On the complexity of hybrid logics with binders

Balder ten Cate[*] Massimo Franceschet[†]

Draft February 1, 2005

Abstract

Hybrid logic refers to a group of logics lying between modal and first-order logic in which one can refer to individual states of the Kripke structure. In particular, the hybrid logic $HL(@, \downarrow)$ is an appealing extension of modal logic that allows one to refer to a state by means of nominals and to dynamically create names for states.

Unfortunately, as for the richer first-order logic, satisfiability for $HL(@, \downarrow)$ is undecidable and model checking for $HL(@, \downarrow)$ is PSpace-complete. We carefully analyze these negative results and establish restrictions (both syntactic and semantic) that make the logic decidable again and that lower the complexity of the model checking problem.

1 Introduction

There is a general interest in well-behaved logical languages in-between the basic modal language and full first-order logic. Ideally, one would like such languages to combine the good properties of both: to be reasonably expressive, to be decidable, and to have other good properties, such as the interpolation property.[1]

Famous examples of fragments that have been studied are the *guarded fragment* [1, 15] and the *two variable fragment* [20, 16]. Both are decidable, reasonably expressive languages, but they lack interpolation. The hybrid logic $HL(@, \downarrow)$ is another example of a language in between the basic modal language and full first-order logic. It extends the basic modal language with three constructs: nominals, which act as names of states of the model, satisfaction operators, which allow one to express that a formula holds at the state named by a nominal, and $\downarrow$, which allows one to give a name the current world. Together, these three elements greatly increase the expressivity of the language.

[*]Informatics Institute, University of Amsterdam, The Netherlands. Email: `balder.tencate@uva.nl`

[†]Informatics Institute, University of Amsterdam, The Netherlands; Department of Sciences, University of Chieti and Pescara, Italy. Email: `francesc@science.uva.nl`

[1]Interpolation is an important property of logics. Logics that have interpolation are in many ways very well-behaved. Due to lack of space, we can only refer the reader to [18] for more information about interpolation and related properties.

Moreover, like the basic modal language and full first-order logic, $HL(@, \downarrow)$ has the interpolation property. Unfortunately, it is undecidable.

In this paper, we given an in-depth analysis of the undecidability of $HL(@, \downarrow)$. We show how decidability can be regained by making a syntactic restriction on the formulas, or by restricting the class of models in a natural way. Moreover, we show how these and similar syntactic and semantic restrictions affect the complexity of the model checking problem for hybrid languages.

Incidentally, it should be emphasized that $HL(@, \downarrow)$ is a proper fragment of first-order logic, and in fact a very natural fragment. It is the generated submodel invariant fragment of first-order logic [4], it is the least expressive extension of the basic hybrid language $HL(@)$ with interpolation [10], and, finally, it can be characterized as the intersection of first-order logic with second order propositional modal logic [9]. Some of the results reported in this paper can be seen as evidence that $HL(@, \downarrow)$ is better behaved than first-order logic.

The paper is as follows. In Section 2 we introduce hybrid logic and we show that it is a fragment of first-order logic, while in Section 3 we revisit the undecidability result for $HL(@, \downarrow)$. We show how decidability can be regained by restricting the language in Section 4 and by restricting the class of models in Section 5. In Section 6 we investigate how these and similar restrictions affect the complexity of the model checking problem for hybrid logic, and we conclude in Section 7.

2 Hybrid Logic

In this section we introduce hybrid logic and we show that it is a fragment of first-order logic. We assume the reader familiar with modal logic [6].

In its basic version, hybrid logic extends modal logic with devices for naming (individual) states and for accessing states by their names. The key idea is the use of *nominals*. Syntactically, nominals behave like ordinary propositions, but they have an important semantic property. A nominal is true at *exactly one state* of the model. In such a way, it gives a name to that point. Besides nominals, the hybrid language $HL(@, \downarrow)$ also contains @-operators, that allow one to state that a formula is true at a state named by a nominal, and the $\downarrow$-binder, that allows one to introduce variables to name points. Formally, $HL(@, \downarrow)$ is defined as follows.

Let $PROP = \{p, q, \ldots\}$ be a set of proposition symbols, $NOM = \{i, j, \ldots\}$ a set of nominals, and $SVAR = \{x, y, \ldots\}$ a set of state variables. We assume that these sets are disjoint. The formulas of the hybrid language $HL(@, \downarrow)$ are given by the following recursive definition.

$$WFF := \top \mid p \mid t \mid \neg\alpha \mid \alpha \wedge \beta \mid \Diamond\alpha \mid @_t\alpha \mid \downarrow x.\alpha$$

where $p \in PROP$, $t \in NOM \cup SVAR$ and $x \in SVAR$. We will use the familiar shorthand notations, such as $\Box\alpha$ for $\neg\Diamond\neg\alpha$. We say that a state variable x is *free* in a hybrid formula α if x is not bound by $\downarrow$. A hybrid *sentence* is a hybrid formula with no free variables. The *width* of a formula α is the maximum number of free variables of any subformula of α.

The binder $\downarrow$ binds a variable to the current state of evaluation. For instance, the formula $\downarrow x.\Diamond x$ says that the current state is reflexive. The @ operator combines naturally with the $\downarrow$ binder: while $\downarrow$ stores the current state of evaluation, @ enables us to retrieve the information stored by shifting the point of evaluation. As an example, the formula $\downarrow x.\Diamond\downarrow y.@_x\Box y$ states that the current point has exactly one successor.

Hybrid logic is interpreted over hybrid Kripke structures (or hybrid models) of the form $M = (W, R, V)$ where W is a set of states, R is a binary relation over W called the accessability relation, and $V : PROP \cup NOM \to \wp(W)$ is a valuation function that assigns to each proposition letter or nominal a set of states, such that $V(i)$ is a singleton set for each nominal i. The pair $F = (W, R)$ is called the *frame* of M and M is said a model based on the frame F.

An assignment for M is a function $g : SVAR \to W$. Given such an assignment g, we define g^x_w to be the assignment that agrees with g on all variables except x, and that maps the latter to w. More precisely,

$$g^x_w(y) = \begin{cases} w & \text{for } x = y \\ g(y) & \text{for } x \neq y \end{cases}$$

Let $M = (W, R, V)$ be a hybrid model and let $w \in W$. For any nominal i, let $[i]^{M,g} = V(i)$, and for any state variable x, let $[x]^{M,g} = \{g(x)\}$. The semantics of the basic hybrid language is as follows:

$$\begin{array}{lll} M, g, w \Vdash \top & & \\ M, g, w \Vdash p & \text{iff} & w \in V(p) \\ M, g, w \Vdash t & \text{iff} & w \in [t]^{M,g} \text{ for } t \in NOM \cup SVAR \\ M, g, w \Vdash \neg\alpha & \text{iff} & M, g, w \nVdash \alpha \\ M, g, w \Vdash \alpha \wedge \beta & \text{iff} & M, g, w \Vdash \alpha \text{ and } M, w \Vdash \beta \\ M, g, w \Vdash \Diamond\alpha & \text{iff} & \exists w'.(wRw' \wedge M, g, w' \Vdash \alpha) \\ M, g, w \Vdash @_t\alpha & \text{iff} & M, w' \Vdash \alpha \text{ where } \{w'\} = [t]^{M,g} \\ M, g, w \Vdash \downarrow x.\alpha & \text{iff} & M, g^x_w, w \Vdash \alpha \end{array}$$

Define the *first-order correspondence language* to be the first-order language with equality that has one binary relation symbol R, a unary relation symbol p for each $p \in PROP$ and a constant i for each nominals $i \in NOM$. Every hybrid Kripke structure (W, R, V) can be viewed as a relational structure for the first-order correspondence language: the binary relation symbol R is interpreted by the accessibility relation R, the unary relation symbols p are interpreted by $V(p)$, and the constants i denote the unique state w such that $V(i) = \{w\}$. Then, the following *Standard Translation*, defined by mutual recursion between two functions ST_x and ST_y, embeds $HL(@, \downarrow)$ into the first-order correspondence language (where $p \in PROP$ and $t \in NOM \cup SVAR$):

