K_0, K_1)$ from the configuration $(s, s_1, \dots, s_k, t, t_1, \dots, t_k)$. We say that $s \sim^k t$ if Duplicator wins $\mathcal{G}^k(K_0, K_1)$ from configuration $(s, s, \dots, s, t, t, \dots, t)$.

It is easy to see that for $k = 0$ and no nominals we get the well-known bisimulation games. Furthermore, as also remarked in [3], these games can be restricted to the hybrid operators in use. For example, the restricted games with $k = 0$ variables and only nominals characterise the expressiveness of the hybrid μ-calculus with only nominals and jumps investigated in [16].

The following lemma states some easy observations that will be of use later on.

▶ **Lemma 14.** *Let K_0, K_1 be as in Definition 13. If $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$, then*

(a) *for every $s \rightarrow_0 s'$ there is a transition $t \rightarrow_1 t'$ such that $(s', s_1, \dots, s_k) \sim^k (t', t_1, \dots, t_k)$,*

(b) *$(s, s_1, \dots, s_{i-1}, s, s_{i+1}, \dots, s_k) \sim^k (t, t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_k)$ for every $i = 1, \dots, k$, and*

(c) *$(s_i, s_1, \dots, s_k) \sim^k (t_i, t_1, \dots, t_k)$ for every $i = 1, \dots, k$.*

4.2 The Hybrid μ-Calculus and k-Bisimulation

We will now show that the expressive power of H_μ^k is limited by k-bisimulations.

▶ **Theorem 15.** *Let $\varphi \in H^k_\mu$ be closed and $K_0 = \langle S_0, \rightarrow_0, L_0\rangle, K_1 = \langle S_1, \rightarrow_1, L_1\rangle$ be two Kripke structures. If $(s, s_1, \dots, s_k) \in S_0^{k+1}$ and $(t, t_1, \dots, t_k) \in S_1^{k+1}$ such that $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$, then $K_0, s, (s_1, \dots, s_k) \models \varphi$ if and only if $K_1, t, (t_1, \dots, t_k) \models \varphi$.*

Proof. To prove this by induction we have to strengthen the hypothesis in order to account for formulas $\varphi(X_1, \dots, X_m)$ with free second-order variables $X_1, \dots, X_m$. Let ρ and ρ' be interpretations for these that respect k-bisimilarity in the sense that for all $(s, s_1, \dots, s_k) \in S_0^{k+1}$ and $(t, t_1, \dots, t_k) \in S_1^{k+1}$ it holds that if $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$ then $(s, s_1, \dots, s_k) \in \rho(X_i)$ if and only if $(t, t_1, \dots, t_k) \in \rho'(X_i)$ for any i.

We will prove the following by induction on the structure of φ: $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$ implies that $K_0, s, (s_1, \dots, s_k), \rho \models \varphi$ if and only if $K_1, t, (t_1, \dots, t_k), \rho' \models \varphi$. For technical convenience, the variable assignments have been split into the parts interpreting the first- resp. second-order variables.

So, assume that $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$ and ρ, ρ' are as stated above. For the base case let $\varphi = p$ for some $p \in Prop$. Then from $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$ we immediately get that $s \in L_0(p) \Leftrightarrow t \in L_1(p)$. The case $\varphi = x$ is similar. For the case $\varphi = X_i$ we can use the assumption for free second-order variables as stated above.

The cases for Boolean operators follow immediately from the hypothesis and the case $\varphi = \Diamond\psi$ follows immediately with Lemma 14 a).

For the hybrid operators suppose that $\varphi = \downarrow x_i.\psi$ for some $i \in \{1, \dots, k\}$. We get that

$$\begin{aligned} K_0, s, (s_1, \dots, s_k), \rho \models \varphi \quad &\Leftrightarrow \quad K_0, s, (s_1, \dots, s_{i-1}, s, s_{i+1}, \dots, s_k), \rho \models \psi \\ &\Leftrightarrow \quad K_1, t, (t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_k), \rho' \models \psi \\ &\Leftrightarrow \quad K_1, t, (t_1, \dots, t_k), \rho' \models \varphi \end{aligned}$$

where the first and last equivalence are simply the semantics of H^k_μ and the second equivalence is Lemma 14 b) and the induction hypothesis. The case for $\varphi = @_{x_i}\, \psi$ then follows with Lemma 14 c).

For the last case suppose that $\varphi = \mu X.\psi(X, X_1, \dots, X_m)$ with free variables in ψ as depicted.

Let $\psi^0 := \psi[\mathtt{ff}/X]$ and $\psi^{\alpha+1} := \psi[\psi^\alpha/X]$. With Proposition 3 and the characterisation of least fixpoints via approximations we have that $[\![\mu X.\psi(X)]\!]^K_\varrho = \bigcup_{\alpha<\omega} [\![\psi^\alpha]\!]^K_\varrho$ for some Kripke structure K and assignment ϱ.

