

Hybrid Logics

Massimo Franceschet and Balder ten Cate

Course notes for the ESSLLI 2004 course

Abstract

Hybrid languages extend traditional modal and temporal languages by providing mechanisms for naming states and referring to states by their names. These mechanisms increase the expressive power of the languages while preserving, in some cases, their nice computational and model-theoretic properties.

This course provides an introduction to hybrid logic, with special emphasis on expressivity and reasoning. We start the course with a gentle introduction to hybrid logic, explaining the motivation behind it and discussing sound and complete proof systems that have been introduced in the literature (in particular tableaux). After this introductory part (first class), the course is divided in two main parts.

- The first part concerns the model-theoretic properties of hybrid logic. The topics we will discuss include expressivity, frame definability and interpolation. Each of these notions will be explained, and no substantial background knowledge on model theory is required (Classes 2 and 3).
- The second part concerns the computational behaviour of hybrid logic. It includes a survey of decidability and complexity of the satisfiability and model checking problem for various hybrid languages. Finally, an example of a concrete application for hybrid logic model checking is presented: the evaluation of queries and constraints for semistructured data (Classes 4 and 5).

This reader contains relevant reading material. More details about the course and slides will be made available on the course website in due time.

Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto

PATRICK BLACKBURN, *Computerlinguistik, Universität des Saarlandes, 66041 Saarbrücken, Germany.*
E-mail: patrick@coli.uni-sb.de

Abstract

This paper is about the good side of modal logic, the bad side of modal logic, and how hybrid logic takes the good and fixes the bad.

In essence, modal logic is a simple formalism for working with relational structures (or multi-graphs). But modal logic has no mechanism for referring to or reasoning about the individual nodes in such structures, and this lessens its effectiveness as a representation formalism. In their simplest form, hybrid logics are upgraded modal logics in which reference to individual nodes is possible.

But hybrid logic is a rather unusual modal upgrade. It pushes one simple idea as far as it will go: represent *all* information as formulas. This turns out to be the key needed to draw together a surprisingly diverse range of work (for example, feature logic, description logic and labelled deduction). Moreover, it displays a number of knowledge representation issues in a new light, notably the importance of sorting.

Keywords: Labelled deduction, description logic, feature logic, hybrid logic, modal logic, sorted modal logic, temporal logic, nominals, knowledge representation, relational structures

1 Modal Logic and Relational Structures

To get the ball rolling, let's recall the syntax and semantics of (propositional) multi-modal logic.

Definition 1.1 (Multimodal languages) *Given a set of propositional symbols $PROP = \{p, q, p', q', \dots\}$, and a set of modality labels $MOD = \{\pi, \pi', \dots\}$, the set of well-formed formulas of the multimodal language (over $PROP$ and MOD) is defined as follows:*

$$WFF := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle\pi\rangle\varphi \mid [\pi]\varphi,$$

for all $p \in PROP$ and $\pi \in MOD$. As usual, $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Definition 1.2 ((Kripke) models) *Such a language is interpreted on models (often called Kripke models). A model $\mathcal{M}$ (for a fixed choice of $PROP$ and MOD) is a triple $(W, \{R_\pi \mid \pi \in MOD\}, V)$. Here W is a non-empty set (I'll call its elements states, or nodes), and each R_π is a binary relations on W . The pair $(W, \{R_\pi \mid \pi \in MOD\})$ is called the frame underlying $\mathcal{M}$, and $\mathcal{M}$ is said to be a model based on this frame. V (the valuation) is a function with domain $PROP$ and range $Pow(W)$; it tells us at which states (if any) each propositional symbol is true.*

Definition 1.3 (Satisfaction and validity) *Interpretation is carried out using the Kripke satisfaction definition. Let $\mathcal{M} = (W, \{R_\pi \mid \pi \in MOD\}, V)$ and $w \in W$. Then:*

$\mathcal{M}, w \Vdash p$	iff	$w \in V(p)$, where $p \in PROP$
$\mathcal{M}, w \Vdash \neg\varphi$	iff	$\mathcal{M}, w \not\Vdash \varphi$
$\mathcal{M}, w \Vdash \varphi \wedge \psi$	iff	$\mathcal{M}, w \Vdash \varphi$ and $\mathcal{M}, w \Vdash \psi$
$\mathcal{M}, w \Vdash \varphi \vee \psi$	iff	$\mathcal{M}, w \Vdash \varphi$ or $\mathcal{M}, w \Vdash \psi$
$\mathcal{M}, w \Vdash \varphi \rightarrow \psi$	iff	$\mathcal{M}, w \not\Vdash \varphi$ or $\mathcal{M}, w \Vdash \psi$
$\mathcal{M}, w \Vdash \langle\pi\rangle\varphi$	iff	$\exists w'(wR_\pi w' \ \& \ \mathcal{M}, w' \Vdash \varphi)$
$\mathcal{M}, w \Vdash [\pi]\varphi$	iff	$\forall w'(wR_\pi w' \Rightarrow \mathcal{M}, w' \Vdash \varphi)$.

If $\mathcal{M}, w \Vdash \varphi$ we say that φ is satisfied in $\mathcal{M}$ at w . If φ is satisfied at all states in all models based on a frame $\mathcal{F}$, then we say that φ is valid on $\mathcal{F}$ and write $\mathcal{F} \Vdash \varphi$. If φ is valid on all frames, then we say that it is valid and write $\Vdash \varphi$.

Now, you've certainly seen these definitions before — but if you want to understand contemporary modal logic you need to think about them in a certain way. Above all, please *don't* automatically think of models as a collection of “worlds” together with various “accessibility relations between worlds”, and *don't* think of modalities as “non-classical logical symbols” suitable only for coping with intensional concepts such as necessity, possibility, and belief. Modal logic can be viewed in these terms, but it's a rather limited perspective. Instead, *think of models as relational structures*, or *multi-graphs*. That is, think of a model as an underlying set together with a collection of binary and unary relations. We use the modalities to talk about the binary relations, and the propositional symbols to talk about the unary relations.

Remark 1.4 (Kripke models are relational structures) *Let's make this precise. Consider a model $\mathcal{M} = (W, \{R_\pi \mid \pi \in MOD\}, V)$. The underlying frame $(W, \{R_\pi \mid \pi \in MOD\})$ is already presented in explicitly relational terms, and it is trivial to present the information in the valuation in same way: in fact $\mathcal{M}$ can be presented as the following relational structure $\mathcal{M} = (W, \{R_\pi \mid \pi \in MOD\}, \{V(p) \mid p \in PROP\})$.*

Why think in terms of relational structures? Two reasons. The first is: relational structures are ubiquitous. Virtually all standard mathematical structures can be viewed as relational structures, as can inheritance hierarchies, transition systems, parse trees, and other structures used in AI, computer science, and computational linguistics. Indeed, anytime you draw a diagram consisting of nodes, arcs, and labels, you have drawn some kind of relational structure. There are no preset limits to the applicability of modal logic: as it is a tool for talking about relational structures, it can be applied just about *anywhere*.

Secondly, relational structures are the models of *classical model theory* (see, for example, Hodges [35]). Thus there is nothing intrinsically “modal” about Kripke models, and we're certainly not forced to talk about them using modal languages. On the contrary, we can talk about models using *any* classical language we find useful (for example, a first-order, infinitary, fixpoint, or second-order language). Unsurprisingly, this means that modal and classical logic are systematically related.

Remark 1.5 (Modal logic is a fragment of classical logic) *To talk about a Kripke model in a classical language, all we have to do is view it as a relational structure (as described in the previous example) and then ‘read off’ from the signature (that is,*

MOD and *PROP*) the non-logical symbols we need, namely a *MOD*-indexed collection of two place relation symbols R_π , and a *PROP*-indexed collection of unary relation symbols P, Q, P', Q' , and so on. We then build formulas in the classical language of our choice.

As modal languages and classical languages both talk about relational structures, it seems overwhelmingly likely that a systematic relationship exists between them. And in fact, the modal language (over *PROP* and *MOD*) can be translated into the best-known classical language of all, namely the first-order language (over *PROP* and *MOD*). Here are some clauses of the Standard Translation, a top-down translation which inductively maps modal to first-order formulas:

$$\begin{aligned} ST_x(p) &= P(x), p \in \text{PROP} \\ ST_x(\neg\varphi) &= \neg ST_x(\varphi) \\ ST_x(\varphi \wedge \psi) &= ST_x(\varphi) \wedge ST_x(\psi) \\ ST_x(\langle\pi\rangle\varphi) &= \exists y(xR_\pi y \wedge ST_y(\varphi)) \\ ST_x([\pi]\varphi) &= \forall y(xR_\pi y \rightarrow ST_y(\varphi)). \end{aligned}$$

Here x is a fixed but arbitrary free variable. In the fourth and fifth clause, the variable y can be any variable not used so far in the translation. The clauses governing ST_y are analogous to those given for ST_x ; in particular, the clauses for the modalities introduce a new variable (say z) and so on. For any modal formula φ , $ST_x(\varphi)$ is a first-order formula containing exactly one free variable (namely x), and it is easy to see that $\mathcal{M}, w \models \varphi$ iff $\mathcal{M} \models ST_x(\varphi)[w]$ (where $\models$ denotes the first-order satisfaction relation and $[w]$ means assign the state w to the free variable x in $ST_x(\varphi)$). The equivalence can be proved by induction, but it should be self-evident: the Standard Translation is simply a reformulation of the clauses of the Kripke satisfaction definition.

There are also non-trivial links between modal logic and infinitary logic, fixed-point logic, and second-order logic; in particular, modal validity is intrinsically second-order. For further discussion, see Blackburn, de Rijke, and Venema [14].

In short, modal logic is not some mysterious non-classical intensional logic, and modalities are not strange new devices. On the contrary, modalities are simply macros that handle quantification over accessible states.

This, of course, leads to another question. OK — so we can use modal logic when working with relational structures — but why bother if it's really just a disguised way of doing classical logic? I think the following two answers are the most important: modal logic brings *simplicity* and *perspective*.

Simplicity comes in a variety of forms. For a start, modal representations are often clean and compact: modalities pack a useful punch into a readable notation. Moreover, modal logic often brings us back to the realms of the computable: while the first-order logic over *MOD* and *PROP* is *undecidable* (whenever *MOD* is non-empty), its modal logic is *decidable* (in fact, PSPACE-complete).

Perspective is more subtle. Modal languages talk about relational structures in a special way: they take an *internal* and *local* perspective on relational structure. When we evaluate a modal formula, we place it *inside* the model, at some particular state w (the *current state*). The satisfaction clause (and in particular, the clause for the modalities) allow us to scan other states for information — but we're only allowed to scan states reachable from the current state. The reader should think of a modal formula as a little automaton, placed at some point on a graph, whose task is to explore

the graph by visiting accessible states. This internal, local, perspective is responsible for many of the attractive mathematical properties of modal logic. Moreover, it makes modal representations ideal for many applications. Here's a classic example:

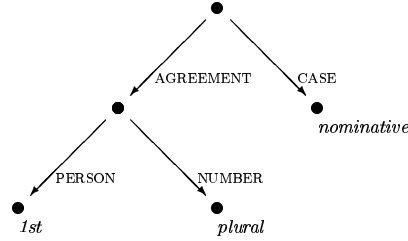
Example 1.6 (Temporal logic) We'll be seeing a lot of the bimodal language with $MOD = \{F, P\}$ in this paper: the modality $\langle F \rangle$ means “at some Future state”, and $\langle P \rangle$ means “at some Past state”. To reflect this temporal interpretation, we usually interpret this language on frames of the form $(T, <)$ that can plausibly be thought of as ‘flows of time’. For example, if we think of time as a branching structure, $(T, <)$ might be some kind of tree, and if we want a linear view of time, $(T, <)$ might be $(\mathbb{Z}, <)$ (the integers in their usual order). When interpreting the language on such frames we insist that R_F is $<$, and R_P is its converse; that is, as required, we ensure that $\langle F \rangle$ looks forward along the flow of time, and $\langle P \rangle$ backwards.

Consider the formula $\langle P \rangle \text{Mia-unconscious}$. This is true iff we can look back in time from the current state and see a state where Mia is unconscious. Similarly $\langle F \rangle \text{Mia-unconscious}$ requires us to scan the states that lie in the future looking for one where Mia is unconscious. Thus these two formulas work similarly to the English sentences Mia has been unconscious and Mia will be unconscious: these sentences don't specify an absolute time for Mia's unconsciousness (which we could do by giving a date and time), rather they locate it relative to the time of utterance. In short, English and other natural languages exploit the fact that human beings live in time, and modal logic models this neatly.

This situated perspective can be lifted to more interesting temporal geometries. For example, we could regard temporal states as unbroken intervals of time, add new modalities such as $\langle \text{SUB} \rangle$ (meaning “at some SUBinterval of the current state”) and $\langle \text{SUP} \rangle$ (meaning “at some SUPERinterval of the current state”). Then a formula of the form $\langle \text{SUB} \rangle \langle F \rangle \langle \text{SUP} \rangle p$ means “by looking down to a subinterval, and then forward to the future, and then up to a superinterval, it is possible to find a state where p is true”. Halpern and Shoham [34] take this idea to its ultimate conclusion: abstracting from the work of James Allen [1], they present a modal logic which allows all possible relationships between two closed intervals over a linear flow of time to be explored ‘from the inside’.

Nowadays, few modal logicians regard modal logic as a non-classical logic, and they certainly don't feel tied to any of the traditional interpretations of modal machinery. On the contrary, since the early 1970s modal logic has been explored as a subsystem of various classical logics, and it is now clear that modal logic are a very special part of classical logic. Indeed, modal languages are in many respects so natural, that — as modal logicians love to point out — it's not particularly surprising that they have been independently reinvented by other research communities that make use of relational structures. Let's look at two well known examples.

Example 1.7 (Feature logic) Feature structures are widely used in unification-based approaches to natural language. In essence, feature structures are multigraphs that represent linguistic information:



Computational linguists have a neat notation for talking about feature structures: Attribute-Value Matrices (AVMs). Here's an example:

$$\left[\begin{array}{l} \text{AGREEMENT} \\ \text{CASE} \end{array} \left[\begin{array}{ll} \text{PERSON} & 1st \\ \text{NUMBER} & plural \\ \neg \text{dative} & \end{array} \right] \right]$$

This AVM is a partial description in the above feature structure — it's satisfied in that structure at the root node. The AVM describes a feature structure in which the AGREEMENT transition leads to a node from which PERSON and NUMBER transitions lead to the information *1st* and *plural* respectively, and if you work down the left hand side of the previous diagram from the root you'll find this structure. The AVM also demands a CASE transition from the root node that does not lead to the information *dative*. The feature structure depicted above also satisfies this requirement, for the CASE transition leads to a node bearing the information *nominative*.

Now, this all sounds very modal — and indeed, the AVM is a notational variant of the following formula:

$$\langle \text{AGREEMENT} \rangle (\langle \text{PERSON} \rangle 1st \wedge \langle \text{NUMBER} \rangle plural) \\ \wedge \langle \text{CASE} \rangle \neg \text{dative}$$

Example 1.8 (Description logic/Terminological logic) In description logic, concept languages are used to build knowledge bases. An important part of the knowledge base is called the *TBox* (or *terminology*). This is a collection of concept macros defined over the primitive concept names using booleans and role names. For example, the concept of being a hired killer for the mob is true of any individual who is a killer and employed by a gangster, and we can define this in the description language *ALC* using the following expression:

$$\text{killer} \sqcap \exists \text{EMPLOYER.gangster}$$

Here *killer* and *gangster* are concept names, *EMPLOYER* is a role name, and $\sqcap$ is a boolean (intersection). This expression means exactly the same thing as the following modal formula:

$$\text{killer} \wedge \langle \text{EMPLOYER} \rangle \text{gangster}$$

Indeed, as Schild [49] pointed out, any *ALC* expression corresponds to a modal formula: simply replace occurrences of $\sqcap$ by $\wedge$, $\sqcup$ by $\vee$, $\exists R$ by $\langle R \rangle$, and $\forall R$ by $[R]$ (both formalisms typically use the symbol $\neg$ to denote boolean-complement/negation, so occurrences of $\neg$ can be left in place). This correspondence lifts to many stronger concept languages: number restrictions correspond to counting modalities, mutually converse roles correspond to mutually converse modalities, and commonly used role constructors (for example, for forming the transitive closure of a role) correspond to the modality constructors of Propositional Dynamic Logic (PDL).

Summing up, modal logic is a well-behaved and intuitively natural fragment of classical logic. Over the past 25 years, modal logicians have explored and extended this fragment in many ways. By introducing modal operators of arbitrary arities, they have made it possible to work with relational structures containing relations of any arity. By evaluating formulas at *sequences* of states (as is done in *multidimensional modal logic*; see Marx and Venema [39]) they have generalized the notion of perspective. By introducing *logical modalities* (see Goranko and Passy [33] and de Rijke [47]) they have shown how to introduce certain forms of *globality* into modal logic while retaining (and in certain respects improving) their desirable properties. Indeed, in recent work on the *guarded fragment* (see Andr  ka, van Benthem, and N  meti [2]) they have shown that it is even possible to “export” the locality intuition back to classical logic; this line of work has unearthed several previously unknown decidable fragments of first-order (and other) classical logics. For a detailed account of contemporary modal logic, see Blackburn, De Rijke, and Venema [14].

So modal logicians have a lot to be proud of. But for all these achievements, something is missing. What exactly?

2 The Trouble with Modal Logic

Carlos Areces summed it up neatly: there is an *asymmetry* at the heart of modal logic. Although states are crucial to Kripke semantics, nothing in modal syntax get to grips with them. This leads to (at least) two kinds of problem. For a start, it means that for many applications modal logic is *not* an adequate representation formalism. Moreover, it makes it difficult to devise usable modal *reasoning* systems.

Example 2.1 (Temporal logic) Although the temporal language with modalities $\langle F \rangle$ and $\langle P \rangle$ neatly captures the perspectival nature of natural language tenses, it fails to get to grips with a linguistic fact of equal importance: many tenses are *referential*. An utterance of Vincent accidentally squeezed the trigger *doesn't* mean that at some completely unspecified past time Vincent did in fact accidentally squeeze the trigger, it means that at some particular, contextually determined, past time he did so. The natural representation, $\langle P \rangle \text{Vincent-accidentally-squeeze-the-trigger}$, fails to capture this.

Similarly, while it's certainly possible to abstract elegant modal logics from the work of James Allen, such abstractions amputate a central feature of his work: reference to specific intervals. Allen's formalism includes the notation $\text{Hold}(P, i)$ meaning “the property *P* holds at the interval *i*”, and **Hold** plays a key role in his approach to temporal knowledge representation. This construction is not present in the modal logic of Halpern and Shoham.

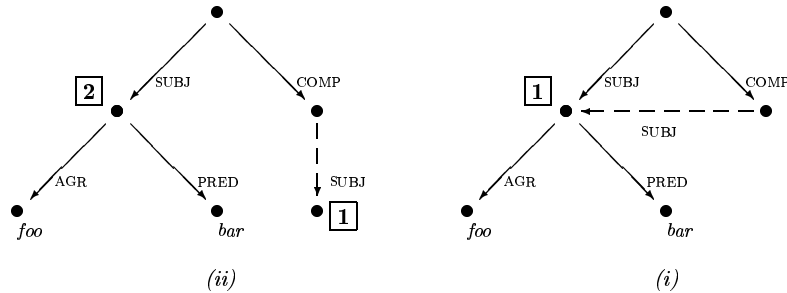
And there are deeper limitations, centered on the notion of validity. Suppose we

are working with the temporal language in $\langle F \rangle$ and $\langle P \rangle$. Can we write down a formula that is valid on every transitive frame, and not valid on any others? That is, can we define transitivity? Sure: $\langle F \rangle \langle F \rangle p \rightarrow \langle F \rangle p$ does so. OK: but can we write down a formula valid on precisely the asymmetric frames (that is, frames $(T, <)$ such that $\forall xy(x < y \rightarrow y \not< x)$)? Try what you like, you won't find any such formula: a central property of flows of time is invisible to modal representations.

Example 2.2 (Feature Logic) While AVM notation is related to modal logic, it offers something new: it lets us name specific nodes in feature structures. Consider the following AVM:

$$\left[\begin{array}{c} \text{SUBJ} \\ \text{COMP} \end{array} \left[\begin{array}{c} \boxed{1} \\ \text{SUBJ} \end{array} \left[\begin{array}{cc} \text{AGR} & \text{foo} \\ \text{PRED} & \text{bar} \end{array} \right] \right] \right]$$

The 'tag' $\boxed{1}$ names a point in the feature structure. This AVM demands that the node we reach by following the SUBJ transition from the root is also the node we reach by first taking a COMP transition from the root and then taking a SUBJ transition. No matter which path we take, we have to end up at the node tagged $\boxed{1}$. For this reason, the AVM is not satisfied on the left hand feature structure below (which otherwise gets everything right) but is satisfied by the right hand feature structure:



Thus AVM notation is *not* a notational variant of ordinary multimodal logic: it's strictly stronger. So are many description logics:

Example 2.3 (Description logic) Description logic lets us reason about specific individuals — in fact, it lets us do so in two distinct ways. First, knowledge bases need not consist of just a TBox — they can also contain an ABox. In the ABox (or assertional component) we specify how properties and roles apply to specific individuals. For example, to assert that Vincent is a gunman we add Vincent.gunman to the ABox, and to insist that Pumpkin loves Honey-Bunny we add $(\text{Pumpkin, Honey-Bunny}).\text{LOVES}$.

Now, the assertional level is a separate level in the knowledge base, so such specifications aren't written in the underlying concept language (in essence they're statements in a constraint language that manipulates formulas of the concept language). But some description languages push matters further: just as feature logic does, they allow reference to individuals to be integrated into the underlying representation formalism itself, thus allowing assertions about individuals to be integrated into the TBox. This is done via the one-of operator $\mathcal{O}$. The notation $\mathcal{O}(\text{Jules}, \dots, \text{Vincent})$ picks out one

of the individuals Jules, $\dots$, Vincent, and $\mathcal{O}(\text{Mia})$ picks out Mia. In short, we now have a concept language rich enough to refer to specific individuals. Such a concept language is not a notational variant of ordinary multimodal logic, or even multimodal logic enriched with (say) counting modalities and PDL-like constructs: it offers a novel form of expressivity.

There is a method (introduced in the late 1960s by Arthur Prior) which allows reference to states to be incorporated into modal logic. But although Prior's idea has attracted a handful of advocates (see the Guide to the Literature at the end of the paper) it's never been part of the modal mainstream. On the other hand, description logicians such as De Giacomo [22] have realized its relevance. This method — hybridization — is central to the paper, and I'll introduce it shortly.

In short, the asymmetry underlying orthodox modal logic translates into obvious weaknesses as a representation formalism. The same asymmetry leads to problems with reasoning. Until recently, modal proof theory was a relatively neglected topic. Traditionally, modal logicians have been content to formulate modal proof systems as Hilbert-style axiomatizations, this being enough to get on with the topics that interested them with a minimum of syntactic fuss. But it resulted in few *usable* modal proof systems available, and little in the way of general proof-theoretical results.

An important exception to this was Fitting's [25] groundbreaking work on *prefixed tableau systems*. Fitting's work can be viewed as a precursor to Gabbay's [26] work on *labelled deduction*. In essence, Gabbay's proposal is to develop a metalinguistic algebra of labels that can act as the motor for modal deduction. Another recent general approach, *display calculus* (see Kracht [37]), though very different from labelled deduction, also makes use of novel metalinguistic machinery. Display calculus is an extension of sequent calculus which introduces additional notation to allow us to freely manipulate object language formulas (in much the same way as a school child rewrites polynomial equations).

Now, first-order proof theory does not require this kind of metalinguistic support. This is because first-order languages are expressive enough to support the key deduction steps at the *object* level. If we find a representation formalism that is *not* capable of doing this, but needs to be augmented by a rich metatheoretic machinery, this is a signal that something is missing. Modal logic seems to be such a formalism — what exactly does it lack?

If we look at the Fitting-Gabbay tradition, an answer practically leaps off the page: we need to be able to deal with states *explicitly*. We need to be able to name them, reason about their identity, and reason about the transitions that are possible between them. In essence, labelled deduction in its various forms supplies metalinguistic equipment for carrying out these tasks, and this leads to modally natural proof systems. In particular, labelled deduction successfully captures the key intuition underlying Kripke semantics, that of a little automaton working its way through a graphlike structure — except that the automaton's *deductive* task is to try and *build* such a structure, not explore a pre-existing one.

Summing up, whether we think about representation or reasoning the conclusion is the same: modal logic's lack of mechanisms for dealing with states explicitly is a genuine weakness.

3 Hybrid Logic

Hybrid languages provide a truly *modal* solution to this problem. Modal logic may not be perfect — but it's certainly a most remarkable fragment of classical logic. How can we add reference to states without destroying it?

Let's go back to basics. Modal logic allows us to form complex formulas out of atomic formulas using booleans and modalities. There's only formulas, nothing else. So if we want to name states and remain modal, we should find a way of naming states using formulas. We can do this by introducing a second sort of atomic formula: *nominals*. Syntactically these will be ordinary atomic formulas, but they will have an important *semantic* property: nominals will be true at exactly *one* point in any model; nominals 'name' this point by being true there and nowhere else. Let's make this idea precise — and improve it in one respect, by adding *satisfaction operators*.

Definition 3.1 (Hybrid multimodal languages) Let NOM be a nonempty set disjoint from $PROP$ and MOD . The elements of NOM are called *nominals*, and we typically write them as i, j, k and l . We define the hybrid multimodal language (over $PROP$, NOM , and MOD) to be the:

$$WFF := i \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle\pi\rangle\varphi \mid [\pi]\varphi \mid @_i\varphi.$$

For any nominal i , we shall call the symbol sequence $@_i$ a *satisfaction operator*.

Remark 3.2 (Nominals and satisfaction operators) As promised, nominals are formulas. What are satisfaction operators? In essence, a simple way of further exploiting the presence of nominals: $@_i\varphi$ means “go to the point named by i (that is, the unique point where i is true) and see if φ is true there”. That is, $@_i\varphi$ is a way of asserting — in the object language — that φ is satisfied at a particular point. Formulas of the form $@_i\varphi$ and $\neg @_i\varphi$ are called *satisfaction statements*.

Definition 3.3 (Hybrid models, satisfaction, and validity) A *hybrid model* is a triple $(W, \{R_\pi \mid \pi \in MOD\}, V)$ where $(W, \{R_\pi \mid \pi \in MOD\})$ is a frame and V is a hybrid valuation. A *hybrid valuation* is a function with domain $PROP \cup NOM$ and range $Pow(W)$ such that for all nominals i , $V(i)$ is a singleton subset of W . We call the unique state in $V(i)$ the *denotation* of i . We interpret hybrid languages on hybrid models by adding the following two clauses to the Kripke satisfaction definition:

$$\begin{aligned} \mathcal{M}, w \Vdash i & \quad \text{iff} \quad w \in V(i), \text{ where } i \in NOM \\ \mathcal{M}, w \Vdash @_i\varphi & \quad \text{iff} \quad \mathcal{M}, w' \Vdash \varphi, \text{ where } w' \text{ is the denotation of } i. \end{aligned}$$

If φ is satisfied at all states in all hybrid models based on a frame $\mathcal{F}$, then we say that φ is *valid* on $\mathcal{F}$ and write $\mathcal{F} \Vdash \varphi$. If φ is valid on all frames, then we say that it is *valid* and write $\Vdash \varphi$.

Remark 3.4 (Hybrid logic is modal) Hybrid languages contain only familiar modal mechanisms: nominals are atomic formulas, and satisfaction operators are actually normal modal operators (that is: for any nominal i , $@_i(\varphi \rightarrow \psi) \rightarrow (@_i\varphi \rightarrow @_i\psi)$ is valid; and if φ is valid, then so is $@_i\varphi$).

Moreover, like multimodal logic, hybrid logic is a fragment of classical logic: indeed, it is easy to extend the Standard Translation to hybrid logic. Divide the first-order

variables into two sets such that one contains the reserved variable x and the variables used to translate familiar modalities, while the other contains a first-order variable x_i for every nominal i . Define:

$$\begin{aligned} ST_x(i) &= x = x_i, i \in NOM \\ ST_x(@_i\varphi) &= (ST_x(\varphi))[x_i/x] \end{aligned}$$

Clearly $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M} \models ST_x(\varphi)[w, V(i), \dots, V(j)]$, where $x, x_i, \dots, x_j$ are the free variables in $ST_x(\varphi)$. Nominals correspond to free variables, and (as the substitution $[x_i/x]$ makes clear) satisfaction operators let us switch our perspective from the current state to named states.

So far, so modal — but what about computational complexity? No change. As Areces, Blackburn and Marx [4] show, hybrid logic is (up to a polynomial) no more complex than multimodal logic: deciding the validity of hybrid formulas is a PSPACE-complete problem.

Remark 3.5 (Hybrid logic is hybrid) Any modal logic is a fragment of classical logic — but hybrid logic takes matters a lot further. The near-atomic satisfaction statement $@_i j$ asserts that the states named by i and j are identical, thus we have incorporated part of the classical theory of equality. Similarly $@_i \langle\pi\rangle j$ means that the state named by j is an R_π -successor of the state named by i , so we've incorporated the classical ability to make assertions about the relations that hold between specific states. Thus hybrid logic is a genuine hybrid: it brings to modal logic the classical concepts of identity and reference.

With this extra classical power at our disposal, it is straightforward to fix the representational problems noted in the previous section.

Example 3.6 (Temporal logic) First, although Vincent accidentally squeezed the trigger can't be correctly represented in the ordinary temporal language in $\langle F \rangle$ and $\langle P \rangle$, it can be with the help of nominals: $\langle P \rangle(i \wedge \text{Vincent-accidentally-squeeze-the-trigger})$ locates the trigger-squeezing not merely in the past, but at a specific temporal state there: the one named by i .

Second, if we want to work with interval-based temporal models, we can now do so in a way that is faithful to the work of James Allen: the satisfaction statement $@_i\varphi$ is a clear analog of Allen's **Hold**(i, φ) construct. More on this in Section 6.

Third, we also solve the deeper issue concerning definability: $i \rightarrow \neg \langle F \rangle \langle F \rangle i$ defines asymmetry (that is, it is valid on all asymmetric frames and no others). More on this in Section 5.

Example 3.7 (Feature logic) Nominals correspond to tags. Consider once more the problematic AVM:

$$\left[\begin{array}{cc} \text{SUBJ} & \boxed{1} \left[\begin{array}{cc} \text{AGR} & \text{foo} \\ \text{PRED} & \text{bar} \end{array} \right] \\ \text{COMP} & [\text{SUBJ} \boxed{1}] \end{array} \right]$$

This corresponds to the following L^N wff:

$$\begin{aligned} & \langle \text{SUBJ} \rangle (i \wedge \langle \text{AGR} \rangle \text{foo} \wedge \langle \text{PRED} \rangle \text{bar}) \\ & \wedge \langle \text{COMP} \rangle \langle \text{SUBJ} \rangle i \end{aligned}$$

And in fact, AVM notation is essentially a two-dimensional notation for multimodal logic with nominals. For more on feature logic as hybrid logic, see Blackburn [10], Blackburn and Spaan [17], and Reape [45, 46] (and see Bird and Blackburn [9] for related ideas in phonology).

Example 3.8 (Description Logic) The TBoxes of the concept language *ALCO* (that is, *ALC* enriched with the *O* operator mentioned in Example 2.3) is a notational variant of the @-free fragment of hybrid multimodal logic. First, every nominal corresponds to an expression of the form $O(i)$. Conversely, every *ALCO* expression of the form $O(i, \dots, j)$ corresponds to the formula $i \vee \dots \vee j$.

Furthermore, @ has a natural description logic interpretation. The ABox specification $i:\varphi$ corresponds to the satisfaction statement $@_i\varphi$, and the specification $(i, j):R$ corresponds to $@_i\langle R \rangle j$. But whereas ABox specifications are constraints stated at a separate representational level, their hybrid equivalents are part of the object language. In effect, hybrid multimodal logic is an extension of *ALCO* which fully integrates ABox specifications into the concept language (without moving us out of PSPACE). For more on description logic as hybrid logic, see De Giacomo [22], Blackburn and Tzakova [19], Areces and de Rijke [6], and (in spite of its title) Areces, Blackburn and Marx [3].

4 Hybrid Reasoning

Nominals and @ make it possible to create names for states, and to reason about state identity and the way states are linked. This give us enough classical power in the object language to capture the modal locality intuition (recall the little automaton exploring/building graphs) *without* requiring elaborate metatheoretic proof machinery. Hybrid deduction is a form of labelled deduction — but it's labelled deduction that has been internalized into the object language. I'll formulate hybrid reasoning as an unsigned tableau system. We'll need two groups of rules. Here's the first:

$$\begin{array}{ll}
\frac{@_s \neg \varphi}{\neg @_s \varphi} [\neg] & \frac{\neg @_s \neg \varphi}{@_s \varphi} [\neg \neg] \\
\frac{@_s(\varphi \wedge \psi)}{@_s \varphi \mid @_s \psi} [\wedge] & \frac{\neg @_s(\varphi \wedge \psi)}{\neg @_s \varphi \mid \neg @_s \psi} [\neg \wedge] \\
\frac{@_s @_t \varphi}{@_t \varphi} [@] & \frac{\neg @_s @_t \varphi}{\neg @_t \varphi} [\neg @] \\
\frac{@_s \langle \pi \rangle \varphi}{@_s \langle \pi \rangle a} [\langle \pi \rangle] & \frac{\neg @_s \langle \pi \rangle \varphi \quad @_s \langle \pi \rangle t}{\neg @_t \varphi} [\neg \langle \pi \rangle] \\
\frac{@_s [\pi] \varphi \quad @_s \langle \pi \rangle t}{@_t \varphi} [[\pi]] & \frac{\neg @_s [\pi] \varphi}{@_s \langle \pi \rangle a} [\neg [\pi]] \\
& \neg @_a \varphi
\end{array}$$

In these rules, s and t are metavariables over nominals, and a is a metavariable over new nominals (that is, nominals not used so far in the tableau construction). The

rules for $\vee$ and $\rightarrow$ are obvious variants of the rules for $\wedge$ (we'll see both rules when we give some examples).

Remark 4.1 (The first group internalizes the satisfaction definition)

These rules use the resources available in hybrid logic to mimic the Kripke satisfaction definition: they draw conclusions from the input to each rule (the formula(s) above the horizontal line) to the output (the formula(s) below the line). For example, the $\wedge$ -rule says that if $\varphi \wedge \psi$ is true at s , then both φ and ψ are true at s , while it's dual rule $\neg \wedge$ (a branching rule) says that if $\varphi \wedge \psi$ is false at s , then either φ or ψ is false at s . Note that both the $[\pi]$ -rule and the $\neg \langle \pi \rangle$ -rule take two input formulas, one of which (the minor premiss) is a formula of the form $@_s \langle \pi \rangle t$. For example, the $[\pi]$ -rule says that if a pair of formulas of the form $@_s [\pi] \varphi$ and $@_s \langle \pi \rangle t$ can be found on some branch of the tableau, we are free to extend that branch by adding $@_t \varphi$ — a clear reflection of the Kripke semantics for $[\pi]$. Already first-order ideas are creeping into the system: this rule trades on the fact that hybrid logic is strong enough to make statements about state succession (using near-atomic satisfaction statements of the form $@_s \langle \pi \rangle t$).

But it is with the $\langle \pi \rangle$ - and $[\pi]$ -rules that first-order ideas really make themselves felt. What do we know when a formula of the form $\langle \pi \rangle \varphi$ is true at s ? The Kripke satisfaction definition gives us the answer: we know that (1) we can make an R_π transition from s to some state, and (2) at this R_π -successor state, φ is true. The $\langle \pi \rangle$ -rule captures this idea: it tells us to (1) introduce a new nominal a to name the successor state, and (2) insist that φ is true at a . Recall that in first-order reasoning, existential quantifiers are eliminated by introducing new parameters. In effect, the $\langle \pi \rangle$ -rule uses nominals to exploit this first-order idea. Incidentally: we don't apply the $\langle \pi \rangle$ -rule to formulas of the form $@_s \langle \pi \rangle \varphi$ where φ is a nominal. Doing so is pointless, for it would simply create a new name for a state that already had a name.