$ST_x(\top)$	$=$	$\top$	$ST_y(\top)$	$=$	$\top$
$ST_x(p)$	$=$	$p(x)$	$ST_y(p)$	$=$	$p(y)$
$ST_x(t)$	$=$	$x = t$	$ST_y(t)$	$=$	$y = t$
$ST_x(\neg\alpha)$	$=$	$\neg ST_x(\alpha)$	$ST_y(\neg\alpha)$	$=$	$\neg ST_y(\alpha)$
$ST_x(\alpha \wedge \beta)$	$=$	$ST_x(\alpha) \wedge ST_x(\beta)$	$ST_y(\alpha \wedge \beta)$	$=$	$ST_y(\alpha) \wedge ST_y(\beta)$
$ST_x(\Diamond\alpha)$	$=$	$\exists y.(xRy \wedge ST_y(\alpha))$	$ST_y(\Diamond\alpha)$	$=$	$\exists x.(yRx \wedge ST_x(\alpha))$
$ST_x(@_t\alpha)$	$=$	$\exists y.(y = t \wedge ST_y(\alpha))$	$ST_y(@_t\alpha)$	$=$	$\exists x.(x = t \wedge ST_x(\alpha))$
$ST_x(\downarrow z.\alpha)$	$=$	$\exists z.(z = x \wedge ST_x(\alpha))$	$ST_y(\downarrow z.\alpha)$	$=$	$\exists z.(z = y \wedge ST_y(\alpha))$

Here, it is assumed that the variables x, y do not occur in α. For each $HL(@,\downarrow)$-formula α with free variables $y_1, \ldots, y_n$, $ST_x(\alpha)$ is a first-order formula with free variables in $\{x, y_1, \ldots, y_n\}$. Moreover, it is easy to show that for any Kripke structure M, assignment g and world w, $M, g, w \Vdash \alpha$ if, and only if, $M, g^x_w \models ST_x(\alpha)$. It follows that $HL(@,\downarrow)$ is a fragment of the first-order correspondence language. In fact, it is the fragment containing (modulo logical equivalence) the formulas that are invariant under generated submodels [4].

It is worth noticing that hybrid sentences of width k are mapped by the above translation to first-order formulas of width at most $k + 2$.

As was pointed out by Guillaume Malod (personal communication), the clause for the $\downarrow$-binder in the Standard Translation for $HL(@,\downarrow)$ given in [4], i.e., $ST_x(\downarrow z.\alpha) = ST_x(\alpha)[z/x]$ and $ST_y(\downarrow z.\alpha) = ST_y(\alpha)[z/y]$, is incorrect. Indeed, consider the formula $\downarrow z.\Diamond\Diamond z$. The Standard Translation of this formula according to the definitions in [4] is $\exists y.(xRy \wedge \exists x.(yRx \wedge x = z))[z/x] = \exists y.(xRy \wedge \exists x.(yRx \wedge x = x))$, which clearly fails to capture the semantics of the hybrid formula.

So far, we have only introduced uni-modal $HL(@,\downarrow)$. This was only for convenience of exposition. It is straightforward to extend the above definitions to the multi-modal case. In fact, in the remainder of this paper, we will frequently make use of multi-modal formulas.

3 The undecidability of $HL(@,\downarrow)$ revisited

In this section, we revisit the negative result that is central to this paper: the undecidability of $HL(@,\downarrow)$ [7]. As a warming up, we give a very simple undecidability proof, by reducing the satisfiability problem for first-order correspondence language to the satisfiability problem for $HL(@,\downarrow)$. Then, we show how the undecidability result can be sharpened using a reduction from an undecidable tiling problem.

Following [8] we call a fragment of first-order logic a *conservative reduction class* if there is a recursive function τ mapping first-order formulas to formulas in the fragment, such that for all formulas α, $\tau(\alpha)$ is satisfiable iff α is, and $\tau(\alpha)$ has a finite model iff α has. Clearly, every conservative reduction class has an undecidable (in fact Π^0_1-complete) satisfiability problem, as well as an undecidable (in fact Σ^0_1-complete) finite satisfiability problem [8]. As was already suggested in [2], $HL(@,\downarrow)$ is a conservative reduction class.

Theorem 3.1 $HL(@,\downarrow)$ *is a conservative reduction class.*

Proof. The class of first-order formulas with equality in a single binary relation is known to be a conservative reduction class [8]. Now, consider the following embedding τ from this first-order language to the hybrid language with @ and $\downarrow$, where s be a fixed nominal:

$$\begin{array}{lcl} \tau(xRy) & = & @_x\Diamond y \\ \tau(p(x)) & = & @_x p \\ \tau(x = y) & = & @_x y \\ \tau(\neg\alpha) & = & \neg\tau(\alpha) \\ \tau(\alpha \wedge \beta) & = & \tau(\alpha) \wedge \tau(\beta) \\ \tau(\exists x.\alpha) & = & @_s\Diamond{\downarrow}x.\ \tau(\alpha) \end{array}$$

Clearly, τ is a recursive function. We claim that for each first-order sentence α, α is has a (finite) model iff $\tau(\alpha)$ is has a (finite) model.

First, suppose $M \models \alpha$. Let the model M' be obtained from M by adding a new state w, labelled with nominal s, and by extending the accessibility relation R such that $(w, v) \in R$ for all states v of M. Then $M', w \models \tau(\alpha)$. Moreover, M' is finite if M is.

Conversely, suppose $M, w \models \tau(\alpha)$. Let v be the state in M labelled by the nominal s. Let M' be the submodel of M consisting of all successors of v. Then $M' \models \alpha$. Moreover, M' is finite if M is. QED

Notice that the nesting degree of the $\downarrow$ binder in $\tau(\alpha)$ corresponds to the quantifier depth of α, which, in general, is not bounded. Hence, from the above proof, it is not clear whether fragments of the full hybrid logic in which formulas have a small nesting degree of $\downarrow$ are decidable.

We now present a different undecidability proof that exploits the $\mathbb{N}\times\mathbb{N}$ tiling problem. The contribution of this new proof is that the hybrid formulas it uses do not nest the $\downarrow$ operator and they contain one state variable only. This proof will be useful to spot the source of complexity of hybrid logic. We will use a hybrid logic with three modalities: $\Diamond_1$ (to move one step up in the grid), $\Diamond_2$ (to move one step to the right in the grid), and $\Diamond$ (to reach all the points of the grid), interpreted by the accessibility relations R_1, R_2 and R, respectively. Moreover, we will take advantage of their converse operators. We will comment later on how to eliminate the converse operators and how to reduce to one accessability relation only.

We first recall the $\mathbb{N} \times \mathbb{N}$ tiling problem. A tile type is a square, fixed in orientation, each side of it has a color. Formally, it can be identified with a 4-tuple of elements of some finite set of colors. To tile a space, we have to ensure that adjacent tiles have the same color on the common side. The $\mathbb{N} \times \mathbb{N}$ tiling problem is as follows: given a finite set of tile types T, can T tile $\mathbb{N} \times \mathbb{N}$? This problem in undecidable (see, e.g., [17]). We now reduce this problem to the satisfiability problem for $HL(@, \downarrow)$ with converse modalities.

Let T be a finite set of tiles, and for each tile $t \in T$ let $left(t), right(t), top(t), bottom(t)$ denote the four colors of t. We will now give a hybrid formula that describes an $\mathbb{N} \times \mathbb{N}$ grid tiled with the tiles in T.

Functionality $\alpha_1 = \Box{\downarrow}x.(\Box_1^-\Box_1 x \wedge \Box_2^-\Box_2 x)$. This property says that the accessability relations R_1 and R_2 are in fact functions.

Grid $\alpha_2 = \Box{\downarrow}x.\Diamond_1\Diamond_2\Diamond_1^-\Diamond_2^- x$. This property says that the accessability relations R_1 and R_2 describe a grid, that is, if, from a given point, we move up and then right, or right and then up, then we end up in the same point.