We show by a separate induction over α that $K_0, s, (s_1, \dots, s_k), \rho \models \psi^\alpha$ if and only if $K_1, t, (t_1, \dots, t_k), \rho' \models \psi^\alpha$. The case for least fixpoints then follows immediately.

For the base case $\alpha = 0$ observe that $[\![\psi[\mathtt{ff}/X]]\!]^K_\varrho = [\![\psi(X)]\!]^K_{\varrho[X\mapsto\emptyset]}$. Thus,

$$\begin{aligned} K_0, s, (s_1, \dots, s_k), \rho \models \psi^0 \quad &\Leftrightarrow K_0, s, (s_1, \dots, s_k), \rho[X \mapsto \emptyset] \models \psi(X) \\ &\Leftrightarrow K_1, t, (t_1, \dots, t_k), \rho'[X \mapsto \emptyset] \models \psi(X) \\ &\Leftrightarrow K_1, t, (t_1, \dots, t_k), \rho' \models \psi^0 \end{aligned}$$

where the second equivalence is by the fact that for all $(s, s_1, \dots, s_k) \sim^k (t, t_1, \dots, t_k)$ it holds that $(s, s_1, \dots, s_k) \in \emptyset \Leftrightarrow (t, t_1, \dots, t_k) \in \emptyset$ so we can use the induction hypothesis of the outer induction for ψ. For the induction step we then get

$$\begin{aligned} K_0, s, (s_1, \dots, s_k), \rho \models \psi^{\alpha+1} \quad &\Leftrightarrow K_0, s, (s_1, \dots, s_k), \rho[X \mapsto [\![\psi^\alpha]\!]^{K_0}_\rho] \models \psi(X) \\ &\Leftrightarrow K_1, t, (t_1, \dots, t_k), \rho'[X \mapsto [\![\psi^\alpha]\!]^{K_1}_{\rho'}] \models \psi(X) \\ &\Leftrightarrow K_1, t, (t_1, \dots, t_k), \rho' \models \psi^{\alpha+1}. \end{aligned}$$

Here, the induction hypothesis for ψ^α makes sure that the conditions for the induction hypothesis for ψ are met which is why the second equivalence holds. This finishes both inductions. ◀

▶ **Corollary 16.** *Let $\varphi \in H^k_\mu$ be a sentence and $K_0 = \langle S_0, \to_0, L_0\rangle, K_1 = \langle S_1, \to_1, L_1\rangle$ be two Kripke structures. If $s \in S_0$ and $t \in S_1$ such that $s \sim^k t$, then $K_0, s \models \varphi$ if and only if $K_1, t \models \varphi$.*

There is another interesting connection between H_μ and bisimulations: H^2_μ can *express* bisimilarity in the sense that there is a fixed formula $\varphi_\sim$ (relative to a fixed *Prop*) which is true in a Kripke structure with a valuation of two variables if and only if these two variables point at bisimilar states. For technical convenience we assume $Nom = \emptyset$ here, since the usual and simple reduction from bisimilarity between two Kripke structures to bisimilarity within a single structure does not work in the presence of nominals. Instead, one has to rename them uniquely in one of them to allow the disjoint union of two Kripke structures with nominals to be seen as one Kripke structure with nominals again.

▶ **Example 17.** The formula

$$\varphi_\sim := \nu X.\Big(\big(\bigwedge_{p\in Prop} @_x\, p \leftrightarrow @_y\, p\big) \wedge (@_x \Box {\downarrow} x.\, @_y \Diamond {\downarrow} y.X) \wedge (@_y \Box {\downarrow} y.\, @_x \Diamond {\downarrow} x.X)\Big)$$

states that x and y are bisimilar. This can be seen using the model checking games of Section 3.2 for instance.

It is even possible to express k-bisimilarity; however this requires $2k+2$ variables.

▶ **Example 18.** The formula

$$\varphi^k_\sim := \nu X.\Big(\bigwedge_{p\in Prop} (@_x\, p \leftrightarrow @_y\, p) \;\wedge\; (@_x \Box {\downarrow} x.\, @_y \Diamond {\downarrow} y.X) \;\wedge\; (@_y \Box {\downarrow} y.\, @_x \Diamond {\downarrow} x.X)$$
$$\bigwedge_{i=1}^{k} \big((@_x\, x_i \leftrightarrow @_y\, y_i) \;\wedge\; (@_x {\downarrow} x_i.\, @_y {\downarrow} y_i.X) \;\wedge\; (@_{x_i} {\downarrow} x.\, @_{y_i} {\downarrow} y.X)\big)\Big)$$

over the variables $\{x, x_1, \ldots, x_k, y, y_1, \ldots, y_k\}$ ordered in this way is true on $(s, s_1, \ldots, s_k, t, t_1, \ldots, t_k)$ in K if and only if $(s, s_1, \ldots, s_k) \sim^k (t, t_1, \ldots, t_k)$.