But we need a second group of rules. Nominals and @ come with a certain amount of logic built in: they provide theories of state equality and state succession. Just as we need to add special rules or axioms to first-order logic to handle the equality symbol correctly, we need additional mechanisms for nominals and @:

$$\frac{[s \text{ on branch}]}{@_s s} [\text{Ref}] \quad \frac{@_t s}{@_s t} [\text{Sym}] \quad \frac{@_s t \quad @_t \varphi}{@_s \varphi} [\text{Nom}] \quad \frac{@_s \langle \pi \rangle t \quad @_t t'}{@_s \langle \pi \rangle t'} [\text{Bridge}]$$

Remark 4.2 (The second group is essentially a classical rewrite system)

The Ref rule says that if a nominal s occurs in any formula on a branch, then we are free to add $@_s s$ to that branch; this is clearly an analog of the first-order reflexivity rule for $=$, just as the Sym rule is an analog of the first-order symmetry rule for $=$. What about transitivity? From $@_s t$ and $@_t t'$ we should be able to conclude $@_s t'$. But this is a special case of Nom, namely when φ is chosen to be a nominal t' . More generally, Nom ensures that identical states carry identical information, while Bridge ensures that states are coherently linked. In first-order terms, these rules ensure that state identity is not merely an equivalence relation but a congruence.

As with any tableau system, we prove formulas by systematically trying to falsify them. Suppose we want to prove φ . We choose a nominal (say i) that does not occur in φ (this acts as a name for the falsifying state that is supposed to exist),

prefix φ with $\neg @_i$, and start applying rules. If the tableau closes (that is, if every branch contains some formula and its negation), then φ is proved. On the other hand, suppose we reach a stage where we have applied the appropriate connective rule to every complex formula (or in the case of $[\pi]$ -formulas, we have applied the $[\pi]$ -rule to every pair of formulas of the form $@_s[\pi]\varphi$, $@_s\langle\pi\rangle t$ on the same branch; and analogously for $\neg\langle\pi\rangle$ -formulas) and no application of the rewrite rules yields anything new. If the tableau we have constructed contains open branches (that is, branches not containing conflicting formulas), then φ is not valid (and hence not provable), and the near-atomic satisfaction statements on the open branch specify a countermodel.

Example 4.3 (A standard multimodal validity) *Let's start with an example from ordinary multimodal logic: $\langle\pi\rangle(p \vee q) \rightarrow \langle\pi\rangle p \vee \langle\pi\rangle q$ is valid (for any modality $\langle\pi\rangle$), hence this formula should be provable. Here's how to do it:*

1	$\neg @_i(\langle\pi\rangle(p \vee q) \rightarrow \langle\pi\rangle p \vee \langle\pi\rangle q)$	
2	$@_i\langle\pi\rangle(p \vee q)$	1, $\neg \rightarrow$
2'	$\neg @_i(\langle\pi\rangle p \vee \langle\pi\rangle q)$	<i>Ditto</i>
3	$\neg @_i\langle\pi\rangle p$	2', $\neg \vee$
3'	$\neg @_i\langle\pi\rangle q$	<i>Ditto</i>
4	$@_i\langle\pi\rangle j$	2, $\langle\pi\rangle$
4'	$@_j(p \vee q)$	<i>Ditto</i>
5	$\neg @_j p$	3, 4, $\neg \langle\pi\rangle$
6	$\neg @_j q$	3', 4, $\neg \langle\pi\rangle$
7	$@_j p$ $@_j q$	4', $\vee$
	$\boxtimes$ 5, 7 $\boxtimes$ $\boxtimes$ 6, 7 $\boxtimes$	

In short, we start with one initial state (namely i) and then use the tableau rules to reason about what must hold there. At line 4 we use the $\langle\pi\rangle$ -rule to introduce a new state name, namely j . We continue to reason about the way information must be distributed across these two states until we are forced to conclude that there is no coherent way of doing so.

Example 4.4 (A genuinely hybrid validity) *The previous example gives only the barest hint of what the system can do. Here's a more interesting example, which shows that hybrid reasoning not merely makes use of nominals and $@$, but also gets to grip with the logic of state identity and succession they embody.*

Suppose we're working with a language with three modalities. To emphasize the geometric intuitions underlying hybrid reasoning, let's call these $\langle\text{VERT}\rangle$, $\langle\text{HOR}\rangle$ and $\langle\text{DIAG}\rangle$ (for vertical, horizontal and diagonal) respectively. Now, $\langle\text{HOR}\rangle\langle\text{VERT}\rangle(i \wedge p) \wedge \langle\text{DIAG}\rangle i \rightarrow \langle\text{DIAG}\rangle p$ is valid (for there's only one state named i) and we can prove

it as follows:

1	$\neg @_j(\langle\text{HOR}\rangle\langle\text{VERT}\rangle(i \wedge p) \wedge \langle\text{DIAG}\rangle i \rightarrow \langle\text{DIAG}\rangle p)$	
2	$@_j(\langle\text{HOR}\rangle\langle\text{VERT}\rangle(i \wedge p) \wedge \langle\text{DIAG}\rangle i)$	1, $\neg \rightarrow$
2'	$\neg @_j\langle\text{DIAG}\rangle p$	<i>Ditto</i>
3	$@_j\langle\text{HOR}\rangle\langle\text{VERT}\rangle(i \wedge p)$	2, $\wedge$
3'	$@_j\langle\text{DIAG}\rangle i$	<i>Ditto</i>
4	$@_j\langle\text{HOR}\rangle k$	3, $\langle\text{HOR}\rangle$
4'	$@_k\langle\text{VERT}\rangle(i \wedge p)$	<i>Ditto</i>
5	$@_k\langle\text{VERT}\rangle l$	4', $\langle\text{VERT}\rangle$
5'	$@_l(i \wedge p)$	<i>Ditto</i>
6	$@_l i$	5', $\wedge$
6'	$@_l p$	<i>Ditto</i>
7	$@_l l$	6, <i>Sym</i>
8	$@_l p$	6', 7, <i>Nom</i>
9	$\neg @_l p$	2', 3', $\neg\langle\text{DIAG}\rangle$
	$\boxtimes$ 8, 9 $\boxtimes$	

Think in terms of a graph-building automaton: it creates an initial state named i , generates successor states j , k and l , and reasons about the way information must be distributed over them until it becomes clear that there is no way to construct a countermodel.

Remark 4.5 (There are other approaches) *I have presented hybrid reasoning as an unsigned tableau system, but we are not forced to do this, and the underlying graph construction intuition come through in a range of proof styles. For example, Seligman [52] presents sequent and natural deduction systems with much the same geometrical flavor (indeed Seligman motivates his rules by discussing what a logic of spatial locations should look like). The same is true of Tzakova's [55] Fitting-style indexed tableau approach, Demri's [23] sequent system for the $\langle\text{F}\rangle$ and $\langle\text{P}\rangle$ language enriched with nominals but without $@$, and Konikowska's [36] sequent based approach to the logic of relative similarity.*

One last point. The link with orthodox modal labelled deduction should now be clear — but there is also a link with description logic: hybrid reasoning is a form of ABox reasoning. The tableau system manipulates satisfaction statements, which are essentially ABox specifications (recall Example 3.8).

5 Other Frame Classes

The tableau system is (sound and) complete in the following sense. Let us say that a formula φ is *tableau provable* iff there is a closed tableau with $\neg @_i \varphi$ as its root (where i is a nominal not occurring in φ). Then:

Theorem 5.1 φ is tableau provable iff φ is valid.

PROOF. Soundness is straightforward. A completeness proof for unimodal languages is given in Blackburn [13] using a Hintikka set argument; it extends straightforwardly to multimodal languages. ■

So far so good — but valid means “true in all states in any hybrid model based on *any* frame”, and often we only care about models based on frames with certain properties, and we want to reason in the stronger logics such frames give rise to.

In many cases hybrid reasoning adapts straightforwardly to cope with such demands. In particular, if we use *pure* formulas (that is, formulas containing no propositional variables) there is a straightforward link between *defining* a class of frames and *reasoning* about the frames in that class. A formula φ *defines* a class of frames F iff φ is valid on all the frames in F and falsifiable on any frame not in F . A formula defines a property of frames (such as transitivity) iff it defines the class of frames with that property. So: what can pure formulas define?

Example 5.2 (Pure formulas and frame definability) *Consider the temporal language in $\langle F \rangle$ and $\langle P \rangle$. Using pure formulas, we can define a number of properties relevant to temporal logic:*

$@_i \neg \langle F \rangle i$	$\forall x \neg (x R_F x)$ (Irreflexivity)
$@_i \neg \langle F \rangle \langle F \rangle i$	$\forall xy (x R_F y \rightarrow \neg y R_F x)$ (Asymmetry)
$@_i [F] (\langle F \rangle i \rightarrow i)$	$\forall xy (x R_F y \wedge y R_F x \rightarrow x = y)$ (Antisymmetry)
$\langle F \rangle \langle F \rangle i \rightarrow \langle F \rangle i$	$\forall xyz (x R_F y \wedge y R_F z \rightarrow x R_F z)$ (Transitivity)
$\langle F \rangle i \rightarrow \langle F \rangle \langle F \rangle i$	$\forall xy (x R_F y \rightarrow \exists z (x R_F z \wedge z R_F y))$ (Density)
$@_i \langle F \rangle j \vee @_i j \vee @_j \langle F \rangle i$	$\forall xy (x R_F y \vee x = y \vee y R_F z)$ (Trichotomy)

The properties just listed only tell us about R_F — but a far more basic property of frames is needed for temporal logic, namely that R_F and R_P be mutually converse relations. This can also be defined using pure formulas. First note that the following relations between R_F and R_P are definable:

$@_i [F] \langle P \rangle i$	$\forall xy (x R_F y \rightarrow y R_P x)$
$@_i [P] \langle F \rangle i$	$\forall xy (x R_P y \rightarrow y R_F x)$

It follows that the conjunction $@_i [F] \langle P \rangle i \wedge @_i [P] \langle F \rangle i$ defines those frames in which R_F and R_P are mutually converse. And once we have this fundamental interaction defined, we can stop thinking in terms of separate R_F and R_P relations, instead viewing $\langle F \rangle$ as looking forward along some binary relation $<$ (the “flow of time”) and $\langle P \rangle$ as looking backwards along the same relation. This enables us to define further temporally interesting properties:

$\langle P \rangle \langle F \rangle i$	$\forall xy \exists z (z < x \wedge z < y)$ (Left-Directedness)
$@_i (\langle F \rangle \top \rightarrow \langle F \rangle [P] \neg i)$	$\forall xy (x < y \rightarrow \exists z (x < z \wedge \neg \exists w (x < w < z)))$ (Right-Discreteness)

I mentioned in Example 2.1 that asymmetry was not definable in ordinary temporal logic. In fact, with the exception of the mutually converse property, transitivity, and density, none of the properties just defined are definable in orthodox temporal logic. Hybrid languages fill a genuine expressive gap when it comes to defining frames.

Remark 5.3 (All we need are satisfaction statements) *Note that if a formula φ defines a class of frames F , then so does the satisfaction statement $@_i \varphi$, where i is any nominal not occurring in φ . The relevance of this for tableaux will soon be clear.*

So nominals and $@$ enable us to define interesting classes of frames, and moreover every definable class of frames is definable using a satisfaction statement. This is pleasant — but the really important point is the way these frame defining powers interact with hybrid reasoning. Roughly speaking, if a pure formula α defines a class

of frames F , and we are free to introduce α as an axiom into our tableau proofs, then the axiom-enriched tableaux system is guaranteed to be complete with respect to F . For pure formulas, definability and completeness match perfectly.

More precisely, let A be a countable set of *pure satisfaction statements*, and $H+A$ be the tableau system that uses the formulas in A as axioms. That is, for any α in A , and any nominals $j, j_1, \dots, j_n$ that occur on a branch of a tableau, we are free to add α or $\alpha[j_1/i_1, \dots, j_n/i_n]$ to the end of that branch (here $i_1, \dots, i_n$ are nominals in α , and $\alpha[j_1/i_1, \dots, j_n/i_n]$ is the pure satisfaction statement obtained by uniformly substituting nominals for nominals as indicated).

Theorem 5.4 *Let A be a finite or countably infinite set of pure satisfaction statements, and let F be the class of frames that A defines (that is, the class of frames on which every formula in A is valid). Then $H+A$ is complete with respect to F .*

PROOF. See Blackburn [13] for the unimodal case. The multimodal case is a straightforward generalization. ■

Example 5.5 (An application in temporal logic) *Suppose we are working with the $\langle F \rangle$ and $\langle P \rangle$ temporal language, and that we are interested in models with a transitive flow of time. Which axioms guarantee completeness?*

The following suffice. First, to ensure that $\langle F \rangle$ and $\langle P \rangle$ really are mutually converse, add the axioms $@_i [F] \langle P \rangle i$ and $@_i [P] \langle F \rangle i$; we know from Example 5.2 that together these formulas define the converse property, and both are pure satisfaction statements. Now to guarantee transitivity. The pure formula $\langle F \rangle \langle F \rangle i \rightarrow \langle F \rangle i$ defines this property. This is not a satisfaction statement, but $@_j (\langle F \rangle \langle F \rangle i \rightarrow \langle F \rangle i)$ is, and this defines transitivity too.

What can we prove in this system? Here’s an illustration. Note that for any choice of formula φ (not just pure formulas), $\langle P \rangle \langle P \rangle \varphi \rightarrow \langle P \rangle \varphi$ is valid on the class of frames our axioms define. Thus, by Theorem 5.4, we should be able to prove any instance of this schema. And we can. In what follows i, j , and k , are chosen to be nominals not occurring in φ :

1	$\neg @_i (\langle P \rangle \langle P \rangle \varphi \rightarrow \langle P \rangle \varphi)$	
2	$@_i \langle P \rangle \langle P \rangle \varphi$	1, $\neg \rightarrow$
2'	$\neg @_i \langle P \rangle \varphi$	Ditto
3	$@_i \langle P \rangle j$	2, $\langle P \rangle$
3'	$@_j \langle P \rangle \varphi$	Ditto
4	$@_j \langle P \rangle k$	3', $\langle P \rangle$
4'	$@_k \varphi$	Ditto
5	$@_j [P] \langle F \rangle j$	Axiom
6	$@_k \langle F \rangle j$	4, 5, $[P]$
7	$@_i [P] \langle F \rangle i$	Axiom
8	$@_j \langle F \rangle i$	3, 7, $[P]$
9	$@_k (\langle F \rangle \langle F \rangle i \rightarrow \langle F \rangle i)$	Axiom
10	$\neg @_k \langle F \rangle \langle F \rangle i$	$@_k \langle F \rangle i$
11	6, 10, $\neg \langle F \rangle$	$\neg @_j \langle F \rangle i$
12	$\nexists 8, 11$	$@_k [F] \langle P \rangle k$
13		$@_i \langle P \rangle k$
		$\neg @_k \varphi$
		$\nexists 4', 13$
		9, $\rightarrow$
		10, 11, $[F]$
		2', 12, $\neg \langle P \rangle$

Once again, it is best to think of this proof in terms of a little graph-building automaton: it stepwise generates a graph and shows (now with the help of the axioms) that there is no coherent way to decorate the resulting structure with information.

In effect, Theorem 5.4 tells us that we can analyze hybrid reasoning in terms of a basic proof engine (such as our tableau rules) together with an axiomatic theory (at least so long as the axiomatic theory is formulated using only *pure* formulas). This is the way things work in first-order logic, and the resemblance is not coincidental. First, recall that the Standard Translation for hybrid languages maps nominals to free first-order variables. It follows that any pure formula φ defines a first-order class of frames (namely the class defined by the universal closure of $ST(\varphi)$). Second, analogous theorems have been proved for various hybrid languages, and although the completeness proofs differ in many respects, they typically have one ingredient in common: they use nominals to integrate the standard first-order model construction technique (the use of Henkin constants) with the standard modal technique (canonical models). As a number of authors emphasize (in particular Bull [21], Passy and Tinchev [41], and Blackburn and Tzakova [20]), such proofs show that hybrid logic genuinely blends modal and classical ideas.

Remark 5.6 (Related work) *Many of the same technical themes (including an essentially identical model construction technique) can be found in Basin, Matthews, and Viganò's [7] approach to labelled deduction for orthodox modal languages. The links between their work and the hybrid tradition deserves further exploration (for a start, many of their proof-theoretical insights may generalize to hybrid languages). Other general completeness results covering first-order definable frame classes have been proved for hybrid languages, such as Demri's [23] extension of the modal Sahlqvist theorem for his nominal-driven temporal sequent system.*

But the emphasis on *first-order* aspects of hybrid logic also point to the limitations of the previous theorem: it doesn't cover *second-order* frame classes — and many such classes are definable with the aid of propositional variables.

Example 5.7 (Second-order frame classes) *By making use of mixed formulas (that is, formulas containing both nominals and ordinary propositional variables) we can define $\mathbb{Z}$, the integers in their usual order, up to isomorphism; this cannot be done in first-order logic.*

The key observation is due to van Benthem [8], who points out that the simple $\langle F \rangle$ and $\langle P \rangle$ language can almost define $\mathbb{Z}$. As he notes, the formula

$$([P]([P]p \rightarrow p) \rightarrow (\langle P \rangle[P]p \rightarrow [P]p)) \wedge ([F]([F]p \rightarrow p) \rightarrow (\langle F \rangle[F]p \rightarrow [F]p))$$

(a bidirectional variant of the Löb formula used in modal provability logic) defines $\mathbb{Z}$ up to isomorphism on the class of strict total orders without endpoints (that is, this Löb variant is valid on a frame $(T, <)$ that is a strict total order without endpoints iff $(T, <)$ is isomorphic to $\mathbb{Z}$.)

But it follows from standard modal results that we can't define strict total order without endpoints using only propositional variables — and this is where nominals come to the rescue. We have already seen that there are (pure) formulas defining the mutual converse property of $\langle F \rangle$ and $\langle P \rangle$, transitivity, irreflexivity and trichotomy. Furthermore, the formulas $\langle F \rangle \top$ and $\langle P \rangle \top$ ensure that there are no endpoints. So the conjunction of all these (pure) formulas defines the class of strict total orders without endpoints — and hence conjoining the Löb variant yields a (mixed) formula valid on precisely the frames isomorphic to $\mathbb{Z}$. In a similar way, using a mixed formula it is

possible to define $\mathbb{N}$, the naturals in their usual order, up to isomorphism; see Blackburn [11] for details. The second-order aspects of hybrid languages deserve further study.

The result has another limitation: it gives no computational information. While the basic satisfaction problem for hybrid languages is PSPACE-complete, adding further axioms can have a wide range of effects: they may lower the problem into NP, leave it in PSPACE or lift it to EXPTIME (see Areces, Blackburn and Marx [3] for examples of all three possibilities). Nor is it difficult to devise axioms which result in logics with undecidable satisfaction problems. So the previous result tells us nothing about proof search or termination: it simply draws attention to a group of logics which are well-behaved from the perspective of completeness theory. It may well be that proof-theoretical and computational insights from the labelled deduction and description logic communities have a role to play in analyzing these logics further.

6 Binding Nominals to States

From the perspective of the Standard Translation, adding nominals to a modal language is in effect to add free variables over states. This immediately suggest a further extension: why not *bind* these “free variables”, thus giving ourselves access to even more expressive power? I'll give a brief sketch of such logics, and then turn to the issue that interests me here: why they are relevant to knowledge representation.

Example 6.1 (Losers, jerks, and politicians) *Let's jump into the realms of pop-psychology and define a loser to be someone with no self-respect. Now, we can't define this concept in the hybrid logics we have seen so far; the closest we get is:*

$$i \wedge \neg \langle \text{RESPECT} \rangle i.$$

This says that a specific individual i lacks self-respect. But we want more: we want a formula that is true at precisely those nodes (individuals) which lack a reflexive RESPECT arc. We can get what we want by binding i out:

$$\exists x (x \wedge \neg \langle \text{RESPECT} \rangle x).$$

This sentence is true at precisely those nodes at which it is possible to bind x to the current state, but impossible to loop back to the current state via the RESPECT relation.

Two remarks. First, the idea of binding nominals to the current state is so important in hybrid logic that a special notation (namely $\downarrow$) has been introduced for it. So the previous sentence would normally be written:

$$\downarrow x. \neg \langle \text{RESPECT} \rangle x.$$

Second, as these examples illustrate, orthodox variable notation (x, y, z , and so on) is usually used for bound nominals.

OK — let's now define a jerk to be an idiot who admires himself:

$$\text{idiot} \wedge \downarrow x. \langle \text{ADMIRE} \rangle x.$$

This sentence is satisfied at precisely those nodes which (1) have the idiot property, and (2) from which it is possible to take a reflexive step via the ADMIRE relation.

Finally, let's define a politician as a smooth talker such that everyone he talks to mistrusts him:

$$\downarrow x.(\text{smooth-talker} \wedge \forall y(\langle \text{TALKS-TO} \rangle y \rightarrow \neg @_y \langle \text{TRUSTS} \rangle x).$$

Note the way the $@_y$ switches the perspective from the node x (the politician) to his audience.

I won't give a precise definition of the syntax and semantics of hybrid languages with $\forall$ and $\exists$ here (you can find all this in Blackburn and Seligman [15, 16] or Blackburn and Tzakova [18, 19]). The previous examples tell you pretty much everything you need to know, and the discussion that follows should clarify things further.

Remark 6.2 (We now have first-order expressivity) Our new hybrid logic is strong enough to express any first-order concept. Here's the Hybrid Translation from first-order representations to our new hybrid logic:

$$\begin{aligned} HT(xR_\pi y) &= @_x \langle \pi \rangle y \\ HT(Px) &= @_x p \\ HT(x = y) &= @_x y \\ HT(\neg \varphi) &= \neg HT(\varphi) \\ HT(\varphi \wedge \psi) &= HT(\varphi) \wedge HT(\psi) \\ HT(\exists v \varphi) &= \exists v HT(\varphi) \\ HT(\forall v \varphi) &= \forall v HT(\varphi). \end{aligned}$$

But although we can jump straight up to full first-order power, we don't have to. For a start, the use of $@$ in the hybrid translation is *crucial*. If we work with the $@$ -free sublanguage, binding nominals to states with $\exists$ and $\forall$ does *not* yield full first-order expressive power; for a counterexample, see Proposition 4.5 of Blackburn and Seligman [15]. Hybrid logic decomposes the action of the classical quantifiers into two subtasks: perspective-shifting (performed by $@$) and binding (performed by the hybrid binders $\exists$ and $\forall$).

Moreover, we've seen that there is a useful restricted form of these binders, namely $\downarrow$. Some recent papers have explored hybrid logics with a primitive $\downarrow$ binder (without $\exists$ or $\forall$), and it turns out that such logics *characterize* the notion of locality; see Areces, Blackburn, and Marx [4].

Remark 6.3 (But even local binding is complex) Be warned: $\downarrow$ may seem simple, but it's not. Even without $@$ (let alone $\forall$ or $\exists$, which are obviously powerful) it has an undecidable satisfaction problem. A detailed analysis is given in Areces, Blackburn, and Marx [5].

Why is this? The following result (taken from Blackburn and Seligman [15]) may help the reader see why local binding is so powerful. We'll see — using a spypoint argument — that a hybrid language containing $\downarrow$ and just a single diamond lacks the finite model property. Let $SCID_4$ be the conjunction of the following formulas:

$$\begin{aligned} S & x \wedge \neg \langle R \rangle x \wedge \langle R \rangle \neg x \wedge [R] \langle R \rangle x \\ C & [R][R] \downarrow y.(\neg x \rightarrow \langle R \rangle (x \wedge \langle R \rangle y)) \end{aligned}$$

$$\begin{aligned} I & [R] \downarrow y. \neg \langle R \rangle y \\ D & [R] \langle R \rangle \neg x \\ 4 & [R] \downarrow y. \langle R \rangle (x \wedge [R] (\langle R \rangle (\neg x \wedge \langle R \rangle y \rightarrow \langle R \rangle y))) \end{aligned}$$

Note that these formulas are pure, and that $\downarrow x.SCID_4$ is a sentence. Moreover, note that this sentence has at least one model. For let $(\omega, <)$ be the natural numbers in their usual order, and suppose $s \notin \omega$ (s is the spypoint). Let $\mathcal{N}^s$ be the model bearing a single binary relation R defined as follows: W is $\omega \cup \{s\}$, R is $< \cup \{(n, s), (s, n) : n \in \omega\}$, and the valuation V is arbitrary. Clearly $\mathcal{N}^s, s \models \downarrow x.SCID_4$.

Obviously $\mathcal{N}^s$ is an infinite model. In fact any model $\mathcal{M} = (W, R, V)$ for $\downarrow x.SCID_4$ is infinite. For suppose $\mathcal{M}, s \models \downarrow x.SCID_4$. Let $B = \{b \in W : sRb\}$. Because S is satisfied, $s \notin B$, $B \neq \emptyset$, and for all $b \in B$, bRs . Because C is satisfied, if $a \neq s$ and a is an R -successor of an element of B then a is also an element of B . As I is satisfied at s , every point in B is irreflexive; as D is satisfied at s , every point in B has an R -successor distinct from s ; and as 4 is satisfied, R is a transitive ordering of B . So B is an unbounded strict partial order, thus B is infinite, hence so is W . So the ability to bind locally really does give us the power to see a lot of structure. And this power leads to undecidability (we can use spypoints to gaze upon the representation of some undecidable problem, such as an unbounded tiling problem).

Thus nominal binding offers (lots!) of new representational power — but how do we reason?

Remark 6.4 ($\forall$ and $\exists$ have classical tableau rules) To cope with hybrid logic enriched with $\forall$ and $\exists$, we add the following rules to our tableau system. Note their form: they are the classical tableau rules for existential and universal quantifiers:

$$\begin{array}{c} \frac{\neg @_s \exists x \varphi}{\neg @_s \varphi[t/x]} \quad \frac{@_s \exists x \varphi}{@_s \varphi[a/x]} \\ \frac{\neg @_s \forall x \varphi}{\neg @_s \varphi[a/x]} \quad \frac{@_s \forall x \varphi}{@_s \varphi[t/x]} \end{array}$$

(Important: recall that a stands for a new nominal.) But while the rules are essentially classical, don't forget that the underlying language is different (after all, $\forall$ and $\exists$ bind formulas!). So as well as being able to prove all the standard classical quantificational principles (for example, $\forall x(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \forall x\psi)$, where x does not occur free in φ) we can also prove intrinsically modal principles. For example, $\exists xx$ is valid (this says: it is always possible to bind a variable to the current state). We can prove it as follows:

$$\begin{array}{ll} 1 & \neg @_i \exists xx \\ 2 & \neg @_i i \quad 1, \neg \exists \\ 3 & @_i i \quad \text{Ref} \\ & \spadesuit 2, 3 \spadesuit \end{array}$$

These rules give us a complete deduction system for hybrid logic with $\forall$ and $\exists$. Moreover, Theorem 5.4 extends to these systems: adding pure axioms yields a system complete with respect to the class of frames the axioms define. As before, “pure” simply means “contains no propositional variables”, so we are free to make use of $\forall$ and

$\exists$ in our axioms. It follows (with the help of the Hybrid Translation) that we have a general completeness result that covers any first-order definable class of frames. Rules for $\downarrow$ can be found in Blackburn [13].

It's time to turn to the link with knowledge representation. I'll approach this topic via James Allen's classic work on temporal representation.

Example 6.5 (Allen style representations) *The core of Allen's system is an orthodox first-order theory of interval structure to which the metapredicate **Hold** has been added: **Hold**(P,i) asserts that property P holds at the interval i.*

Allen then goes on to elaborate his account of properties. He introduces function symbols suggestively named and, or, not, exists, and all, for combining property symbols, together with axioms governing them: for example

$$\mathbf{Hold}(\text{and}(\mathbf{P},\mathbf{Q}),i) \leftrightarrow \mathbf{Hold}(\mathbf{P},i) \wedge \mathbf{Hold}(\mathbf{Q},i).$$

It's clear Allen wants an 'internal' logic of terms that mirrors the 'external' logic of formulas. To put it another way, although he represents properties using terms, he wants them to behave like formulas.

This aspect of Allen's system has been criticized (some of the axioms governing the logical functions are rather odd; furthermore, as the structure of property terms is never fully specified, it's rather unclear what can and cannot be done with them; see Turner [54] and Shoham [53]). But I'm not so much interested in the details as the general strategy — for this is now standard in AI.

For example, if you look at Russell and Norvig [48] (in particular, the discussion of ontological engineering in Chapter 8) you'll see that Allen's approach has been generalized into a multistep methodology: (1) start with a first-order language; (2) reify the language heavily (that is, treat categories as individuals); (3) add metapredicates; and (4) when handling temporal aspects of ontology, induce boolean structure on the terms by adding and axiomatizing the logical functions and, or, and not (exists and all are not discussed).

Why is the methodology pioneered by Allen so popular? In my view, the point is the following. Knowledge representation is ultimately about representing information in a usable form — and this means bringing a variety of information types into a precise framework in which it can be manipulated as flexibly as possible. In essence, Allen's strategy is to start with first-order logic (because it's well understood) and then to mould it to the requirements of knowledge representation. Heavy use of reification and metapredicates allows general statements about a wide range of category types to be made. Logical functions are an attempt to soften the rigid distinction first-order logic draws between terms (which code referential information) and formulas (which code other types of information), thereby making more flexible representations possible. It's an interesting strategy — but it's *not* the only one.

Why not *start* with the intuition that all types of information should be treated democratically — or more accurately, *polymorphically*? This is the intuition behind hybrid logic. Hybrid logic begins with the observation that we *can* freely combine referential and non-referential information if we represent both types of information as *formulas*. Because this is our starting point, we don't need to introduce special logical functions and axioms to govern them — there is no term/formula distinction:

the standard connectives are responsible for combining all information right from the start. (Note that $@_i(p \wedge q) \leftrightarrow @_i p \wedge @_i q$, the hybrid analog of Allen's axiom for the *and* function, isn't something extra that needs to be stipulated: it's just a validity of hybrid logic, and can easily be proved in the basic tableau system.) Nor is there any mystery about what “property terms” are: Allen seems to have wanted properties to have a formula-like structure, and of course, that's *exactly* the form all representations take in hybrid logic. And binding nominals with $\forall$ and $\exists$ (which seems to correspond to Allen's intentions regarding the logical functions *exists* and *all*) will take us all the way up to first-order expressivity (if that's where we want to go).

7 The Sorting Strategy

In horticulture, hybrids are crossbreeds between distinct but related strains: ideally they combine the desirable properties of the parent strains in interesting new ways. Hybrid logic is certainly hybrid in this sense. Enriching modal logic with nominals and $@$ leads to systems that draw on both modal and first-order logic: we retain the locality and decidability of modal logic, gain the ability to name states and reason about their identity and their interrelationships, and (via nominal binding) open a novel route to first-order expressivity.

But hybrid logic is also a *sociological* hybrid: it's a meeting place for ideas from many traditions. We've seen that feature logic, description logic, and labelled deduction have independently developed key ideas of hybrid logic, and I've argued that the Allen-style ontological engineering languages can be viewed as strong hybrid languages. In short, a number of research communities, faced with similar problems (how best to represent and reason about graphlike structures) have come up with similar answers independently. Not only do they draw (consciously or unconsciously) on modal logic, they even moved beyond the barriers of modal orthodoxy in much the same way — the way encapsulated in hybrid logic.

But there is a third sense in which hybrid languages are hybrid, and this is perhaps the most important of all: hybrid languages are *intrinsically* hybrid. They allow us combine different sorts of information in a single formalism. In a nutshell, hybrid logics are *sorted modal logics*.

The importance of sorting has long been recognized in AI, linguistics, and philosophy: knowing that a piece of information is of a particular kind may allow us to draw useful conclusions swiftly and easily. But sorting has been neglected in the logical tradition: many useful kinds of sortal reasoning (for example, chaining through an inheritance hierarchy) are regarded as too simple to be of logical interest, and every logician knows that sorted first-order languages offer no new expressive power.

But sorted *modal* languages certainly do. As we have seen, by adding a second sort of atomic formula (nominals) and a new construct to exploit it (satisfaction operators), we can describe models in more detail and define new classes of frames. Moreover, we can create a basic reasoning system that is modally natural and supports a wide range of richer logics. But the hybrid languages of this paper have been simple two-sorted systems. Why stop there?

Example 7.1 (Sorting and fine-grained temporal reference)

Blackburn [12] presents multisorted modal logics with atomic formulas ranging over intervals of different lengths (seconds, hours, years, ...). This lets us build repre-

sentations like

$\langle P \rangle (3.05 \wedge P.M. \wedge \text{Friday} \wedge 26\text{th} \wedge \text{March} \wedge 1999 \wedge \text{Vincent-accidentally-squeeze-the-trigger}),$

which locates the trigger-squeezing event at the specific day and time the notation suggests. These logics are then extended to deal with indexical expressions (such as now, yesterday, today, and tomorrow), enabling us to build representations such as

$\langle P \rangle (\text{Yesterday} \wedge \text{Marvin's-head-explode}),$

which locates the exploding-head event yesterday. Doing this properly means we have to sort two-dimensional modal logic (among other things, we need to guarantee that $\langle F \rangle (\text{yesterday} \wedge \varphi)$ is false at every state in every model, for yesterday always lies in the past), and sorting turns out to be an effective way of exploiting two-dimensional semantics. The resulting logics are decidable (in fact, NP-complete) in many cases of interest.

Example 7.2 (Sorting and paths) When reasoning about branching time we often want to assert that that some event will take place in all possible paths into the future. This cannot be done in the temporal language in $\langle F \rangle$ and $\langle P \rangle$, even with the help of nominals and @.

Bull [21] solved this problem by further sorting. He introduced a three-sorted modal language: in addition to propositional variables and nominals, his language contained path nominals, atomic formulas true at precisely the points on some path through a frame. He allowed explicit quantification over path nominals, and hence could define a “true at some state in every future” modality:

$$\langle \text{EVERY-FUT} \rangle \varphi := \forall \rho (\rho \rightarrow \langle F \rangle \exists x (x \wedge \rho \wedge \varphi)).$$

Here ρ is a bound path nominal, and x a bound nominal, so this says that on every path ρ through the current state, there is some future state x at which φ is true. See Goranko [32] and Blackburn and Tzakova [20] for more on hybrid languages for paths.

I believe such examples point the way to an interesting line of work: dealing with all ontological distinctions in multisorted modal languages. At present little is known about what can and cannot be done in such systems, but interesting questions abound. I hope some equally interesting answers will soon be forthcoming.

A Brief Guide to the Literature

I have said little about the history of hybrid logic; these notes are an attempt to put this right, and provide a route into the hybrid literature. I'll omit references to applications of hybrid logic (such as feature logic) as these were given in the main text.

Hybrid logic was invented by Arthur Prior, the inventor of $\langle F \rangle$ and $\langle P \rangle$ based temporal logic (that is, *tense logic*). The germs of the idea seem to have emerged in discussion with C.A. Meredith in the 1950s, but the first detailed account is in Chapter V and Appendix B3 of Prior's 1967 book *Past, Present, and Future* [42]. Several of the papers collected in *Paper on Time and Tense* [43] allude to or discuss hybrid languages, and the posthumously published book *Worlds, Times and Selves* [44]

is solely devoted to the topic (unfortunately, the book is only an approximation to Prior's intentions: it's essentially a reconstruction, by Kit Fine, of notes found after Prior's death in 1969). Prior called nominals *world propositions*, typically worked with very rich hybrid languages (he bound nominals using $\forall$ and $\exists$) and made heavy use of near-atomic satisfaction statements like the ones used in our tableau systems.

The next big step was Robert Bull's 1970 paper “An Approach to Tense Logic” [21]. Bull introduced a three-sorted hybrid language (propositional variables, nominals, and path nominals), noted that the presence of $\forall$ and $\exists$ made it easy to combine the modal canonical model construction with the first-order Henkin construction (and thus proved the earliest version of Theorem 5.4), and re-thought modal and hybrid completeness theory in terms of Robinson's non-standard set theory. It's a (too long overlooked) classic. Tough going in places, it repays careful reading.

I know of no more papers on the subject till the 1980s, when hybrid logic was independently reinvented by a group of Bulgarian logicians (Solomon Passy, Tinko Tinchev, George Gargov, and Valentin Goranko). The locus classicus of this work is Passy and Tinchev's “An Essay on Combinatoric Dynamic Logic” [41], a detailed study of hybrid Propositional Dynamic Logic. Like Bull's paper, it's one of the must reads of the hybrid literature (but don't overlook the many other excellent papers by these authors, such as [40, 40, 30, 28, 29].) The Sofia School did discuss nominal binding with $\forall$ and $\exists$, but one of their enduring legacies is that they initiated the study of binder-free systems. Gargov and Goranko's “Modal Logic with Names” [27] studies such systems in the setting of unimodal logic, and my own “Nominal Tense Logic” [11] does so in tense logic.

During the 1990s, the emphasis has been on understanding the hybrid hierarchy in more detail. Goranko [31] introduced $\downarrow$, Blackburn and Seligman [15, 16] examined the interrelationships between a number of different binders, and Blackburn and Tzakova [18, 20] mapped hybrid completeness theory for many of these systems. Intuitions about locality hinted at in some of these papers are placed on a firm mathematical footing in Areces, Blackburn and Marx [4]; the paper also proves some fundamental interpolation and complexity results (see also [5], by the same authors, for a detailed discussion of undecidability in $\downarrow$ based logics). The late 1990's also saw a number of papers of hybrid proof theory: Blackburn [13], Demri [23], Demri and Goré [24], Konikowska [36], Seligman [52] and Tzakova [55]. Actually, pioneering work had been done by Seligman at the beginning of the decade (see [50, 51]); unfortunately his work was overlooked.