Tiling $\beta = \Box(\beta_1 \wedge \beta_2 \wedge \beta_3)$, where $\beta_1 = \bigvee_{t \in T}(p_t \wedge \bigwedge_{t,t' \in T; t \neq t'} \neg p_{t'})$ states that exactly one tile is placed at each node of the grid, $\beta_2 = \bigwedge_{t \in T}(p_t \rightarrow \Box \bigvee_{t' \in T; left(t') = right(t)} p_{t'})$ says that horizontally adjacent tiles must match, $\beta_3 = \bigwedge_{t \in T}(p_t \rightarrow \Box \bigvee_{t' \in T; bottom(t') = top(t)} p_{t'})$ says that vertically adjacent tiles must match. Hence, β states that the space is well-tiled.

Spypoint $\gamma = s \wedge \Diamond s \wedge \Box\Box_1\Diamond^- s \wedge \Box\Box_2\Diamond^- s$, where s is a nominal. This property says that there is a spypoint labelled with nominal s that can reach each point of the grid trough the relation R.

It is easy to prove that there is a solution to the tiling problem if, and only if, the hybrid formula $\pi = \alpha_1 \wedge \alpha_2 \wedge \beta \wedge \gamma$ is satisfiable. The formulas α_1 and α_2 make use of converse operators (also called *tense operators* or *backward looking operators*). Alternatively, $\Diamond_1^-$ and $\Diamond_2^-$ could be considered as independent modalities, in which case π must be extended with additional conjuncts $\Box{\downarrow}x.(\Box_k\Diamond_k^- x \wedge \Box_k^-\Diamond_k x)$ (for $k = 1, 2$).

It is worth noticing that the formula π does not nest the $\downarrow$ binder and it contains only one variable. Moreover, the *functionality statement* α_1 in the only conjunct in π containing a $\Box{\downarrow}\Box$-pattern, that is, a $\Box$-operator that has scope over a $\downarrow$ that in turn has scope over a $\Box$-operator. We can conclude that the source of undecidability for hybrid logic is not the nesting degree of $\downarrow$ nor the number of state variables used the formulas. As we will show in the next section, the source of complexity of the satisfiability problem for hybrid logic is instead the $\Box{\downarrow}\Box$-pattern.

We conclude this section by briefly surveying the undecidability proofs for hybrid logic with $\downarrow$ binder. The first undecidability proofs appear in [7, 14]. Both the proofs embed the tiling problem into the satisfiability for hybrid logic with $\downarrow$ binder. However, Goranko takes advantage of a primitive global modality to create the grid. Blackburn and Seligman, instead, make use of a spy point technique: each point in the grid is seen by a spy point s (not belonging to the grid) and it sees s back. Since they do not use converse operators, their proof is a bit more complicated than ours and the encoding formulas do nest the $\downarrow$ binder. Areces, Blackburn, and Marx give another proof in which they embed the global satisfaction problem for K_{23} (the class of frames in which every state has at most 2 successors and at most 3 two-step successors) [4]. This proof is very simple but it makes us of a deep nesting of $\downarrow$. Finally, Marx gives another proof by embedding the tiling problem into a fragment of description logic extended with $\downarrow$ [19]. The formulas used in this proof do not nest $\downarrow$ and contain only one state variable. Moreover, only one relation and no converse operators are used. The encoding of the tiling problem with only one relation is made possible by adding an additional point labelled with the proposition *up* (respectively, *right*) between any two consecutive vertical (respectively, horizontal) points in the grid and by using a symmetric accessibility relation. However, this proof far more involving then ours. All the above proofs use formulas showing the $\Box\!\downarrow\!\Box$-pattern. We think that the proof we gave above is simple and, since it also uses very simple formulas, it spots the complexity source of the problem.

4 Syntactic restrictions

The undecidability proofs in the previous section involve formulas containing a $\Box\!\downarrow\!\Box$-pattern. In this section, we show that such formulas are actually necessary for the undecidability of the satisfiability problem.

We first extend the hybrid language with the global modality $E\alpha$ and with the converse operator $\Diamond^{-}\alpha$, and their duals $A\alpha = \neg A \neg \alpha$ and $\Box^{-}\alpha = \neg \Diamond^{-} \neg \alpha$, respectively. The new semantics clauses are as follows:

$$\begin{array}{lll} M, g, w \Vdash E\alpha & \text{iff} & \exists w'.\ M, g, w' \Vdash \alpha \\ M, g, w \Vdash \Diamond^{-}\alpha & \text{iff} & \exists w'.\ (w'Rw \wedge M, g, w' \Vdash \alpha) \end{array}$$

The Standard Translation of the new operators into the first-order correspondence language is as follows:

$$\begin{array}{lll|lll} ST_x(E\alpha) & = & \exists y.(y = y \wedge ST_y(\alpha)) & ST_y(E\alpha) & = & \exists x.(x = x \wedge ST_x(\alpha)) \\ ST_x(\Diamond^{-}\alpha) & = & \exists y.(yRx \wedge ST_y(\alpha)) & ST_y(\Diamond^{-}\alpha) & = & \exists x.(xRy \wedge ST_x(\alpha)) \end{array}$$

We call the resulting language the full hybrid language (FHL, for short). Let us define $HL(\theta_1, \ldots, \theta_n)$ as the extension of the modal language with nominals and operators $\theta_1, \ldots, \theta_n$ (if $\downarrow$ is among $\theta_1, \ldots, \theta_n$ then it is understood that the languages contains state variables as well). For instance, $HL(@)$ is the basic hybrid language with nominals and @, while $HL(@, \Diamond^{-}, E, \downarrow)$ is the full hybrid language. It is known that $HL(@)$ is PSPACE-complete and $HL(@, \Diamond^{-}, E)$ is ExpTime-complete [3]. As we already know, $HL(@, \downarrow)$ is undecidable (even without @ and nominals) [4].

In this section, we will consider the universal operators $\Box$ and $\Box^{-}$ as well as the disjunction $\vee$ as part of the language (and not just as shorthand definitions). Moreover, we will restrict ourselves to hybrid sentences, that is hybrid formulas with no free variables. This is not a limitation for our purpose, since, given a formula with free variables, we can always replace the free variables with fresh nominals obtaining an equisatisfiable formula.

We say that a hybrid formula α is in *negation normal form* (NNF) if the negation symbol $\neg$ appears in front of atomic formulas only. Notice that each hybrid formula in equivalent to a hybrid formula in NNF. For instance, we have that $\neg\!\downarrow\!x.\Diamond(x \wedge \neg p) \equiv \downarrow\!x.\Box(\neg x \vee p)$.

We call *universal operators* the modalities $\Box$, $\Box^{-}$ and A, and *existential operators* the modalities $\Diamond$, $\Diamond^{-}$ and E. We define a $\Box\!\downarrow$-formula (respectively, $\Diamond\!\downarrow$-formula) as a hybrid formula in NNF in which some occurrence of $\downarrow$ is in the scope of a universal (respectively, existential) operator. Moreover, we define a

$\downarrow\Box$-formula (respectively, $\downarrow\Diamond$-formula) as a hybrid formula in NNF in which an occurrence of a universal (respectively, existential) operator is in the scope of a $\downarrow$. Similar definitions hold for different patterns. For example, $\Box\downarrow\Box$-formula is a formula in NNF containing a universal operator in contains in its scope a $\downarrow$ that contains in its scope a universal operator. A $\downarrow$-formula is simply a formula in NNF containing a $\downarrow$ binder. Given a pattern π, we define $FHL \setminus \pi$ as the fragment of the full hybrid language consisting of all formulas in NNF that are not of the form π. Notice that languages $FHL \setminus \pi$ are not necessarily closed under negation.

Theorem 4.1 *There is a polynomial satisfiability-preserving translation from* $FHL \setminus \Box\downarrow$ *to* $HL(@, \Diamond^-, E)$*. Moreover, the translation preserve satisfiability relative to any class of frames.*

Proof. It is convenient to introduce a new hybrid binder $\exists$. We add to the language formulas of the form $\exists x.\alpha$, where x is a state variable, interpreted as follows:

$$M, g, w \Vdash \exists x.\alpha \text{ iff } M, g^x_{w'}, w \Vdash \alpha \text{ for some state } w'$$

Notice that $\downarrow$ can be defined in terms of $\exists$ as follows: $\downarrow x.\alpha \equiv \exists x.(x \wedge \alpha)$.