▶ **Theorem 19.** *H^{2k+2}_μ can express k-bisimilarity for any $k \geq 0$.*

4.3 Comparing H_μ with Hybrid Branching Time Logics

A natural question that arises when studying the expressiveness of H_μ is its relationship to other hybrid logics. It is easy to see that H_μ is more expressive than hybrid modal logic using the same example which shows that L_μ is more expressive than modal logic. Candidates for an interesting comparison are hybrid extensions of branching time logics. There are three hybrid extensions of CTL* [12] defined by constraints on how hybrid operators can be used on path formulas which are not state formulas, giving rise to the syntactical hierarchy $\text{HCTL}^*_{\mathsf{ss}} \leq \text{HCTL}^*_{\mathsf{ps}} \leq \text{HCTL}^*_{\mathsf{pp}}$. The smallest already subsumes the previously studied hybrid extensions of CTL and CTL$^+$ [11]. We will show that $\text{HCTL}^*_{\mathsf{ps}}$ already is not subsumed by H_μ. This is somewhat surprising given that L_μ is known to subsume CTL* [7].

Let $C_k = \langle S_k, \to, L\rangle$ be the complete clique over k states and $C_\infty = \langle S_\infty, \to, L\rangle$ be the complete clique over $\mathbb{N}$, as depicted for $k = 8$ in Figure 3, as Kripke structures over $Prop = Nom = \emptyset$.

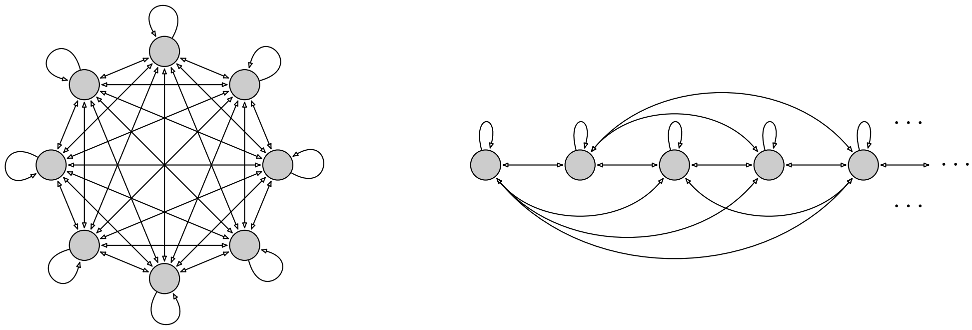


Figure 3 C_8 and C_∞.

▶ **Lemma 20.** *Let $(s, s_1, \ldots, s_k, t, t_1, \ldots, t_k)$ be a configuration in the k-bisimulation game $\mathcal{G}^k(C_{k+1}, C_\infty)$ such that $s = s_i$ if and only if $t = t_i$ for all $i = 1, \ldots, k$. Then Duplicator has a winning strategy for this game starting in this configuration.*

Proof. We call a configuration $(s, s_1, \ldots, s_k, t, t_1, \ldots, t_k)$ *consistent* if for all i we have $s = s_i$ if and only if $t = t_i$. By assumption, the game starts in a consistent configuration. There are two observations to be made:

1. Regardless of Spoiler's choices in a consistent configuration, Duplicator can always answer such that the next configuration is also consistent. This is particularly easy for moves in C_∞, and in C_{k+1} it is possible because every state is reachable from every other in one step, and there is always at least one state which is not inhabited by an inactive pebble.
2. Spoiler wins the k-bisimulation game only when an inconsistent configuration has been reached.

Hence, the simple strategy of preserving consistency is a winning strategy for Duplicator in $\mathcal{G}^k(C_{k+1}, C_\infty)$. ◀

Duplicator especially wins the game starting in configurations of the form $(s, s, \ldots, s, t, t, \ldots, t)$, since they are obviously consistent, which means that $s \sim^k t$ for any $s \in C_{k+1}$ and $t \in C_\infty$.