Here's three suggestions for further reading. First, Chapter 7 of Blackburn, de Rijke, and Venema [14] contains a textbook level discussion on how to blend the canonical model and Henkin constructions (the idea behind Theorem 5.4 and its analogs). Second, “Complexity Results for Hybrid Temporal Logics” [3] a recent paper by Areces, Blackburn and Marx studies complexity issues in some detail. The proofs make heavy use of relational structures and have a strong geometric content. The paper relates the results to issues in temporal (and, in spite of the title, description) logic; for many readers this would be a good place to learn more about the expressivity hybrid languages offer. Third, Marx [38] is a review of *HyLo'99* (the First International Workshop on Hybrid Logic). This will give you a birds-eye-view of current issues in the field. In addition, Carlos Areces has recently created a hybrid logic website at <http://www.illc.uva.nl/~carlos/hybrid>. You can find the papers just mentioned

(and others) there.

Acknowledgments

In writing this paper I have tried to remain faithful to the spirit of my *M4M* talk, which means I have drawn on many discussions of modal logic, relational structures, hybrid logic, and related topics. I particularly wish to thank Carlos Areces, Johan van Benthem, Valentin Goranko, Edith Hemaspaandra, Maarten Marx, Maarten de Rijke, Jerry Seligman, Miroslava Tzakova, and Yde Venema, who have deeply influenced the way I think about these topics. I am also grateful to the participants of *M4M* and *HyLo'99* for interesting discussion, and to David Basin and Luca Viganò for helping me understand labelled deduction better. Thanks to Claire Gardent and Ewan Klein for their comments on the first version. Finally, a big thank you to the two referees for their detailed comments, and to Carlos Areces for his painstaking editorial help.

References

- [1] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, July 1984.
- [2] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
- [3] C. Areces, P. Blackburn, and M. Marx. Complexity results for hybrid temporal logics. To appear in *Logic Journal of the IGPL*, 1999.
- [4] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. Technical Report CLAUS-Report 104, Computerlinguistik, Universität des Saarlandes, 1999. <http://www.coli.uni-sb.de/cl/clus>. To appear in *Journal of Symbolic Logic*.
- [5] C. Areces, P. Blackburn, and M. Marx. A roadmap on the complexity of hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, number 1683 in LNCS, pages 307–321. Springer, 1999. Proceedings of the 8th Annual Conference of the EACSL, Madrid, September 1999.
- [6] C. Areces and M. de Rijke. Accounting for assertional information. Manuscript, 1999.
- [7] D. Basin, S. Matthews, and L. Viganò. Labelled propositional modal logics: theory and practice. *Journal of Logic and Computation*, 7:685–717, 1997.
- [8] J. van Benthem. *The Logic of Time*. Kluwer Academic Publishers, Dordrecht, second edition, 1991.
- [9] S. Bird and P. Blackburn. A logical approach to arabic phonology. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–29, 1991.
- [10] P. Blackburn. Modal logic and attribute value structures. In M. de Rijke, editor, *Diamonds and Defaults*, Synthese Language Library, pages 19–65. Kluwer Academic Publishers, Dordrecht, 1993.
- [11] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 14:56–83, 1993.
- [12] P. Blackburn. Tense, temporal reference, and tense logic. *Journal of Semantics*, 11:83–101, 1994.
- [13] P. Blackburn. Internalizing labelled deduction. Technical Report CLAUS-Report 102, Computerlinguistik, Universität des Saarlandes, 1998. <http://www.coli.uni-sb.de/cl/clus>. To appear in *Journal of Logic and Computation*.
- [14] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. 1999. A draft version of this book is available at <http://www.coli.uni-sb.de/cl/clus>.
- [15] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.
- [16] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyashev, editors, *Advances in Modal Logic*, volume 1, pages 41–62. CSLI Publications, Stanford University, 1998.
- [17] P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information*, 2:129–169, 1993.
- [18] P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 6:625–650, 1998.
- [19] P. Blackburn and M. Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24:23–49, 1998.
- [20] P. Blackburn and M. Tzakova. Hybrid languages and temporal logics. *Logic Journal of the IGPL*, 7(1):27–54, 1999.
- [21] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.
- [22] G. De Giacomo. *Decidability of class-based knowledge representation formalisms*. PhD thesis, Università di Roma “La Sapienza”, 1995.
- [23] S. Demri. Sequent calculi for nominal tense logics: a step towards mechanization? In N. Murray, editor, *Conference on Tableau Calculi and Related Methods (TABLEAUX)*, Saratoga Springs, USA, volume 1617 of *LNAI*, pages 140–154. Springer Verlag, 1999.
- [24] S. Demri and R. Goré. Cut-free display calculi for nominal tense logics. In N. Murray, editor, *Conference on Tableau Calculi and Related Methods (TABLEAUX)*, Saratoga Springs, USA, volume 1617 of *LNAI*, pages 155–170. Springer Verlag, 1999.
- [25] M. Fitting. *Proof Methods for Modal and Intuitionistic Logic*. Reidel, 1983.
- [26] D. Gabbay. *Labelled Deductive Systems*. Clarendon Press, Oxford, 1996.
- [27] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22:607–636, 1993.
- [28] G. Gargov and S. Passy. Determinism and looping in combinatory PDL. *Theoretical Computer Science*, 61:259–277, 1988.
- [29] G. Gargov and S. Passy. A note on Boolean modal logic. In P. Petkov, editor, *Mathematical Logic. Proceedings of the 1988 Heyting Summerschool*, pages 311–321. Plenum Press, New York, 1990.
- [30] G. Gargov, S. Passy, and T. Tinchev. Modal environment for Boolean speculations. In D. Skořdev, editor, *Mathematical Logic and its Applications*, pages 253–263. Plenum Press, 1987.
- [31] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language and Information*, 5(1):1–24, 1996.
- [32] V. Goranko. An interpretation of computational tree logics into temporal logics with reference pointers. Technical Report Verslagreeks van die Department Wiskunde, RAU, Nummer 2/96, Department of Mathematics, Rand Afrikaans University, Johannesburg, South Africa, 1996.
- [33] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.
- [34] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the Association for Computing Machinery*, 38:935–962, 1991.
- [35] W. Hodges. *Model Theory*. Cambridge University Press, Cambridge, 1993.
- [36] B. Konikowska. A logic for reasoning about relative similarity. *Studia Logica*, 58:185–226, 1997.
- [37] M. Kracht. Power and weakness of the modal display calculus. In H. Wansing, editor, *Proof Theory of Modal Logic*, pages 93–121. Kluwer Academic Publishers, Dordrecht, 1996.
- [38] M. Marx. First International Workshop on Hybrid Logic (HyLo ’99). *Logic Journal of the IGPL*, 7:665–669, 1999. Conference Report.
- [39] M. Marx and Y. Venema. *Multidimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, Dordrecht, 1997.
- [40] S. Passy and T. Tinchev. Quantifiers in combinatory PDL: completeness, definability, incompleteness. In *Fundamentals of Computation Theory FCT 85*, volume 199 of *LNCS*, pages 512–519. Springer, 1985.
- [41] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93:263–332, 1991.
- [42] A. Prior. *Past, Present and Future*. Oxford University Press, 1967.

- [43] A. Prior. *Papers on Time and Tense*. University of Oxford Press, 1968.
- [44] A. Prior and K. Fine. *Worlds, Times and Selves*. University of Massachusetts Press, 1977.
- [45] M. Reape. An introduction to the semantics of unification-based grammar formalisms. Technical Report DYANA deliverable R3.2.A, ESPRIT basic research action BR 3175, Center for Cognitive Science, University of Edinburgh, 1991.
- [46] M. Reape. A feature value logic. In C. Rupp, M. Rosner, and R. Johnson, editors, *Constraints, Language and Computation*, Synthese Language Library, pages 77–110. Academic Press, 1994.
- [47] M. de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57:566–584, 1992.
- [48] S. Russell and P. Norvig. *Artificial Intelligence*. Prentice Hall, 1995.
- [49] K. Schild. A correspondence theory for terminological logics. In *Proceedings of the 12th IJCAI*, pages 466–471, 1991.
- [50] J. Seligman. A cut-free sequent calculus for elementary situated reasoning. Technical Report HCRC-RP 22, HCRC, Edinburgh, 1991.
- [51] J. Seligman. Situated consequence for elementary situation theory. Technical Report Logic Group Preprint IULG-92-16, Indiana University, 1992.
- [52] J. Seligman. The logic of correct description. In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.
- [53] Y. Shoham. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*. The MIT Press, Cambridge, MA, 1988.
- [54] R. Turner. *Logics for Artificial Intelligence*. Ellis Horwood: Chichester, 1984.
- [55] M. Tzakova. Tableaux calculi for hybrid logics. In N. Murray, editor, *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 278–292. Springer Verlag, 1999.

Received February 10, 2000

Hybrid Logics. Characterization, Interpolation and Complexity

Carlos Areces	Patrick Blackburn	Maarten Marx
University of Amsterdam	Universität des Saarlandes	University of Amsterdam
The Netherlands	Germany	The Netherlands
carlos@wins.uva.nl	patrick@coli.uni-sb.de	marx@wins.uva.nl

March 3, 1999

Abstract

Hybrid languages are extended modal languages which can refer to (or even quantify over) worlds. The use of strong hybrid languages dates back to at least [Pri67], but recent work (for example [BS98, BT99]) has focussed on a more constrained system called $\mathcal{H}(\downarrow, @)$. The purpose of the present paper is to show in detail that $\mathcal{H}(\downarrow, @)$ is a modally natural system. We begin by studying its expressivity, and provide both model theoretic characterizations (via a restricted notion of Ehrenfeucht-Fraïssé game, and an enriched notion of bisimulation) and a syntactic characterization (in terms of bounded formulas). The key result to emerge is that $\mathcal{H}(\downarrow, @)$ corresponds precisely to the first-order fragment which is invariant for generated submodels. We further establish that $\mathcal{H}(\downarrow, @)$ has (strong) interpolation, and provide failure results in the finite variable fragments. We also show that weak interpolation holds for the sublanguage $\mathcal{H}(@)$, and provide complexity results for $\mathcal{H}(@)$ and other fragments and variants (the full logic being undecidable).

1 Introduction

In their simplest form, hybrid languages are modal languages which use *formulas* to refer to worlds. To build a simple hybrid language, take an ordinary language of propositional modal logic (built over some collection of propositional variables p, q, r , and so on) and add a second type of atomic formula. These new atoms are called *nominals*, and are typically written i, j and k . Both types of atom can be freely combined to form more complex formulas in the usual way; for example,

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q)$$

is a well formed formula. And now for the key idea: insist that *each nominal must be true at exactly one world in any model*. Thus a nominal names a world by being true there and nowhere else. This simple idea gives rise to richer logics (note, for example, that the previous formula is valid: if the antecedent is satisfied at a world m , then the unique world named by i must be accessible from m , and both p and q must be true there) and enables us to define classes of frames that ordinary modal languages cannot (we'll see some examples later).

Once the idea of using “formulas as terms” has been noted (Arthur Prior [Pri67], influenced by unpublished work of C. A. Meredith, seems to have been the first to grasp its potential) the way lies open for further enrichments. The most obvious is to regard nominals not as names but as *variables over individual worlds*, and to add quantifiers. That is, we now allow expressions like

$$\forall x. \Diamond(x \wedge \exists y. \Diamond(y \wedge \Diamond y))$$

to be formed. This sentence is satisfied at a world m if and only if from every world x that is accessible from m we can reach at least one reflexive world y . No formula with this property exists in ordinary modal languages, or even in modal languages enriched with nominals. Unsurprisingly, if we are allowed to quantify over worlds in this manner, it is straightforward to define hybrid languages that offer first-order expressivity over models. Early work on hybrid languages (notably that of Bull [Bul70] and Passy and Tinchev [PT85, PT91]) was largely concerned with such systems.

The idea of binding variables to worlds underlies much current work on hybrid languages, but for many purposes the $\forall$ binder is arguably too strong: languages which employ $\forall$ obscure the *locality* intuition central to Kripke semantics. Fundamental to Kripke semantics is the relativization of semantic evaluation to worlds. That is, to evaluate a modal formula we need to specify some world m (the *current* world) and begin evaluation *there*. The function of the modalities is to scan the worlds accessible from m , the worlds accessible from those worlds, and so on; in short, m is the starting point for step-wise local exploration of the model. Languages which allow variables to be bound to *arbitrary* worlds (some of which may not be reachable from m by chaining through the accessibility relation) don't mesh well with this intuition.

For this reason, recent work on hybrid languages has focussed on a local language called $\mathcal{H}(\downarrow, @)$. This extends the simplest type of hybrid language (propositional modal logic plus nominals) with two new mechanisms, $\downarrow$ and $@$. Now $\downarrow$ does bind variables to worlds, but (unlike $\forall$) it does so in an intrinsically *local* way:

The $\downarrow$ binder binds variables to the current world. In essence it enables us to create a name for the here-and-now.

The $@$ operator (which does *not* bind variables) is a natural counterpart to $\downarrow$. Whereas $\downarrow$ “stores” the current world (by binding a variable to it), $@$ enables us to “retrieve” worlds. More precisely, a formula of the form $@_x \varphi$ is an instruction to move to the world labeled by the variable x and evaluate φ there. Previous work on $\mathcal{H}(\downarrow, @)$ has concentrated on relating it to other hybrid languages [BS98], studying it axiomatically [BT99], and developing analytic proof techniques [Bla98, Tza98]. Taken together, this work suggests $\mathcal{H}(\downarrow, @)$ and certain of its sublanguages (notably $\mathcal{H}(@)$) are important systems. The purpose of the present paper is to demonstrate in detail that this impression is justified.

We do so as follows. After defining $\mathcal{H}(\downarrow, @)$ and noting some basic results in Section 2, we turn in Section 3 to the task of characterizing its expressivity. The key result to emerge is this: $\mathcal{H}(\downarrow, @)$ is not merely local, it is the language which *characterizes* locality. More precisely, $\mathcal{H}(\downarrow, @)$ corresponds to the fragment of first-order logic which is invariant under generated submodels. Previous discussions of $\mathcal{H}(\downarrow, @)$ have stressed that it is “modally natural”; our characterization confirms this impression and makes it precise. In Section 4 we discuss the consequences of this characterization for frame-definability, completeness, and tense logic. In Section 5, we show that $\mathcal{H}(\downarrow, @)$ is well-behaved in yet another way: it has the strong (arrow) interpolation property, and the sublanguage $\mathcal{H}(@)$ has weak interpolation

In Section 6 we turn to complexity. It is known that $\mathcal{H}(\downarrow, @)$ has an undecidable satisfaction problem (indeed, this is clear from the characterization result); but it is also known that $\mathcal{H}(@)$ is decidable, that this sublanguage can define more classes of frames than ordinary propositional modal logic can, and that it is proof-theoretically well-behaved. Intriguingly, these properties come at no computational cost: we shall show that $\mathcal{H}(@)$ has a PSPACE-complete satisfaction problem, just as ordinary propositional modal logic does. However we also show that there are limits on how far this result can be generalized: in particular we show that if we start with a language of tense logic, then adding even a single nominal moves the complexity of the local satisfaction problem from PSPACE to EXPTIME. We close the paper with a discussion of a key open problem.

The paper is largely self contained, but as the existing literature on hybrid languages is small and scattered it seems appropriate to give the reader a swift overview of what is available. First, two early papers on $\forall$ -based hybrid languages (namely [Bul70] and [PT91]) deserve to be far more widely read: both contain important technical ideas and interesting motivation for the use of hybrid languages. Second, some work has been done on very basic hybrid languages (that is, modal or tense languages enriched with nominals, but with no additional mechanisms such as $\downarrow$, $@$, or $\forall$); the main references here are [GG93] and [Bla93]. Third, while [BS98] and [BT99] are the basic references for $\mathcal{H}(\downarrow, @)$, an interesting discussion of $\downarrow$ as part of a stronger system can be found in [Gor96]. Finally, in addition to the proof theoretical investigations of [Bla98] and [Tza98], there is also [Sel91, Sel97]. Seligman's work deals with stronger ($\forall$ -based) systems, but many of the key ideas underlying hybrid deduction (in particular, the deductive significance of $@$) were first explored here, and they are still the only source we know of for discussion of natural deduction techniques for hybrid logics.

2 Preliminaries

In this section we define the syntax and semantics of $\mathcal{H}(\downarrow, @)$ and note some of its basic properties.

Definition 2.1 (Language) Let $PROP = \{p_1, p_2, \dots\}$ be a countable set of propositional variables, $NOM = \{i_1, i_2, \dots\}$ a countable set of nominals, and $WVAR = \{x_1, x_2, \dots\}$ a countable set of world variables. We assume that $PROP$, NOM and $WVAR$ are pairwise disjoint. We call $WSYM = NOM \cup WVAR$ the set of world symbols, and $ATOM = PROP \cup NOM \cup WVAR$ the set of atoms. The well-formed formulas of the hybrid language (over the signature $(PROP, NOM, WVAR)$) are

$$\varphi := \top \mid \perp \mid a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box\varphi \mid \downarrow x_j.\varphi \mid @_s\varphi$$

where $a \in ATOM$, $x_j \in WVAR$ and $s \in WSYM$. Let $\mathcal{L}$ be the set of all well-formed formulas. For $\varphi \in \mathcal{L}$, $PROP(\varphi)$, $NOM(\varphi)$ and $WVAR(\varphi)$ denote, respectively, the set of propositional variables, nominals, and world variables which occur in φ . $IP(\varphi)$ will denote $PROP(\varphi) \cup NOM(\varphi)$, and will be called the language of φ .

In what follows we assume that a signature $(PROP, NOM, WVAR)$, and hence $\mathcal{L}$, has been fixed. We usually write p, q and r for propositional variables, i, j and k for nominals, and x, y and z for world variables. As usual, $\Diamond\varphi$ is defined to be $\neg\Box\neg\varphi$.

Note that all three types of atomic symbol are *formulas*. Further, note that the above syntax is simply that of ordinary unimodal propositional modal logic extended by the clauses

for $\downarrow x_j.\varphi$ and $@_s\varphi$. Finally, the difference between nominals and world variables is simply this: nominals cannot be bound by $\downarrow$, whereas world variables can. In fact, nominals could be dispensed with (it is always possible to make do with free world variables instead) but for some purposes it is useful to have a special kind of world symbol that can't be accidentally bound.

Definition 2.2 The notions of free and bound world variable are defined in the manner familiar from first-order logic, with $\downarrow$ as the only binding operator. Similarly, other syntactic notions (such as substitution, and of a world symbol t being substitutable for x in φ) are defined as the corresponding notions in first-order logic. We use $\varphi[t/s]$ to denote the formula obtained by replacing all free instances of the world symbol t by the world symbol s .

A sentence is a formula containing no free world variables. Furthermore, a formula is pure if it contains no propositional variables, and nominal-free if it contains no nominals.

Definition 2.3 (Semantics) A (hybrid) model $\mathfrak{M}$ for $\mathcal{L}$ is a triple $\mathfrak{M} = \langle M, R, V \rangle$ such that M is a non-empty set, R a binary relation on M , and $V : PROP \cup NOM \rightarrow Pow(M)$ such that for all nominals $i \in NOM$, $V(i)$ is a singleton subset of M . (We use calligraphic letters $\mathfrak{M}$ for models, italic roman M for their domains. When necessary, we use sub or supra-indices.) We usually call the elements of M worlds (though sometimes we call them times or states), and we call R the accessibility relation and V the valuation. A frame is a pair $\mathfrak{M} = \langle M, R \rangle$; that is, a frame is a model without a valuation.

An assignment g for $\mathfrak{M}$ is a mapping $g : WVAR \rightarrow M$. Given an assignment g , we define the assignment g_m^i by $g_m^i(x_j) = g(x_j)$ for $j \neq i$ and $g_m^i(x_i) = m$. We say that g_m^i is an x_i -variant of g .

Let $\mathfrak{M} = \langle M, R, V \rangle$ be a model, $m \in M$, and g an assignment. For any atom a , let $[V, g](a) = \{g(a)\}$ if a is a world variable, and $V(a)$ otherwise. Then the forcing relation is defined as follows

$$\begin{array}{ll} \mathfrak{M}, g, m \Vdash \top & \\ \mathfrak{M}, g, m \nVdash \perp & \\ \mathfrak{M}, g, m \Vdash a & \text{iff } m \in [V, g](a), a \in ATOM \\ \mathfrak{M}, g, m \Vdash \neg\varphi & \text{iff } \mathfrak{M}, g, m \nVdash \varphi \\ \mathfrak{M}, g, m \Vdash \varphi \wedge \psi & \text{iff } \mathfrak{M}, g, m \Vdash \varphi \text{ and } \mathfrak{M}, g, m \Vdash \psi \\ \mathfrak{M}, g, m \Vdash \Box\varphi & \text{iff } \forall m' (Rmm' \Rightarrow \mathfrak{M}, g, m' \Vdash \varphi) \\ \mathfrak{M}, g, m \Vdash \downarrow x_i.\varphi & \text{iff } \mathfrak{M}, g_m^i, m \Vdash \varphi \\ \mathfrak{M}, g, m \Vdash @_s\varphi & \text{iff } \mathfrak{M}, g, m' \Vdash \varphi, \text{ where } [V, g](s) = \{m'\}, s \in WSYM. \end{array}$$

When $\mathfrak{M}$ and g are understood by context we will simply write $m \Vdash \varphi$ for $\mathfrak{M}, g, m \Vdash \varphi$. We write $\mathfrak{M}, g \Vdash \varphi$ iff for all $m \in M$, $\mathfrak{M}, g, m \Vdash \varphi$. We write $\mathfrak{M} \models \varphi$ iff for all g , $\mathfrak{M}, g \Vdash \varphi$.

The first six clauses are essentially the standard Kripke forcing relation for propositional modal logic; in fact the only difference is that whereas the standard definition relativizes semantic evaluation to worlds m , we relativize to variable assignments g as well. Note that the clause for atoms covers all three types of symbol (propositional variables, nominals, and world variables) and that given any model $\mathfrak{M}$ and assignment g , any world symbol (whether it is a nominal or a world variable) will be forced at a *unique* world; this is an immediate consequence of the way we defined valuations and assignments. As promised in the introduction, $\downarrow$ binds

world variables to the world where evaluation is being performed (the *current world*), and $@_s$ shifts evaluation to the world named by s . Just as in first-order logic, if φ is a *sentence* it is irrelevant which assignment g is used to perform evaluation: $\mathfrak{M}, g, m \Vdash \varphi$ for *some* assignment g iff $\mathfrak{M}, g, m \Vdash \varphi$ for *all* assignments g . Hence for sentences the relativization to assignments of the forcing relation can be dropped, and we simply write $\mathfrak{M}, m \Vdash \varphi$ instead of $(\forall g) \mathfrak{M}, g, m \Vdash \varphi$.

A formula φ is *satisfiable* if there is a model $\mathfrak{M}$, an assignment g on $\mathfrak{M}$, and a world $m \in M$ such that $\mathfrak{M}, g, m \Vdash \varphi$. A formula φ is *valid* if for all models $\mathfrak{M}$, $\mathfrak{M} \models \varphi$. A formula φ is a *local consequence* of a set of formulas T if for some finite subset $\{\varphi_1, \dots, \varphi_n\}$ of T , $\varphi_1 \wedge \dots \wedge \varphi_n \rightarrow \varphi$ is valid. A formula φ is a *global consequence* of a set of formulas T if for all models $\mathfrak{M}$, $\mathfrak{M} \models \varphi$ only if for all $\psi \in T$, $\mathfrak{M} \models \psi$. We denote local consequence by $T \models \varphi$ and global consequence by $T \models^{gl} \varphi$. As in ordinary propositional modal logic, local consequence is strictly stronger than global consequence.

$\mathcal{H}(\downarrow, @)$ offers us considerable expressive power over models. For example we can define the *Until* operator:

$$Until(\varphi, \psi) := \downarrow x. \Diamond \downarrow y. @_x (\Diamond(y \wedge \varphi) \wedge \Box(\Diamond y \rightarrow \psi)).$$

Note how this works: we name the current world x , use $\Diamond$ to move to an accessible world, which we name y , and then use $@$ to jump us back to x . We then use the modalities to insist that (1) φ holds at the world named y , and (2) ψ holds at all successors of the current world that precede this y -labeled world.

But there is an obvious (and modally natural) limit to the expressive power $\mathcal{H}(\downarrow, @)$ gives us: any nominal-free sentence is preserved under the formation of *point-generated* (or *rooted*) submodels. That is, if a sentence φ is satisfied at a world m in a model $\mathfrak{M}$, and we form a submodel $\mathfrak{M}_m$ by discarding from $\mathfrak{M}$ all the worlds that are *not* reachable by making a finite (possibly empty) sequence of transitions from m , then $\mathfrak{M}_m$ also satisfies φ at m . (The key point to observe is that in any subformula of φ of the form $@_t \psi$, t must be a world variable bound by some previous occurrence of $\downarrow$. As $\downarrow$ binds to the current world, t is bound to some world in the submodel generated by m , thus φ is unaffected by the restriction to $\mathfrak{M}_m$.) That is, $\mathcal{H}(\downarrow, @)$ is genuinely local: only reachable worlds are relevant to semantic evaluation. In the following section we shall generalize this observation (we have not merely preservation, but *invariance*) and show that it characterizes the expressivity of $\mathcal{H}(\downarrow, @)$.

$\mathcal{H}(\downarrow, @)$ also offers us considerable expressive power with respect to *frames*. Modal logicians like to view modal languages as tools for talking about frames, and they do so via the concept of frame validity:

Definition 2.4 A formula φ is valid on a frame $\mathfrak{F} = \langle M, R \rangle$ if for every valuation V on $\mathfrak{F}$, and every assignment g on $\mathfrak{F}$, and every $m \in M$, $\langle \mathfrak{F}, V \rangle, g, m \Vdash \varphi$. A formula is valid on a class of frames F if it is valid on every frame $\mathfrak{F}$ in F . A formula φ defines a class of frames if it is valid on precisely the frames in F , and it defines a property of frames (for example, transitivity) if it defines the class of frames with that property.

Many interesting properties are definable using pure, nominal-free, sentences:

$\downarrow x. \Box \neg x$	<i>Irreflexivity</i>
$\downarrow x. \Box \Box \neg x$	<i>Asymmetry</i>
$\downarrow x. \Box (\Diamond x \rightarrow x)$	<i>Antisymmetry</i>
$\downarrow x. \Box \downarrow y. @_x \Diamond y$	<i>Density</i>
$\downarrow x. \Box \Box \downarrow y. @_x \Diamond y$	<i>Transitivity</i>
$\downarrow x. \Diamond \downarrow y. @_x (\Box \Box \neg y \wedge \Box \downarrow z. @_y (z \vee \Diamond z))$	<i>Right-Discreteness</i>

With the exception of transitivity and density, none of these properties are definable in ordinary modal logic. In Section 4 we shall characterize the classes of frames that pure, nominal-free, sentences can define.

[BT99] provides the following complete axiom system for $\mathcal{H}(\downarrow, @)$:

Definition 2.5 (Axiomatization) Let φ, ψ be formulas, v a metavariable over world variables and s, t metavariables over world symbols. The hybrid logic $K[\mathcal{H}(\downarrow, @)]$ is the smallest subset of $\mathcal{L}$ containing all instances of propositional tautologies, all instances of the following axiom schemes and closed under the following deduction rules

- K.* $\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$.
- N.* $\vdash \varphi \Rightarrow \vdash \Box\varphi$.
- MP.* $\vdash \varphi \rightarrow \psi, \vdash \varphi \Rightarrow \vdash \psi$.
- Q1.* $\downarrow v. (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \downarrow v. \psi)$, φ without free occurrences of v .
- Q2.* $\downarrow v. \varphi \rightarrow (s \rightarrow \varphi[v/s])$, s substitutable for v in φ .
- Q3.* $\downarrow v. (v \rightarrow \varphi) \rightarrow \downarrow v. \varphi$.
- Self Dual.* $\downarrow v. \varphi \leftrightarrow \neg \downarrow v. \neg \varphi$.
- N_↓.* $\vdash \varphi \Rightarrow \vdash \downarrow v. \varphi$.
- K_@.* $@_s(\varphi \rightarrow \psi) \rightarrow (@_s\varphi \rightarrow @_s\psi)$.
- Self Dual_@.* $@_s\varphi \leftrightarrow \neg @_s\neg\varphi$.
- Introduction.* $(s \wedge \varphi) \rightarrow @_s\varphi$.
- Label.* $@_ss$.
- Nom.* $@_st \rightarrow (@_t\varphi \rightarrow @_s\varphi)$.
- Swap.* $@_st \leftrightarrow @_ts$.
- Scope.* $@_t @_s\varphi \leftrightarrow @_s\varphi$.
- N_@.* $\vdash \varphi \Rightarrow \vdash @_s\varphi$.
- Back.* $\Diamond @_s\varphi \rightarrow @_s\varphi$.
- Bridge.* $(\Diamond s \wedge @_s\varphi) \rightarrow \Diamond\varphi$.
- Paste-0.* $\vdash @_s(t \wedge \varphi) \rightarrow \psi \Rightarrow \vdash @_s\varphi \rightarrow \psi, t \in WSYM \setminus WSYM(\varphi, \psi, s)$.
- Paste-1.* $\vdash @_s\Diamond(t \wedge \varphi) \rightarrow \psi \Rightarrow \vdash @_s\Diamond\varphi \rightarrow \psi, t \in WSYM \setminus WSYM(\varphi, \psi, s)$.

We use this axiomatization to help prove strong interpolation in Section 5. For this purpose, the following derived theorems will be useful (their derivation from the above axiomatization is a simple exercise; see [BT99] (Lemma 4.1) and [BT98] (Lemma 7) for details).

Proposition 2.6 (Derived Theorems and Properties)

- i. $K\downarrow. \downarrow v.(\varphi \rightarrow \psi) \rightarrow (\downarrow v. \varphi \rightarrow \downarrow v. \psi)$.
- ii. $Distr_{@}. @_s(\varphi \wedge \psi) \leftrightarrow (@_s \varphi \wedge @_s \psi)$.
- iii. *Free variables and nominals.* If $\vdash \varphi$ and i is a nominal in φ , then for some world variable x not occurring in φ , $\vdash \downarrow x. \varphi[i/x]$. If $\vdash \varphi$ and x is a free variable in φ , then for some nominal i not occurring in φ , $\vdash \varphi[x/i]$.

But we are interested in this axiomatization for another reason. The completeness result proved in [BT99] is very general: not only does this axiomatization generate all valid formulas, but it automatically extends to many stronger logics. In particular, if we add a pure, nominal-free, sentence φ as an additional axiom, the resulting system is strongly complete with respect to the class of frames that φ defines. In Section 4 we shall characterize the completeness results covered by such extensions.

Before starting our investigations, one final remark. We have defined $\mathcal{H}(\downarrow, @)$ to be an extension of unimodal propositional modal logic. But of course, it would have been equally straightforward to extend a multimodal language (that is, a language containing an indexed collection of modalities, each interpreted by a separate relation) or a language of tense logic (that is, propositional modal logic enriched with modalities which scan the converse of the relation R) with nominals, variables, $@$ and $\downarrow$ in exactly the same way. For the most part we will work with the above version of $\mathcal{H}(\downarrow, @)$ (most results go through for other formulations essentially unchanged) but sometimes it will be interesting to switch to a richer underlying modal language, especially when we discuss computational complexity in Section 6.

3 Characterizing $\mathcal{H}(\downarrow, @)$

In this section we characterize (the first-order language corresponding to) $\mathcal{H}(\downarrow, @)$. We begin by providing a *syntactic* characterization. In particular, we shall first extend the *standard translation* ST of modal logic into first-order logic (cf. [Ben83]) to $\mathcal{H}(\downarrow, @)$. It will be clear that the range of our translation lies in a certain *bounded fragment*, and we shall define a reverse translation HT which maps the bounded fragment back into the hybrid language. Thus we are free to think either in terms of $\mathcal{H}(\downarrow, @)$ or the corresponding bounded fragment.

But how are these languages characterized *semantically*? It should be clear that $\mathcal{H}(\downarrow, @)$ is a genuine hybrid of modal and first-order ideas (after all, it combines Kripke semantics with the idea of binding variables to worlds) thus there are two obvious ways to proceed. The first is essentially first-order: we could look for a weaker notion of Ehrenfeucht-Fraïssé game. The second is essentially modal: we could try looking for a stronger notion of bisimulation. We shall pursue *both* options. As we shall see, both yield natural notions of equivalence between models, and by relating them (and drawing on our syntactic characterization) we can provide a detailed picture of what $\mathcal{H}(\downarrow, @)$ offers.

3.1 Translations

We focus on two kinds of signature for first-order logic with equality. First we have *modal* signatures (familiar from modal correspondence theory [Ben83]) which consist of one binary predicate R , countably many unary predicates, and no constant symbols. It will be convenient to make the set of first-order variables at our disposal explicit in the signature

(just as we did when we defined hybrid signatures in Definition 2.1) thus, a modal signature has the form $\langle \{R\} \cup \text{UN-REL}, \{\}, \text{VAR} \rangle$. A *hybrid* signature is an expansion of the modal signature with countably many constant symbols, thus hybrid signatures have the form $\langle \{R\} \cup \text{UN-REL}, \text{CONS}, \text{VAR} \rangle$. Note that any hybrid model $\mathfrak{M} = \langle M, R, V \rangle$ can be regarded as a first-order model with domain M , for the accessibility relation R can be used to interpret the binary predicate R , unary predicates can be interpreted by the subsets V assigns to propositional variables, and constants (if any) can be interpreted by the worlds that nominals name. So we let the context determine whether we are thinking of first-order or hybrid models, and continue to use the notation $\mathfrak{M} = \langle M, R, V \rangle$.

We first extend the well-known *standard translation* to $\mathcal{H}(\downarrow, @)$. The translation ST from the hybrid language over $\langle \text{PROP}, \text{NOM}, \text{WVAR} \rangle$ into first-order logic over the signature $\langle \{R\} \cup \{P_j \mid p_j \in \text{PROP}\}, \text{NOM}, \text{WVAR} \cup \{x, y\} \rangle$ is defined by mutual recursion between two functions ST_x and ST_y . Recall that $\varphi[x/y]$ means “replace all free instances of x by y ”.

$$\begin{array}{ll|ll}
 ST_x(p_j) & = & P_j(x), p_j \in \text{PROP}. & ST_y(p_j) & = & P_j(y), p_j \in \text{PROP}. \\
 ST_x(i_j) & = & x = i_j, i_j \in \text{NOM}. & ST_y(i_j) & = & y = i_j, i_j \in \text{NOM}. \\
 ST_x(x_j) & = & x = x_j, x_j \in \text{WVAR}. & ST_y(x_j) & = & y = x_j, x_j \in \text{WVAR}. \\
 ST_x(\neg\varphi) & = & \neg ST_x(\varphi). & ST_y(\neg\varphi) & = & \neg ST_y(\varphi). \\
 ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\psi). & ST_y(\varphi \wedge \psi) & = & ST_y(\varphi) \wedge ST_y(\psi). \\
 ST_x(\Diamond\varphi) & = & \exists y. (Rxy \wedge ST_y(\varphi)). & ST_y(\Diamond\varphi) & = & \exists x. (Ryx \wedge ST_x(\varphi)). \\
 ST_x(\downarrow x_j. \varphi) & = & (ST_x(\varphi))[x_j/x]. & ST_y(\downarrow x_j. \varphi) & = & (ST_y(\varphi))[x_j/y]. \\
 ST_x(@_s \varphi) & = & (ST_x(\varphi))[x/s]. & ST_y(@_s \varphi) & = & (ST_y(\varphi))[y/s].
 \end{array}$$

For m an element in the domain of a given model $\mathfrak{M}$ we will often write $ST_m(\varphi)$ as shorthand for $ST_x(\varphi)[m]$. This translation differs from the one given in [BS98]; these authors handle $\downarrow$ and $@$ as follows:

$$\begin{array}{ll}
 ST_x(\downarrow x_j. \varphi) & = \exists x_j. (x = x_j \wedge ST_x(\varphi)). \\
 ST_x(@_s \varphi) & = ST_s(\varphi), s \in \text{NOM} \cup \text{VAR}.
 \end{array}$$

Now the [BS98] translation makes the quantificational effect of $\downarrow$ clear, but our translation draws attention to another perspective: in adding $\downarrow$ and $@$ we have in effect enriched the modal language with an *explicit substitution operator*. Such operators are used in the study of cylindric algebras, and were added to cylindric modal logic in [Ven94].

The link between $\downarrow$ and explicit substitution can be made even more clear if we expand the first-order language with an explicit substitution operator (like s_j^i in the theory of cylindric algebras) and adjust our definition of ST to take advantage of it. We do this as follows. Add the following clause to the grammar generating the first-order language:

if φ is a formula and x, y are variables, then $S_y^x \varphi$ is a formula.