Let us proceed with the proof. Let α_0 be a hybrid formula in $FHL \setminus \Box\downarrow$. We show how to polynomially translate α_0 into a formula α_3 in $HL(@, \Diamond^-, E)$ such that α_0 is satisfiable if, and only if, α_3 is satisfiable. The translation consists of three steps:

1. rewrite each subformula of α_0 of the form $\downarrow x.\varphi$ as $\exists x(x \wedge \varphi)$ and let α_1 be the resulting equivalent formula. Since no occurrence of the $\downarrow$ binder in α_0 is in the scope of a universal operator, the same holds for the occurrences of the $\exists$ binder in α_1;

2. rewrite α_1 in prenex normal form, which means with all the existential binders $\exists$ in front of the formula. This is possible using the following equivalences: $\Diamond\exists x.\varphi \equiv \exists x.\Diamond\varphi$, $\Diamond^-\exists x.\varphi \equiv \exists x.\Diamond^-\varphi$, $E\,\exists x.\varphi \equiv \exists x.E\varphi$, $@\exists x.\varphi \equiv \exists x.@\varphi$, $\psi \wedge \exists x.\varphi \equiv \exists x.(\psi \wedge \varphi)$, $\psi \vee \exists x.\varphi \equiv \exists x.(\psi \vee \varphi)$. Let's α_2 be the resulting equivalent formula;

3. replace each state variable in α_2 by a fresh nominal and drop the corresponding existential quantifier. Let's call α_3 the resulting formula.

Notice that α_3 is in $HL(@, \Diamond^-, E)$ and the size of α_3 is linear in the size of α_0. We claim that α_0 is satisfiable if, and only if, α_3 is satisfiable. Since α_0 is equivalent to α_2, it is sufficient to prove that α_2 and α_3 are equi-satisfiable.

Assume that $\alpha_2 = \exists x_1. \ldots \exists x_n.\beta$, and α_3 is obtained from β by replacing, for $j = 1, \ldots n$, the state variable x_j by the nominal i_j.

If α_2 is satisfiable, then there is a hybrid model $M = (W, R, V)$, an assignment g, and a state w such that $M, g, w \vDash \alpha_2$. Hence, there is an sequence $(w_1, \ldots, w_n) \in W^n$ such that $M, g', w \vDash \beta$, where $g' = g[x_1/w_1, \ldots, x_n/w_n]$.

Let $M' = (W, R, V')$, where V' is such that $V'(i_j) = \{w_j\}$ and $V'(t) = V(t)$ for $t \notin \{i_1, \ldots i_n\}$. It follows that $M', g, w \vDash \alpha_3$, hence α_3 is satisfiable.

Conversely, if α_3 is satisfiable, then there is a hybrid model $M = (W, R, V)$, an assignment g, and a state w such that $M, g, w \vDash \alpha_3$. Let $V(i_j) = \{w_j\}$, for $j = 1, \ldots n$. Then, $M, g', w \vDash \beta$, where $g' = g[x_1/w_1, \ldots, x_n/w_n]$. It follows that $M, g, w \vDash \exists x_1 \ldots \exists x_n.\beta$, that is, $M, g, w \vDash \alpha_2$, hence α_2 is satisfiable. QED

Corollary 4.2 *The satisfiability problem for $FHL \setminus \Box\downarrow$ is* ExpTime*-complete.*

Proof. The lower bound follows from the fact that $FHL \setminus \Box\downarrow$ embeds the basic modal language with global modality, which is known to have an ExpTime-complete satisfiability problem [12]. The upper bound follows from Theorem 4.1 since satisfiability of $HL(@, \Diamond^-, E)$-formulas can be decided in ExpTime. QED

We now prove the mirror image of Theorem 4.1: satisfiability for $FHL \setminus \downarrow\Box$ is decidable. We use a technique similar to the one used to prove Theorem 1 in [19], embedding $FHL \setminus \downarrow\Box$ into the universally guarded fragment. Formulas in this fragment are constructed from atoms and their negations by conjunction, disjunction, unrestricted existential quantification and guarded universal quantification. Hence only universal quantification is constrained. The satisfiability problem for the universal guarded fragment is 2ExpTime-complete. The satisfiability problem becomes ExpTime-complete when there is a uniform bound on the width of the formula. For more details, cf. [11].

Theorem 4.3 *The satisfiability problem for $FHL \setminus \downarrow\Box$ is in 2*ExpTime*. The satisfiability problem for $FHL \setminus \downarrow\Box$-formulas of bounded width is* ExpTime*-complete.*

Proof. Let α be any $FHL \setminus \downarrow\Box$-sentence. We will show by induction on α that $ST_x(\alpha)$ is universally guarded. Since $ST_x(\alpha)$ can be obtained from α in polynomial time, this proved that the satisfiability problem for $FHL \setminus \downarrow\Box$ is in 2ExpTime.

If α is a (negated) atomic formula, then $ST_x(\alpha)$ is quantifier-free, hence universally guarded. If α is of the form $\alpha_1 \wedge \alpha_2$, then it follows from the induction hypothesis that $ST_x(\alpha)$ is the conjunction of two universally guarded formulas, hence itself universally guarded. Similarly for α is of the form $\alpha_1 \vee \alpha_2$.

Next, suppose α is of the form $X\alpha_1$, where X is an existential operator. By induction hypothesis, $ST_y(\alpha_1)$ is universally guarded. Inspection of the relevant clauses of the Standard Translation shows that $ST_x(\alpha)$ is also universally guarded. Similarly if α is of the form $@_t\alpha_1$.

Consider the case where α is of the form $X\alpha_1$, for some universal operator X. Again, by induction hypothesis, $ST_y(\alpha_1)$ is universally guarded. Also, since our original formula α is a $FHL \setminus \downarrow\Box$-formula, α_1 does not contain any free state variables. It follows that $ST_y(\alpha_1)$ contains no free variables besides (possibly) y. Inspection of the relevant clauses of the Standard Translation shows that this variable y is appropriately guarded in $ST_x(\alpha)$, hence $ST_x(\alpha)$ is universally guarded.

Finally, suppose α is of the form $\downarrow z.\alpha_1$. Then, $ST_x(\alpha) = \exists z.(z = x \wedge ST_x(\alpha_1))$. Since α is a $FHL \setminus \downarrow\Box$-formula, α_1 does not contain any universal operator, hence $ST_x(\alpha_1)$ does not contain any universal quantifier, and thus it is universally guarded. It follows that $ST_x(\alpha)$ is also universally guarded.

As was already mentioned in Section 2, if a hybrid formula α has width w, then the width of $ST_x(\alpha)$ is at most $w + 2$. Hence, a bound on the width of the $FHL \setminus \downarrow\Box$-formula α implies a bound on the width of its universally guarded translation $ST_x(\alpha)$. Since the satisfiability problem for universally guarded formulas of bounded width is ExpTime-complete, this gives us an ExpTime upper bound. The lower bound follows from the ExpTime-hardness of the basic modal logic extended with the global modality [12]. QED

Satisfiability for $FHL \setminus \downarrow\Box$ is ExpTime-hard, since satisfiability for modal logic with the global modality is already ExpTime-hard [12]. We don't know the exact complexity of $FHL \setminus \downarrow\Box$, but we conjecture that is it ExpTime-complete.

By combining the techniques used to prove Theorems 4.1 and 4.3, we have the main result of this section:

Theorem 4.4 *The satisfiability problem for $FHL \setminus \Box\downarrow\Box$ is in 2*ExpTime*. The satisfiability problem for $FHL \setminus \Box\downarrow\Box$-formulas of bounded width is* ExpTime*-complete.*

Proof. Let $\alpha \in FHL \setminus \Box\downarrow\Box$. If $\alpha \in FHL \setminus \downarrow\Box$, then the satisfiability of α can be decided in 2ExpTime by Theorem 4.3. Suppose therefore that $\alpha \notin FHL \setminus \downarrow\Box$. Let β be a minimal $\downarrow\Box$-subformula of α. Since $\alpha \in FHL \setminus \Box\downarrow\Box$, β cannot be in the scope of a universal operator in α. It follows that this occurrence of $\downarrow$ can be removed as in the proof of Theorem 4.1. Repeating this step for each minimal $\downarrow\Box$-subformula of α, we obtain a formula $\beta \in FHL \setminus \downarrow\Box$ that is satisfiable iff α is satisfiable. By Theorem 4.3, satisfiability of β can be checked in 2ExpTime. The ExpTime-completeness in the case of bounded width follows from the bounded width case of Theorem 4.3. QED

Since the negation of an $FHL \setminus \Diamond\downarrow\Diamond$-formula is equivalent to an $FHL \setminus \Box\downarrow\Box$-formula, we have as a corollary the following dual result.