▶ **Theorem 21.** *There is a formula $\varphi \in HCTL^*_{\mathsf{ps}}$ that cannot be expressed by any formula in H_μ.*

Proof. The formula $\varphi := \mathsf{EG}(\downarrow x.\mathsf{XG}\neg x) \in \mathrm{HCTL}^*_{\mathsf{ps}}$ states that there is an infinite path such that no state on this path is seen twice. Clearly, we have $C_\infty, t \models \varphi$ for any state t and $C_k, s \not\models \varphi$ for any state s and any $k \in \mathbb{N}$. Now suppose there was a formula $\psi \in H_\mu$ expressing this property. Then we would have $\psi \in H^k_\mu$ for some k. By equivalence we would get $C_\infty, t \models \psi$ and $C_{k+1}, s \not\models \psi$ for any states s, t. On the other hand, according to Lemma 20 we have $C_{k+1}, s \sim^k C_\infty, t$, and by Theorem 15, no H_μ formula – in particular not ψ – can distinguish these two structures. Hence, no such ψ can exist. ◀

5 Conclusion and Further Work

We have introduced a hybrid extension of the μ-calculus with nominals, binders and jumps and have shown that the model checking problem for this logic is ExpTime-complete for the general case and only polynomially worse than model checking the modal μ-calculus for a fixed number of variables. We have investigated the expressiveness of the fully hybrid μ-calculus and have shown that it is invariant under hybrid k-bisimulations introduced in [3] when restricted to k variables. We used this result to show that – contrary to the pure modal case – the hybrid extension of the full branching time logic CTL* is not a fragment of H_μ.

Future work will investigate the relationship between hybrid branching time logics and the μ-calculus in more detail. We have shown here that the second level of the syntactic hierarchy of hybrid branching time logics introduced in [12] cannot be translated to the fully hybrid μ-calculus. We believe that the variant which allows binders and jumps over state formulas only could be a fragment of H_μ and will investigate this further.

Another interesting connection that needs to be explored is that between H_μ and the polyadic μ-calculus [2, 15], another extension of L_μ which can express bisimilarity in the sense of Example 17. Its formulas are interpreted in tuples of states of fixed arity, rather than single states as in the case of L_μ. This is reminiscent of the mechanisms in H_μ, especially under the semantics developed here, where a formula with k first-order variables is essentially interpreted by a $(k+1)$-tuple of states. We believe that the polyadic μ-calculus can be embedded into H_μ. The opposite direction cannot hold because the polyadic μ-calculus is known to be bisimulation-invariant. However, this obviously breaks when it is equipped with an equality predicate [14], and we believe that then it becomes strong enough to embed H_μ.

References

1 R. Alur and T. A. Henzinger. A really temporal logic. *J. of the ACM*, 41(1):181–204, 1994.

2 H. R. Andersen. A polyadic modal μ-calculus. Technical Report ID-TR: 1994-195, Dept. of Computer Science, Technical University of Denmark, Copenhagen, 1994.

3 C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *J. of Symb. Log.*, 66:2001, 1999.

4 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.

5 J. van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume II: Extensions of Classical Logic*, pages 167–247. D. Reidel, 1984.

6 B. S. Chlebus. Domino-tiling games. *J. of Comp. and Sys. Sc.*, 32:374–392, 1986.

7 M. Dam. CTL* and ECTL* as fragments of the modal μ-calculus. *TCS*, 126(1):77–96, 1994.

8 E. A. Emerson, C. S. Jutla, and A. P. Sistla. On model-checking for fragments of μ-calculus. In *Proc. 5th Conf. on Computer Aided Verification, CAV'93*, volume 697 of *LNCS*, pages 385–396. Springer, 1993.

9 M. Franceschet and M. de Rijke. Model checking hybrid logics (with an application to semistructured data). *J. of Applied Logic*, 4(3):279–304, 2006.

10 M. Jurdziński. Deciding the winner in parity games is in $UP \cap$co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.

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. Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, 1983.

14 M. Lange and E. Lozes. Model checking the higher-dimensional modal μ-calculus. In *Proc. 8th Workshop on Fixpoints in Computer Science, FICS'12*, volume 77 of *Electr. Proc. in Theor. Comp. Sc.*, pages 39–46, 2012.

15 M. Otto. Bisimulation-invariant PTIME and higher-dimensional μ-calculus. *Theor. Comput. Sci.*, 224(1–2):237–265, 1999.

16 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.

17 C. Stirling. Local model checking games. In *Proc. 6th Conf. on Concurrency Theory, CONCUR'95*, volume 962 of *LNCS*, pages 1–11. Springer, 1995.

18 C. Stirling. Bisimulation, modal logic and model checking games. *Logic J. of the IGPL*, 7(1):103–124, 1999.

19 B. ten Cate. *Model theory for extended modal languages.* PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands, 2005.

20 J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *Proc. 12th Int. Conf. on Computer Aided Verification, CAV'00*, volume 1855 of *LNCS*, pages 202–215. Springer, 2000.