Interpret S_y^x as follows:

$$\mathfrak{M} \models S_y^x \varphi[g] \Leftrightarrow \begin{cases} \mathfrak{M} \models \varphi[g] & \text{for } x = y \\ \mathfrak{M} \models \varphi[g_y^x] & \text{for } x \neq y. \end{cases}$$

Clearly $S_y^x \varphi$ and $\varphi[x/y]$. This enrichment of first-order logic can be axiomatized by adding the following axiom schemata to a complete axiomatization of first-order logic with equality:

$$\begin{array}{ll}
 S_x^x \varphi & \leftrightarrow \varphi \\
 S_y^x \varphi & \leftrightarrow \exists x. (x = y \wedge \varphi) \quad \text{for } x \neq y.
 \end{array}$$

And now we can give transparent translations of $\downarrow$ and $@$:

$$\begin{aligned} ST_x(\downarrow x_j.\varphi) &= S_x^{xj}ST_x(\varphi) \\ ST_x(@_s\varphi) &= S_s^sST_x(\varphi). \end{aligned}$$

Note that theorems like $\downarrow v.@_v\varphi \leftrightarrow \downarrow v.\varphi$ follow more or less immediately, for $ST_x(\downarrow v.@_v\varphi) = S_x^vS_v^xST_x(\varphi)$, but $S_x^vS_v^x\varphi \equiv S_x^vS_x^v\varphi = S_x^v\varphi$, which is equivalent to $S_x^vST_x(\varphi)$. However we shall stick with our original formulation of ST in what follows.

Proposition 3.1 (ST Preserves Truth) *Let φ be a hybrid formula, then for all hybrid models $\mathfrak{M}$, $m \in M$ and assignments g , $\mathfrak{M}, g, m \models \varphi$ iff $\mathfrak{M} \models ST_m(\varphi)[g]$.*

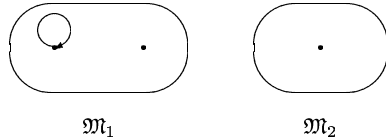
PROOF. A straightforward extension of the induction familiar from ordinary modal logic. The only cases which are new are $ST_x(\downarrow x_j.\varphi)$ and $ST_x(@_s\varphi)$. But $\mathfrak{M}, g, m \models \downarrow x_j.\varphi$, iff $\mathfrak{M}, g_m^{xj}, m \models \varphi$, by IH iff $\mathfrak{M} \models ST_m(\varphi)[g_m^{xj}]$, iff $\mathfrak{M} \models (ST_m(\varphi))[x_j/x] [g]$. The argument for $ST_x(@_s\varphi)$ is similar. QED

Now for the interesting question: what is the *range* of ST ? In fact it belongs to a *bounded* fragment of first-order logic. Given a first-order signature $\langle \{R\} \cup \text{UN-REL}, \text{CONS}, \text{VAR} \rangle$ we define the *bounded* fragment as the set of formulas generated by the following grammar:

$$\varphi = Rtt' \mid P_jt \mid t = t' \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \exists x_i.(Rtx_i \wedge \varphi) \text{ (for } x_i \neq t).$$

where $x_i \in \text{VAR}$ and $t, t' \in \text{VAR} \cup \text{CONS}$.

The side-condition on the generation of existentially quantified formulas is crucial: it prevents sentences like $\exists x.(Rxx \wedge x = x)$ from falling into the fragment. The sentence $\exists x.Rxx$ is probably the simplest example of a first-order sentence which is *not invariant for generated submodels* (or subframes). In fact it is not even *preserved* under the formation of generated submodels, for it is true in the model $\mathfrak{M}_1$ but not in the generated submodel $\mathfrak{M}_2$:



Clearly ST generates formulas in the bounded fragment. Crucially, however, we can also translate the bounded fragment into $\mathcal{H}(\downarrow, @)$. The translation HT from the bounded fragment over $\langle \{R\} \cup \text{UN-REL}, \text{CONS}, \text{VAR} \rangle$ into the hybrid language over $\langle \text{UN-REL}, \text{CONS}, \text{VAR} \rangle$ is defined as follows:

$$\begin{aligned} HT(Rtt') &= @_t\Diamond t'. \\ HT(P_jt) &= @_j\varphi_j. \\ HT(t = t') &= @_t t'. \\ HT(\neg\varphi) &= \neg HT(\varphi). \\ HT(\varphi \wedge \psi) &= HT(\varphi) \wedge HT(\psi). \\ HT(\exists v.(Rtv \wedge \varphi)) &= @_t\Diamond \downarrow v.HT(\varphi). \end{aligned}$$

(Here $t, t' \in \text{VAR} \cup \text{CONS}$.)

By construction, $HT(\varphi)$ is a hybrid formula, but furthermore it is a boolean combination of $@$ -formulas (formulas whose main operator is $@$). So, because $@$ commutes with all the booleans (recall the Self Dual $@$ axiom and Proposition 2.6), it is equivalent to a formula whose main operator is $@$. We can now prove the following strong truth preservation result. Recall that $\mathfrak{M}, g \Vdash \phi$, abbreviates “(for all $m \in M$) $\mathfrak{M}, g, m \models \phi$ ”.

Proposition 3.2 (HT Preserves Truth) *Let φ be a bounded formula. Then for every first-order model $\mathfrak{M}$ and for every assignment g , $\mathfrak{M} \models \varphi[g]$ iff $\mathfrak{M}, g \Vdash HT(\varphi)$.*

PROOF. The proof uses the following fact about formulas whose main operator is $@$:

$$(\text{there exists an } m), \mathfrak{M}, g, m \Vdash @_s\varphi \text{ if and only if } \mathfrak{M}, g \Vdash @_s\varphi.$$

Again there is only one interesting case: $HT(\exists v.(Rtv \wedge \varphi))$. Now $\mathfrak{M} \models \exists v.(Rtv \wedge \varphi)[g]$ iff $\mathfrak{M} \models (Rtv \wedge \varphi)[g_m^v]$ for some $\mathbf{m} \in M$. Let $\mathbf{t}$ be the interpretation of t in $\mathfrak{M}$ under g_m^v . Because of the restriction on variables in bounded quantification, $t \neq v$, whence $\mathbf{t}$ is also the interpretation of t in $\mathfrak{M}$ under g . So $Rt\mathbf{m}$ holds in $\mathfrak{M}$ and $\mathfrak{M} \models \varphi[g_m^v]$. By the inductive hypothesis, $\mathfrak{M}, g_m^v \Vdash HT(\varphi)$ iff $\mathfrak{M}, g, \mathbf{m} \Vdash \downarrow v.HT(\varphi)$, iff $\mathfrak{M}, g, \mathbf{t} \Vdash \Diamond \downarrow v.HT(\varphi)$ iff $\mathfrak{M}, g, \mathbf{t} \Vdash @_t\Diamond \downarrow v.HT(\varphi)$ iff $\mathfrak{M}, g \Vdash @_t\Diamond \downarrow v.HT(\varphi)$. QED

As simple corollaries we have:

Corollary 3.3 *Let $\varphi(x)$ be a bounded formula with only x free, then for all models $\mathfrak{M}$ and for all $m \in M$, $\mathfrak{M} \models \varphi(m)$ iff $\mathfrak{M}, m \Vdash \downarrow x.HT(\varphi)$.*

Corollary 3.4 *Let φ be a first-order formula in the hybrid signature. Then the following are equivalent.*

- i. φ is equivalent to the standard translation of a hybrid formula.
- ii. φ is equivalent to a formula in the bounded fragment.

Moreover, there are effective translations between $\mathcal{H}(\downarrow, @)$ and the bounded fragment.

3.2 Generated back-and-forth systems

We now turn to the problem of providing semantic characterizations of $\mathcal{H}(\downarrow, @)$ (or equivalently, of the bounded fragment). In this section we shall adopt an essentially first-order approach: we define *generated back-and-forth systems*, basically a restricted form of Ehrenfeucht-Fraïssé game, and link it to the concept of *generated submodels*.

Definition 3.5 (Partial Isomorphism) *Let $\mathfrak{M}$ and $\mathfrak{N}$ be two hybrid models. A function h from a subset of M to a subset of N is called a partial isomorphism if*

1. h is a bijection;
2. for all x in the domain of h , for all $a \in \text{PROP} \cup \text{NOM}$, $x \in V^{\mathfrak{M}}(a)$ iff $h(x) \in V^{\mathfrak{N}}(a)$;
3. for all x, y in the domain of h , $R^{\mathfrak{M}}xy$ iff $R^{\mathfrak{N}}h(x)h(y)$.

Generated back-and-forth systems Let $\mathfrak{M}$ and $\mathfrak{N}$ be two first-order models in the hybrid signature. A generated back-and-forth system between $\mathfrak{M}$ and $\mathfrak{N}$ is a non-empty family F of finite partial isomorphisms between $\mathfrak{M}$ and $\mathfrak{N}$ satisfying the following two extension rules:

($\Diamond$ -extension)

- (forth) if $h \in F$, x in its domain and $R^{\mathfrak{M}}xy$, then $h \cup \{\langle y, y' \rangle\} \in F$ for some $y' \in N$.
- (back) if $h \in F$, x in its range and $R^{\mathfrak{N}}xy$, then $h \cup \{\langle y', y \rangle\} \in F$ for some $y' \in M$.

(nominal extension)

- (forth) if $h \in F$ and there exists an $x \in M$ such that $V^{\mathfrak{M}}(i) = \{x\}$ for some nominal i , then there exists an $x' \in N$ such that $h \cup \{\langle x, x' \rangle\} \in F$.
- (back) a similar condition backwards.

Let $m \in M$ and $n \in N$. Then $(\mathfrak{M}, m) \equiv_R (\mathfrak{N}, n)$ means that there exists a generated back-and-forth system between $\mathfrak{M}$ and $\mathfrak{N}$ which connects m and n . More generally, let $\bar{m}$ ($\bar{n}$) be a finite M -tuple (N -tuple). Then $(\mathfrak{M}, \bar{m}) \equiv_R (\mathfrak{N}, \bar{n})$ means that there exists a generated back-and-forth system between $\mathfrak{M}$ and $\mathfrak{N}$, containing a partial isomorphism which sends $m(i)$ to $n(i)$.

Note how closely this definition follows the familiar one from first-order logic (cf. e.g., [Hod93]). In fact, if we think of such a system as describing an Ehrenfeucht-Fraïssé game, then the sole difference is that in the “generated back-and-forth game” the universal player ($\forall$ elard) must choose his moves from R -successors or worlds named by a nominal, whereas he can play whatever he likes in the full first-order game. Unsurprisingly, restricting the play to *accessible* worlds means that generated back-and-forth systems and *generated submodels* are closely connected.

Definition 3.6 (Generated Submodel) Let $\mathfrak{M} = \langle M, R, V \rangle$ be a hybrid model and $S \subseteq M$. Let NOM denote the subset of M whose elements are in the interpretation of some nominal. The submodel of $\mathfrak{M}$ generated by S is the substructure of $\mathfrak{M}$ with domain $\{m \in M \mid \exists s \in S \cup NOM (R^*sm)\}$ (R^* is the reflexive and transitive closure of R). This is also called the S -generated submodel of $\mathfrak{M}$.

Note that if $NOM = \emptyset$, we obtain the familiar modal notion of a generated submodel; and that if in addition S is a singleton set, we have the usual modal notion of a point-generated (or rooted) submodel.

We now define two notions of invariance. The first is taken from [Ben83]. A first-order formula $\varphi(\bar{x})$ in free variables $\bar{x}$ in a signature with one binary relation R , unary predicates and constants (and equality) is *invariant for generated submodels* if for all models $(\mathfrak{M}, \bar{m})$ and $(\mathfrak{M}', \bar{m}')$ such that $\mathfrak{M}'$ is the $\bar{m}$ -generated submodel of $\mathfrak{M}$,

$$\mathfrak{M} \models \varphi(\bar{m}) \text{ if and only if } \mathfrak{M}' \models \varphi(\bar{m}').$$

In a similar spirit, we shall say that a first-order formula $\varphi(\bar{x})$ in the same signature is invariant for generated back-and-forth systems if for all models $(\mathfrak{M}, \bar{m})$ and $(\mathfrak{N}, \bar{n})$, $(\mathfrak{M}, \bar{m}) \equiv_R (\mathfrak{N}, \bar{n})$ implies

$$\mathfrak{M} \models \varphi(\bar{m}) \text{ if and only if } \mathfrak{N} \models \varphi(\bar{n}).$$

Theorem 3.7 Let $\varphi(\bar{x})$ be a first-order formula in the hybrid signature. Then the following are equivalent.

- i. $\varphi(\bar{x})$ is equivalent to a formula in the bounded fragment.
- ii. $\varphi(\bar{x})$ is invariant for generated submodels.
- iii. $\varphi(\bar{x})$ is invariant for generated back-and-forth systems.

PROOF.

i. $\Rightarrow$ ii. is obvious.

ii. $\Rightarrow$ iii. First note that $\varphi(\bar{x})$ is invariant for generated submodels if and only if $\neg\varphi(\bar{x})$ is. Now suppose $\varphi(\bar{x})$ is invariant for generated submodels but *not* preserved under generated back-and-forth systems. Then we have models $(\mathfrak{M}, \bar{m})$ and $(\mathfrak{N}, \bar{n})$, a generated back-and-forth system linking $\bar{m}$ and $\bar{n}$, and $\mathfrak{M} \models \varphi(\bar{m})$ while $\mathfrak{N} \models \neg\varphi(\bar{n})$.

Let $\mathfrak{M}'$ ($\mathfrak{N}'$) be the $\bar{m}$ - ($\bar{n}$) generated submodel of $\mathfrak{M}$ ($\mathfrak{N}$). Then still $\mathfrak{M}' \models \varphi(\bar{m})$ and $\mathfrak{N}' \models \neg\varphi(\bar{n})$, by invariance. Clearly $(\mathfrak{M}', \bar{m}) \equiv_R (\mathfrak{N}', \bar{n})$. But then $(\mathfrak{M}', \bar{m})$ and $(\mathfrak{N}', \bar{n})$ have the same first-order theory by the following argument. Because $(\mathfrak{M}', \bar{m}) \equiv_R (\mathfrak{N}', \bar{n})$ holds, Eloise has a winning strategy in all games where $\forall$ elard only plays *immediate* R -successors or points named by a nominal. But since the models are generated, in the first-order back-and-forth game he can only play worlds which are accessible by a finite R -transition from either the root or one of the named worlds. But then she can compute a winning answer for the classic Ehrenfeucht-Fraïssé from her winning generated back-and-forth strategy. This contradicts the claim that $\mathfrak{M}' \models \varphi(\bar{m})$ and $\mathfrak{N}' \models \neg\varphi(\bar{n})$.

iii. $\Rightarrow$ i. We use a van Benthem style diagram-chasing argument [Ben96]. We only provide the outline. Let $\varphi(\bar{x})$ be as in the hypothesis and $BC(\varphi(\bar{x}))$ the bounded consequences of $\varphi(\bar{x})$ (that is, the consequences of $\varphi(\bar{x})$ that belong to the bounded fragment). We will show that $BC(\varphi(\bar{x})) \models \varphi(\bar{x})$, from which the result follows by compactness. (In this notation we interpret the $\bar{x}$ as constants, or equivalently we use the local version of first-order consequence, cf. [End72].)

If $BC(\varphi(\bar{x}))$ is inconsistent we are done. Otherwise, let $\mathfrak{M}, \bar{m}$ be a model of $BC(\varphi(\bar{x}))$ and $\mathfrak{N}, \bar{n}$ be a model of $\varphi(\bar{x})$ together with the bounded theory of $\mathfrak{M}, \bar{m}$. (Such a model can easily be shown to exist.) Take ω -saturated extensions $\mathfrak{M}^+, \bar{m}$ and $\mathfrak{N}^+, \bar{n}$. Create a family F of finite functions between M^+ and N^+ as follows: $f : \bar{x} \rightarrow \bar{y}$ is in F iff $\mathfrak{M}^+, \bar{x}$ and $\mathfrak{N}^+, \bar{y}$ make the same bounded formulas true. It is easy to show that F is a generated back and forth system linking $\bar{m}$ and $\bar{n}$. Now we can start diagram chasing: $\mathfrak{N} \models \varphi(\bar{n})$ then (by elementary extension) $\mathfrak{N}^+ \models \varphi(\bar{n})$, then (by invariance) $\mathfrak{M}^+ \models \varphi(\bar{m})$, then (passing to an elementary submodel) $\mathfrak{M} \models \varphi(\bar{m})$ as desired. QED

3.3 Hybrid bisimulations

We have just seen that by weakening the notion of an Ehrenfeucht-Fraïssé game we can link the bounded fragment (and hence $\mathcal{H}(\downarrow, @)$) with generated submodels. But in spite of its binding apparatus, $\mathcal{H}(\downarrow, @)$ has a distinctly modal flavor. Is it not also possible to strengthen the notion of *bisimulation* (the standard notion of equivalence between models used in modal logic) with clauses for $\downarrow$ and $@$, and so characterize $\mathcal{H}(\downarrow, @)$ in intrinsically modal terms? That's what we will do in this section. The approach has an advantage over the use of generated back-and-forth systems: preservation results can be easily obtained for reducts as well.

Recall that for ordinary propositional modal logics, bisimulations are non-empty binary relations linking the domains of models, with the restriction that only worlds with identical atomic information and matching accessibility relations should be connected (see Definition 3.7 [Ben83]; here bisimulations are called p -relations). Now, if we want to extend this notion to $\mathcal{H}(\downarrow, @)$, we need to take care of assignments to world variables as well. To this end, hybrid bisimulations will not simply link worlds, rather they will link pairs $(\bar{m}, m)$, where m is a world and $\bar{m}$ is an assignment. We start by defining k -bisimulations, which are the correct notion of bisimulation for formulas φ such that $\text{WVAR}(\varphi) \subseteq \{x_1, \dots, x_k\}$.

k -bisimulation. Let $\mathfrak{M}$ and $\mathfrak{N}$ be two hybrid models. Let $\stackrel{k}{\sim}$ be a binary relation between ${}^kM \times M$ and ${}^kN \times N$. So $\stackrel{k}{\sim}$ relates tuples $((m_1, \dots, m_k), m)$ with tuples $((n_1, \dots, n_k), n)$. We write these tuples as $(\bar{m}, m)$. Notice that $\bar{m}$ can be seen as an assignment over $(x_1, \dots, x_k)$. A non-empty relation $\stackrel{k}{\sim}$ is called a k -bisimulation if it satisfies the following properties

- (prop)** If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, then $m \in V^{\mathfrak{M}}(a)$ iff $n \in V^{\mathfrak{N}}(a)$, for $a \in \text{PROP} \cup \text{NOM}$.
- (var)** If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, then for all $j \leq k$, $m_j = m$ iff $n_j = n$.
- (forth)** If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$ and $R^{\mathfrak{M}}mm'$, then there exists an $n' \in N$ such that $R^{\mathfrak{N}}nn'$ and $(\bar{m}, m') \stackrel{k}{\sim} (\bar{n}, n')$.
- (back)** A similar condition from $\mathfrak{N}$ to $\mathfrak{M}$.
- (@)** If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, then for every nominal $i \in \text{NOM}$, if $m' \in V^{\mathfrak{M}}(i)$ and $n' \in V^{\mathfrak{N}}(i)$ then $(\bar{m}, m') \stackrel{k}{\sim} (\bar{n}, n')$, and for every $j \leq k$, $(\bar{m}, m_j) \stackrel{k}{\sim} (\bar{n}, n_j)$
- ($\downarrow$)** If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, then for every $j \leq k$, $(\bar{m}_m^j, m) \stackrel{k}{\sim} (\bar{n}_n^j, n)$.

Note that since $\downarrow$ and $@$ are self-dual, we can collapse the back and forth clauses for these modalities into one. We write $\mathfrak{M} \stackrel{k}{\sim} \mathfrak{N}$ if there exists a k -bisimulation between the two models.

To extend the notion to the full language we need to add only one further condition.

ω -bisimulation. Let $\mathfrak{M}$ and $\mathfrak{N}$ be two hybrid models. An ω -bisimulation between $\mathfrak{M}$ and $\mathfrak{N}$ is a non-empty family of k -bisimulations satisfying the following *storage rule*:

- (sto)** If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, then $(\bar{m} * m, m) \stackrel{k+1}{\sim} (\bar{n} * n, n)$.

Here and elsewhere, $\bar{m} * m$ denotes the tuple obtained from concatenating $\bar{m}$ and m . Let $\bar{m}$ ($\bar{n}$) be an M -tuple (N -tuple). Then $(\mathfrak{M}, \bar{m}) \stackrel{\omega}{\sim} (\mathfrak{N}, \bar{n})$ means that there exists an ω -bisimulation between $\mathfrak{M}$ and $\mathfrak{N}$ such that $(\bar{m}, m(0)) \stackrel{k}{\sim} (\bar{n}, n(0))$, for k the length of $\bar{m}$.

Some remarks. First, k and ω -bisimulations can be restricted to a given set of propositional variables and nominals $\text{PROP} \cup \text{NOM}$ by restricting **(prop)** and **(@)** accordingly. Second, the modular definition of k and ω -bisimulation will lead to results for reducts of the language as well. For instance if we delete $\downarrow$ from the language, we just delete the **($\downarrow$)** clause from the definition of bisimulation and we obtain the appropriate notion for $\mathcal{H}(@)$. Of course, if we delete the variables from the language, we don't need the assignment tuples anymore, and the

bisimulation becomes just a relation between worlds, as usual. Then for the language without $\downarrow, @$ and variables, the standard definition of bisimulation applies (the condition **(prop)** takes care of the nominals). If we add $@$ to this language, we just have to add the following clause

- (@')** For all nominals i , if $V^{\mathfrak{M}}(i) = \{m\}$ and $V^{\mathfrak{N}}(i) = \{n\}$, then $m \sim n$.

Preservation results for all these alternatives can be given (the required proofs follow much the same lines as the proofs below) and we shall prove one such result in Section 7.

The first important fact about hybrid bisimulations is that they preserve truth:

Proposition 3.8 *Let $\mathfrak{M}$ and $\mathfrak{N}$ be two hybrid models, $m \in M$, $n \in N$. Then,*

- i. If $\mathfrak{M} \stackrel{k}{\sim} \mathfrak{N}$, with $\stackrel{k}{\sim}$ over a given set $\text{PROP} \cup \text{NOM}$, then for all formulas φ over the signature $\langle \text{PROP}, \text{NOM}, \{x_1, \dots, x_k\} \rangle$, $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$ implies $\mathfrak{M}, \bar{m}, m \Vdash \varphi \Leftrightarrow \mathfrak{N}, \bar{n}, n \Vdash \varphi$.*
- ii. If $(\mathfrak{M}, m) \stackrel{\omega}{\sim} (\mathfrak{N}, n)$, with $\stackrel{\omega}{\sim}$ over a given set $\text{PROP} \cup \text{NOM}$, then for all sentences φ over the signature $\langle \text{PROP}, \text{NOM}, \text{WVAR} \rangle$, $\mathfrak{M}, m \Vdash \varphi \Leftrightarrow \mathfrak{N}, n \Vdash \varphi$. (Recall that for sentences the choice of assignment is irrelevant.)*

PROOF.

- i.* By a straightforward inductive argument.
- ii.* Let $(\mathfrak{M}, m) \stackrel{\omega}{\sim} (\mathfrak{N}, n)$ and let φ be a hybrid sentence. Then it contains variables (after renaming) say $\{x_1, \dots, x_k\}$. We have $(\langle m \rangle, m) \stackrel{1}{\sim} (\langle n \rangle, n)$, so $k-1$ applications of the storage rule give us $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, where $\bar{m}$ is a k -tuple consisting of m 's and similarly for $\bar{n}$. But then, by *i.*, $\mathfrak{M}, \bar{m}, m \Vdash \varphi \Leftrightarrow \mathfrak{N}, \bar{n}, n \Vdash \varphi$, whence since φ is a sentence $\mathfrak{M}, m \Vdash \varphi \Leftrightarrow \mathfrak{N}, n \Vdash \varphi$. QED

The notion of k -bisimulation has a distinct modal flavor. But a very first-order notion is hidden inside: partial isomorphism.

Proposition 3.9 *Let $k \geq 2$, and let $\mathfrak{M} \stackrel{k}{\sim} \mathfrak{N}$. If $(\bar{m}, m) \stackrel{k}{\sim} (\bar{n}, n)$, then the function f defined as $f(m) = n$ and $f(m(i)) = n(i)$ is a partial isomorphism between $\{m(1), \dots, m(k), m\}$ and $\{n(1), \dots, n(k), n\}$.*

PROOF. The map f is a bijection by **(var)** and **(@)**. By **(prop)** and **(@)**, f preserves nominals and propositional variables. To see that it preserves the accessibility relation suppose $R^{\mathfrak{M}}xy$. There are three cases.

- (Case 1: $x = m$, $y = m_i$.) Then by **(forth)** there exists an n' such that $R^{\mathfrak{N}}nn'$ and $(\bar{m}, m_i) \stackrel{k}{\sim} (\bar{n}, n')$. But $\bar{m}(i) = m_i$, so by **(var)**, $n' = \bar{n}(i)$, whence $R^{\mathfrak{N}}nf(m(i))$.
 - (Case 2: $x = m_i$, $y = m$.) Let $j \neq i$. Such a j exists because we assumed that $k \geq 2$. By **($\downarrow$)**, $(\bar{m}_m^j, m) \stackrel{k}{\sim} (\bar{n}_n^j, n)$. Then by **(@)**, $(\bar{m}_m^j, m_i) \stackrel{k}{\sim} (\bar{n}_n^j, n_i)$. Now continue as in case 1.
 - (Case 3: $x = m_i$, $y = m_j$.) By **(@)**, $(\bar{m}, m_i) \stackrel{k}{\sim} (\bar{n}, n_i)$. Now continue as in case 1.
- Thus $R^{\mathfrak{M}}xy$ implies $R^{\mathfrak{N}}f(x)f(y)$. For the other direction use **(back)** in the same way. QED

Note that the condition $k \geq 2$ is crucial. We use it together with **($\downarrow$)** to store the information about m . Of course in a model where Rm_im holds, we have $\bar{m}, m \Vdash \downarrow x_j. @_{x_i} \Diamond x_j$.

Thus there is a clear link between our earlier work on generated back-and-forth systems, and the next theorem shouldn't come as a surprise:

Theorem 3.10 *Let $(\mathfrak{M}, \bar{m})$ and $(\mathfrak{N}, \bar{n})$ be two models. Then the following are equivalent:*

- i. $(\mathfrak{M}, \bar{m}) \lesssim (\mathfrak{N}, \bar{n})$.
- ii. $(\mathfrak{M}, \bar{m}) \equiv_R (\mathfrak{N}, \bar{n})$.

PROOF.

i. $\Rightarrow$ ii. Let $(\mathfrak{M}, \bar{m}) \lesssim (\mathfrak{N}, \bar{n})$. Define a family F of maps as follows: $f \in F$ if there exists $(\bar{x}, x') \lesssim (\bar{y}, y')$ and f is defined as in Proposition 3.9.

Clearly $\bar{m}$ and $\bar{n}$ are connected by a map. By Proposition 3.9 all maps are partial isomorphisms. We show the forth side of **(nominal extension)**; all other conditions have a similar proof. Suppose $f \in F$ and $\mathbf{z} \in M$ and $V^{\mathfrak{M}}(i) = \{\mathbf{z}\}$, for some nominal i . Then for some $\bar{x}$, $x, \bar{y}, y, (\bar{x}, x') \lesssim (\bar{y}, y')$ by definition of F . Then $(\bar{x} * x', x') \lesssim^{k+1} (\bar{y} * y', y')$ by **(sto)**. But then by **(@)**, $(\bar{x} * x', \mathbf{z}) \lesssim^{k+1} (\bar{y} * y', \mathbf{z}')$ for $V^{\mathfrak{N}}(i) = \{\mathbf{z}'\}$. Thus the wanted extension is in F .

ii. $\Rightarrow$ i. Let $(\mathfrak{M}, \bar{m}) \equiv_R (\mathfrak{N}, \bar{n})$. We define an ω -bisimulation in the obvious way: for any $f \in F$, for any k , for any tuple $\bar{m}$ in the k -th power of the domain of f and for any m in the domain of f , we set $(\bar{m}, m) \lesssim (f\bar{m}, f(m))$. It is easy to check that this is an ω -bisimulation. QED

It is possible to prove a direct characterization result for $\mathcal{H}(\downarrow, @)$ in terms of invariance for k -bisimulations, using again a diagram chasing argument. We're not going to do this here since in the next section we shall take a detour via the bounded fragment to reach the same result. It is also possible to develop k -pebble versions of generated back-and-forth systems; this notion takes the exact number of variables used in formulas into account. It is not difficult to see that $k+1$ -pebble generated back-and-forth systems correspond to k -bisimulations, but we won't follow up these matters here.

4 Harvest

It is time to draw together the threads developed in the previous section. First we note their consequences for $\mathcal{H}(\downarrow, @)$ expressivity over models. Then we note the consequences for frames and what this tells us about hybrid completeness. Finally we note the consequences for hybrid *tense* logic.

4.1 Expressivity over models

We have the following five-fold characterization of $\mathcal{H}(\downarrow, @)$:

Theorem 4.1 *Let $\varphi(\bar{x})$ be a first-order formula in the hybrid signature (with equality). Then the following are equivalent.*

- i. $\varphi(\bar{x})$ is equivalent to the standard translation of a $\mathcal{H}(\downarrow, @)$ formula.
- ii. $\varphi(\bar{x})$ is invariant for generated submodels.
- iii. $\varphi(\bar{x})$ is invariant for generated back-and-forth systems.
- iv. $\varphi(\bar{x})$ is invariant for ω -bisimulation.
- v. $\varphi(\bar{x})$ is equivalent to a formula in the bounded fragment of first-order logic.

PROOF. By Corollary 3.4, Theorem 3.7 and Proposition 3.8.

QED

But these have obvious consequences for the ordinary modal correspondence language. In particular, if we consider *nominal-free* hybrid sentences, then we obtain a five-fold characterization of the fragment of first-order logic in the classical modal signature which is invariant for generated submodels:

Corollary 4.2 *Let $\varphi(x)$ be a first-order formula in the modal signature with equality. Then the following are equivalent.*

- i. $\varphi(x)$ is equivalent to the standard translation of a nominal-free $\mathcal{H}(\downarrow, @)$ sentence.
- ii. $\varphi(x)$ is invariant for generated submodels (now in the standard modal sense).
- iii. $\varphi(x)$ is invariant for R -generated back-and-forth systems, where an R -generated back-and-forth system is a back-and-forth system satisfying only the $\Diamond$ -**extension** rule.
- iv. $\varphi(x)$ is invariant for ω -bisimulation.
- v. $\varphi(x)$ is equivalent to a formula in the bounded fragment of first-order logic without constants.

4.2 Frames and completeness

Recall that a frame $\mathfrak{F}$ is a pair $\langle W, R \rangle$ (that is, a model without a valuation). Since the late 1950s, one of the central topics in modal logic has been linking modal formulas to properties of frames and investigating when they give rise to complete axiomatizations for the frame classes they define. The work of the previous section easily yields a characterization of the frame-defining abilities of pure nominal-free sentences. Moreover, the axiomatic investigations of [BT99] (and indeed, the tableaux-based investigations of [Bla98]) show that there is a perfect match between definability and completeness for pure nominal-free sentences. By combining these results we obtain matching definability and completeness results for a wide range of first-order definable frame classes.

In modal correspondence theory, the first-order language (with equality) over the signature consisting simply of a binary symbol R is called the (first-order) *frame language*. We shall call a formula φ in the frame language containing exactly one free variable a *frame condition*. The class of frames defined by a frame condition $\varphi(x)$ is the class in which the universal closure $\forall x \varphi(x)$ is true; we call this class $\text{FRAMES}(\forall x \varphi(x))$.

Before proceeding further, two simple observations are in order. First, note that if we apply the standard translation ST to a pure nominal-free sentence α , then $ST(\alpha)$ is a frame condition with free-variable x . Furthermore, note that for any frame $\mathfrak{F} = \langle W, R \rangle$ we have that $\mathfrak{F} \models \alpha$ iff $\mathfrak{F} \models \forall x.ST(\alpha)$; this is an immediate consequence of the definition of frame validity (see Definition 2.4).

Theorem 4.3 *Let $\mathcal{K}[\mathcal{H}(\downarrow, @)]$ be the axiomatization given in Section 2, and for any hybrid sentence α let $\mathcal{K}[\mathcal{H}(\downarrow, @)] + \alpha$ be the system obtained by adding α as an additional axiom. Then, if $\varphi(x)$ is a frame condition and $\varphi(x)$ is invariant under generated submodels (in the usual modal sense) we have that:*

- i. If $\varphi(x)$ is in the bounded fragment, then the pure nominal free sentence $\downarrow x.HT(\varphi(x))$ defines $\text{FRAMES}(\forall x \varphi(x))$. Moreover, $\mathcal{K}[\mathcal{H}(\downarrow, @)] + \downarrow x.HT(\varphi(x))$ is strongly complete with respect to $\text{FRAMES}(\forall x \varphi(x))$.
- ii. If $\varphi(x)$ is not in the bounded fragment, there is a pure nominal free sentence α such that α defines $\text{FRAMES}(\forall x \varphi(x))$, and $ST(\alpha)$ is equivalent to $\varphi(x)$. Moreover, $\mathcal{K}[\mathcal{H}(\downarrow, @)] + \alpha$ is strongly complete with respect to $\text{FRAMES}(\forall x \varphi(x))$.

Conversely, if α is pure nominal-free sentence, then α defines $\text{FRAMES}(\forall x.ST(\alpha(x)))$, and $K[\mathcal{H}(\downarrow, @)] + \alpha$ is complete with respect to $\text{FRAMES}(\forall x.ST(\alpha(x)))$.

PROOF. The converse condition was proved in [BT98], so let's examine the other direction.

For item *i.*, we first remark that as $\varphi(x)$ belongs to the *frame* language, it contains no unary predicate symbols, hence $HT(\varphi(x))$ is a *pure* formula; that $\downarrow x.HT(\varphi(x))$ is a pure nominal-free sentence is thus clear. Now, by Corollary 3.3, for any model $\mathfrak{M} = (\mathfrak{F}, V)$ and any $m \in M$,

$$(\mathfrak{F}, V) \models \varphi(m) \text{ iff } (\mathfrak{F}, V), m \Vdash \downarrow x.HT(\varphi).$$

But this means that

$$(\mathfrak{F}, V) \models \forall x.\varphi \text{ iff } (\mathfrak{F}, V) \Vdash \downarrow x.HT(\varphi).$$

But as $\varphi(x)$ contains no unary predicate symbols (and $\downarrow x.HT(\varphi)$ no propositional variables) V is irrelevant, and hence

$$\mathfrak{F} \models \forall x.\varphi(x) \text{ iff } \mathfrak{F} \Vdash \downarrow x.HT(\varphi).$$

But this means that $\downarrow x.HT(\varphi(x))$ defines $\text{FRAMES}(\forall x.\varphi(x))$. Completeness follows using the arguments of [BT98].

For item *ii.*, we know that $\varphi(x)$ being invariant under generated submodels is equivalent to a formula in the bounded fragment — but is it equivalent to a *frame condition* $\varphi'(x)$? In fact, this can be established by modifying the diagram chasing argument used in the proof of Theorem 3.7. The key point to observe is that instead of showing that $BC(\varphi(x)) \models \varphi(x)$, we can show by the same method that $FC(\varphi(x)) \models \varphi(x)$, where FC are all the frame conditions implied by $\varphi(x)$. Thus there is an equivalent frame condition $\varphi'(x)$, and we can take α to be $\downarrow x.HT(\varphi'(x))$. The remainder of the proof is as for item *i.* QED

4.3 Hybrid tense logic

The characterization results have a particularly natural interpretation in the setting of hybrid *tense* logic. Recall that in tense logic we write $\Box$ as G , $\Diamond$ as F , and that we also have at our disposal an operator H (a $\Box$ -operator that scans the *converse* of the accessibility relation) and its dual P (a $\Diamond$ -operator that scans the converse of the accessibility relation). It is straightforward to hybridize tense logic by adding nominals, $@$ and $\downarrow$ (though now it is more natural to talk of *time* variables rather than world variables) thus forming the language $\mathcal{H}_t(\downarrow, @)$. To cope with the backward looking operators, tense logic requires a slightly more liberal notion of generated submodel: a point t belongs to the submodel *temporally generated* by a subset S if t is reachable from some point $s \in S$ by making a finite sequence of moves through the accessibility relation, where both forward *and backward* steps are allowed. The characterization results we have proved hold for $\mathcal{H}_t(\downarrow, @)$ under this notion of generated submodel.