Corollary 4.5 *The validity problem for $FHL \setminus \Diamond\downarrow\Diamond$ is in 2*ExpTime*. The validity problem for $FHL \setminus \Diamond\downarrow\Diamond$-formulas of bounded width is* ExpTime*-complete.*

In particular, if a hybrid formula ϕ contains neither the $\Box\downarrow\Box$ pattern and nor the $\Diamond\downarrow\Diamond$ pattern, then both satisfiability and validity of ϕ are decidable.

5 Semantic restrictions

In this section, we restrict attention to uni-modal models of bounded width, i.e., models with only one binary relation R, in which each node is R-related only to

Table 1: Complexity of the satisfiability problem on κ-models

	$HL(@, \downarrow)$	first-order correspondence language
$\kappa = 1$	NP-complete	NExpTime-complete
$\kappa = 2$	NP-complete	Decidable, not elementary recursive
$3 \leq \kappa < \omega$	NExpTime-complete	Π^0_1-complete (co-r.e., not decidable)
$\kappa = \omega$	Σ^0_1-complete (r.e., not decidable)	Σ^1_1-complete (highly undecidable)
$\kappa > \omega$	Π^0_1-complete (co-r.e., not decidable)	Π^0_1-complete (co-r.e., not decidable)

a restricted number of points. More precisely, for any cardinal κ, let K_κ be the class of uni-modal models in which for every node d there are strictly less than κ nodes e such that $(d, e) \in R$. In particular, K_2 is the class of models in which every points has at most one R-successor, and K_ω is the class of models in which every node has only finitely many R-successors. We will refer to elements of K_κ as κ-models for short. In what follows we will consider the satisfiability problem of $HL(@, \downarrow)$ and of the first-order correspondence language on κ-models, for particular κ. Our results are summarized in Table 1. All results generalize to to case with multiple modalities, except for the decidability of the first-order correspondence language on K_2.

The terminology and results used in this section can be found in [8] and [17], or in other texts on computational complexity. In particular, we follow the usual terminology from recursion theory: the language of second-order arithmetic is the second-order language with constants 0, 1, function symbols + and $\times$, and equality. Formulas of second-order arithmetic are interpreted over the natural numbers. A Σ^1_1 formula of second order arithmetic is a formula of the form $\exists R_1 \ldots R_n.\phi$ where ϕ contains no second-order quantifiers. A set A of natural numbers is said to be in Σ^1_1 if it is defined by a Σ^1_1 formula that has one free first-order variable and no free second-order variables. A set A of natural numbers is Σ^1_1-hard if for every B in Σ^1_1 there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, $n \in B$ iff $f(n) \in A$. A set of natural numbers is Σ^1_1-complete if it is both in Σ^1_1 and Σ^1_1-hard. It is well known that Σ^1_1-hard sets are not recursively enumerable. When one speaks of an arbitrary decision problem as being in Σ^1_1 or Σ^1_1-hard, it is implicitly understood that the instances of the decision problem are coded into natural numbers (under some computable encoding).

Following [8], we call a decidable problem *elementary recursive* if the time complexity can be bounded by a constant number of iterations of the exponential function.

Finally, given a formula ϕ and a unary predicate P, we will use the notation ϕ^P to refer to the *relativisation of ϕ by P*, i.e., the result of replacing all subformulas in ϕ of the form $\exists x.\psi$ or $\forall x.\psi$ by $\exists x.(Px \wedge \psi)$ resp. $\forall x.(Px \to \psi)$.

Theorem 5.1 *The satisfiability problem of $HL(@, \downarrow)$ on the class of models K_κ is*

1. NP*-complete, for $\kappa = 1, 2$*

2. NExpTime-*complete, for* $3 \leq \kappa < \omega$.

3. *Recursively enumerable but not decidable, for* $\kappa = \omega$

4. *Co-recursively enumerable but not decidable, for* $\kappa > \omega$

Proof.

1. The lower bound follows from the NP-hardness of propositional satisfiability. The upper bound is proved by establishing the polynomial size model property.

   For $\kappa = 1, 2$, every κ-satisfiable $HL(@, \downarrow)$-formula is satisfiable in a κ-model with at most $O(|\phi|^2)$ nodes. For, suppose $\mathfrak{M}, w \models \phi$ for some κ-model $\mathfrak{M} = (W, R, V)$. Let $W' \subseteq W$ consist of all worlds that are reachable from w or from a world named by one of the nominals occurring in ϕ in at most $md(\phi)$ steps, where $md(\phi)$ is the modal depth of ϕ. Let $\mathfrak{M}'$ be the submodel of $\mathfrak{M}$ with domain W'. Clearly, $\mathfrak{M}'$ is a κ-model and $\mathfrak{M}'$ satisfies the cardinality requirements. Furthermore, a straightforward induction argument shows that $\mathfrak{M}', w \models \phi$.

   This leads to a non-deterministic polynomial time algorithm for testing satisfiability of an $HL(@, \downarrow)$-formula ϕ on κ-models, for $\kappa = 1, 2$. The algorithm first non-deterministically chooses a candidate model $(\mathfrak{M}, w)$ of size $O(|\phi|^2)$, and then it tests whether $\mathfrak{M}, w \models \phi$ and $\mathfrak{M} \in \mathsf{K}_\kappa$. The latter tests can be performed in polynomial time using a top down model checking algorithm (cf. Theorem 6.1 below).

2. **[Upper bound]** For $3 \leq \kappa < \omega$, every formula satisfiable on a κ-model is satisfiable on a κ-model with at most $O(|\phi| \cdot \kappa^{md(\phi)})$ nodes. For, suppose $\mathfrak{M}, w \models \phi$ for some κ-model $\mathfrak{M} = (W, R, V)$. Let $W' \subseteq W$ consist of all worlds that are reachable from w or from a world named by one of the nominals occurring in ϕ in at most $md(\phi)$ steps. Let $\mathfrak{M}'$ be the submodel of $\mathfrak{M}$ with domain W'. Note that the cardinality of $\mathfrak{M}'$ is $O(|\phi| \cdot \kappa^{|\phi|})$, and $\mathfrak{M}'$ is still a κ-model. Furthermore, a straightforward induction argument shows that $\mathfrak{M}', w \models \phi$.

   This leads to a non-deterministic ExpTime algorithm for testing satisfiability of an $HL(@, \downarrow)$-formula ϕ on κ-models. The algorithm first non-deterministically chooses a candidate model $(\mathfrak{M}, w)$ of size $O(|\phi| \cdot \kappa^{|\phi|})$, and then tests whether $\mathfrak{M}, w \models \phi$. The latter test can be performed in time $O(|\mathfrak{M}|^{|\phi|})$ [13], which is $O((|\phi| \cdot \kappa^{|\phi|})^{|\phi|}) = O(|\phi|^{|\phi|} \cdot \kappa^{(|\phi|^2)})$.

   [Lower bound] Consider monadic first-order formulas without equality, i.e., first-order formulas containing unary predicates only, without equality. Any such satisfiable formula ϕ of length n has a model with at most 2^n nodes, and the satisfiability problem for such formulas is NExpTime-complete [8, Section 6.2.1]. We will reduce this problem to the satisfiability problem for $HL(@, \downarrow)$-formulas on κ-models (for $3 \leq \kappa < \omega$), thus showing that the latter problem is NExpTime-hard.