But let's press matters a little further. Note that in *nominal-free sentences* of $\mathcal{H}_t(\downarrow, @)$, all occurrences of $@$ are eliminable. As a simple example, consider the definition of the *Until* operator:

$$\text{Until}(\varphi, \psi) := \downarrow x.F \downarrow y.@_x(F(y \wedge \varphi) \wedge G(Fy \rightarrow \psi)).$$

(This is simply the definition given in Section 2 written in tense logical notation.) But observe that the following nominal-free sentence has the same effect:

$$\text{Until}(\varphi, \psi) := \downarrow x.F \downarrow y.P(x \wedge (F(y \wedge \varphi) \wedge G(Fy \rightarrow \psi))).$$

That is, instead of retrieving the point named x using the $@$ operator, we can “reach back” for this point using P .

This observation (first made in [BS98]) is completely general. As long as a $\mathcal{H}_t(\downarrow, @)$ formula doesn't contain nominals or free time variables, it will always be possible to simulate $@$ by zig-zagging back to the binding point using the tense operators. More precisely, suppose a nominal free sentence φ has a temporal depth of n (that is, the maximal depth of embedding of tense operators is n) and that φ is satisfied at a time m . Then when we evaluate a subformula of φ of the form $@_x\psi$ at some point m' (note that m' cannot be more than n forward and back steps from m) then we know that x must bound to a point m'' (which is also not more than n forward and back steps from m). Hence m' and m'' are separated by at most $2n$ (forward and back) steps. We can define an operator $@^{2n}_x$ that allows us to zig-zag to a named time lying within $2n$ steps as follows. Let $ZZ2n$ be the set of all non-empty finite sequences of F and P operators of length at most $2n$. Then for any formula ψ and any variable x we define:

$$@^{2n}_x\psi := (x \wedge \psi) \vee \bigvee_{Z \in ZZ2n} Z(x \wedge \psi).$$

Hence, given a nominal free sentence φ of temporal depth n , we eliminate all occurrences of $@$ as follows. Let $@_x\psi$ be a subformula of φ where ψ contains no occurrences of $@$. Replace $@_x\psi$ by $@^{2n}_x\psi$ to form φ' . Repeating this procedure (starting with φ') produces an equivalent nominal-free sentence containing no occurrences of $@$. Thus, in the setting of tense logic, our characterization results for nominal free sentences go through without the help of $@$.

This is a pleasing result, for there are also non-technical reasons for viewing $\mathcal{H}_t(\downarrow)$ as a key system: this language can be viewed as a marriage between the ideas of Arthur Prior and Hans Reichenbach.

That $\mathcal{H}_t(\downarrow)$ captures the idea of Arthur Prior is clear: tense logic was invented by Prior precisely to capture the “internal” perspective on time which underlies temporal discourse in natural language (see in particular [Pri67]). But while the Priorean perspective gets a lot right, it misses a crucial fact about temporal discourse: tenses are very often *referential*. That is, tenses in natural language often achieve their effect by referring to specific points of time, and many semantic distinctions between natural language tenses cannot be drawn without taking referential effects into account. The importance of temporal reference was first made clear in the work of Hans Reichenbach, and many modern theories of tense (for example, [Com85]) adopt a fundamentally Reichenbachian stance.

It should be clear where this is heading: $\downarrow$ can be seen as the Reichenbachian device *par excellence*. In a sense, $\downarrow$ gives us a sort of *generalized present tense*; it enables us to “store” an evaluation point, thereby making it possible to insist later that certain events happened at *that* time, or that certain other events must be viewed from that particular perspective. This is precisely the kind of expressive power we need to encode Reichenbach's ideas. And crucially, the use of $\downarrow$ does not in any sense conflict with Prior's use of tense operators. Quite the reverse: F , P , and $\downarrow$ work together beautifully. No auxiliary apparatus (not even $@$) is required to blend the two approaches, and the result is a language which exactly captures first-order temporal reachability.

5 Interpolation

In this section we show that $\mathcal{H}(\downarrow, @)$ is well behaved in yet another sense: it has the interpolation property. Interpolation is a much studied notion. Originally considered a property

of deductive systems, it was proved for first-order logic in [Cra57] using a proof theoretic argument. We shall view interpolation as a property of *consequence relations* and will prove it using semantic arguments (as is done, for example, in [CK90]).¹

Before plunging into the details, note that in modal logic we can distinguish between *strong arrow interpolation* (AIP) and *weak turnstile interpolation* (TIP). AIP implies TIP, but not conversely. We will prove AIP for $\mathcal{H}(\downarrow, @)$, disprove AIP and TIP for its finite variable fragments (our earlier work on k -bisimulations will enable us to construct counterexamples fairly straightforwardly) and show that TIP interpolation holds for the sublanguage $\mathcal{H}(@)$. Here are the definitions of these concepts:

AIP A logic L has the *Arrow Interpolation Property* (AIP) if, whenever $\models_L \varphi \rightarrow \psi$, there exists a formula θ such that $\models_L \varphi \rightarrow \theta$, $\models_L \theta \rightarrow \psi$ and $IP(\theta) \subseteq IP(\varphi) \cap IP(\psi)$.

TIP A logic L has the *Turnstile Interpolation Property* (TIP) if, whenever $\varphi \models_L \psi$, there exists a formula θ such that $\varphi \models_L \theta$, $\theta \models_L \psi$ and $IP(\theta) \subseteq IP(\varphi) \cap IP(\psi)$.

For first-order logic these notions are equivalent, but in modal logic this is not the case (as we see below, equivalence depends on both compactness and the availability of a deduction theorem; cf. also [Cze82]). Further, note that the meaning of TIP depends on the way we define the consequence relation $\varphi \models \psi$. In Section 2 we introduced two consequence relations: local consequence and global consequence (see [Ben83] or [MV97] for a discussion of their relative merits). So there are two plausible definitions of the turnstile interpolation property, TIP^{loc} and TIP^{glo} .

In modal logic these different notions of interpolation are related as follows:

Proposition 5.1

- i. *AIP and TIP^{loc} are equivalent.*
- ii. *If the local consequence relation is compact, then AIP implies TIP^{glo} .*

For this reason, from now on we take TIP to be defined using the *global* consequence relation. AIP and TIP are often referred to as the strong and weak interpolation properties respectively, and we shall sometimes use this terminology.

We turn to the technicalities of the interpolation result. As is usual in interpolation proofs, where language related issues require special care, we replace the standard notion of consistency

Definition 5.2 (Consistency) *Let T be a set of formulas in $\mathcal{L}$. Then T is consistent iff there is a model $\mathfrak{M}$, an $m \in M$, and an assignment g such that for all $\varphi \in T$, $\mathfrak{M}, g, m \Vdash \varphi$.*

by the finer-grained notion of separability

Definition 5.3 (Separability) *Let T, U, L be sets of formulas in $\mathcal{L}$. We say that the pair $\langle T, U \rangle$ is separable with respect to L if there exists a formula $\theta \in L$ such that $T \models \theta$ and $U \models \neg\theta$. $\langle T, U \rangle$ is inseparable with respect to L if it is not separable with respect to L .*

We are ready to prove the main result of this section.

¹Jerry Seligman has recently announced a proof-theoretic proof of interpolation for $\mathcal{H}(\downarrow, @)$. However his proof has not yet been made available.

Theorem 5.4 (Arrow Interpolation for $\mathcal{H}(\downarrow, @)$) *Let φ and ψ be formulas in $\mathcal{L}$ such that $\models \varphi \rightarrow \psi$. Then there exists a formula θ such that*

- i. $\models \varphi \rightarrow \theta$ and $\models \theta \rightarrow \psi$.
- ii. $IP(\theta) \subseteq IP(\varphi) \cap IP(\psi)$.

PROOF. Suppose we are given formulas φ_0 and ψ_0 such that there is no interpolant for $\varphi_0 \rightarrow \psi_0$. We will prove that $\not\models \varphi_0 \rightarrow \psi_0$ by producing a model $\mathfrak{M} = \langle M, R, V \rangle$ and an assignment g such that for some $m \in M$, $\mathfrak{M}, g, m \Vdash \varphi_0 \wedge \neg\psi_0$ (the proof follows the method of [CK90] in which two connected models are simultaneously built using fresh constants). Hence we will have shown the contrapositive.

We can assume that $\{\varphi_0\}$ and $\{\neg\psi_0\}$ are consistent (for if they are not, then either $\perp$ or $\top$ is an interpolant). Furthermore they must be inseparable over the formulas in $\mathcal{L}$ with propositional variables and nominals in $IP(\varphi_0) \cap IP(\psi_0)$.

Let $\mathcal{L}'$ be the hybrid language over the signature $\langle \text{PROP}, \text{NOM} \cup \text{N}, \text{WVAR} \rangle$, where $\text{N} = \{n_0, \dots, n_k, \dots\}$ is a countably infinite set of new nominals. For any formula φ define the restricted language $\mathcal{L}_\varphi$ as $\{\xi \in \mathcal{L} \mid IP(\xi) \subseteq IP(\varphi)\}$ and similarly for $\mathcal{L}'$. Let $\mathcal{L}'_{\varphi_0\psi_0} = \mathcal{L}'_{\varphi_0} \cap \mathcal{L}'_{\psi_0}$.

Let $\varphi_1, \dots, \varphi_k, \dots$ be an enumeration of all formulas in $\mathcal{L}'_{\varphi_0}$,
 $\psi_1, \dots, \psi_k, \dots$ be an enumeration of all formulas in $\mathcal{L}'_{\psi_0}$.

We define the sequences,
 $\{n_0\} \cup \{\varphi_0\} = T_0 \subseteq T_1 \subseteq T_2 \subseteq \dots$
 $\{n_0\} \cup \{\neg\psi_0\} = U_0 \subseteq U_1 \subseteq U_2 \subseteq \dots$

as follows:

- If $T_j \cup \{\varphi_j\}$ and U_j are separable over $\mathcal{L}'_{\varphi_0\psi_0}$ then $T_{j+1} = T_j$, else
 - if $\varphi_j \neq @_s s$ and $\varphi_j \neq @_s \Diamond \varphi'$ for $s \in \text{WSYM}$, then $T_{j+1} = T_j \cup \{\varphi_j\}$,
 - if $\varphi_j = @_s s$, then $T_{j+1} = T_j \cup \{\varphi_j\} \cup \{@_s(n_k \wedge s)\}$, for $n_k \in \text{N} \setminus \text{NOM}(T_j \cup U_j)$,
 - if $\varphi_j = @_s \Diamond \varphi'$, then $T_{j+1} = T_j \cup \{\varphi_j\} \cup \{@_s \Diamond(n_k \wedge \varphi')\}$, for $n_k \in \text{N} \setminus \text{NOM}(T_j \cup U_j \cup \{\varphi_j\})$.
- If T_{j+1} and $U_j \cup \{\psi_j\}$ are separable over $\mathcal{L}'_{\varphi_0\psi_0}$ then $U_{j+1} = U_j$, else
 - if $\psi_j \neq @_s s$ and $\psi_j \neq @_s \Diamond \psi'$ for $s \in \text{WSYM}$, then $U_{j+1} = U_j \cup \{\psi_j\}$,
 - if $\psi_j = @_s s$, then $U_{j+1} = U_j \cup \{\psi_j\} \cup \{@_s(n_k \wedge s)\}$, for $n_k \in \text{N} \setminus \text{NOM}(T_{j+1} \cup U_j)$,
 - if $\psi_j = @_s \Diamond \psi'$, then $T_{j+1} = T_j \cup \{\psi_j\} \cup \{@_s \Diamond(n_k \wedge \psi')\}$, for $n_k \in \text{N} \setminus \text{NOM}(T_{j+1} \cup U_j \cup \{\psi_j\})$.

The fresh nominals play the same role as *Henkin witnesses* in first-order proofs: they ensure that we obtain models in which every world has a name. Define

$$T_\omega = \bigcup_{j \in \omega} T_j \quad \text{and} \quad U_\omega = \bigcup_{j \in \omega} U_j.$$

Claim 1 *For all $j \in \omega$, $\langle T_j, U_j \rangle$ is inseparable with respect to $\mathcal{L}'_{\varphi_0\psi_0}$. Whence $\langle T_\omega, U_\omega \rangle$ is an inseparable pair with respect to $\mathcal{L}'_{\varphi_0\psi_0}$. Furthermore T_ω (resp. U_ω) is maximal consistent in $\mathcal{L}'_{\varphi_0}$ (resp. $\mathcal{L}'_{\psi_0}$). In particular, for all $\theta \in \mathcal{L}'_{\varphi_0\psi_0}$: $\theta \in T_\omega \Leftrightarrow \theta \in U_\omega$.*

PROOF OF CLAIM. The proof is by induction on j . Separability/inseparability below is with respect to $\mathcal{L}'_{\varphi_0\psi_0}$ except when otherwise mentioned.

BASE CASE $j = 0$. Suppose $\langle T_0, U_0 \rangle$ is separable. Then there is a formula $\theta \in \mathcal{L}'_{\varphi_0\psi_0}$ such that $\models n_0 \wedge \varphi_0 \rightarrow \theta$ and $\models n_0 \wedge \neg\psi_0 \rightarrow \neg\theta$. θ might contain some nominals of N , say $\{n_{i_1}, \dots, n_{i_k}\}$, $k \geq 0$. Let $x_0, x_{i_1}, \dots, x_{i_k} \in \text{WVAR}$ which don't occur in $\varphi_0, \psi_0, \theta$. We will write $\theta[x_0x_{i_1} \dots x_{i_k}]$ for the formula obtained from θ by replacing n_{i_j} by x_{i_j} , and n_0 by x_0 . Then, making use of the complete axiomatization given in Section 2, we have:

$$\begin{array}{ll}
\vdash \varphi_0 \rightarrow (n_0 \rightarrow \theta) & \\
\vdash \downarrow x_{i_k}.(\varphi_0 \rightarrow (n_0 \rightarrow \theta[x_{i_k}])) & \text{Proposition 2.6} \\
\vdash \varphi_0 \rightarrow (n_0 \rightarrow \downarrow x_{i_k}.\theta[x_{i_k}]) & Q1 \text{ twice} \\
\vdash \varphi_0 \rightarrow (n_0 \rightarrow \downarrow x_{i_1} \dots \downarrow x_{i_k}.\theta[x_{i_1} \dots, x_{i_k}]) & \text{Similarly} \\
\vdash \downarrow x_0.(\varphi_0 \rightarrow (x_0 \rightarrow \downarrow x_{i_1} \dots \downarrow x_{i_k}.\theta[x_0x_{i_1} \dots, x_{i_k}])) & \text{Proposition 2.6} \\
\vdash \varphi_0 \rightarrow \downarrow x_0.(x_0 \rightarrow \downarrow x_{i_1} \dots \downarrow x_{i_k}.\theta[x_0x_{i_1} \dots, x_{i_k}])) & Q1 \\
\vdash \varphi_0 \rightarrow \downarrow x_0.\downarrow x_{i_1} \dots \downarrow x_{i_k}.\theta[x_0x_{i_1} \dots, x_{i_k}] & Q3 \\
\\
\vdash \neg\psi_0 \rightarrow (n_0 \rightarrow \neg\theta) & \\
\vdash \neg\psi_0 \rightarrow \downarrow x_0.\downarrow x_{i_1} \dots \downarrow x_{i_k}.\neg\theta[x_0x_{i_1} \dots, x_{i_k}] & \text{As before} \\
\vdash \neg\psi_0 \rightarrow \downarrow x_0.\downarrow x_{i_1} \dots \downarrow x_{i_k}.\theta[x_0x_{i_1} \dots, x_{i_k}] & \text{Self Dual}\downarrow
\end{array}$$

$\theta[x_0x_{i_1} \dots, x_{i_k}]$ is a formula in $\mathcal{L}_{\varphi_0} \cap \mathcal{L}_{\psi_0}$ and thus $\langle \{\varphi_0\}, \{\neg\psi_0\} \rangle$ is separable over $\mathcal{L}_{\varphi_0} \cap \mathcal{L}_{\psi_0}$. Contradiction.

By using the inductive hypothesis “ $\langle T_j, U_j \rangle$ is an inseparable pair” and going step by step through the construction, the inseparability of $\langle T_{j+1}, U_{j+1} \rangle$ is easily established. $\blacktriangleleft$

To start the construction of our model, we recall the notion of pasted maximal consistent sets (MCS) and labeled models from [BT99]. A maximal consistent set Γ is *pasted* if

- $\@_s\varphi \in \Gamma$ implies for some nominal i , $\@_s(i \wedge \varphi) \in \Gamma$ and
- $\@_s\Diamond\varphi \in \Gamma$ implies for some nominal i , $\@_s\Diamond(i \wedge \varphi) \in \Gamma$.

A pasted MCS Γ is *labeled* by a nominal i precisely when $i \in \Gamma$. Let Γ be a pasted MCS labeled by a nominal, then for all world symbols s appearing in Γ , let $\Delta_s \stackrel{\text{def}}{=} \{\varphi \mid \@_s\varphi \in \Gamma\}$. Then the *labeled model* yielded by Γ is $\mathfrak{M} = \langle M, R, V \rangle$, where

- $M = \{\Delta_s \mid s \text{ is a world symbol in } \Gamma\}$,
- $R\Delta\Delta' \text{ iff } \{\varphi \mid \Box\varphi \in \Delta\} \subseteq \Delta'$,
- $\Delta \in V(p) \text{ iff } p \in \Delta$, for p a propositional variable or nominal.

We define the natural assignment $g : \text{WVAR} \rightarrow M$ by $g(x) = \{m \in M \mid x \in m\}$.

By construction T_ω and U_ω are pasted MCSs labeled by the nominal $n_0 \in N$. Let $\mathfrak{M}_{\varphi_0} = \langle M_{\varphi_0}, R_{\varphi_0}, V_{\varphi_0} \rangle$ be the labeled model obtained from T_ω and $\mathfrak{M}_{\psi_0} = \langle M_{\psi_0}, R_{\psi_0}, V_{\psi_0} \rangle$ the one obtained from U_ω . We denote by $\Delta_s^{\varphi_0}$ (resp. $\Delta_s^{\psi_0}$) the elements of M_{φ_0} (resp. M_{ψ_0}).

Claim 2

- i. $\Delta_{n_0}^{\varphi_0} = T_\omega$ and $\Delta_{n_0}^{\psi_0} = U_\omega$.
- ii. For $\Delta_s^{\varphi_0} \in M_{\varphi_0}$ there is $n \in N$ such that $n \in \Delta_s^{\varphi_0}$ (or equivalently $\Delta_s^{\varphi_0} = \Delta_n^{\varphi_0}$). Similarly for $\Delta_s^{\psi_0} \in \mathfrak{M}_{\psi_0}$.

PROOF OF CLAIM.

i. We show that $\Delta_{n_0}^{\varphi_0} = T_\omega$; the other case is similar. $\Delta_{n_0}^{\varphi_0}$ is an MCS because $\@_{n_0}$ is self-dual. So it is sufficient to show that $\Delta_{n_0}^{\varphi_0} \supseteq T_\omega$. Let $\varphi \in T_\omega$, by *Introduction* $\models n_0 \wedge \varphi \rightarrow \@_{n_0}\varphi$. Because $\@_{n_0}\varphi \in \mathcal{L}'_{\varphi_0}$, $n_0 \in T_\omega$ and T_ω is maximal in this language, $\@_{n_0}\varphi \in T_\omega$. By definition $\varphi \in \Delta_{n_0}^{\varphi_0}$.

ii. By [BT99] Lemma 4.3.5 we have $n \in \Delta_s^{\varphi_0} \Rightarrow \Delta_s^{\varphi_0} = \Delta_n^{\varphi_0}$. We prove the case for $\mathfrak{M}_{\varphi_0}$. $\@_ss$ is a formula in $\mathcal{L}'_{\varphi_0}$, hence $\@_ss = \varphi_j$ for some j , and furthermore a theorem. Hence $\{\@_ss\}$ will be added to T_{j+1} together with $\@_s(s \wedge n_k)$ for a new nominal n_k . It follows that $n_k \in \Delta_s^{\varphi_0}$ and hence $\Delta_{n_k}^{\varphi_0} = \Delta_s^{\varphi_0}$. $\blacktriangleleft$

From Claim 2 using results in [BT99] it follows that $\mathfrak{M}_{\varphi_0}$ and $\mathfrak{M}_{\psi_0}$ are hybrid models satisfying a Truth Lemma, and the natural assignment g defined as $g(x) = \{m \in M \mid x \in m\}$ is an assignment. Thus we obtain

$$\mathfrak{M}_{\varphi_0}, g_{\varphi_0}, \Delta_{n_0}^{\varphi_0} \Vdash \varphi \text{ and } \mathfrak{M}_{\psi_0}, g_{\psi_0}, \Delta_{n_0}^{\psi_0} \Vdash \neg\psi. \quad (1)$$

Furthermore the two models are very closely related.

Claim 3 Let a function $h : M_{\varphi_0} \rightarrow M_{\psi_0}$ be defined as $h(\Delta_n^{\varphi_0}) = \Delta_n^{\psi_0}$, for $n \in N$. Then h is a bijection which respects the accessibility relation and the propositional variables and nominals in the common language $\mathcal{L}'_{\varphi_0\psi_0}$. Moreover, $g^{\psi_0} = h \circ g^{\varphi_0}$, for g the natural assignment.

PROOF OF CLAIM. h is defined at every member of the domain of $\mathfrak{M}_{\varphi_0}$ by Claim 2.ii and the fact that for any $n \in N$, both $\Delta_n^{\varphi_0}$ and $\Delta_n^{\psi_0}$ are uniquely defined.

Moreover, h is a bijection because $\@_n \top \in T_\omega$ iff $\@_n \top \in U_\omega$ and in $\mathfrak{M}_{\varphi_0}$ and $\mathfrak{M}_{\psi_0}$ nominals are interpreted as singletons.

For all proposition variables p in $\mathcal{L}'_{\varphi_0\psi_0}$ we have $\Delta_n^{\varphi_0} \in V^{\varphi_0}(p)$ iff $\@_np \in T_\omega$ iff $\@_np \in U_\omega$ iff $h(\Delta_n^{\varphi_0}) \in V^{\psi_0}(p)$. For the relation R , $(\Delta_n^{\varphi_0}, \Delta_{n'}^{\varphi_0}) \in R^{\varphi_0}$ iff $\@_n\Diamond n' \in T_\omega$ iff $\@_n\Diamond n' \in U_\omega$ iff $(h(\Delta_n^{\varphi_0}), h(\Delta_{n'}^{\varphi_0})) \in R^{\psi_0}$. A similar argument shows $g^{\psi_0} = h \circ g^{\varphi_0}$. $\blacktriangleleft$

Since the two models share the same frame, and agree on the common language, there is a model $\mathfrak{M}$ for the union of the two languages which has $\mathfrak{M}_{\varphi_0}$ and $\mathfrak{M}_{\psi_0}$ as reducts. But then by (1), $\mathfrak{M}, g, \Delta_{n_0} \Vdash \varphi_0 \wedge \neg\psi_0$, and we have proved the contrapositive. QED

Actually, we can prove a stronger result: we can restrict the free variables occurring in the interpolant θ to only those appearing both in φ_0 and ψ_0 . Reason as follows. Assume $\models \varphi_0(\bar{x}, \bar{z}) \rightarrow \psi_0(\bar{y}, \bar{z})$; here we explicitly show the world variables free in φ_0 and ψ_0 and indicate which are shared. Now by Proposition 2.6 we can replace free variables by new nominals $\bar{i}_x, \bar{i}_y$ and $\bar{i}_z$ such that $\models \varphi_0(\bar{i}_x, \bar{i}_z) \rightarrow \psi_0(\bar{i}_y, \bar{i}_z)$. Use Theorem 5.4 to find an interpolant $\theta(\bar{s}, \bar{i}_z)$. The $\bar{s}$ are free variables that might appear in θ ; we can assume $\bar{s}$ disjoint from $\bar{x}, \bar{y}, \bar{z}$. By *Nec $\downarrow$* and *Q $_1$* we obtain $\models \varphi_0(\bar{i}_x, \bar{i}_z) \rightarrow \downarrow \bar{s}.\theta(\bar{i}_z)$ and $\models \downarrow \bar{s}.\theta(\bar{i}_z) \rightarrow \psi_0(\bar{i}_y, \bar{i}_z)$. We now rename again the formulas to the original $\bar{x}, \bar{y}, \bar{z}$ and the interpolant will only contain free world variables common to φ_0 and ψ_0 .

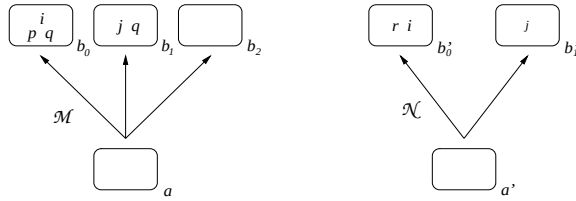
Furthermore, note that nothing in the proof is intrinsically tied to the number of modalities in the language. In other words, arrow interpolation also holds for the multi-modal versions of $\mathcal{H}(\downarrow, @)$ if modalities are allowed freely in the interpolant.

Corollary 5.5 Multi-modal $\mathcal{H}(\downarrow, @)$ has arrow interpolation if no restriction on occurrences of modalities is imposed for the interpolant.

However when we restrict the interpolant to contain only the modalities in the common language, then interpolation does *not* follow immediately, especially if the modalities interact (for example, if the theory contains axioms involving more than one modality; see [Mar99] for examples of this type). Nonetheless, we advance the following conjecture:

Conjecture 5.6 *Multi-modal $\mathcal{H}(\downarrow, @)$ has arrow interpolation even if the interpolant's modalities are restricted to the common language.*

On the other hand, it is clear from the proof that the number of world variables needed cannot be bounded (they are used to quantify away the nominals in the proof of Claim 1). Indeed, if we restrict $\mathcal{H}(\downarrow, @)$ to only a finite number of variables, then arrow interpolation *fails*. Because we have the notion of a k -bisimulations at our disposal, it is fairly straightforward to provide counterexamples. Let's consider first the case of $\mathcal{H}(\downarrow, @)$ restricted to only one world variable. Take the models



and the formulas

$$\begin{aligned}\varphi &= \Diamond(p \wedge q) \wedge \Diamond(\neg p \wedge q) \wedge \Diamond(\neg p \wedge \neg q) \\ \psi &= (\Diamond r \wedge \Box(r \rightarrow i)) \rightarrow (\Diamond(\neg r \wedge j) \rightarrow \Diamond(\neg r \wedge \neg j)).\end{aligned}$$

It is easy to prove that $\varphi \rightarrow \psi$ is valid: in any model having a world with at least three different accessible worlds, if there is a unique accessible r -world and one of the accessible $\neg r$ -worlds is named by the nominal j , then the second accessible $\neg r$ -world is named $\neg j$. Furthermore, $\mathfrak{M}$ and $\mathfrak{N}$ 1-bisimulate in the common (empty) language via the relation

$$(m, m') \stackrel{1}{\sim} (n, n') \text{ iff } \text{depth}(m) = \text{depth}(n) \ \& \ \text{depth}(m') = \text{depth}(n') \ \& \ (m = m' \Leftrightarrow n = n'),$$

where depth is the distance from the root.

Finally, φ is true in $\mathfrak{M}, a$, while ψ is false at $\mathfrak{N}, a'$ which proves that an interpolant on only one variable does not exist². Actually, not even an interpolant in two variables exists as also a 2-bisimulation between $\mathfrak{M}$ and $\mathfrak{N}$ can be defined.

Notice that the heart of the counterexample is just a counting argument, which can be reproduced for the other finite variable fragments of $\mathcal{H}(\downarrow, @)$ by taking bigger and bigger models $\mathfrak{M}$ and $\mathfrak{N}$ exhibiting the same basic pattern. Hence:

Theorem 5.7 *Strong interpolation (AIP) fails in all finite world variable fragments of $\mathcal{H}(\downarrow, @)$.*

²The formulas $\varphi = \Diamond p \wedge \Diamond \neg p$ and $\psi = \Diamond i \rightarrow \Diamond \neg i$ provide a simpler counterexample to strong interpolation in the one-variable fragment, but they have $\Diamond \top \wedge (\Diamond x \rightarrow \Diamond \neg x)$ as weak interpolant. The more complex example proves failure in the weak case also.

A more complex counterexample based on the same idea can be set up to prove failure of weak (turnstile) interpolation. Take the same formulas φ and ψ as before; clearly $\varphi \models^{glo} \psi$. Take now the model $\mathfrak{M}$ and define $\mathfrak{M}'$ by linking new copies of b_0, b_1 and b_2 to each terminal world in $\mathfrak{M}$. Let $\mathfrak{M}_\omega$ be the infinite model obtained by iterating this operation ω times and similarly for $\mathfrak{N}_\omega$. Now $\mathfrak{M}_\omega$ makes φ globally true. Suppose θ is an interpolant on one variable. Then as $\varphi \models^{glo} \theta$, θ is globally true at $\mathfrak{M}_\omega$.

We need something stronger than a mere 1-bisimulations linking $\mathfrak{M}_\omega$ and $\mathfrak{N}_\omega$, as we want to transfer *global* truth. With ordinary modal languages, requiring $\sim$ to be total and surjective is enough, but we have to take care of assignments as well. We shall say that a k -bisimulation between $\mathfrak{M}$ and $\mathfrak{N}$ is *full* if for every pair $\langle \bar{m}, m \rangle \in {}^k M \times M$ there exists $\langle \bar{n}, n \rangle \in {}^k N \times N$ such that $\langle \bar{m}, m \rangle \stackrel{k}{\sim} \langle \bar{n}, n \rangle$ and vice versa. If we can define a full 1-bisimulation between $\mathfrak{M}_\omega$ and $\mathfrak{N}_\omega$ then $\mathfrak{N}_\omega \models^{glo} \theta$. But $\stackrel{1}{\sim}$ defined as in the previous case is indeed full. Hence, as $\theta \models^{glo} \psi$, ψ should be globally true in $\mathfrak{N}_\omega$ — but it is not.

Theorem 5.8 *Weak interpolation (TIP) fails in all finite world variable fragments of $\mathcal{H}(\downarrow, @)$.*

It is not difficult to see that in both cases (arrow and turnstile)

$$\theta = \downarrow x. \Diamond(\downarrow y. @_x(\Diamond(\downarrow y'. @_x(\Diamond(y \wedge \neg y') \wedge \Diamond(\neg y \wedge y') \wedge \Diamond(\neg y \wedge \neg y')))))$$

(a sentence in 3 variables) is an interpolant.

Finally, we see from the proof of our main result that the $\downarrow$ binder is needed in Claim 1. So what about interpolation in the sublanguage $\mathcal{H}(@)$? We can again use models $\mathfrak{M}$ and $\mathfrak{N}$ to prove that arrow interpolation fails. We use the restricted version of k -bisimulation which leaves out condition ($\downarrow$). In this framework we can define for any k , a k -bisimulation between $\mathfrak{M}$ and $\mathfrak{N}$ such that for any $\bar{m} \in {}^k M$ and any $\bar{n} \in {}^k N$, $(\bar{m}, a) \stackrel{k}{\sim} (\bar{n}, a')$. This proves that there is no arrow interpolant for $\varphi \rightarrow \psi$ in $\mathcal{H}(@)$.

Theorem 5.9 *Strong interpolation (AIP) fails in $\mathcal{H}(@)$.*

But *weak* interpolation holds for $\mathcal{H}(@)$ because the role of $\downarrow$ is played by the implicit quantification in the definition of $\varphi \models^{glo} \psi$. (Another way of looking at it is to say that the definition of $\models^{glo}$ sneaks in implicit quantification using the global hybrid binder $\forall$ mentioned in the introduction.)

Theorem 5.10 *Let φ and ψ be sentences of $\mathcal{H}(@)$ such that $\varphi \models^{glo} \psi$. Then there is a formula θ , which may contain additional free variables, such that*

- i. $\varphi \models^{glo} \theta$ and $\theta \models^{glo} \psi$.
- ii. $IP(\theta) \subseteq IP(\varphi) \cap IP(\psi)$.

OUTLINE OF PROOF. We outline how the proof of arrow interpolation for $\mathcal{H}(\downarrow, @)$ should be modified to obtain the result.

First, the construction of the pasted sets T_ω and U_ω needs to be adjusted as we need to ensure that the labeled models obtained from them make φ_0 and $\neg\psi_0$ *globally* true. To that end, whenever we run into a formula of the form $@_s s$ or $@_s \Diamond \xi$ we paste not only a new nominal n_k but also the formulas we want to make globally true. For example one clause in the definition of T_{j+1} would read

- if $\varphi_j = @_s s$, then $T_{j+1} = T_j \cup \{\varphi_j\} \cup \{@_s(n_k \wedge s \wedge \varphi_0)\}$, for $n_k \in N \setminus \text{NOM}(T_j \cup U_j)$.

We will need to show that for all $j \in \omega$, $\langle T_j, U_j \rangle$ is (globally) inseparable with respect to $\mathcal{L}'_{\varphi_0 \psi_0}$. The base case is simple: if θ (including perhaps some new nominals $\{n_{i_1}, \dots, n_{i_k}\}$) separates $\langle T_0, U_0 \rangle$ on $\mathcal{L}'_{\varphi_0 \psi_0}$, then $\theta[x_{i_1} \dots x_{i_k}]$ separates $\langle \{\varphi_0\}, \{\neg \psi_0\} \rangle$, for new variables $\{x_{i_1}, \dots, x_{i_k}\}$; this is precisely where the free variables in the interpolant are needed.

What about the inductive step? Consider, for example, the case of $\varphi_j = @_s s$. Assume that $\langle T_j \cup \{\varphi_j\}, U_j \rangle$ is inseparable in $\mathcal{L}'_{\varphi_0 \psi_0}$; we want to prove that $\langle T_j \cup \{\varphi_j, @_s(n_k \wedge s \wedge \varphi_0)\}, U_j \rangle$ is inseparable. Suppose θ separates this last pair. Then $U_j \models^{glo} \neg \theta$ while $T_j \cup \{@_s s, @_s(n_k \wedge s \wedge \varphi_0)\} \models^{glo} \theta$. Because $@_s(n_k \wedge s \wedge \varphi_0)$ is an $@$ -formula, this is the case iff $T_j \cup \{@_s s\} \models^{glo} @_s(n_k \wedge s \wedge \varphi_0) \rightarrow \theta$. Furthermore, as $\varphi_0 \in T_j$ and n_k is a new nominal by definition, for all $\mathfrak{M}, \mathfrak{M} \models T_j$ implies $\mathfrak{M} \models @_s(n_k \wedge s \wedge \varphi_0)$. Hence $T_j \cup \{@_s s\} \models^{glo} \theta$. Contradiction.

From now on the proof follows the same lines as before. We obtain labeled models such that $\mathfrak{M}_{\varphi_0} \models \varphi_0$ and $\mathfrak{M}_{\psi_0} \models \neg \psi_0$ “sharing” the same frame, from which we build a model $\mathfrak{M}$ where $\varphi_0 \wedge \neg \psi_0$ holds globally. QED

We will prove in the next section that the $\mathcal{H}(@)$ is well behaved with respect to complexity: like ordinary unimodal logic it has a PSPACE-complete satisfaction problem. On the other hand, $\mathcal{H}(\downarrow, @)$ is known to be undecidable. Thus we have a decidable system with TIP, an undecidable system with AIP, and it is natural to ask:

Open Question 5.11 *Is there any decidable hybrid language extending $\mathcal{H}(@)$ that enjoys arrow interpolation?*

To close this section, some remarks on Beth Definability. The Beth Definability Property [Bet53] is commonly studied together with interpolation (and in many cases interpolation is considered just a step in the proof of the Beth Property). Loosely speaking, a logic has the Beth property if any implicit definition has also an explicit definition. More precisely, for hybrid languages we define:

Definition 5.12 (Beth Definability) *A hybrid logic has the Beth Definability Property if for all formulas $\varphi(\bar{a}, a)$ whose propositional letters, free world variables and nominals occur among $\bar{a}, a$, if $\models \varphi(\bar{a}, a/b_1) \wedge \varphi(\bar{a}, a/b_2) \rightarrow (b_1 \leftrightarrow b_2)$ then there is a formula $\psi(\bar{a})$ such that $\models \varphi(\bar{a}, a) \rightarrow (\psi(\bar{a}) \leftrightarrow a)$.*

Using a standard argument [Cra57] we can easily derive the Beth definability property for $\mathcal{H}(\downarrow, @)$ from Theorem 5.4.

Theorem 5.13 *$\mathcal{H}(\downarrow, @)$ has the Beth definability property.*

A careful analysis of weaker versions of the Beth property can be carried out for fragments of $\mathcal{H}(\downarrow, @)$ as we did with the interpolation property.

6 Complexity

Beautiful as $\mathcal{H}(\downarrow, @)$ is, it has a drawback: its satisfaction problem is undecidable (see [BS95]). The culprit is the binder $\downarrow$; $@$ is not needed. However $\mathcal{H}(\downarrow, @)$ has an interesting sublanguage, namely $\mathcal{H}(@)$. As we have just seen, although $\mathcal{H}(@)$ does not enjoy strong interpolation,

it does have weak interpolation. Moreover, as is shown in [Bla98], simple tableaux and sequent systems for $\mathcal{H}(@)$ can be defined by exploiting the interplay between nominals (or free variables) and $@$; the underlying idea traces back to [Sel91]. Furthermore, while $\mathcal{H}(@)$ doesn't offer any exciting new expressivity at the level of *models* (for example, without the $\downarrow$ binder we can't define *Until*) it does provide new expressivity at the level of *frames*: we can define many properties that are not definable in ordinary propositional modal logic, including irreflexivity ($@_i \Box \neg i$), asymmetry ($@_i \Diamond j \rightarrow @_j \neg \Diamond i$) and antisymmetry ($@_i \Diamond j \wedge @_j \Diamond i \rightarrow @_i j$); these correspondences are easy to check using the standard translation. Moreover, pure formulas such as these automatically yield complete axiomatizations for the frame classes they define (see [BT99, Bla98]).

Thus there are many reasons for being interested in $\mathcal{H}(@)$, and a natural question to ask is: how high a computational price do we pay for these benefits? It turns out that (up to a polynomial) there are no extra computational costs when expanding unimodal logic (or even multimodal logic) with $@$ and nominals and/or free variables, but expanding tense logic with even a single nominal (or free variable) leads to an increase in complexity (assuming PSPACE $\neq$ EXPTIME).

For the most part we will study (local) *K-satisfaction problems*: given a formula φ , does there exist a model $\mathfrak{M}$, a world m and an assignment g , such that $\mathfrak{M}, g, m \models \varphi$? (The K reflects the fact that we place no restrictions on the satisfying models; thus in effect we're measuring the complexity of the minimal logic K in whatever language we're working with.) Note that as far as local satisfaction problems are concerned, if we replace all variables in φ by nominals, obtaining φ' , then φ is satisfiable if and only if φ' is satisfiable. For this reason we can restrict ourselves in the proofs to formulas without variables, and we won't need to mention variable assignments. However at the end of this section we examine the complexity of the *global* K-satisfaction problem and an interesting difference between nominals and world variables emerges.

We will consider expansions with $@$ and nominals of the following modal languages:

1. the unimodal language; that is, the language with one operator $\Box$ and its dual $\Diamond$;
2. the basic tense language; that is, the language with two operators G and H and their respective duals F and P, where the accessibility relation for H and P is the converse of the relation for G and F;
3. their multimodal versions: frames come with a family of accessibility relations $\{R_i\}_{i \in I}$, each has its own forward looking modality $[i]$ and dual $\langle i \rangle$, and in the tense case, also their backward looking counterparts $[i]^\smile$ and $\langle i \rangle^\smile$.

The complexity of the K-satisfaction problem for all these modal languages is known:

Theorem 6.1

- i. [Lad77] *The K-satisfaction problem in the unimodal language is PSPACE-hard.*
- ii. [Spa93b] *The K-satisfaction problem for the tensed multimodal languages is in PSPACE.*

That is, for ordinary modal languages the additional expressive power obtained by adding a backwards looking operator does not lead to an increase in complexity, and neither does the addition of more modalities, including multiple backwards looking operators. However if we hybridize these languages, the situation is rather different. For a start, adding even just one nominal to tense logic causes a jump in complexity from PSPACE to EXPTIME:

Theorem 6.2 *The K-satisfaction problem for a basic tense language containing at least one nominal is EXPTIME-hard.*

PROOF. We shall reduce the EXPTIME-complete *global* K-satisfaction problem for unimodal languages to the (local) K-satisfaction problem for a basic tense language that contains at least one nominal. The global K-satisfaction problem for unimodal languages is this: given a formula φ in the unimodal language, does there exist a Kripke model $\mathfrak{M}$ such that $\mathfrak{M} \models \varphi$ (in other words, where φ is true in *all* worlds)? The EXPTIME-completeness of this problem is an easy consequence of (the proof of) the EXPTIME-completeness of modal logic K expanded with the universal modality in [Spa93a].

Our reduction uses the *spy-point* technique from [BS95]. Define the following translation function $(\cdot)^t$ from ordinary unimodal formulas to formulas in a tense language that contains at least one nominal i :

$$\begin{aligned} p^t &= p \\ (\neg\varphi)^t &= \neg\varphi^t \\ (\varphi \wedge \psi)^t &= \varphi^t \wedge \psi^t \\ (\Diamond\varphi)^t &= F(Pi \wedge \varphi^t), \end{aligned}$$

Note that i is a *fixed* nominal in this translation. Clearly $(\cdot)^t$ is a linear reduction. We claim that for any formula φ , φ is globally K-satisfiable if and only if $i \wedge G(Pi \rightarrow \varphi^t)$ is K-satisfiable.

For the left to right direction, let $\mathfrak{M} \models \varphi$, where $\mathfrak{M} = \langle M, R, V \rangle$ is an ordinary Kripke model. Define $\mathfrak{M}^*$ as follows:

$$\begin{aligned} M^* &= M \cup \{i\}, \\ R^* &= R \cup \{(i, m) \mid m \in M\}, \\ V^* &= V \cup \{(n, \{i\}) \mid \text{for all nominals } n\}. \end{aligned}$$

$\mathfrak{M}^*$ is a hybrid model, for all nominals (including i) are interpreted by the singleton set $\{i\}$, our spy-point. We claim that for all $m \in M$, for all ψ , we have $\mathfrak{M}, m \models \psi$ if and only if $\mathfrak{M}^*, m \models \psi^t$. This follows by a simple induction. The only interesting step is for $\Diamond$:

$$\begin{aligned} &\mathfrak{M}, m \models \Diamond\psi \\ \Leftrightarrow & (\exists m' \in M) : Rmm' \ \& \ \mathfrak{M}, m' \models \psi \\ \Leftrightarrow & (\exists m' \in M^*) : R^*mm' \ \& \ \mathfrak{M}^*, m' \models \psi^t \ \& \ R^*im' \text{ (by IH and definition of } R^*) \\ \Leftrightarrow & \mathfrak{M}^*, m \models F(Pi \wedge \psi^t) \\ \Leftrightarrow & \mathfrak{M}^*, m \models (\Diamond\psi)^t \end{aligned}$$

It follows that $\mathfrak{M}^*, i \models i \wedge G(Pi \rightarrow \varphi^t)$, as desired.

For the other direction, let $\mathfrak{M}, w \models i \wedge G(Pi \rightarrow \varphi^t)$, where $\mathfrak{M} = \langle M, R, V \rangle$ is a hybrid model. Define $\mathfrak{M}^*$ as follows

$$\begin{aligned} M^* &= \{m \in M \mid Rwm\}, \\ R^* &= R_{\upharpoonright M^*}, \\ V^* &= V_{\upharpoonright M^*}. \end{aligned}$$

We claim that for all $m \in M^*$, for all ψ , $\mathfrak{M}, m \models \psi^t$ if and only if $\mathfrak{M}^*, m \models \psi$. Again we only present the inductive step for $\Diamond$:

$$\begin{aligned} &\mathfrak{M}, m \models F(Pi \wedge \psi^t) \\ \Leftrightarrow & (\exists m' \in M) : Rmm' \ \& \ Rwm' \ \& \ \mathfrak{M}, m' \models \psi^t \\ \Leftrightarrow & (\exists m' \in M^*) : Rmm' \ \& \ Rwm' \ \& \ \mathfrak{M}, m' \models \psi^t \\ \Leftrightarrow & (\exists m' \in M^*) : R^*mm' \ \& \ \mathfrak{M}^*, m' \models \psi \text{ (by IH and definition of } M^*) \\ \Leftrightarrow & \mathfrak{M}^*, m \models \Diamond\psi. \end{aligned}$$

For all $m \in M^*$, Rwm holds, whence for all $m \in M^*$, $\mathfrak{M}, m \models Pi$. So, since $\mathfrak{M}, w \models G(Pi \rightarrow \varphi^t)$, for all $m \in M^*$, $\mathfrak{M}, m \models \varphi^t$. Hence by our last claim $\mathfrak{M}^* \models \varphi$, which is what we needed to show. QED

We obtain the corresponding upper bound by translating into first-order logic. It turns out that we end up in (a version of) the guarded fragment with two variables (see [AvBN98]). The satisfaction problem of this fragment is known to be EXPTIME-complete, whence the desired result.

In fact we will prove a stronger result. The *universal modality* E is a modality with the following interpretation: in any model $\mathfrak{M}$, at any $m \in M$, $\mathfrak{M}, m \models E\varphi$ iff for some $m' \in M$, $\mathfrak{M}, m' \models \varphi$. (That is, with the aid of the universal modality we can scan the entire model for information.) The universal modality is very strong (see [GP92] for a detailed discussion) but for present purposes we need merely observe that, given E , we can define $@_i\varphi$ to be $E(i \wedge \varphi)$. In spite of its strength, we have the following upper bound:

Theorem 6.3 *The K-satisfaction problem for a multimodal tense language expanded with both nominals and the universal modality is EXPTIME-complete.*

PROOF. The lower bound was proved in the previous theorem. For the upper bound, use the standard translation to first-order logic with the following additional clause for the universal modality:

$$ST_x(E\varphi) = \exists x.ST_x(\varphi).$$

The translation only uses 2 variables. Moreover, inspection shows that the range of the translation belongs to the packed fragment defined in [Mar97]. (It does not fall in the guarded fragment as defined in [AvBN98] because the translation of E is not guarded, though a translation into the guarded fragment could be obtained by relativizing with a unary predicate $M(x)$ uniformly in all clauses of the translation.) But for a fixed number of variables, the satisfaction problem of the packed fragment is EXPTIME-complete; see [Mar97]. QED

Summing up: the addition of even one nominal to a basic tense language yields an EXPTIME-hard local K-satisfaction problem. However, adding further nominals, multiple forwards and backwards looking modalities, and even the universal modality, doesn't take us any higher in the complexity hierarchy.

This completes our complexity analysis of tense languages, so let's turn to a positive result: (multimodal) $\mathcal{H}(@)$ has the same complexity as ordinary propositional modal logic.

Theorem 6.4 *The K-satisfaction problem of multimodal language enriched with nominals and $@$ is PSPACE-complete.*

PROOF. The lower bound follows from Fact 6.1. We show the upper bound by defining the notion of a ξ -game between two players. We will show that the existential player has a winning strategy for the ξ -game if and only if ξ is satisfiable. Moreover every ξ -game stops after at most as many rounds as the modal depth of ξ and the information on the playing board is polynomial in the length of ξ . Using the close correspondence between Alternating Turing Machines (ATM's) and two player games [Chl86], it is straightforward to implement the problem of whether the existential player has a winning strategy in the ξ -game on a PTIME ATM. Because any PTIME ATM algorithm can be turned into a PSPACE Turing Machine program, we obtain our desired result. We present the proof only for unimodal $\mathcal{H}(@)$; it can be straightforwardly extended to the multimodal case.

Fix a formula ξ . A ξ -HS (for Hintikka Set) is a maximal consistent set of subformulas of ξ . We denote the set of subformulas of ξ by $SF(\xi)$. The ξ -game is played as follows. There are two players, $\forall$ belard (male) and $\exists$ loise (female). She starts the game by playing a collection $\{X_0, \dots, X_k\}$ of Hintikka Sets, and she specifies a relation R on them.

$\exists$ loise loses immediately if one of the following conditions is false:

1. X_0 contains ξ , and all the other X_l contain at least one nominal occurring in ξ .
2. no nominal occurs in two different Hintikka Sets.
3. for all X_l , for all $@_i\varphi \in SF(\xi)$, $@_i\varphi \in X_l$ iff $\{i, \varphi\} \subseteq X_k$, for some k .
4. for all $\Diamond\varphi \in SF(\xi)$, if RX_lX_k and $\Diamond\varphi \notin X_l$, then $\varphi \notin X_k$.

Now $\forall$ belard may choose an X_l and a “defect-formula” $\Diamond\varphi \in X_l$. $\exists$ loise must respond with a Hintikka Set Y such that

1. $\varphi \in Y$ and for all $\Diamond\psi \in SF(\xi)$, $\Diamond\psi \notin X_l$ implies that $\psi \notin Y$.
 2. for all $@_i\varphi \in SF(\xi)$, $@_i\varphi \in Y$ iff $\{i, \varphi\} \subseteq X_k$, for some k .
 3. if $i \in Y$ for some nominal i , then Y is one of the Hintikka Sets she played at the start.
- In this case the game stops and $\exists$ loise wins.

If $\exists$ loise cannot find a suitable Y , the game stops and $\forall$ belard wins. If $\exists$ loise does find a suitable Y (one that is not covered by the halting clause in item 3 above) then Y is added to the list of played sets, and play continues.

$\forall$ belard must now choose a defect $\Diamond\varphi$ from the last played Hintikka Set. He can't choose any defect though: in round k he can only choose defects $\Diamond\varphi$ such that the modal depth of $\Diamond\varphi$ is less than or equal to the modal depth of ξ minus k . $\exists$ loise must respond as before. She wins if she can survive all his challenges (in other words, he loses if he reaches a situation where he can't choose any more defects).

It is clear that the ξ -game stops after at most modal depth of ξ many rounds. The size of the information on the board is at any stage of the game polynomial in the length of ξ , as Hintikka Sets are polynomial in the length of ξ and ξ can only contain polynomially many nominals. We claim that $\exists$ loise has a winning strategy iff ξ is satisfiable.

Now the right-to-left direction is clear: $\exists$ loise has a winning strategy if ξ is satisfiable, for she need simply play by reading the required Hintikka sets off the model. The other direction requires more work. Suppose $\exists$ loise has a winning strategy for the ξ -game. We shall create a model $\mathfrak{M}$ for ξ as follows. The domain M is build in steps by following her winning strategy.

M_0 consists of her initial move $\{X_0, \dots, X_n\}$. Suppose M_j is defined. Then M_{j+1} consists of a copy of those Hintikka Sets she plays when using her winning strategy for each of $\forall$ belard's possible moves played in the Hintikka Sets from M_j (except when she plays a HS from her initial move, then of course we do not make a copy). Let M be the disjoint union of all M_j for j smaller than the modal depth of ξ . Set Rmm' iff for all $\Diamond\varphi \in SF(\xi)$, $\Diamond\varphi \notin m \Rightarrow \varphi \notin m'$ holds, and set $V(p) = \{m \in M \mid p \in m\}$. Note that the rules of the game guarantee that nominals are interpreted as singletons.

We claim that the following truth-lemma holds.

Claim 1 *For all $m \in M$ which she plays in round j (i.e., $m \in M_j$), for all φ of modal depth less than or equal to the modal depth of ξ minus j , $\mathfrak{M}, m \Vdash \varphi$ if and only if $\varphi \in m$.*

PROOF OF CLAIM. By induction on the structure of formulas. For atoms, the booleans and $@$ the proof is easy. For $\Diamond$, if $\Diamond\varphi \in m$, then $\forall$ belard challenged this defect, so $\exists$ loise could respond with an m' containing φ . Since for all $\Diamond\varphi \in SF(\xi)$, $\Diamond\varphi \notin m \Rightarrow \varphi \notin m'$ holds, we have Rmm' and by induction hypothesis $\mathfrak{M}, m \Vdash \Diamond\varphi$. If $\Diamond\varphi \notin m$ but Rmm' holds, then by our definition of R , $\varphi \notin m'$, so again $\mathfrak{M}, m \not\Vdash \Diamond\varphi$. $\blacktriangleleft$

Since she plays a Hintikka Set containing ξ in the first round, $\mathfrak{M}$ satisfies ξ . QED

To close this section we discuss the *global K-satisfaction* problem; that is, the problem of deciding whether $\mathfrak{M} \models \varphi$. In the proof of Theorem 6.2 we made use of the fact that the global K-satisfaction problem for ordinary unimodal logic is EXPTIME complete. Indeed, for ordinary unimodal logic we have that the global K-consequence problem —does $\varphi \models^{glo} \psi$ hold?— is EXPTIME complete. It follows from Theorem 6.3 above that the same result holds when we expand even a multimodal tense language with *nominals*, $@$, and the universal modality. This is because the universal modality allows us to capture $\models^{glo}$ for this language. In particular, note that $\varphi \models^{glo} \psi$ iff $\models A\varphi \rightarrow \psi$, where A is the dual of E .

But this is not the end of the matter: what if we were to extend the unimodal language with *variables* instead of nominals? This *does* make a difference. Recall from Section 2 that $\mathfrak{M} \models \varphi$ is defined to hold iff for all g , $\mathfrak{M}, g \Vdash \varphi$. That is, g is *not* held constant. Now, if there are no world variables in φ , g is irrelevant and (as we have just observed) global satisfaction is EXPTIME-complete. But if φ is allowed to contain free variables, the implicit universal quantification in the definition certainly does change matters. In effect, we are surreptitiously using the global hybrid binder $\forall$ mentioned in the introduction, and we wind up with an undecidable global satisfaction problem.

This can be shown as follows. Let φ be the formula $(\Diamond x_1 \wedge \Diamond x_2 \wedge \Diamond x_3) \rightarrow \Box(x_1 \leftrightarrow x_2 \vee x_1 \leftrightarrow x_3 \vee x_2 \leftrightarrow x_3)$. Then for every model $\mathfrak{M} = \langle M, R, V \rangle$, $\mathfrak{M} \models \varphi$ if and only if $\mathfrak{M} \models \forall y x_1 x_2 x_3. ((Ry x_1 \wedge Ry x_2 \wedge Ry x_3) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3))$. In other words φ expresses that every world has at most two R -successors. Similarly we can create a formula without propositional variables expressing that every world has at most three two-step R -successors. But in [Spa93b] it is shown that the global satisfaction problem of the uni-modal logic of the class of frames satisfying these two conditions is undecidable (see the proof of Theorem 4.2.1 and its preliminaries). Note that this argument makes no use of $@$.

Pressing this line of thought further, we can show that if we have at least three modalities in the language, then we obtain an undecidable global consequence problem for the sublanguage containing only one variable; again, no use will be made of $@$. Suppose we are working

in a language with three modalities. Consider the normal modal logic obtained by adding the following axioms to the minimal logic K:

$$\langle r \rangle \langle u \rangle p \rightarrow \langle d \rangle p \qquad \langle d \rangle p \rightarrow [u] \langle r \rangle p.$$

These (Sahlqvist) axioms correspond to the first-order frame conditions $\forall xyz.(Rxy \wedge Uyz) \rightarrow Dxz$ and $\forall xy.(Dxy \rightarrow \forall z.(Uxz \rightarrow Rzy))$, respectively. [Mar97] shows that the global satisfaction problem of this logic is undecidable. But then the problem of deciding whether a modal formula ϕ in the language with diamonds $\{\langle r \rangle, \langle u \rangle, \langle d \rangle\}$ globally follows from $\langle r \rangle \langle u \rangle x \rightarrow \langle d \rangle x$ and $\langle d \rangle x \rightarrow [u] \langle r \rangle x$ (that is, the axioms with the *propositional* variable p replaced by the *world* variable x) must also be undecidable.

It is at present unknown precisely how many world variables are needed to obtain an undecidable global satisfaction problem for the unimodal case.

7 Further Work

In their long (if sparse) history, hybrid languages have attracted a number of enthusiastic advocates. Some have claimed that hybridization is a natural way to “power-up” the expressive power of modal languages, others have been impressed by the proof theoretical options they open up, or the ease with which general completeness can be proved. And underlying most of this work lies a simple (and seductive) idea: that by exploiting the notion of formulas as terms to the full, it should be possible to define systems which in some sense combine the best of modal and classical techniques.

We believe that the results of this paper confirm the interest of hybridization. For a start, our characterization of $\mathcal{H}(\downarrow, @)$ shows that relatively simple tools are capable of capturing first-order fragments that are central from a modal perspective; invariance under generated submodels mirrors the key notion of locality, and it is pleasing that it can be pinned down so simply. Furthermore, the results on interpolation and complexity tend to confirm that we are dealing with a natural collection of ideas, ideas that are well behaved even in relatively weak sublanguages. Finally in writing this paper it has become very clear to us that working with hybrid languages involves a genuine interplay of modal and classical methods (for example, both Ehrenfeucht-Fraïssé games and bisimulations were involved in the expressivity result, and interpolation was proved by blending the modal notion of canonical models with the classical idea of Henkin-models). This is something previous writers on hybrid languages have emphasized (see for example [PT91]) and the natural way these methods blend bodes well for further developments.

Nonetheless, to close this paper it seems more appropriate to emphasize what remains to be done; for the blunt fact remains that compared with orthodox modal languages, the study of hybrid languages is in its infancy. Many fundamental questions have not been satisfactorily resolved, and to close the paper we’re going to discuss a very obvious one that we regard as particularly important, namely:

Which classes of frames are definable using $\mathcal{H}(@)$ formulas whose only atoms are world variables?

In a sense, the standard translation ST already gives us an answer to this question. Let F be a class of frames defined by a sentence φ of the first-order frame language. Then F is

definable by a formula of $\mathcal{H}(@)$ whose only atoms are variables iff there is some formula α in this fragment such that φ is equivalent to the universal closure of $ST(\alpha)$.

Unfortunately, this is not very helpful. Ideally we would like a syntactic characterization of the range of ST when restricted to $\mathcal{H}(@)$ formulas whose only atoms are world variables, together with a reverse translation (like our earlier HT). But at present we only have partial results in this direction.

What about a semantic characterization? Here we can do a little better by introducing the concept of an *@-bisimulation*. Let $\mathfrak{M}$ and $\mathfrak{N}$ be models and $\bar{m} \in {}^k M$ and $\bar{n} \in {}^k N$. A relation $B \subseteq M \times N$ is called a *@-k-bisimulation* between $(\mathfrak{M}, \bar{m})$ and $(\mathfrak{N}, \bar{n})$ if it satisfies the following conditions:

1. B is a modal bisimulation;
2. $(\forall 0 \leq i < k) : (m_i, n_i) \in B$;
3. $(\forall x \in N) : (m_i, x) \in B \Rightarrow x = n_i$;
4. $(\forall x \in M) : (x, n_i) \in B \Rightarrow x = m_i$.

We denote this by $(\mathfrak{M}, \bar{m}) \stackrel{B}{\sim} (\mathfrak{N}, \bar{n})$. A first-order formula $\varphi(\bar{x}, y)$ is called invariant for *@-k-bisimulation*, if for all models $\mathfrak{M}, \mathfrak{N}$, for all k -tuples $\bar{m}$ and $\bar{n}$, for all B such that $(\mathfrak{M}, \bar{m}) \stackrel{B}{\sim} (\mathfrak{N}, \bar{n})$,

$$(m, n) \in B \Rightarrow \mathfrak{M} \models \varphi(\bar{m}, m) \Leftrightarrow \mathfrak{N} \models \varphi(\bar{n}, n).$$

Theorem 7.1 *A first-order formula $\varphi(\bar{x}, y)$ is invariant for @-k-bisimulation if and only if it is equivalent to the standard translation of an $\mathcal{H}(@)$ -formula containing the variables $\bar{x}$.*

PROOF. The proof follows the standard modal lines. We only sketch the differences due to the new operator and the variables. For the preservation part, suppose $(\mathfrak{M}, \bar{m}) \stackrel{B}{\sim} (\mathfrak{N}, \bar{n})$ and $(m, n) \in B$. Then $\mathfrak{M}, \bar{m}, m \Vdash v_i \Leftrightarrow m = m_i$; but this implies that $n = n_i$, whence $\mathfrak{N}, \bar{n}, n \Vdash v_i$. For @-formulas, $\mathfrak{M}, \bar{m}, m \Vdash @_{v_i} \varphi \Leftrightarrow \mathfrak{M}, \bar{m}, m_i \Vdash \varphi$; but then since $(m_i, n_i) \in B$ and inductive hypothesis, $\mathfrak{N}, \bar{n}, n_i \Vdash \varphi \Leftrightarrow \mathfrak{N}, \bar{n}, n \Vdash @_{v_i} \varphi$.

For the characterization part we also copy the standard diagram-chasing proof. Let $\mathfrak{M}, \bar{m}, m$ and $\mathfrak{N}, \bar{n}, n$ have the same hybrid theory. Define a relation B between elements in the ω -saturated extensions $\mathfrak{M}^+$ and $\mathfrak{N}^+$ as follows:

$$(x, y) \in B \Leftrightarrow (\forall \varphi) : \mathfrak{M}^+, \bar{m}, x \Vdash \varphi \Leftrightarrow \mathfrak{N}^+, \bar{n}, y \Vdash \varphi.$$

The standard proof shows that B is a modal bisimulation. We check the extra conditions. For all i , $(m_i, n_i) \in B$ holds by the following argument. $\mathfrak{M}^+, \bar{m}, m_i \Vdash \varphi \Leftrightarrow \mathfrak{M}^+, \bar{m}, m \Vdash @_{v_i} \varphi \Leftrightarrow \mathfrak{N}^+, \bar{n}, n \Vdash @_{v_i} \varphi \Leftrightarrow \mathfrak{N}^+, \bar{n}, n_i \Vdash \varphi$. The other two conditions are satisfied because of the following. Let $(m_i, y) \in B$. Since $\mathfrak{M}^+, \bar{m}, m_i \Vdash v_i$, also $\mathfrak{N}^+, \bar{n}, y \Vdash v_i$. But then $y = n_i$. This finishes the outline of the proof. QED

Using this notion it is easy to show that $\exists y.(Rxy \wedge Ryy)$ is not equivalent to an @-formula with one free variable, and that $Rxy \wedge Rxz \rightarrow \exists w.(Ryw \wedge Rzw)$ is not equivalent to an @-formula in three free variables. And it does tell us something about frame definability:

Corollary 7.2 *Let F be a class of frames defined by a sentence φ of the first-order frame language. Then F is definable by a formula of $\mathcal{H}(\downarrow, @)$ whose only atoms are variables iff φ is equivalent to the universal closure of a formula that is invariant under $@-k$ -bisimulations.*

But in practice this characterization does not seem to be particularly helpful. One of the nicer aspects of our characterization of $\mathcal{H}(\downarrow, @)$ was the way the notion of k -bisimulation linked with the notion of generated submodels to yield a natural “geometric” characterization of definable frame classes. It is hard to see what the definition of $@-k$ -bisimulation invariance tells us about frame geometry. Clearly new ideas are called for here.

Acknowledgments The authors want to thank Johan van Benthem. His suggestion of using the bounded fragment greatly improved the presentation of the results in Section 3. Maarten Marx is supported by the Spinoza Project ‘Logic in Action.’ This article arose from a visit of Marx to Blackburn partially funded by the Dutch research School in Logic and the ‘Internationaliserings’-program of the Netherlands Organization for Scientific Research.

References

- [AvBN98] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
- [Ben83] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.
- [Ben96] J. van Benthem. *Exploring Logical Dynamics*. Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996.
- [Bet53] E. Beth. On Padoa’s method in the theory of definition. *Nederl. Akad. Wetensch. Proc. Ser. A*. **56** = *Indagationes Math.*, 15:330–339, 1953.
- [Bla93] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 34(1):56–83, 1993.
- [Bla98] P. Blackburn. Internalizing labeled deduction. Technical Report CLAUS-Report 102, Computerlinguistik, Universität des Saarlandes, 1998. <http://www.coli.uni-sb.de/cl/clus>.
- [BS95] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.
- [BS98] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyashev, editors, *Advances in Modal Logic*, volume 1, pages 41–62. CSLI Publications, Stanford University, 1998.
- [BT98] P. Blackburn and M. Tzakova. Hybrid languages and temporal logics (full version). Technical Report CLAUS-Report 96, Computerlinguistik, Universität des Saarlandes, 1998. <http://www.coli.uni-sb.de/cl/clus>.
- [BT99] P. Blackburn and M. Tzakova. Hybrid languages and temporal logics. *Logic Journal of the IGPL*, 7(1):27–54, 1999.

- [Bul70] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.
- [Chl86] B. Chlebus. Domino-tiling games. *Journal of Computer and System Sciences*, 32(3):374–392, 1986.
- [CK90] C. Chang and H. Keisler. *Model theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, third edition, 1990.
- [Com85] B. Comrie. *Tense*. Cambridge University Press, 1985.
- [Cra57] W. Craig. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.
- [Cze82] J. Czelakowski. Logical matrices and the amalgamation property. *Studia Logica*, XLI(4):329–341, 1982.
- [End72] H. Enderton. *A mathematical introduction to logic*. Academic Press, New York, 1972.
- [GG93] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, pages 607–636, 1993.
- [Gor96] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language and Information*, 5(1):1–24, 1996.
- [GP92] V. Goranko and S. Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.
- [Hod93] W. Hodges. *Model theory*. Cambridge University Press, Cambridge, 1993.
- [Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, 1977.
- [Mar97] M. Marx. Complexity of modal logics of relations. Technical Report ML-97-02, Institute for Logic, Language and Computation, University of Amsterdam, May 1997.
- [Mar99] M. Marx. Interpolation in (fibered) modal logic. In A. Haeberer, editor, *Proceedings of AMAST 1998*, 1999.
- [MV97] M. Marx and Y. Venema. *Multi-dimensional Modal Logic*. Applied Logic Series. Kluwer Academic Publishers, 1997.
- [Pri67] A. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.
- [PT85] S. Passy and T. Tinchev. Quantifiers in combinatory PDL: completeness, definability, incompleteness. In *Fundamentals of computation theory (Cottbus, 1985)*, pages 512–519. Springer, Berlin, 1985.
- [PT91] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.

- [Rei47] H. Reichenbach. *Elements of Symbolic Logic*. Random House, New York, 1947.
- [Sel91] J. Seligman. A cut-free sequent calculus for elementary situated reasoning. Technical Report HCRC/RP-22, HCRC, Edinburgh, 1991.
- [Sel97] J. Seligman. The logic of correct description. In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.
- [Spa93a] E. Spaan. *Complexity of modal logics*. PhD thesis, University of Amsterdam, Institute for Logic, Language and Computation, 1993.
- [Spa93b] E. Spaan. The complexity of propositional tense logics. In *Diamonds and defaults (Amsterdam, 1990/1991)*, pages 287–307. Kluwer Acad. Publ., Dordrecht, 1993.
- [Tza98] M. Tzakova. Tableaux calculi for hybrid logics. Manuscript. Will be made available at <http://www.mpi-sb.mpg.de/~tzakova/>, 1998.
- [Ven94] Y. Venema. A modal logic for quantification and substitution. *Bulletin of the IGPL*, 2(1):31–45, 1994.

A Road-map on Complexity for Hybrid Logics

Carlos Areces¹, Patrick Blackburn², and Maarten Marx³

¹ ILLC. UvA. Plantage Muidergracht 24, 1018TV. The Netherlands.

phone: +31 (20) 525-6925 e-mail: carlos@wins.uva.nl

² Computerlinguistik. Universität des Saarlandes. D-66041 Saarbrücken. Germany.

phone: +49 (681) 302-4501 e-mail: patrick@coli.uni-sb.de

³ Department of Artificial Intelligence, Free University, Amsterdam, and

ILLC. UvA. Plantage Muidergracht 24, 1018TV. The Netherlands.

phone: +31 (20) 525-2459 e-mail: marx@wins.uva.nl

Abstract. Hybrid languages are extended modal languages which can refer to (or even quantify over) states. Such languages are better behaved proof theoretically than ordinary modal languages for they internalize the apparatus of labeled deduction. Moreover, they arise naturally in a variety of applications, including description logic and temporal reasoning. Thus it would be useful to have a map of their complexity-theoretic properties, and this paper provides one.

Our work falls into two parts. We first examine the basic hybrid language and its multi-modal and tense logical cousins. We show that the basic hybrid language (and indeed, multi-modal hybrid languages) are no more complex than ordinary uni-modal logic: all have PSPACE-complete K-satisfiability problems. We then show that adding even one nominal to tense logic raises complexity from PSPACE to EXPTIME. In the second part we turn to stronger hybrid languages in which it is possible to *bind* nominals. We prove a general expressivity result showing that even the weak form of binding offered by the $\downarrow$ operator easily leads to undecidability.

Keywords. Computational Complexity, Modal and Temporal Logic, Description Logic, Labeled Deduction.

1 Introduction

Hybrid languages are modal languages which use atomic formulas called nominals to name states. Nominals are true at exactly one state in any model; they “name” this state by being true there and nowhere else. Although a wide range of hybrid languages have been studied, including hybrid languages in which it is possible to bind nominals in various ways, little is known about their computational complexity. This paper is an attempt to fill the gap.

Before going further, let’s be precise about the syntax and semantics of the *basic hybrid language* $\mathcal{H}(@)$, the weakest language we shall consider in the paper.

Definition 1 (Syntax). Let $\text{PROP} = \{p, q, r, \dots\}$ be a countable set of *propositional variables* and $\text{NOM} = \{i, j, k, \dots\}$ a countable set of *nominals*, disjoint

from PROP . We call $\text{ATOM} = \text{PROP} \cup \text{NOM}$ the set of *atoms*. The *well-formed formulas* of the hybrid language (over ATOM) are

$$\varphi := a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box\varphi \mid @_i\varphi$$

where $a \in \text{ATOM}$, and $i \in \text{NOM}$. As usual, $\Diamond\varphi$ is defined to be $\neg\Box\neg\varphi$. A formula which contains no symbols from PROP is called *pure*.

Thus, syntactically speaking, the basic hybrid language is a two-sorted uni-modal language which contains a NOM indexed collection of operators $@_i$. Now for the semantics.

Definition 2 (Semantics). A (hybrid) *model* $\mathfrak{M}$ is a triple $\mathfrak{M} = \langle M, R, V \rangle$ such that M is a non-empty set, R is a binary relation on M , and $V : \text{ATOM} \rightarrow \text{Pow}(M)$ is such that for all $i \in \text{NOM}$, $V(i)$ is a singleton subset of M . We usually call the elements of M states, R is the *transition relation*, and V is the *valuation*. A *frame* is a pair $\mathfrak{F} = \langle M, R \rangle$, that is, a model without a valuation.

Let $\mathfrak{M} = \langle M, R, V \rangle$ be a model and $m \in M$. Then the *satisfaction relation* is defined by:

$$\begin{aligned} \mathfrak{M}, m \Vdash a & \quad \text{iff } m \in V(a), a \in \text{ATOM} \\ \mathfrak{M}, m \Vdash \neg\varphi & \quad \text{iff } \mathfrak{M}, m \not\Vdash \varphi \\ \mathfrak{M}, m \Vdash \varphi \wedge \psi & \quad \text{iff } \mathfrak{M}, m \Vdash \varphi \text{ and } \mathfrak{M}, m \Vdash \psi \\ \mathfrak{M}, m \Vdash \Box\varphi & \quad \text{iff } \forall m' (Rmm' \Rightarrow \mathfrak{M}, m' \Vdash \varphi) \\ \mathfrak{M}, m \Vdash @_i\varphi & \quad \text{iff } \mathfrak{M}, m' \Vdash \varphi, \text{ where } V(i) = \{m'\}, i \in \text{NOM}. \end{aligned}$$

A formula φ is *satisfiable* if there is a model $\mathfrak{M}$, and a state $m \in M$ such that $\mathfrak{M}, m \Vdash \varphi$. We write $\mathfrak{M} \models \varphi$ iff for all $m \in M$, $\mathfrak{M}, m \Vdash \varphi$. If $\mathfrak{F}$ is a frame, and for all valuations V on $\mathfrak{F}$ we have $\langle \mathfrak{F}, V \rangle \models \varphi$, we say φ is *valid* on $\mathfrak{F}$ and write $\mathfrak{F} \models \varphi$.

Because valuations assign *singletons* to nominals, it is clear that each nominal is satisfied at exactly one state in any model. And the clause for formulas of the form $@_i\varphi$ simply says: to evaluate $@_i\varphi$, jump to the unique state named by i and evaluate φ there.

There are at least two reasons for being interested in hybrid languages. First, they can be seen as modal languages which internalize the ideas underlying labeled deduction systems. Second, hybrid languages arise naturally in many applications.

Hybrid languages and labeled deduction Labeled deduction (see [Gab96]) is built around the notation $l:\varphi$. Here the meta linguistic symbol $:$ associates the meta linguistic label l with the formula φ . This has a natural modal interpretation: regard labels as names for states and read $l:\varphi$ as asserting that φ is satisfied at l . Labeled deduction proceeds by manipulating such labels to guide proof search; the approach has become an important way of handling modal proof theory.

The basic hybrid language places the apparatus of labeled deduction in the *object language*: nominals are essentially object-level labels, and the formula $@_i\varphi$

asserts in the object language what $i:\varphi$ asserts in the metalanguage. And indeed, hybrid languages turn out to be proof-theoretically well behaved. For a start, the basic hybrid language enables us to directly “internalize” labeled deduction (see [Bla98]), and to define sequent calculi and natural deduction systems (see [Sel97]). In fact, even if @ is dropped from the language, elegant Fitting-style systems which exploit the presence of nominals can be defined (see [Tza98]).