Fix a nominal i, and for any monadic first-order formula ϕ without equality, define ϕ^+ inductively, such that $(x = y)^+ = @_x y$, $(Px)^+ = @_x p$, $(\cdot)^+$ commutes with the Boolean connectives and $(\exists x.\psi)^+ = @_i \Diamond^{|\phi|} {\downarrow} x.\psi$. In words, ϕ^+ states that ϕ holds in the submodel consisting of all points reachable from the point named i in exactly $|\phi|$ many steps. In general, there can be up to $(\kappa - 1)^{|\phi|}$ many points reachable from the point named i in exactly $|\phi|$ many steps (in particular, this will be the case if the submodel generated by i is a $(\kappa - 1)$-ary tree). It follows that ϕ is satisfiable iff ϕ is satisfiable in a model with at most $2^{|\phi|}$ nodes iff ϕ^+ is satisfiable in a κ-model, for $\kappa \geq 3$.

3. We will provide polynomial reductions between this problem and the finite satisfiability problem for first-order logic. The satisfiability problem for first-order logic on finite models is Σ^0_1-complete, even in the case with only a single, binary relation [8, Section 3.2].

   Trivially, if an $HL(@, \downarrow)$-formula is satisfiable in a finite model, it in a ω-model. Conversely, if an $HL(@, \downarrow)$-formula is satisfiable in an ω-model then is satisfiable in a finite model, since the modal depth of the formula provides a bound on the depth of the model. Hence, the satisfiability problem of $HL(@, \downarrow)$ on ω-models reduces (by the Standard Translation) to the satisfiability problem for first-order logic on finite models.

   Conversely, the finite satisfiability problem for first-order logic can be reduced to satisfiability of $HL(@, \downarrow)$ on ω-models. Fix a nominal i, and for any first-order formula ϕ, define ϕ^+ inductively, such that $(x = y)^+ = @_x y$, $(Rxy)^+ = @_x \Diamond y$, $(\cdot)^+$ commutes with the Boolean connectives and $(\exists x.\psi)^+ = @_i \Diamond {\downarrow} x.\psi^+$. In words, ϕ^+ states that ϕ holds in the submodel consisting of the successors of the point named i. It follows that ϕ is satisfiable in a finite model iff the $HL(@, \downarrow)$-formula ϕ^+ is satisfiable on an finitely branching ω-model.

4. By the Löwenheim-Skolem theorem, a first-order formula is satisfiable if and only if it is satisfiable on a finite or countably infinite model. Since $HL(@, \downarrow)$ is a fragment of first-order logic, the Löwenheim-Skolem theorem also applies to $HL(@, \downarrow)$-formulas. It follows that the satisfiability problem for $HL(@, \downarrow)$ on countably branching models coincides with the general satisfiability problem of $HL(@, \downarrow)$, which is Π^0_1 complete by Theorem 3.1.

QED

Theorem 5.2 *The satisfiability problem of first-order sentences of the correspondence language on* K_κ *is*

1. NExpTime *complete, for* $\kappa = 1$
2. *decidable but not elementary recursive, for* $\kappa = 2$
3. *Co-recursively enumerable but not decidable, for* $3 \leq \kappa < \omega$

4. *Σ^1_1-hard, and hence neither recursively enumerable nor co-recursively enumerable, for $\kappa = \omega$*

5. *Co-recursively enumerable but not decidable, for $\kappa > \omega$*

PROOF.

1. This case coincides with the satisfiability problem for monadic first-order logic (on 1-models, every formula of the form Rst is equivalent to $\bot$), which is known to be NEXPTIME complete [8].

2. Consider the satisfiability problem for first-order logic with one unary function symbol, an arbitrary number of unary relation symbols and equality ("the Rabin class"). This problem is decidable, but not elementary recursive [8]. We will provide reductions between this problem and the satisfiability problem for first-order logic on 2-models.

   - Let ϕ be any first-order formula containing one unary function symbol f and any number of unary relation symbols and equality. Let R be a binary relation symbol, and let ϕ_R be obtained from ϕ by repeatedly applying the rewrite rules
     - replace atomic formulas of the form $Pf(t)$ by $\exists x.(Rtx \wedge Px)$
     - replace atomic formulas of the form $f(s) = t$ or $t = f(s)$ by $\exists x.(Rsx \wedge x = t)$

     until the function symbol f does not occur in the formula anymore (in case of nested function symbols, the above rules might need to be applied several times). It is not hard to see that ϕ is satisfiable iff $\phi_R \wedge \forall x \exists y.Rxy$ is satisfiable on a 2-model.
   - Let ϕ be any first-order formula with one binary relation symbol R and any number of unary relation symbols. Let f be a unary function symbol and let P be a new unary relation, and let ϕ_f be the result of replacing all subformulas of ϕ of the form Rst by $Ps \wedge (t = fs)$. Intuitively, the unary predicate P represents the existence of a successor, and the unary function f encodes the successor of a node, if it exists. One can easily see that ϕ is satisfiable on a 2-model iff ϕ_f is satisfiable (simply let R denote the graph of f, or vice versa).

   It follows that the satisfiability problem of first-order logic on 2-models is decidable but not elementary recursive.

3. It is known that the satisfiability problem for first-order sentences with a single binary relation R is Π^0_1-complete [8]. For any such first-order formula ϕ define ϕ^* as follows:

$$\begin{array}{lcl}
(x = y)^* & = & x = y \\
(Rxy)^* & = & \exists x'y'.(\neg Rx'x' \wedge \neg Ry'y' \wedge Rx'y' \wedge Rx'x \wedge Ry'y) \\
(\neg\phi)^* & = & \neg\phi^* \\
(\phi \wedge \psi)^* & = & \phi^* \wedge \psi^* \\
(\exists x.\phi)^* & = & \exists x(Rxx \wedge \phi^*)
\end{array}$$

We claim that ϕ is satisfiable in a model $\mathfrak{M}$ iff ϕ^* is satisfiable on a 3-model $\mathfrak{M}'$. Intuitively, the reflexive nodes of $\mathfrak{M}'$ will correspond to the nodes of $\mathfrak{M}$, and the irreflexive nodes of $\mathfrak{M}'$ will be used to encode the binary relation of $\mathfrak{M}$: we think of reflexive points d, e as standing in the binary relation iff there are irreflexive points d', e' such that $(d', d) \in R$, $(d', e') \in R$ and $(e', e) \in R$. More precisely, the argument can be spelled out as follows.

[⇒] Suppose $\mathfrak{M} \models \phi$, with $\mathfrak{M} = (D, R)$. Let D' be a set of objects obtained from D by adding by adding new objects $(d, e)_1$ and $(d, e)_2$ for all $d, e \in D$. Let $R' = \{(d, d), ((d, e)_1, d), ((d, e)_2, d) \mid d \in D\} \cup \{((d, e)_1, (d, e)_2) \mid (d, e) \in R\}$. The model (D', R') is a 3-model, and by induction on can easily show that $\mathfrak{M}' \models \phi^*$.

[⇐] Suppose $\mathfrak{M} \models \phi^*$ for some 3-model $\mathfrak{M} = (D, I)$. Let $D' = \{d \in D \mid (d, d) \in R\}$. Let $R' = \{(d, e) \in (D')^2 \mid (d', d') \notin R$ and $(e', e') \notin R$ and $(d', d) \in R$ and $(e', e) \in R$ and $(d', e') \in R$, for some $d', e' \in D\}$. Let $\mathfrak{M}' = (D', R')$. A straightforward induction shows that $\mathfrak{M}' \models \phi$.

For $3 < \kappa < \omega$, it follows that a first-order formulas ϕ with one binary relation R is satisfiable iff $\phi^* \wedge \forall x \exists^{\leq 2} y.Rxy$ is satisfiable on a κ-model. Hence, satisfiability of first-order formulas on κ-models is Π^0_1-hard. Finally, membership of Π^0_1 follows from the fact that the satisfiability problem for first-order formulas is in Π^0_1, since ϕ is satisfiable on a κ-model iff $\phi \wedge \forall x \exists^{\leq \kappa} y.Rxy$ is satisfiable.

4. We will provide reductions between that the satisfiability problem for first-order formulas on ω-models and the problem of deciding whether an existential second order sentence holds in the model $(\mathbb{N}, <)$. This proves the result, since the latter problem is Σ^1_1-complete [9].