Furthermore, such calculi *automatically* handle the logics of a wide range of frame classes, including many that are awkward for ordinary modal logic. To give a simple example, no ordinary modal formula defines irreflexivity (that is, no ordinary modal formula is valid on precisely the irreflexive frames). But the (pure) formula $@_i \neg \Diamond i$ does so, as the reader can easily check. Moreover, when used as an additional axiom, this formula (and indeed, *any* pure formula) is complete with respect to the class of frames it defines. For a full discussion of these issues, see [Bla98, BT99].

Hybrid languages and applied logic Modal logicians like to claim that notational variants of modal logics are often reinvented by workers in artificial intelligence, computational linguistics, and other fields — in this case, it would be more accurate to say that it is *hybrid languages* which are reinvented in this way. For example, it is well known that the description language $\mathcal{ALC}$ (see [SSS91]) is a notational variant of multi-modal logic (see [Sch91]). But this relation is established at the level of what is called the TBox reasoning. TBox reasoning is complemented with ABox assertions, which corresponds to the addition of nominals (see [AdR99, BS98]). Moreover, many authors have pointed out how natural hybrid languages are for temporal reasoning (see [Bul70, Gor96, BT99]). Among other things, hybrid languages make it possible to introduce specific times (days, dates, etc.), and to define many temporally relevant frame properties (such as irreflexivity, asymmetry, trichotomy, and directedness) that ordinary modal languages cannot handle. Furthermore, if one starts with a modal interval language and adds nominals and @, one obtains variants of the Holds(t, φ)-driven interval logics discussed in [All84] (with @ playing the role of Holds).

The emergence of hybrid languages in applied logic is not particularly surprising. Modal languages offer a simple notation for modeling many problems — but the ability to reason about what happens at a *particular* state is often important and this is precisely what orthodox modal languages lack. This seems to have encouraged a drift (often implicit) towards hybrid languages.

Our work falls into two parts. We first examine the basic hybrid language and its multi-modal and tense logical variants. We show that the basic and even the multi-modal hybrid languages are no more complex than ordinary uni-modal logic: all have PSPACE-complete K-satisfiability problems. We also show that adding even one nominal to tense logic raises complexity from PSPACE to EXPTIME. In the second part of the paper we turn to stronger hybrid languages in which it is possible to *bind* nominals. We shall show, via a general expressivity result called the *Spypoint Theorem*, that even the restricted form of binding offered by the $\downarrow$ operator easily leads to undecidability.

2 Complexity of the basic hybrid language

We begin with a positive result. We know from [Lad77] that ordinary propositional uni-modal logic has a PSPACE-complete K-satisfaction problem (the K meaning that no restrictions are placed on the transition relation R). What happens when we add nominals and @ to form the basic hybrid language? The answer (up to a polynomial) is: *nothing*.

Theorem 1. *The K-satisfaction problem for the basic hybrid language is PSPACE-complete.*

Proof. The lower bound follows from [Lad77]. We show the upper bound by defining the notion of a ξ -game between two players. We will show that the existential player has a winning strategy for the ξ -game if and only if ξ is satisfiable. Moreover every ξ -game stops after at most as many rounds as the modal depth of ξ and the information on the playing board is polynomial in the length of ξ . Using the close correspondence between Alternating Turing Machines (ATM’s) and two player games [Chl86], it is straightforward to implement the problem of whether the existential player has a winning strategy in the ξ -game on a PTIME ATM. Because any PTIME ATM algorithm can be turned into a PSPACE Turing Machine program, we obtain our desired result. We present the proof only for uni-modal $\mathcal{H}(@)$; it can be straightforwardly extended to the multi-modal case.

Fix a formula ξ . A ξ -Hintikka set is a maximal consistent set of subformulas of ξ . We denote the set of subformulas of ξ by $SF(\xi)$. The ξ -game is played as follows. There are two players, $\forall$ belard (male) and $\exists$ loise (female). She starts the game by playing a collection $\{X_0, \dots, X_k\}$ of Hintikka sets and specifying a relation R on them.

$\exists$ loise loses immediately if one of the following conditions is false:

1. X_0 contains ξ , and all others X_l contain at least one nominal occurring in ξ .
2. no nominal occurs in two different Hintikka sets.
3. for all X_l , for all $@_i \varphi \in SF(\xi)$, $@_i \varphi \in X_l$ iff $\{i, \varphi\} \subseteq X_k$, for some k .
4. for all $\Diamond \varphi \in SF(\xi)$, if $RX_l X_k$ and $\Diamond \varphi \notin X_l$, then $\varphi \notin X_k$.

Now $\forall$ belard may choose an X_l and a “defect-formula” $\Diamond \varphi \in X_l$. $\exists$ loise must respond with a Hintikka set Y such that

1. $\varphi \in Y$ and for all $\Diamond \psi \in SF(\xi)$, $\Diamond \psi \notin X_l$ implies that $\psi \notin Y$.
2. for all $@_i \varphi \in SF(\xi)$, $@_i \varphi \in Y$ iff $\{i, \varphi\} \subseteq X_k$, for some k .
3. if $i \in Y$ for some nominal i , then Y is one of the Hintikka sets she played at the start. In this case the game stops and $\exists$ loise wins.

If $\exists$ loise cannot find a suitable Y , the game stops and $\forall$ belard wins. If $\exists$ loise does find a suitable Y (one that is not covered by the halting clause in item 3 above) then Y is added to the list of played sets, and play continues.

$\forall$ belard must now choose a defect $\Diamond \varphi$ from the last played Hintikka set with the following restriction: in round k he can only choose defects $\Diamond \varphi$ such that

the modal depth of $\Diamond\varphi$ is less than or equal to the modal depth of ξ minus k . Eloise must respond as before. She wins if she can survive all his challenges (in other words, he loses if he reaches a situation where he can't choose any more defects).

It is clear that the ξ -game stops after at most modal depth of ξ many rounds. The size of the information on the board is at any stage of the game polynomial in the length of ξ , as Hintikka sets are polynomial in the length of ξ and ξ can only contain polynomially many nominals. We claim that Eloise has a winning strategy iff ξ is satisfiable.

Now the right-to-left direction is clear: Eloise has a winning strategy if ξ is satisfiable, for she need simply play by reading the required Hintikka sets off the model. The other direction requires more work. Suppose Eloise has a winning strategy for the ξ -game. We shall create a model $\mathfrak{M}$ for ξ as follows. The domain M is build in steps by following her winning strategy. M_0 consists of her initial move $\{X_0, \dots, X_n\}$. Suppose M_j is defined. Then M_{j+1} consists of a copy of those Hintikka sets she plays when using her winning strategy for each of $\forall$ belard's possible moves played in the Hintikka sets from M_j (except when she plays a Hintikka set from her initial move, then of course we do not make a copy). Let M be the disjoint union of all M_j for j smaller than the modal depth of ξ . Set Rmm' iff for all $\Diamond\varphi \in SF(\xi)$, $\Diamond\varphi \notin m \Rightarrow \varphi \notin m'$ holds, and set $V(p) = \{m \in M \mid p \in m\}$. Note that the rules of the game guarantee that nominals are interpreted as singletons.

We claim that the following truth-lemma holds. For all $m \in M$ which she plays in round j (i.e., $m \in M_j$), for all φ of modal depth less than or equal to the modal depth of ξ minus j , $\mathfrak{M}, m \Vdash \varphi$ if and only if $\varphi \in m$.

PROOF OF CLAIM. By induction on the structure of formulas. For atoms, the booleans and $@$ the proof is easy. For $\Diamond$, if $\Diamond\varphi \in m$, then $\forall$ belard challenged this defect, so Eloise could respond with an m' containing φ . Since for all $\Diamond\varphi \in SF(\xi)$, $\Diamond\varphi \notin m \Rightarrow \varphi \notin m'$ holds, we have Rmm' and by induction hypothesis $\mathfrak{M}, m' \Vdash \varphi$. If $\Diamond\varphi \notin m$ but Rmm' holds, then by our definition of R , $\varphi \notin m'$, so again $\mathfrak{M}, m \not\Vdash \Diamond\varphi$.

Since she plays a Hintikka set containing ξ in the first round, $\mathfrak{M}$ satisfies ξ .

This result generalizes to the multi-modal case. Recall that in a multi-modal language we have an indexed collection of modalities $[\alpha]$, each interpreted by some relation R_α . From [HM92] we know that the K-satisfaction problem for multi-modal languages is PSPACE-complete (here the K means that no restrictions are placed on the individual R_α , or on the way they are inter-related). If we add nominals and $@$ to such a language, the previous proof straightforwardly extends to show that we are still PSPACE-complete.

We have already mentioned that the description language $\mathcal{ALC}$ with assertional axioms is a restriction of multi-modal logic enriched with nominals an $@$: nominals cannot be freely used in formulas and can only act as subindices of the $@$ operator. The logic $\mathcal{ALCO}$ [Sch94] moves closer to $\mathcal{H}(@)$ by allowing the

formation of concepts by means of sets of nominals. Eliminating the restrictions on $@$ from such a language in effect would give us an equational calculus for reasoning about individuals, and make it possible to specify additional frame properties.

3 Hybrid Tense Logic

The language of tense logic is a bimodal language; its $\Box$ -modalities are written G and H and the respective $\Diamond$ -modalities F and P . But these modalities are inter-related: while G and F look forward along the transition relation R , the H and P modalities look *backwards* along this relation (that is, H and P are interpreted using the converse of R). Now, we know from [Spa93b] that the K-satisfaction problem for tense logic is PSPACE-complete. However because G and H are inter-related the results of the previous section are not applicable. And in fact, adding even one nominal to tense logic causes a jump in complexity from PSPACE to EXPTIME, and we *don't* need to add $@$ to obtain this result. Our proof uses the *spy-point* technique from [BS95]; we will be exploring this technique in great detail in the following section when we discuss undecidable systems.

Theorem 2. *The K-satisfaction problem for a language of tense logic containing at least one nominal is EXPTIME-hard.*

Proof. We shall reduce the EXPTIME-complete *global* K-satisfaction problem for uni-modal languages to the (local) K-satisfaction problem for a basic tense language that contains at least one nominal. The global K-satisfaction problem for uni-modal languages is this: given a formula φ in the uni-modal language, does there exist a Kripke model $\mathfrak{M}$ such that $\mathfrak{M} \models \varphi$ (in other words, where φ is true in *all* states)? The EXPTIME-completeness of this problem is an easy consequence of (the proof of) the EXPTIME-completeness of modal logic K expanded with the universal modality in [Spa93a].

Define the following translation function $(\cdot)^t$ from ordinary uni-modal formulas to formulas in a tense language that contains at least one nominal i : $p^t = p$, $(\neg\varphi)^t = \neg\varphi^t$, $(\varphi \wedge \psi)^t = \varphi^t \wedge \psi^t$, $(\Diamond\varphi)^t = F(Pi \wedge \varphi^t)$. Note that i is a *fixed* nominal in this translation. Clearly $(\cdot)^t$ is a linear reduction. We claim that for any formula φ , φ is globally K-satisfiable if and only if $i \wedge G(Pi \rightarrow \varphi^t)$ is K-satisfiable.

For the left to right direction, let $\mathfrak{M} \models \varphi$, where $\mathfrak{M} = \langle M, R, V \rangle$ is an ordinary Kripke model. Define $\mathfrak{M}^*$ as follows: $M^* = M \cup \{i\}$, $R^* = R \cup \{(i, m) \mid m \in M\}$, $V^* = V \cup \{(n, \{i\}) \mid \text{for all nominals } n\}$. $\mathfrak{M}^*$ is a hybrid model where all nominals (including i) are interpreted by the singleton set $\{i\}$, our spy-point. We claim that for all $m \in M$, for all ψ , we have $\mathfrak{M}, m \Vdash \psi$ if and only if $\mathfrak{M}^*, m \Vdash \psi^t$. This follows by a simple induction. The only interesting step is for $\Diamond$:

$$\begin{aligned} & \mathfrak{M}, m \Vdash \Diamond\psi \\ \iff & (\exists m' \in M) : Rmm' \ \& \ \mathfrak{M}, m' \Vdash \psi \\ \iff & (\exists m' \in M^*) : R^*mm' \ \& \ \mathfrak{M}^*, m' \Vdash \psi^t \ \& \ R^*im' \text{ (by IH and def. of } R^*) \\ \iff & \mathfrak{M}^*, m \Vdash F(Pi \wedge \psi^t) \\ \iff & \mathfrak{M}^*, m \Vdash (\Diamond\psi)^t. \end{aligned}$$

It follows that $\mathfrak{M}^*, i \Vdash i \wedge G(Pi \rightarrow \varphi^t)$, as desired.

For the other direction, let $\mathfrak{M}, w \Vdash i \wedge G(Pi \rightarrow \varphi^t)$, where $\mathfrak{M} = \langle M, R, V \rangle$ is a hybrid model. Define $\mathfrak{M}^*$ as follows: $M^* = \{m \in M \mid Rwm\}$, $R^* = R|_{M^*}$, $V^* = V|_{M^*}$. We claim that for all $m \in M^*$, for all ψ , $\mathfrak{M}, m \Vdash \psi^t$ if and only if $\mathfrak{M}^*, m \Vdash \psi$. Again we only present the inductive step for $\Diamond$:

$$\begin{aligned} \mathfrak{M}, m \Vdash F(Pi \wedge \psi^t) \\ \iff (\exists m' \in M) : Rmm' \ \& \ Rwm' \ \& \ \mathfrak{M}, m' \Vdash \psi^t \\ \iff (\exists m' \in M^*) : Rmm' \ \& \ Rwm' \ \& \ \mathfrak{M}, m' \Vdash \psi^t \\ \iff (\exists m' \in M^*) : R^*mm' \ \& \ \mathfrak{M}^*, m' \Vdash \psi \text{ (by IH and definition of } M^*) \\ \iff \mathfrak{M}^*, m \Vdash \Diamond\psi. \end{aligned}$$

For all $m \in M^*$, Rwm holds, whence for all $m \in M^*$, $\mathfrak{M}, m \Vdash Pi$. So, since $\mathfrak{M}, w \Vdash G(Pi \rightarrow \varphi^t)$, for all $m \in M^*$, $\mathfrak{M}, m \Vdash \varphi^t$. Hence by our last claim $\mathfrak{M}^* \models \varphi$, which is what we needed to show.

A matching upper bound can be obtained by interpreting the fragment in the guarded fragment with two variables [Grä97].

4 Binding nominals

Once we are used to treating labels as formulas, it is easy to obtain further expressivity. For example, instead of viewing nominals as *names*, we could think of them as *variables over states* and bind them with quantifiers. That is, we could form expressions like

$$\exists x. \Diamond(x \wedge \forall y. \Diamond(y \wedge \Diamond y \wedge p)).$$

This sentence is satisfied at a state w if and only if there is some state x accessible from w such that all states y accessible from x are reflexive and satisfy p . Historically, hybrid languages offering quantification over states were the first to be explored ([Bul70, PT85]). In their multi-modal version, they are essentially description languages which offer full first-order expressivity (see [BS98]). If the underlying modal language is taken to be the modal interval logic described in [Ben83a], the resulting system is essentially the full version of Allen's $\text{Holds}(t, \varphi)$ -based interval logic in which quantification over t is permitted (see [All84]). But because they offer full first-order expressivity over states, such hybrid languages are obviously undecidable.

More recently, there has been interest in hybrid languages which use a weaker binder called $\downarrow$ (see [Gor96, BS95]). Unlike $\exists$ and $\forall$, this is not a quantifier: it is simply a device which binds a nominal to the state where evaluation is being performed (that is, the *current state*). For example, the interplay between $\downarrow$ and $@$ allows us to define the *Until* operator:

$$\text{Until}(\varphi, \psi) := \downarrow x. \Diamond \downarrow y. @_x(\Diamond(y \wedge \varphi) \wedge \Box(\Diamond y \rightarrow \psi)).$$

This works as follows: we name the current state x , use $\Diamond$ to move to an accessible state, which we name y , and then use $@$ to jump us back to x . We then use $\Diamond$

to insist that φ holds at the state named y , while ψ holds at all successors of the current state that precede this y -labeled state.

$\mathcal{H}(\downarrow, @)$, the extension of $\mathcal{H}(@)$ with the $\downarrow$ binder, is proof theoretically well behaved, and completeness results for a wide class of frames can be obtained automatically (see [BT99, Bla98, Tza98]). But $\downarrow$ turns out to be extremely powerful: not only is $\mathcal{H}(\downarrow, @)$ undecidable, the sublanguage $\mathcal{H}(\downarrow)$ containing *only* the $\downarrow$ binder is too. However the only published undecidability result for $\mathcal{H}(\downarrow)$ is the one in [BS95], and this makes use of $\downarrow$ over a modal language with four modalities. In unpublished work, Valentin Goranko, and Blackburn and Seligman have proved undecidability in the uni-modal case, but these proofs make use of propositional variables to carry out the encoding. We are now going to prove the sharpest undecidability result yet for $\mathcal{H}(\downarrow)$ through a general expressivity result called the *Spypoint Theorem*. Roughly speaking, the Spypoint Theorem shows that $\downarrow$ is powerful enough to encode modal satisfaction over a wide range of Kripke models, and that it doesn't need the help of propositional variables or multiple modalities to do this.

4.1 The language $\mathcal{H}(\downarrow, @)$

Let's first make the syntax and semantics of $\mathcal{H}(\downarrow, @)$ precise.

Definition 3 (Syntax). As in Definition 1, $\text{PROP} = \{p, q, r, \dots\}$ is a countable set of propositional variables, and $\text{NOM} = \{i, j, k, \dots\}$ is a countable set of nominals. To this we add $\text{SVAR} = \{x_1, x_2, \dots\}$ a countable set of *state variables*. We assume that PROP , NOM and SVAR are pairwise disjoint. We call $\text{SSYM} = \text{NOM} \cup \text{SVAR}$ the set of *state symbols*, and $\text{ATOM} = \text{PROP} \cup \text{NOM} \cup \text{SVAR}$ the set of *atoms*. The *well-formed formulas* of $\mathcal{H}(\downarrow, @)$ (over ATOM) are

$$\varphi := a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box\varphi \mid @_s\varphi \mid \downarrow v.\varphi$$

where $a \in \text{ATOM}$, $v \in \text{SVAR}$ and $s \in \text{SSYM}$.

The difference between nominals and state variables is simply this: nominals cannot be bound by $\downarrow$ whereas state variables can. The notions of *free* and *bound* state variable are defined as in first-order logic, with $\downarrow$ the only binding operator. A *sentence* is a formula containing no free state variables. A formula is *pure* if it contains no propositional variables, and *nominal-free* if it contains no nominals. In what follows we assume that some choice of PROP , NOM , and SVAR has been fixed.

Definition 4 (Semantics). Hybrid models $\mathfrak{M}$ are defined as in Definition 2. An *assignment* g for $\mathfrak{M}$ is a mapping $g : \text{SVAR} \rightarrow M$. Given an assignment g , we define the assignment g_m^v by $g_m^v(v') = g(v')$ for $v' \neq v$ and $g_m^v(v) = m$. We say that g_m^v is a *v-variant* of g .

Let $\mathfrak{M} = \langle M, R, V \rangle$ be a model, $m \in M$, and g an assignment. For any atom a , let $[V, g](a) = \{g(a)\}$ if a is a state variable, and $V(a)$ otherwise. Then:

$$\begin{aligned}
\mathfrak{M}, g, m \Vdash a & \quad \text{iff } m \in [V, g](a), a \in \text{ATOM} \\
\mathfrak{M}, g, m \Vdash \neg \varphi & \quad \text{iff } \mathfrak{M}, g, m \nVdash \varphi \\
\mathfrak{M}, g, m \Vdash \varphi \wedge \psi & \quad \text{iff } \mathfrak{M}, g, m \Vdash \varphi \text{ and } \mathfrak{M}, g, m \Vdash \psi \\
\mathfrak{M}, g, m \Vdash \Box \varphi & \quad \text{iff } \forall m' (Rmm' \Rightarrow \mathfrak{M}, g, m' \Vdash \varphi) \\
\mathfrak{M}, g, m \Vdash \downarrow v. \varphi & \quad \text{iff } \mathfrak{M}, g_m^v, m \Vdash \varphi \\
\mathfrak{M}, g, m \Vdash @_s \varphi & \quad \text{iff } \mathfrak{M}, g, m' \Vdash \varphi, \text{ where } [V, g](s) = \{m'\}, s \in \text{SSYM}.
\end{aligned}$$

We write $\mathfrak{M}, g \Vdash \varphi$ iff for all $m \in M$, $\mathfrak{M}, g, m \Vdash \varphi$, and $\mathfrak{M} \models \varphi$ iff for all g , $\mathfrak{M}, g \Vdash \varphi$.

Thus, as promised, $\downarrow$ enables us to bind a state variables to the current state. Note that, just as in first-order logic, if φ is a *sentence* it is irrelevant which assignment g is used to perform evaluation. Hence for sentences the relativization to assignments of the satisfaction relation can be dropped. A formula φ is *satisfiable* if there is a model $\mathfrak{M}$, an assignment g on $\mathfrak{M}$, and a state $m \in M$ such that $\mathfrak{M}, g, m \Vdash \varphi$.

We can now get down to business. First, we shall present a fragment of first-order logic (the *bounded fragment*) which is precisely as expressive as $\mathcal{H}(\downarrow, @)$ and provide explicit translations between these two languages. Secondly, we shall give an easy proof that (uni-modal) $\mathcal{H}(\downarrow, @)$ is undecidable. Third, we shall show how the dependency in this proof on $@$ and propositional variables can be systematically eliminated (in particular, we will show how to encode the valuation V so that the use of propositional variables can be simulated) and how we can encode any frame-condition expressible inside the pure fragment of $\mathcal{H}(\downarrow)$. This leads directly to the Spypoint Theorem and our undecidability result.

4.2 $\mathcal{H}(\downarrow, @)$ and the bounded fragment

We first relate $\mathcal{H}(\downarrow, @)$ to a certain bounded fragment of first-order logic. We shall work with a first-order language which contains a binary relation symbol R , a unary relation symbol P_j for each $p_j \in \text{PROP}$, and whose constants are the elements of NOM . Obviously any hybrid model $\mathfrak{M} = \langle M, R, V \rangle$ can be regarded as a first-order model for this language: the domain of the model is M , the accessibility relation R is used to interpret the binary predicate R , unary predicates are interpreted by the subsets that V assigns to propositional variables, and constants are interpreted by the states that nominals name. Conversely, any model for our first-order language can be regarded as a hybrid model. So we shall let context determine whether we are referring to first-order or hybrid models, and continue to use the notation $\mathfrak{M} = \langle M, R, V \rangle$ for models.

First the easy part: we extend the well-known *standard translation* ST of modal correspondence theory (see [Ben83b]) to $\mathcal{H}(\downarrow, @)$. We assume that the first-order variables are $\text{SVAR} \cup \{x, y\}$ (where x and y are distinct new variables) and define the required translation by mutual recursion between two functions ST_x and ST_y . Here $\varphi[x/y]$ means “replace all free instances of x by y in φ .”

$$\begin{array}{ll|ll}
ST_x(p_j) & = P_j(x), p_j \in \text{PROP}. & ST_y(p_j) & = P_j(y), p_j \in \text{PROP}. \\
ST_x(i_j) & = x = i_j, i_j \in \text{NOM}. & ST_y(i_j) & = y = i_j, i_j \in \text{NOM}. \\
ST_x(x_j) & = x = x_j, x_j \in \text{SVAR}. & ST_y(x_j) & = y = x_j, x_j \in \text{SVAR}. \\
ST_x(\neg \varphi) & = \neg ST_x(\varphi). & ST_y(\neg \varphi) & = \neg ST_y(\varphi). \\
ST_x(\varphi \wedge \psi) & = ST_x(\varphi) \wedge ST_x(\psi). & ST_y(\varphi \wedge \psi) & = ST_y(\varphi) \wedge ST_y(\psi). \\
ST_x(\Diamond \varphi) & = \exists y. (Rxy \wedge ST_y(\varphi)). & ST_y(\Diamond \varphi) & = \exists x. (Ryx \wedge ST_x(\varphi)). \\
ST_x(\downarrow x_j. \varphi) & = (ST_x(\varphi))[x_j/x]. & ST_y(\downarrow x_j. \varphi) & = (ST_y(\varphi))[x_j/y]. \\
ST_x(@_s \varphi) & = (ST_x(\varphi))[x/s]. & ST_y(@_s \varphi) & = (ST_y(\varphi))[y/s].
\end{array}$$

Proposition 1. *Let φ be a hybrid formula, then for all hybrid models $\mathfrak{M}$, $m \in M$ and assignments g , $\mathfrak{M}, g, m \Vdash \varphi$ iff $\mathfrak{M} \models ST_x(\varphi)[g_m^x]$.*

Proof. Induction on the structure of φ .

Now for the interesting question: what is the *range* of ST ? In fact it belongs to a *bounded* fragment of our first-order language. This fragment consists of the formulas generated as follows:

$$\varphi := Rtt' \mid P_j t \mid t = t' \mid \neg \varphi \mid \varphi \wedge \varphi' \mid \exists x_i. (Rtx_i \wedge \varphi) \text{ (for } x_i \neq t \text{)}.$$

where x_i is a variable and t, t' are either variables or constants.

Clearly ST generates formulas in the bounded fragment. Crucially, however, we can also translate any formula in the bounded fragment into $\mathcal{H}(\downarrow, @)$ as follows:

$$\begin{aligned}
HT(Rtt') & = @_t \Diamond t'. \\
HT(P_j t) & = @_t P_j. \\
HT(t = t') & = @_t t'. \\
HT(\neg \varphi) & = \neg HT(\varphi). \\
HT(\varphi \wedge \psi) & = HT(\varphi) \wedge HT(\psi). \\
HT(\exists v. (Rtv \wedge \varphi)) & = @_t \Diamond \downarrow v. HT(\varphi).
\end{aligned}$$

By construction, $HT(\varphi)$ is a hybrid formula, but furthermore it is a boolean combination of $@$ -formulas (formulas whose main operator is $@$). We can now prove the following strong truth preservation result.

Proposition 2. *Let φ be a bounded formula. Then for every first-order model $\mathfrak{M}$ and for every assignment g , $\mathfrak{M} \models \varphi[g]$ iff $\mathfrak{M}, g \Vdash HT(\varphi)$.*

Proof. Induction on the structure of φ .

To summarize, there are effective translations between $\mathcal{H}(\downarrow, @)$ and the bounded fragment.

4.3 Undecidability of $\mathcal{H}(\downarrow, @)$

We are now ready to discuss undecidability. The result we want to prove is this:

The fragment of $\mathcal{H}(\downarrow)$ consisting of pure nominal-free sentences has an undecidable satisfaction problem.

However we begin by quickly sketching an easy undecidability proof for the full language $\mathcal{H}(\downarrow, @)$. The proof uses the spypoint technique from the previous section together with results from [Spa93a]. By generalizing the methods used in this simple proof, we will be lead to the Spypoint Theorem and the undecidability result just stated.

Hemaspaandra shows in [Spa93a] that the global satisfaction problem of the uni-modal logic of the class K_{23} of frames is undecidable. K_{23} consists of all modal frames $\langle W, R \rangle$ in which every state has at most 2 R -successors and at most 3 two-step R -successors. We will show that we can reduce the satisfiability problem of this logic to $\mathcal{H}(\downarrow, @)$.

Let *Grid* be the conjunction of the following formulas:

$$\begin{aligned} G_1 & @_s \neg \Diamond s \\ G_2 & @_s \Diamond \top \\ G_3 & @_s (\Box \Box \downarrow x. @_s \Diamond x) \\ G_4 & @_s (\Box \downarrow y. \Box \downarrow x_1. @_y \Box \downarrow x_2. @_y \Box \downarrow x_3. (@_{x_1} x_2 \vee @_{x_1} x_3 \vee @_{x_2} x_3)) \\ G_5 & @_s (\Box \downarrow y. \Box \Box \downarrow x_1. @_y \Box \Box \downarrow x_2. @_y \Box \Box \downarrow x_3. @_y \Box \Box \downarrow x_4. (\bigvee_{1 \leq i \neq j \leq 4} @_{x_i} x_j)). \end{aligned}$$

What does *Grid* express? Suppose it is satisfied in a model $\mathfrak{M}$ on a frame $\langle W, R \rangle$. Then there exists a state which is named by s (the spypoint). By G_1 , s is not related to itself. By G_2 , s is related to some state, and by G_3 , every state which can be reached from s in two steps can also be reached from s in one step. This means that in $\mathfrak{M}_s$ —the submodel of $\mathfrak{M}$ generated by s — every state is reachable from s in one step. Now G_4 and G_5 express precisely the two conditions characterizing the class K_{23} on successors of s . Instead of spelling out this proof we show that the similar formula $@_s \Box \downarrow y. \Box \downarrow x_1. @_y \Box \downarrow x_2. @_{x_1} x_2$ expresses that every successor of s in $\mathfrak{M}_s$ has at most one R -successor. As G_4 and G_5 follow the same pattern, it is easy to extend the argument below to verify their meaning.

$$\begin{aligned} & \mathfrak{M}, g, s \models \Box \downarrow y. \Box \downarrow x_1. @_y \Box \downarrow x_2. @_{x_1} x_2 \\ \iff & (\forall w : sRw) : \mathfrak{M}, g_w^y, w \models \Box \downarrow x_1. @_y \Box \downarrow x_2. @_{x_1} x_2 \\ \iff & (\forall u : wRu) : \mathfrak{M}, (g_w^y)_{u_1}^{x_1}, u \models @_y \Box \downarrow x_2. @_{x_1} x_2 \\ \iff & \mathfrak{M}, (g_w^y)_{u_1}^{x_1}, w \models \Box \downarrow x_2. @_{x_1} x_2 \\ \iff & (\forall v : wRv) : \mathfrak{M}, ((g_w^y)_{u_1}^{x_1})_v^{x_2}, v \models @_{x_1} x_2 \\ \iff & (\forall w : sRw)(\forall u : wRu)(\forall v : wRv) : u = v. \end{aligned}$$

Now we are ready to complete the proof. We claim that for every formula φ ,

φ is globally satisfiable on a K_{23} -frame iff $Grid \wedge @_s \Box \varphi$ is satisfiable.

The proof of the claim is a simple copy of the two constructions given in the proof of Theorem 2.

4.4 Undecidability of pure nominal-free sentences of $\mathcal{H}(\downarrow)$

We are ready to prove our main result. We do so by analysing the previous proofs and generalizing the underlying ideas. The models used in the proof of Theorem 2 and the undecidability proof just given both had a certain characteristic form. Let's pin this down:

Definition 5. A model $\mathfrak{M} = \langle W, R, V \rangle$ is called a *spypoint model* if there is an element $s \in W$ (the spypoint) such that

- i. $\neg sRs$;
- ii. For all $w \in W$, if $w \neq s$, then sRw and wRs .

Notice that by ii above, any spypoint model is generated by its spy point. We will now show that with $\downarrow$ we can easily create spypoint models. On these models we can create for every variable x introduced by $\downarrow x$, a formula which has precisely the meaning of $@_x$.

Proposition 3. Let $\mathfrak{M} = \langle M, R, V \rangle$ and $s \in M$ be such that $\mathfrak{M}, s \models \downarrow s. (\neg \Diamond s \wedge \Box \Box \downarrow x. \Diamond (s \wedge \Diamond x) \wedge \Box \Diamond s)$. Then,

- i. $\mathfrak{M}_s$, the submodel of $\mathfrak{M}$ generated by s , is a spypoint model with s the spypoint.
- ii. $@_s \varphi$ is definable on $\mathfrak{M}_s$ by $(s \wedge \varphi) \vee \Diamond (s \wedge \varphi)$.
- iii. Let g be any assignment. Then for all $u \in M$, $\mathfrak{M}_s, g, u \models @_x \varphi$ iff $\mathfrak{M}_s, g, u \models @_s (\varphi \vee \Diamond (x \wedge \varphi))$.

Proof. i is immediate. ii and iii follow from the properties of a spypoint model.

Now, spypoint models are very powerful: we can encode lots of information about Kripke models (for finitely many propositional variables) inside a spypoint model. More precisely, for each Kripke model $\mathfrak{M}$, we define the notion of a *spypoint model of $\mathfrak{M}$* .

Definition 6. Let $\mathfrak{M} = \langle M, R, V \rangle$ be a Kripke model in which the domain of V is a finite set $\{p_1, \dots, p_n\}$ of propositional variables. The spypoint model of $\mathfrak{M}$ (notation $\text{Spy}[\mathfrak{M}]$) is the structure $\langle M', R', V' \rangle$ in which

- i. $M' = M \cup \{s\} \cup \{w_{p_1}, \dots, w_{p_n}\}$, for $s, w_{p_1}, \dots, w_{p_n} \notin M$
- ii. $R' = R \cup \{(s, x), (x, s) \mid x \in M' \setminus \{s\}\} \cup \{(x, w_{p_i}) \mid x \in M \text{ and } x \in V(p_i)\}$
- iii. $V' = \emptyset$.

Let $\{s, x_{p_1}, \dots, x_{p_n}\}$ be a set of state variables. A *spypoint assignment* for this set is an assignment g which sends s to the spypoint s and x_{p_i} to w_{p_i} . We use $\mathbf{m}$ as an abbreviation for $\neg s \wedge \neg x_{p_1} \wedge \dots \wedge \neg x_{p_n}$. Note that when evaluated under the spypoint assignment, the denotation of $\mathbf{m}$ in $\text{Spy}[\mathfrak{M}]$ is precisely M .

$\text{Spy}[\mathfrak{M}]$ encodes the valuation on $\mathfrak{M}$ and we can take advantage of this fact. Define the following translation from uni-modal formulas to hybrid formulas:

$$\begin{aligned} IT(p_i) &= \Diamond(x_{p_i}) \\ IT(\neg \varphi) &= \neg IT(\varphi) \\ IT(\varphi \wedge \psi) &= IT(\varphi) \wedge IT(\psi) \\ IT(\Diamond \varphi) &= \Diamond(\mathbf{m} \wedge IT(\varphi)). \end{aligned}$$

Proposition 4. Let $\mathfrak{M}$ be a Kripke model and φ a uni-modal formula. Then for any spypoint assignment g ,

$$\mathfrak{M} \models \varphi \text{ if and only if } \text{Spy}[\mathfrak{M}], g, s \models \Box(\mathbf{m} \rightarrow IT(\varphi)).$$

Proof. Immediate by the fact that the spypoint is R -related to all states in the domain of $\mathfrak{M}$, and the interpretation of $\mathbf{m}$ under any spypoint assignment g .

We modify the hybrid translation HT to its relativized version $HT^{\mathbf{m}}$ which also defines away occurrences of $@$. Define $HT^{\mathbf{m}}(\exists v.(Rtv \wedge \varphi))$ as $@_t \diamond \downarrow v.(\mathbf{m} \wedge HT^{\mathbf{m}}\varphi)$ and replace all $@$ symbols by their definition as indicated in Proposition 3.ii and 3.iii.

The crucial step is now the fact that $\downarrow$ is strong enough to encode many frame-conditions.

Proposition 5. *Let $\mathfrak{M} = \langle M, R, V \rangle$ be a Kripke model. Let $C(y)$ be a formula in the bounded fragment in the signature $\{R, =\}$. Then for any assignment g ,*

$$\langle M, R \rangle \models \forall y.C(y) \text{ if and only if } \text{Spy}[\mathfrak{M}], g, s \Vdash \Box \downarrow y.(\mathbf{m} \rightarrow HT^{\mathbf{m}}(C(y))).$$

Proof. Immediate by the properties of HT , Proposition 3, and the fact that the spypoint is R -related to all states in the domain of $\mathfrak{M}$.

Theorem 3 (Spypoint theorem). *Let φ be a uni-modal formula in $\{p_1, \dots, p_n\}$ and $\forall y.C(y)$ a first-order frame condition in $\{R, =\}$ with $C(y)$ in the bounded fragment. The following are equivalent.*

i. *There exists a Kripke model $\mathfrak{M} = \langle M, R, V \rangle$ such that $\langle M, R \rangle \models \forall y.C(y)$ and $\mathfrak{M} \models \varphi$.*

ii. *The pure hybrid sentence F in the language $\mathcal{H}(\downarrow)$ is satisfiable. F is*

$$\downarrow s.(SPY \wedge \diamond \downarrow x_{p_1}.@_s \diamond \downarrow x_{p_2}.@_s \dots \diamond \downarrow x_{p_n}.@_s(DIS \wedge VAL \wedge FR)),$$

where

$$\begin{aligned} SPY &= \neg \diamond s \wedge \Box \Box \downarrow x. \diamond (s \wedge \diamond x) \wedge \Box \diamond s \\ DIS &= \Box (\bigwedge_{1 \leq i \leq n} (x_{p_i} \rightarrow \bigwedge \{\neg x_{p_j} \mid 1 \leq j \neq i \leq n\})) \\ VAL &= \Box (\mathbf{m} \rightarrow IT(\varphi)) \\ FR &= \Box \downarrow y.(\mathbf{m} \rightarrow HT^{\mathbf{m}}(C(y))). \end{aligned}$$

Proof. The way we have written it, F contains occurrences of $@_s$; but this does not matter, by Proposition 3 all these occurrences can be term-defined. So let's check that F works as claimed.