   Let $\phi_{(\mathbb{N},>)}$ be a first-order sentence expressing that R is a strict linear order and $\forall x \exists y.Ryx$. Then a finitely branching model satisfies $\phi_{(\mathbb{N},>)}$ precisely if the model is isomorphic to $(\mathbb{N}, >)$. For any existential second order sentence $\phi = \exists R_1 \ldots R_n.\psi(R_1, \ldots, R_n, >)$, let ϕ^* be the defined as follows, where $P_1, \ldots, P_n, N$ are new, distinct unary predicates.

$$\begin{array}{lcl}
(x = y)^* & = & x = y \\
(x > y)^* & = & Rxy \\
(R_k x_1 \ldots x_n)^* & = & \exists y_1 \ldots y_n.\big(\bigwedge_{m=1\ldots n}(P_k y_m \wedge R y_m x_m) \wedge \\
 & & \qquad \bigwedge_{m=1\ldots n-1}(R y_m y_{m+1})\big) \\
(\neg\phi)^* & = & \neg\phi^* \\
(\phi \wedge \psi)^* & = & \phi^* \wedge \psi^* \\
(\exists x.\phi)^* & = & \exists x(Nx \wedge \phi^*)
\end{array}$$

   We claim that $(\mathbb{N}, >) \models \phi$ iff $\phi^* \wedge \phi^N_{(\mathbb{N},>)}$ is satisfiable in a finitely branching model, where $\phi^N_{(\mathbb{N},>)}$ is the result of relativising all quantifiers in $\phi_{(\mathbb{N},>)}$ by

N. This can be seem as follows. The submodel consisting of the points satisfying N is the "intended model", while the elements satisfying one of the unary predicates P_k are only used to encode which tuples stand in the R_k relation. More specifically, a tuple $(d_1, \dots, d_n)$ of points satisfying N is thought to stand in the R_k relation iff there are points $e_1, \dots, e_n$ satisfying P_k such that $e_m R d_m$ for all $m \leq n$ and $e_m R e_{m+1}$ for all $m < n$. We will omit the details of the proof here.

Now for the other direction. First, observe that whenever a first-order formula has a finitely branching model $\mathfrak{M}$, then it has a countable such model (indeed, it suffices to take any countable elementary submodel of $\mathfrak{M}$). Now, for any first-order formula $\phi(R, P_1, \dots, P_n)$, let ϕ' be the existential second order sentence $\exists R, P_1, \dots, P_n.(\phi \wedge \forall x \exists y \forall z.(Rxz \rightarrow z < y))$. Observe how, on the natural numbers, the second conjunct enforces that each point has only finite many R-successors). It follows that ϕ is satisfiable in a countable ω-model iff ϕ' is true in a submodel of $(\mathbb{N}, <)$. The latter in turn holds iff $\exists Q.(\phi')^Q$ is true in $(\mathbb{N}, <)$, where $(\phi')^Q$ is the result of relativising all quantifiers in ϕ' by Q.

5. By the Löwenheim-Skolem theorem, a first-order formula is satisfiable if and only if it is satisfiable on a finite or countably infinite model. Hence, the satisfiability problem on countably branching models coincides with the general satisfiability problem, which is known to be Π^0_1-complete [8].

QED

6 Model checking

So far, we only studied the satisfiability and the validity problems. It is natural to ask how our syntactic and semantic restrictions affect the complexity of the model checking problem.

Given a hybrid model M, an assignment g, a state w, and a hybrid formula α, the *model checking problem* is to check whether $M, g, w \vDash \alpha$. We will restrict ourselves to hybrid sentences. This is not a limitation, since we can always replace a free variable x by a fresh nominal i_x such that the valuation V of i_x is the state associated to x by the assignment g.

In [13], the authors give a polynomial time model checker for $HL(@, \Diamond^-, E)$. Moreover, they prove that the model checking problem for $HL(@, \downarrow)$ is PSPACE-complete, as is the case for the full first-order correspondence language. This result holds even for formulas without @, nominals, and propositions.

Theorem 6.1 *The model checking problem for $HL(@, \downarrow)$ on κ-models can be solved in polynomial time for $\kappa \leq 2$, and is* PSPACE*-complete for $\kappa \geq 3$.*

PROOF. The first part of the theorem can be proved using a straightforward top-down model checking algorithm. Since each state in the model has at most one successor, the algorithm takes time linear in the length of the input formula.

As for the second part, the proof of PSPACE-hardness of model checking for $HL(@, \downarrow)$ given in [13] uses a model with out-degree 2. It follows that the model checking problem for $HL(@, \downarrow)$ on κ-models, with $\kappa \geq 3$, is PSPACE-complete. QED

On the contrary, model checking for $HL(@, E, \downarrow)$ and for first-order logic is PSPACE-complete even on 1-models [13].

In the following, we investigate how to restrict the syntax of hybrid languages in order to lower down the complexity of model checking. A first result is that, if formulas do not show the $\downarrow\Box\downarrow$ pattern, then the model checking problem drops from PSPACE-complete to NP-complete.

Theorem 6.2 *The model checking problem for $FHL \setminus \downarrow\Box\downarrow$ is* NP*-complete.*

PROOF. To prove NP-hardness, we embed the satisfiability problem for propositional formulas (SAT) into the model checking problem for $HL \setminus \downarrow\Box\downarrow$. Let $\phi(p_1, \ldots, p_n)$ be any propositional formula, and let $M = (W, R, V)$, where $W = \{0, 1\}$, $R = W \times W$. For each p_k occurring in ϕ, pick a corresponding state variable x_k. Furthermore, let y be a state variable distinct from all $x_1, \ldots, x_n$. Let ϕ' be obtained from ϕ by replacing each occurrence of p_k by $\Diamond(x_k \wedge y)$, for $k = 1 \ldots n$. Intuitively, the two states of M represent truth and falsity, and among these two states the variable y denotes the truth state. It is easily seen that the propositional formula ϕ is satisfiable iff $\Diamond\downarrow y\Diamond\downarrow x_1\Diamond\downarrow x_2 \ldots \Diamond\downarrow x_n.\phi'$ is true in M (at any of the nodes $0, 1$). The latter formula contains no $\Box$ operators, and hence belongs to $FHL \setminus \downarrow\Box\downarrow$.

To prove membership of NP, we give a nondeterministic algorithm that solves the model checking problem in polynomial time. Let α be an $FHL \setminus \downarrow\Box\downarrow$ sentence, $M = (W, R, V)$ be a model and $v \in W$. Replace each subformula of α of the form $\downarrow x.\varphi$ by $\exists x.(x \wedge \varphi)$, and apply the equivalences given in the proof of Theorem 4.1 in order to move the existential quantifiers out of the scope of as many connectives as possible. The resulting sentence α' is equivalent to α and has the following properties:

1. α' is built up from literals (i.e., formulas of the form $(\neg)p$, $(\neg)i$ or $(\neg)x$) using conjunction, disjunction, existential operators ($\Diamond, \Diamond^-, E$), universal operators ($\Box, \Box^-, A$) and existential quantifiers.

2. All existential quantifiers in α' either immediately follow a universal operator (e.g., as in $\Box\exists x_1 \ldots x_n\beta$) or occur at the start of the formula.

3. For all subformulas of α' of the form $X\exists x_1 \ldots x_n\beta$, with X a universal operator, β contains no free variables besides $x_1, \ldots, x_n$.

List all subformulas of α' of the form $X\beta$, with X a universal operator and $\beta = \exists x_1 \ldots \exists x_m.\gamma(x_1 \ldots x_m)$, in order of increasing length, and do the following for each:

Create a new proposition symbol p_β and replace β by p_β in α'. For each state $w \in W$, check whether $M, w \vDash \beta$, and if so, insert the state w in $V(p_\beta)$.

Finally, apply the usual model checking algorithm to check if in polynomial time if v satisfies the resulting $HL(@, \Diamond^-, E)$ formula. If so, return true, else return false.

The nondeterminism is hidden in the test $M, w \vDash \beta$ in step 3. To check $M, w \vDash \exists x_1 \ldots \exists x_m . \gamma(x_1 \ldots x_m)$, the algorithm guesses an assignment g for the variables $x_1, \ldots, x_n$ and checks whether $M, g, w \vDash \gamma(x_1 \ldots x_m)$. Since γ does not contain any existential quantifiers (the subformulas were processed in order of increasing length), it belongs to $HL(@, \Diamond^-, E)$. Hence, the check whether $M, g, w \models \gamma$ can be performed in polynomial time. All in all, our model checking algorithm runs in nondeterministic polynomial time. QED

Notice that the NP-hardness holds even for formulas without proposition letters, nominals and @-operators. Also note that both $FHL \setminus \Box\downarrow$ and $FHL \setminus \downarrow\Box$ are subsets of $FHL \setminus \downarrow\Box\downarrow$. Hence, the model checking for both $FHL \setminus \Box\downarrow$ and $FHL \setminus \downarrow\Box$ is NP-complete. A typical example of a formula to which Theorem 6.2 does not apply is $\downarrow x.\Box\Box\downarrow y.@_x\Diamond y$, which expresses a local form of transitivity.

In Section 4, we saw that $FHL \setminus \Box\downarrow\Box$ has a decidable satisfiability problem. We leave it as an open question whether the model checking complexity of that fragment also below PSpace (since the SAT problem can be embedded into the model checking problem for $FHL \setminus \Box\downarrow\Box$ as done in the proof of Theorem 6.2, the problem is at least NP-hard). Conversely, the fragment $FHL \setminus \downarrow\Box\downarrow$ for which we have just proved that the model checking problem is NP-complete, has an undecidable satisfiability problem: it suffices to note that the encoding of the tiling problem given in Section 3 does not make use of $\downarrow\Box\downarrow$-formulas.

We conclude this section with a hierarchy of fragments of the full hybrid language with $\downarrow$ binder that admits polynomial time model checking. As we remarked already in Section 2, if a hybrid formula α has width w, then $ST_x(\alpha)$ has width at most $w + 2$. Hence, a bound on the width of the hybrid formulas implies a bound on the width of the standard translations. Moreover, model checking for first-order formulas using a bounded number of variables can be performed in polynomial time [21]. It is known that first-order formulas of a bounded width can be rewritten using a bounded number of variables (cf. [11] for an explicit proof). Thus, we obtain the following.

Theorem 6.3 *The model checking problem for formulas of the full hybrid language of bounded width can be solved in polynomial time.*

Proof. Let M be a hybrid model with n nodes and α be a formula of length k and width w. Applying the Standard Translation to α, we obtain $ST_x(\alpha)$, a first-order formula with width at most $w + 2$. The Standard Translation can be implemented in linear time $O(k)$. Each first-order formula of width w can be translated in quadratic time $O(k^2)$ into a formula using at most w variables [11]. Finally, model checking a first-order formula of length k containing v variables on a model of n nodes costs $O(k \cdot n^v)$ [21]. Hence, we can model check α in time $O(k^2 + k \cdot n^{w+2})$. This is polynomial since w is constant. QED

Notice that a formula of bounded width can use an arbitrary number of variables and can have an arbitrary nesting degree of $\downarrow$. For instance, let $\alpha_0 = x_1$, and, for $n > 0$, let $\alpha_n = E{\downarrow}x_n.(\Diamond x_{n+1} \wedge \alpha_{n-1})$. For $n > 0$, the formula α_n says that there are points $x_1, \ldots, x_{n+1}$ such that x_i reaches x_{i+1} for $i = 1, \ldots, n$. It is easy to see that, for $n > 0$, the width of α_n is 2. Moreover, α_n uses $n + 1$ variables and the nesting degree of $\downarrow$ in α_n is n.

7 Conclusion

In this paper, we described two ways to tame the hybrid logic $HL(@, \downarrow)$, and in fact the full hybrid language $HL(@, E, \downarrow, \Diamond^-)$. By taming a logic we mean restricting it in such a way that it becomes decidable. These two ways are:

1. Restricting the syntax by excluding formulas containing the pattern $\Box{\downarrow}\Box$.
2. Restricting the class of models by assuming a bound on the branching degree of the models.

Furthermore, we showed that similar restrictions can be used to lower the complexity of model checking task for these logics.

Some results in this paper show that, under certain natural conditions, $H(@, \downarrow)$ behaves better than the first-order correspondence language, computationally speaking. Incidentally, the full hybrid language $HL(@, E, \downarrow, \Diamond^-)$ has the same expressive power as the full first-order correspondence language, as is shown by the following translation [5]:

$$\begin{array}{lcl}
HT(x = y) & = & @_x y \\
HT(Px) & = & @_x p \\
HT(Rxy) & = & @_x \Diamond y \\
HT(\neg\phi) & = & \neg HT(\phi) \\
HT(\phi \wedge \psi) & = & HT(\phi) \wedge HT(\psi) \\
HT(\exists x.\phi) & = & E{\downarrow}x.HT(\phi)
\end{array}$$

The last clause of this translation shows that, in some sense, the first-order quantifier $\exists x$ consist of two parts, namely the *picking a state of the model* part, which is captured by the global modality, and the *variable binding* part, which is captured by the $\downarrow$. The syntax of $HL(@, E, \downarrow, \Diamond^-)$ allows us to distinguish these two parts. Hence, one could say that some of our results identify computationally tractable fragments of first-order logic that can only be distinguished once these two parts of the quantifiers are split. In this sense, our paper can be seen as a fine study of the structure of first-order quantifiers.

By analogy to the study of decidable quantifier prefix classes [], one could also view our decidability result for $FHL \setminus \Box{\downarrow}\Box$ from a more systematic perspective. For any sequence σ of elements of $\{\Box, \Diamond, \downarrow, @\}$ (where $\Box$ and $\Diamond$ now stand for a sequence of universal resp. existential operators), one could consider the fragment $FHL \setminus \sigma$. Then it follows from the undecidability proof by tiling in Section 3 that there is no such sequence σ that contains $\Box\!\downarrow\!\Box$ as a proper subsequence and such that $FHL \setminus \sigma$ is decidable. In other words, our decidability result is optimal.

Finally, the outcomes of our investigation show once more that, from a computational point of view, the satisfiability problem and the model checking problem for a logic are sensitive to different sources of complexity. Restricting the width (i.e., the out-degree) of the model makes the satisfiability problem decidable, but it does not lower the complexity of the model checking problem (unless the width is less than two). On the other hand, restricting the width of the formula makes the model checking problem more tractable, but it does not affect the undecidability of the satisfiability problem (unless the width is 0).

References

[1] H. Andréka, J. van Benthem, and I. Németi. Modal logics and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[2] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez Artalejo, editors, *Proceedings of the 8th Annual Conference of the EACSL*, Madrid, 1999.

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

[4] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation, and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

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

[6] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2001.

[7] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4:251–272, 1995.

[8] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Springer, Berlin, 1997.

[9] B. ten Cate. *Model theory for Extended Modal Languages.* PhD thesis, ILLC, University of Amsterdam, 2005.

[10] B. ten Cate. Interpolation for extended modal languages. *Journal of Symbolic Logic*, To appear. A preliminary version is available from `http://staff.science.uva.nl/~bcate`

[11] B. ten Cate and M. Franceschet. Guarded fragments with constants. Technical Report PP-2004-32, ILLC, University of Amsterdam, 2004.

[12] M. Fisher and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[13] M. Franceschet and M. de Rijke. Model checking for hybrid logics. In *Proceedings of the Workshop Methods for Modalities*, 2003.

[14] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language, and Information*, 5(1):1–24, 1996.

[15] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.

[16] E. Grädel and M. Otto. On logics with two variables. *Theoretical computer science*, 224(1-2):73–113, 1999.

[17] D. Harel. Recurring dominoes: making the highly undecidable highly understandable. *Annals of Discrete Mathematics*, 24:51–72, 1985.

[18] E. Hoogland. *Definability and Interpolation*. PhD thesis, University of Amsterdam, 2001.

[19] M. Marx. Narcissists, stepmothers and spies. In *Proceedings of the International Workshop on Description Logics*, 2002.

[20] M. Mortimer. On languages with two variables. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

[21] M. Y. Vardi. On the complexity of bounded-variable queries. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 266–276, 1995.