For the implication from i to ii, let $\mathfrak{M}$ be a Kripke model as in i. We claim that $\text{Spy}[\mathfrak{M}], s \Vdash F$. The first conjunct of F is true in $\text{Spy}[\mathfrak{M}]$ at s by Proposition 3. The diamond part of the second disjunct can be satisfied using any spypoint assignment g . In the spypoint model all w_{p_i} are pairwise disjoint, whence $\text{Spy}[\mathfrak{M}], g, s \Vdash DIS$. By Propositions 4 and 5, also $\text{Spy}[\mathfrak{M}], g, s \Vdash VAL \wedge FR$.

For the other direction, let $\mathfrak{M}, s \Vdash F$. By Proposition 3, the submodel $\mathfrak{M}_s = \langle M_s, R_s, V_s \rangle$ generated by s is a spypoint model. Let g be the assignment such that $\mathfrak{M}, g, s \Vdash DIS \wedge VAL \wedge FR$. By DIS , $g(x_{p_i}) \neq g(x_{p_j})$ for all $i \neq j$, and (since $\neg sRs$) also $g(x_{p_i}) \neq s$, for all i . Define the following Kripke model $\mathfrak{M}' = \langle M', R', V' \rangle$, where

$$\begin{aligned} M' &= M \setminus \{g(s), g(x_{p_1}), \dots, g(x_{p_n})\} \\ R' &= R \upharpoonright_{M'} \\ V'(p_i) &= \{w \mid wRg(x_{p_i})\}. \end{aligned}$$

Note that $\text{Spy}[\mathfrak{M}']$ is precisely $\mathfrak{M}_s$, and g is a spypoint assignment. But then by Propositions 4 and 5 and the fact that $\mathfrak{M}_s, g, s \Vdash VAL \wedge FR$, we obtain $\mathfrak{M}' \models \varphi$ and $\langle M', R' \rangle \models \forall y.C(y)$.

The proof of the claimed undecidability result is now straightforward.

Corollary 1. *The fragment of $\mathcal{H}(\downarrow)$ consisting of all pure nominal-free sentences has an undecidable satisfaction problem.*

Proof. We will reduce the undecidable global satisfaction problem in the uni-modal language over the class K_{23} , just as we did in our easy undecidability result for $\mathcal{H}(\downarrow, @)$. The first-order frame conditions defining K_{23} are of the form $\forall y.C(y)$ with $C(y)$ in the bounded fragment. (This is easy to check. For instance, y has at most two successors can be written as $\forall x_1.(yRx_1 \rightarrow \forall x_2.(\rightarrow \forall x_3. \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3)))$.) Now apply the Spypoint Theorem. The formula F (after all occurrences of $@_s$ have been term-defined) is a pure nominal-free sentence of $\mathcal{H}(\downarrow)$, and the result follows.

Because of the generality of the Spypoint Theorem, it seems unlikely that even restricted forms of label binding will lead to decidable systems. For this reason, much of our ongoing research is focusing on binder free systems, such as *Until*-based languages enriched with nominals and $@$, and modal languages with counting modalities (these are widely used in description logic) enriched in the same way.

5 Concluding remarks

In this paper we have examined the complexity of a number of hybrid languages. Our results have been both positive and negative and we sum them up here:

1. Adding nominals and $@$ to the uni-modal language, or even the multi-modal language, does not lead to an increase in complexity: K-satisfiability remains PSPACE-complete.
2. On the other hand, adding even one nominal to the language of tense logic takes the complexity from PSPACE-complete to EXPTIME-complete.
3. We provide a simple proof of the known fact that $\mathcal{H}(\downarrow, @)$ is undecidable. Furthermore, we prove that very restricted use of $\downarrow$ leads already to undecidability. In fact, undecidability strikes even in the sentential fragment of the uni-modal language without $@$ or propositional variables.

Furthermore, a simple extension of the undecidability proof provided in this paper shows that this last fragment is even a conservative reduction class in the sense of [BGG97].

Needless to say, the results we presented conform just a preliminary sketch of the complexity-theoretic territory occupied by hybrid languages. The spectrum of plausible directions for further work is huge. As an example, we have only considered logics with full Boolean expressive power. In the description logic community fragments which restrict negation or disallow disjunctions (aiming to obtain good computational behavior) are standard. Again, the generality of the Spypoint Theorem will be of much help in mapping this new variations.

References

- [AdR99] C. Areces and M. de Rijke. Accounting for assertional information. Manuscript, 1999.
- [All84] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, July 1984.
- [Ben83a] J. van Benthem. *The logic of time*. Reidel, 1983.
- [Ben83b] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.
- [BG97] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer Verlag, 1997.
- [Bla98] P. Blackburn. Internalizing labeled deduction. Technical Report CLAUS-Report 102, Computerlinguistik, Universität des Saarlandes, 1998. <http://www.coli.uni-sb.de/cl/claus>.
- [BS95] P. Blackburn and J. Seligman. Hybrid languages. *J. of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.
- [BS98] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyashev, editors, *Advances in Modal Logic*, volume 1, pages 41–62. CSLI Publications, Stanford University, 1998.
- [BT99] P. Blackburn and M. Tzakova. Hybrid languages and temporal logics. *Logic J. of the IGPL*, 7(1):27–54, 1999.
- [Bul70] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.
- [Chl86] B. Chlebus. Domino-tiling games. *J. Comput. System Sci.*, 32(3):374–392, 1986.
- [Gab96] D. Gabbay. *Labeled deductive systems*. Oxford University Press, 1996.
- [Gor96] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *J. Logic Lang. Inform.*, 5(1):1–24, 1996.
- [Grä97] E. Grädel. On the restraining power of guards. To appear in *J. of Symbolic Logic*, 1997.
- [HM92] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
- [Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. of Computing*, 6(3):467–480, 1977.
- [PT85] S. Passy and T. Tinchev. Quantifiers in combinatory PDL: completeness, definability, incompleteness. In *Fundamentals of computation theory (Cottbus, 1985)*, pages 512–519. Springer, Berlin, 1985.
- [Sch91] K. Schild. A correspondence theory for terminological logics. In *Proc. of the 12th IJCAI*, pages 466–471, 1991.
- [Sch94] A. Schaerf. *Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues*. PhD thesis, Dipartimento di Informatica e Sistemistica, Univ. di Roma “La Sapienza”, 1994.
- [Sel97] J. Seligman. The logic of correct description. In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.
- [Spa93a] E. Spaan. *Complexity of modal logics*. PhD thesis, ILLC. University of Amsterdam, 1993.
- [Spa93b] E. Spaan. The complexity of propositional tense logics. In *Diamonds and defaults (Amsterdam, 1990/1991)*, pages 287–307. Kluwer Acad. Publ., Dordrecht, 1993.
- [SSS91] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.
- [Tza98] M. Tzakova. Tableaux calculi for hybrid logics. Manuscript, 1998.

The Computational Complexity of Hybrid Temporal Logics

CARLOS ARECES, *ILLC, Universiteit van Amsterdam, Plantage Muidergracht 24, 1018TV, Amsterdam, The Netherlands.*
E-mail: carlos@wins.uva.nl

PATRICK BLACKBURN, *Computerlinguistik, Universität des Saarlandes, 66041, Saarbrücken, Germany.*
E-mail: patrick@coli.uni-sb.de

MAARTEN MARX, *CCSOM, ILLC, Universiteit van Amsterdam, Plantage Muidergracht 24, 1018TV, Amsterdam, The Netherlands.*
E-mail: marx@wins.uva.nl

Abstract

In their simplest form, hybrid languages are propositional modal languages which can refer to states. They were introduced by Arthur Prior, the inventor of tense logic, and played an important role in his work: because they make reference to specific times possible, they remove the most serious obstacle to developing modal approaches to temporal representation and reasoning. However very little is known about the computational complexity of hybrid temporal logics.

In this paper we analyze the complexity of the satisfiability problem of a number of hybrid temporal logics: the *basic hybrid language* over transitive frames; *nominal tense logic* over transitive frames, strict total orders, and transitive trees; *nominal Until logic*; and *referential interval logic*. We discuss the effects of including nominals, the @ operator, the somewhere modality E, and the difference operator D. Adding nominals to tense logic leads for several frame-classes to an increase in complexity of the satisfiability problem from PSPACE to EXPTIME. On transitive trees, however, the satisfiability problem for this language can be decided in PSPACE. Along the way we make a detour through *hybrid* propositional dynamic logic: we establish upper bounds for a number of temporal logics by generalizing results due to Passy and Tinchev [PT91] and De Giacomo [De 95]. We conclude with some remarks on the relevance of our results to description logic, and draw attention to the utility of the *spypoint technique* for proving upper and lower bounds.

Keywords: Hybrid Logic, Computational Complexity, Modal and Temporal Logic, Propositional Dynamic Logic, Description Logic, Nominals, At Operator, Difference Operator, Universal Modality

1 Introduction

The simplest hybrid language is a uni-modal language that uses special atomic formulas (called *nominals*) to name states. Nominals are true at exactly one state in any model; they “name” this state by being true there and nowhere else. A wide range of hybrid languages have been studied, including hybrid languages in which it is possible to bind nominals in various ways, but although a number of fundamental complexity and undecidability results are known (see [ABM99]), the complexity of hybrid formalisms based on richer modal logics have rarely been studied. Two

2 The Computational Complexity of Hybrid Temporal Logics

important exceptions to this are the EXPTIME-completeness result for Propositional Dynamic Logic (PDL) enriched with nominals and the somewhere modality E (a Sofia school result from the 1980s; see [PT91]), and the EXPTIME-completeness result for PDL with converse enriched with nominals (but without E) due to De Giacomo (see [De 95]). Moreover, most known results deal with the satisfiability problem over *arbitrary* frames; the only exception seems to be the results for deterministic frames proved in [BS93]. This is a real gap, as in practice we are often interested in richer modal languages and frames with certain properties. In particular, in temporal logic, we almost always work with transitive frames, and it is common to demand that temporal frames be treelike or even linear. This paper provides complexity results for various hybrid temporal logics over such structures.

Before going further, let’s be precise about the syntax and semantics of the *basic hybrid language*, the weakest language we shall consider in the paper:

DEFINITION 1.1 (Syntax)

Let $\text{PROP} = \{p, q, r, \dots\}$ be a countable set of *propositional variables* and $\text{NOM} = \{i, j, k, \dots\}$ a countable set of *nominals*, disjoint from PROP . We call $\text{ATOM} = \text{PROP} \cup \text{NOM}$ the set of *atoms*. The *well-formed formulas* of the basic hybrid language (over ATOM) are

$$\varphi := a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Diamond\varphi \mid @_i\varphi$$

where $a \in \text{ATOM}$, and $i \in \text{NOM}$. As usual, $\Box\varphi := \neg\Diamond\neg\varphi$.

Thus, syntactically speaking, the basic hybrid language is a two-sorted extension of uni-modal logic which contains a NOM indexed collection of operators $@_i$. Now for the semantics. Ordinary (unsorted) modal languages are interpreted on Kripke models, but hybrid languages are interpreted on *hybrid models*. These are Kripke models in which the valuations are constrained to ensure that each nominal names a unique state.

DEFINITION 1.2 (Semantics)

A *hybrid model* is a triple $\mathfrak{M} = \langle M, R, V \rangle$ such that M is a non-empty set, R is a binary relation on M , and $V : \text{ATOM} \rightarrow \text{Pow}(M)$ is such that for all $i \in \text{NOM}$, $V(i)$ is a singleton subset of M . The elements of M are called *states*, R is the *transition relation*, and V is the *valuation*. A *frame* is a pair $\mathfrak{F} = \langle M, R \rangle$, that is, a model without a valuation.

Let $\mathfrak{M} = \langle M, R, V \rangle$ be a model and $m \in M$. Then the *satisfaction relation* is defined by:

$$\begin{aligned} \mathfrak{M}, m \Vdash a & \quad \text{iff} \quad m \in V(a), a \in \text{ATOM} \\ \mathfrak{M}, m \Vdash \neg\varphi & \quad \text{iff} \quad \mathfrak{M}, m \nVdash \varphi \\ \mathfrak{M}, m \Vdash \varphi \wedge \psi & \quad \text{iff} \quad \mathfrak{M}, m \Vdash \varphi \text{ and } \mathfrak{M}, m \Vdash \psi \\ \mathfrak{M}, m \Vdash \Diamond\varphi & \quad \text{iff} \quad \exists m' \in M (Rmm' \ \& \ \mathfrak{M}, m' \Vdash \varphi) \\ \mathfrak{M}, m \Vdash @_i\varphi & \quad \text{iff} \quad \mathfrak{M}, m' \Vdash \varphi, \text{ where } V(i) = \{m'\}. \end{aligned}$$

A formula φ is *satisfiable* if there is a hybrid model $\mathfrak{M}$, and a state $m \in M$ such that $\mathfrak{M}, m \Vdash \varphi$. We write $\mathfrak{M} \Vdash \varphi$ if for all $m \in M$, $\mathfrak{M}, m \Vdash \varphi$, and say that φ is *globally satisfied* in $\mathfrak{M}$. If $\mathfrak{F}$ is a frame, and for all hybrid valuations V on $\mathfrak{F}$ we have $\langle \mathfrak{F}, V \rangle \Vdash \varphi$, we say that φ is *valid* on $\mathfrak{F}$ and write $\mathfrak{F} \Vdash \varphi$.

Because hybrid valuations assign *singletons* to nominals, each nominal is satisfied at exactly one state in any model; this means that nominals are essentially a mechanism

for referring to states in a propositional modal language. In what follows we often call the unique state in $V(i)$ the *denotation* of i . Note that the satisfaction clause for formulas of the form $@_i\varphi$ says: to evaluate $@_i\varphi$, jump to the unique state named by i and evaluate φ there. It follows that a wff of the form $@_ij$ expresses the identity of the states named by i and j , and that a wff of the form $@_i\Diamond j$ says that the state named by i *R*-precedes the state named by j .

So far we have said nothing about why we are interested in hybrid languages in the first place. There are at least two good reasons. First, they are proof-theoretically well behaved. Second, hybrid languages arise naturally in many applications.

Hybrid Languages and Modal Proof Theory Probably the most straightforward way to see why hybrid languages are proof-theoretically natural is to observe that the basic hybrid language offers precisely what is needed to capture the main ideas of labelled deduction. Labelled deduction (see [Gab96]) is built around the notation $l:\varphi$. Here the metalinguistic symbol $:$ associates the metalinguistic label l with the object language formula φ . This has a natural modal interpretation: regard labels as names for states and read $l:\varphi$ as asserting that φ is satisfied at l . Labelled deduction proceeds by manipulating such labels to guide proof search, and is today an important approach to modal proof theory (see, for example, the work of Basin, Matthews and Viganò [BMV97]).

The basic hybrid language places the apparatus of labelled deduction in the *object* language: nominals are essentially object-level labels, and the formula $@_i\varphi$ asserts in the object language what $i:\varphi$ asserts in the metalanguage. And indeed, the basic hybrid language turns out to be proof-theoretically natural, for we can directly “internalize” standard labelled deduction (see [Sel97, Bla98]). Nonetheless, though important, $@$ is not indispensable. For example, [Tza99] presents a general approach to hybrid proof theory using Fitting-style prefix calculi. And when the underlying modal logic is tense logic (that is, when we have a backward looking modality to match the forward looking one) even more flexibility is possible: [Dem99] presents a sequent system for nominal tense logic *without* $@$ that has much in common with $@$ -based internalized labelled deductive systems, and [DG99] discusses display calculi for nominal tense logic with the difference operator D . For this reason, when we discuss nominal tense logic, we will be interested in the complexity of $@$ -free systems, system containing $@$, and even richer systems containing the universal modality E or the difference modality D .

Hybrid Languages and Applied Logic Modal logicians like to point out that many of the formalisms invented by researchers in artificial intelligence, computational linguistics, and other fields are actually notational variants of modal logic. But in many cases it would be more accurate to say that it is *hybrid languages* which are reinvented in this way. For example, it is well known that the description language *ALC* (see [SSS91]) is a notational variant of multi-modal logic (see [Sch91]). But this relation only holds at the level of T-Box reasoning (that is, general reasoning about concepts). It is often important to perform A-Box (or assertional) reasoning as well — that is, to reason about how concepts apply to particular individuals. A-Box reasoning corresponds to a restricted use of nominals and the $@_i$ operators (see [BT98, AdR99]). In a similar vein, while the basic Attribute Value Matrices (AVMs) used in computational linguistics are an obvious notational variant of deterministic multi-

modal logic, full AVM notation (which provides a special device for specifying that two sequences of features lead to exactly the same node) corresponds to deterministic multi-modal logic enriched with nominals (see [BS93] and [Rea94]).

But in fact hybrid languages were introduced in the mid-1960s: they were developed by Arthur Prior, the founder of tense logic, and played a crucial role in his modal analysis of temporal representation and reasoning (see [Pri67], especially Chapter 5 and Appendix B3). Since then, many different hybrid temporal logics have been explored for different purposes (see [Bul70, Has91, Bla94, Gor96, BT99]). Among other things, hybrid languages make it possible to introduce specific times (days, dates, years, etc.), to cope with temporal indexicals (such as yesterday, today, tomorrow and now) and to define many temporally relevant frame properties (such as irreflexivity, asymmetry, trichotomy, and directedness) that ordinary modal languages cannot. Furthermore, as we discuss in Section 5, when nominals and $@$ are added to interval-based logic, the result is a $\text{Holds}(t, \varphi)$ -driven interval logic in the style of those introduced into AI by James Allen, with $@$ playing the role of Holds . In a nutshell, because they make temporal reference possible, nominals (aided, perhaps, by $@$, E or D) remove the most serious obstacle to a modal analysis of temporal representation and reasoning.

Unfortunately, very little is known about the complexity of hybrid logics, and in particular of hybrid temporal logics. This paper is a first attempt to fill the gap. We will concentrate on complexity results for nominal tense logic, presenting a series of results which show that the EXPTIME-completeness of nominal tense logic reported in [ABM99] can be tamed by either choosing a more appropriate class of frames, or by working with a fragment of the language.

Overview of the paper. In the next section we discuss the relations between nominals, the universal modality and the difference operator. Section 3 contains our main results. We first show that the satisfiability problem for the basic hybrid language is complete for PSPACE when interpreted on transitive frames. Then we show that adding just one nominal to the language of tense logic leads to EXPTIME hardness on this class of frames. We then consider frame classes which are more appropriate to temporal logic: linear structures and transitive trees. In both cases we establish that the complexity of the satisfiability problem remains the same after expansion with nominals and the $@$ operator. The last two sections discuss two stronger languages: nominal Until/Since logic, and a referential version of Halpern and Shoham’s interval logic. Along the way we make a brief detour through hybrid PDL: we establish upper bounds for a number of temporal logics by generalizing results due to Passy and Tinchev [PT91] and De Giacomo [De 95]. We conclude with some remarks on the relevance of our results to description logic, and draw attention to the *spypoint technique*, which is used in this paper to prove both upper and lower bounds. We believe this technique may be useful in other settings.

2 Nominals, $@$, E and D

The somewhere modality E , and to a lesser extent the difference operator D , have played an important role in the development of hybrid languages. These operators can be added to any (modal or hybrid) language, and on any (Kripke or hybrid)

model $\mathfrak{M} = \langle M, R, V \rangle$ they are interpreted as follows:

$$\begin{aligned} \mathfrak{M}, m \Vdash E\varphi & \quad \text{iff} \quad \exists m' \in M(\mathfrak{M}, m' \Vdash \varphi) \\ \mathfrak{M}, m \Vdash D\varphi & \quad \text{iff} \quad \exists m' \in M(m' \neq m \wedge \mathfrak{M}, m' \Vdash \varphi). \end{aligned}$$

We will see a lot more of these operators in what follows, so let's note some useful facts right away. First, the dual of E, namely $A\varphi := \neg E\neg\varphi$, asserts that φ is true at *all* points in a model; A is called the *universal modality*. Second, note that E can mimic @, for $E(i \wedge \varphi)$ means exactly the same thing as $@_i\varphi$; this is why practically everyone who has worked with nominals has also experimented with E. In a sense, @ amounts to *guarded* use of E, where “guarded” is used in the sense of [AvBN98]. As we shall see, this kind of guarding can be effective: hybrid logics with @ are often less complex than those which allow unrestricted use of E.

The difference operator D is stronger than E, for we can define $E\varphi$ as $\varphi \vee D\varphi$ but E is not strong enough to define D. Actually, D is so strong that it can even simulate nominals: clearly the formula $Ep \wedge A(p \rightarrow \neg Dp)$ forces p to be true at only one point in the model.¹ More interesting is the following fact:

THEOREM 2.1

There is a polynomial reduction which preserves satisfaction from any hybrid language containing D to the fragment containing only E and nominals.

This result is relevant to our work on hybrid complexity, for it immediately yields:

COROLLARY 2.2

Let F be any class of Kripke frames, and let $\mathcal{L}$ be a modal language. The F-satisfiability problem is the problem of satisfying a formula on a model based on a frame in F. Then:

1. if $\mathcal{L}$ contains D then adding nominals, @, and E does not modify (except by a polynomial) the complexity of the F-satisfiability problem,
2. if $\mathcal{L}$ contains nominals and E then adding D does not modify (except by a polynomial) the complexity of the F-satisfiability problem.

For this reason we won't explicitly mention results concerning D. We now prove the theorem.

PROOF. Let φ be a formula in the full language. We construct in two steps a formula $\varphi' \wedge \theta'$ without D such that φ is satisfiable iff $\varphi' \wedge \theta'$ is. We take care that this construction can be performed in time polynomial in the length of φ . We use the fact that in any Kripke model $\mathfrak{M}$, the denotation of $D\varphi$ (denoted by $[D\varphi]_{\mathfrak{M}} = \{m \in M \mid \mathfrak{M}, m \Vdash D\varphi\}$) can only take three values, namely:

$$[D\varphi]_{\mathfrak{M}} = \begin{cases} M & \text{if } |[D\varphi]_{\mathfrak{M}}| > 1 \\ \emptyset & \text{if } [D\varphi]_{\mathfrak{M}} = \emptyset \\ M \setminus \{m\} & \text{if } [D\varphi]_{\mathfrak{M}} = \{m\}. \end{cases}$$

We now delete all occurrences of D, replacing them with nominals and A; we proceed inductively in the number of D operators in φ . If φ contains no D we are done. Otherwise, consider a subformula of the form $D\psi$ where ψ contains no occurrences

¹This is not a new observation. For example, proof systems for D-logics based on rules for the undefinable trade on this; see [Rij92, Ven93].

6 The Computational Complexity of Hybrid Temporal Logics

of D. Let φ' be φ with this subformula replaced by a new variable p_k and let $\theta_k = A(p_k \leftrightarrow D\psi)$. Clearly φ is satisfiable iff $\varphi' \wedge \theta_k$ is. An inductive application of this procedure eventually yields a formula φ' without occurrences of D and a conjunction θ of formulas of the form $A(p_k \leftrightarrow D\psi)$ with ψ not containing D.

Now we “axiomatize” all the $D\psi$ using nominals, E, and A. For every conjunct $\theta_k = A(p_k \leftrightarrow D\psi)$, we create a formula θ'_k , the conjunction of:

$$\begin{aligned} & Ap_k \vee A\neg p_k \vee (A(p_k \leftrightarrow \neg i_k) \wedge Ep_k) \\ & Ap_k \rightarrow E(\psi \wedge i_k) \wedge E(\psi \wedge \neg i_k) \\ & A\neg p_k \rightarrow A\neg\psi \\ & (A(p_k \leftrightarrow \neg i_k) \wedge Ep_k) \rightarrow A(\psi \rightarrow i_k). \end{aligned}$$

For every conjunct θ_k , we use a new nominal i_k . Note that the three disjuncts making up the first conjunct are mutually exclusive: each reflects one of the three possible denotations of $D\psi$. Let θ' be the conjunction of all θ'_k . It is clear that $\varphi' \wedge \theta$ is satisfiable iff $\varphi' \wedge \theta'$ is, and θ' contains no occurrences of D. Finally, the translation is a PTIME-reduction with at most a quadratic blow-up in the size of the formula. ■

3 Nominal tense logic

The classic language for temporal reasoning is Priorean tense logic [Pri67]. This is a bimodal language whose $\Diamond$ -modalities are F (for reasoning about the *future*) and P (for reasoning about the *past*). P is interpreted using the converse of the transition relation for F. That is:

$$\begin{aligned} \mathfrak{M}, m \Vdash F\varphi & \quad \text{iff} \quad \exists m' \in M(Rmm' \ \& \ \mathfrak{M}, m' \Vdash \varphi) \\ \mathfrak{M}, m \Vdash P\varphi & \quad \text{iff} \quad \exists m' \in M(Rm'm \ \& \ \mathfrak{M}, m' \Vdash \varphi). \end{aligned}$$

The respective $\Box$ -modalities are written G and H. We know from [Spa93b] that the satisfiability problem for Priorean tense logic over arbitrary frames is PSPACE-complete, and that it remains PSPACE-complete over transitive frames.

When nominals (and possibly @) are added to tense logic, the resulting language is called *nominal tense logic*. Nominal tense logic is well-behaved in many respects, and its good behavior does not depend on @. For example, [Dem99] shows how to define well-behaved @-free sequent systems. Moreover, nominal tense logic is highly expressive at the level of frames. For example, it can define both $\langle \mathbb{N}, < \rangle$ (the natural numbers in their usual order) and $\langle \mathbb{Z}, < \rangle$ (the integers in their usual order) up to isomorphism (see [Bla93]) and the definitions don't require @. In what follows, when stating complexity results we will explicitly say whether they presuppose the presence of @ or not.

What about complexity? The following result, proved in [ABM99], is a good starting point as we can consider the basic hybrid language as the purely future fragment of nominal tense logic.

THEOREM 3.1

The satisfiability problem for the basic hybrid language over arbitrary frames is PSPACE-complete.

Now, we know from [Lad77] that ordinary propositional uni-modal logic has a PSPACE-complete satisfiability problem over arbitrary frames. Hence, Theorem 3.1 shows that

we can strengthen the expressivity of the logic by adding nominals and @ without any complexity cost (up to a polynomial). Furthermore, it shows that for arbitrary frames, the guarding strategy (that is, using @ instead of E) pays off: adding the somewhere modality E (and no nominals) to ordinary uni-modal logic results in an EXPTIME-hard satisfiability problem (see [FL79, HM92]).

But these results are of limited *temporal* interest for two reasons. First, while the purely future fragment is useful in certain applications, we often want the past operator too. Moreover, if we think of states in a Kripke model as *time points*, and view R as the *temporal precedence* relation, reading $\Diamond\varphi$ as “ φ occurs in the future,” then this means we should require R to be (at least) *transitive*.

In [ABM99] it was shown that the past operator does make a difference. We pay a price for the ability to look backwards:

THEOREM 3.2

The satisfiability problem over arbitrary frames for the Priorean tense language expanded with just one nominal (and no @ operator) is EXPTIME-hard.

This sounds like bad news — but because the class of arbitrary frames is of little relevance in temporal logic, it is merely a warning sign. The real task that faces us is to examine the complexity of hybrid logics over the frame classes that really are important in temporal logic. We start by looking at the minimal requirement we can impose on R , transitivity.

3.1 Transitive frames.

We start again with the simple case of the pure future fragment (that is, the basic hybrid language). We know (also from [Lad77]) that the satisfiability problem for ordinary uni-modal logic over transitive frames is PSPACE-complete. What happens when we add nominals and @? Again, nothing. In fact we can even add the somewhere modality E while staying in PSPACE.

THEOREM 3.3

The satisfiability problem over transitive frames for the basic hybrid language expanded with E is PSPACE-complete.

PROOF. The lower bound follows from [Lad77]. For the proof of the upper bound, we only consider formulas without occurrences of @. We lose no generality because $@_i\varphi$ is equivalent to $E(i \wedge \varphi)$.

For a formula ξ we will define a two player ξ -game. We will show that Eloise (the existential player) has a winning strategy in the ξ -game if and only if ξ is satisfiable. The ξ -game is designed so that it halts after at most $|SF(\xi)|$ rounds, where $SF(\xi)$ is the set of all subformulas in ξ . Moreover, at each stage of the game at most $|SF(\xi)|$ Hintikka sets (maximal consistent sets of subformulas of ξ) are on the board. Coding this information takes space polynomial in the length of ξ . Given the close correspondence between Alternating Turing Machines (ATM's) and two player games (see [Chl86]), it is straightforward to implement the problem of deciding whether Eloise has a winning strategy on a PTIME ATM. As any PTIME ATM algorithm can be turned into a PSPACE Turing Machine program, we obtain the desired upper bound.

Fix a formula ξ . The ξ -game is played as follows. There are two players, $\forall$ belard (male) and Eloise (female). She starts the game by playing a collection $\{X_0, \dots, X_k\}$

8 The Computational Complexity of Hybrid Temporal Logics

of Hintikka sets. She must choose these Hintikka sets so that the following three conditions hold:

1. X_0 contains ξ , and $|\{X_0, \dots, X_k\}|$ is smaller than $|SF(\xi)|$,
2. no nominal occurs in two different Hintikka sets,
3. for all X_l , for all $E\varphi \in SF(\xi)$, $E\varphi \in X_l$ iff $\varphi \in X_k$, for some k .

If Eloise cannot find Hintikka sets satisfying these conditions, she loses the game immediately. If she can, the game continues. $\forall$ belard chooses an X_l and a “defect formula” $\Diamond\varphi \in X_l$. Eloise must respond with a Hintikka set Y such that

1. $\varphi \in Y$ and for all $\Diamond\psi \in SF(\xi)$, $\Diamond\psi \notin X_l$ implies that $\Diamond\psi \notin Y$ and $\psi \notin Y$,
2. for all $E\varphi \in SF(\xi)$, if $\varphi \in Y$ then $E\varphi \in Y$, and $E\varphi \in Y$ iff $E\varphi \in X_l$,
3. if $i \in Y$ for some nominal i , then Y is one of the Hintikka sets she played at the start. In this case the game stops and Eloise wins.

If Eloise cannot find a suitable Y , the game stops and $\forall$ belard wins. If Eloise does find a suitable Y (one that is not covered by the halting clause in item 3 above) then Y is added to the list of played sets, and play continues, the players moving alternately. At each stage, $\forall$ belard must choose a defect $\Diamond\varphi$ from the last played Hintikka set. To ensure that the length of the game is bounded by $|SF(\xi)|$, we keep a list of the $\Diamond$ -formulas $\forall$ belard plays, and we insist that if he plays a formula $\Diamond\varphi$ a second time, Eloise has to respond with the Hintikka set she played when he challenged with $\Diamond\varphi$ the first time. If this (forced) response does not meet the three criteria just listed, she loses; but if it does meet these criteria, she wins. Either way, the game stops immediately.

CLAIM: Eloise has a winning strategy in the ξ -game iff ξ is satisfiable in the class of transitive hybrid models.

PROOF OF CLAIM.

[$\Rightarrow$]. Suppose Eloise has a winning strategy for the ξ -game. We build a model $\mathfrak{M}$ for ξ as follows. The domain M is built in steps by following her winning strategy. M_0 consists of her initial move $\{X_0, \dots, X_k\}$. Suppose M_j is defined. Then M_{j+1} consists of a copy of those Hintikka sets she plays when using her winning strategy for each of $\forall$ belard's possible moves played in the Hintikka sets from M_j (except when she is required to play a Hintikka set from her initial move that contains a nominal; we don't copy these sets). Let M be the disjoint union of all the M_j . Set Rmm' iff for all $\Diamond\varphi \in SF(\xi)$, $\Diamond\varphi \notin m \Rightarrow [\Diamond\varphi \notin m' \ \& \ \varphi \notin m']$. Let $V(a) = \{m \in M \mid a \in m\}$, for all atoms a in $SF(\xi)$; and if i is any nominal *not* in $SF(\xi)$, then $V(i)$ is an arbitrary singleton subset of M . Clearly $\mathfrak{M}$ is a transitive hybrid model.

We claim that the following truth-lemma holds: for all $\psi \in SF(\xi)$, for all $m \in M$, $\mathfrak{M}, m \models \psi$ if and only if $\psi \in m$. This follows by induction on the structure of formulas. The steps for atoms, Booleans and E are obvious. For $\Diamond$, first suppose that $\Diamond\psi \in m$. Then, as $\forall$ belard can challenge this defect, it must be possible for Eloise to respond with an m' containing ψ . Since for all $\Diamond\psi \in SF(\xi)$, $\Diamond\psi \notin m \Rightarrow [\Diamond\psi \notin m' \ \& \ \varphi \notin m']$ holds, we have Rmm' and by induction hypothesis $\mathfrak{M}, m' \models \Diamond\psi$. For the converse, we argue by contraposition. Suppose that $\Diamond\psi \notin m$. Let m' be an arbitrary R -successor of m . Then by our definition of R , $\psi \notin m'$, so $\mathfrak{M}, m' \not\models \psi$. As m' was arbitrary, $\mathfrak{M}, m \not\models \Diamond\psi$ as required. Thus the truth-lemma holds, and as Eloise played

a Hintikka set containing ξ in the first round, ξ is satisfiable in $\mathfrak{M}$. We have proved the left to right direction of our claim.

[$\Leftarrow$]. Suppose ξ is satisfiable. That is, suppose there is some hybrid model $\mathfrak{M}$ and a point m_0 such that $\mathfrak{M}, m_0 \Vdash \xi$. Let $\mathfrak{M}^f = \langle M^f, R^f, V^f \rangle$ be the transitive filtration of $\mathfrak{M}$ through $SF(\xi)$.² Then $\mathfrak{M}^f$ is a *finite* model, $\mathfrak{M}^f, |m_0| \Vdash \xi$, and moreover each state in $\mathfrak{M}^f$ is a Hintikka set over ξ . Think of Eloise as consulting $\mathfrak{M}^f$ as she plays, and choosing her moves from its states. For her first move, Eloise chooses $|m_0|$ (as this is a point in $\mathfrak{M}^f$ that contains ξ), and each state of $\mathfrak{M}^f$ that contains a nominal and for every $E\psi \in SF(\xi)$, if ψ is satisfied in $\mathfrak{M}^f$, just one $|m|$ such that $\mathfrak{M}^f, |m| \Vdash \psi$. Clearly her first move satisfies the required conditions. Now for the crucial point: when $\forall$ belard chooses a defect $\Diamond_\varphi$ from a Hintikka set X , Eloise responds with a *maximal* R^f -successor Y of X such that Y contains φ ; as $\mathfrak{M}^f$ is finite, such a Y exists. It is this choice that enables Eloise to successfully play the same Hintikka set twice if $\forall$ belard chooses a defect $\Diamond_\varphi$ twice. For suppose Eloise has played Y in response to a defect $\Diamond_\varphi$ in X . Then suppose that at some later stage $\forall$ belard points to the defect $\Diamond_\varphi$ in Z , for Z a successor of X . Eloise consults the model, looking for a maximal R^f successor of Z that contains φ . But as $R^f X Z$, any such point will also be a maximal R^f successor of X that contains φ . Thus Y is a suitable choice, and by playing it again she wins immediately. Thus Eloise has a winning strategy for ξ -game. This completes the proof of the right to left direction of our claim. $\blacktriangleleft$

The theorem follows directly from the claim. $\blacksquare$

So far, so good: nothing strange happens when we add transitivity to the basic hybrid language. But as we will now show, transitivity is not enough to tame hybridization when the backward looking modality P is added. The proof that follows uses the *spypoint technique* from [BS95, ABM99].

THEOREM 3.4

The satisfiability problem over transitive frames for the Priorean tense language expanded with just one nominal (and no $@$ operator) is EXPTIME-hard.

PROOF. We will reduce the EXPTIME-complete *global* K-satisfiability problem for uni-modal languages to the (local) satisfiability problem over transitive frames for a language of Priorean tense logic that contains at least one nominal (and no $@$ operator). The global K-satisfiability problem for uni-modal languages is this: given a formula φ in the uni-modal language, does there exist a Kripke model $\mathfrak{M}$ such that $\mathfrak{M} \Vdash \varphi$ (in other words, where φ is true in *all* states)? The EXPTIME-completeness of this problem is an immediate consequence of the EXPTIME-completeness of the satisfiability problem over arbitrary frames for uni-modal logic expanded with the somewhere modality E proved in [Spa93a].

We define a translation function $(\cdot)^t$ from ordinary uni-modal formulas to formulas in a language of tense logic containing one nominal i . We assume without loss of generality that the propositional variables in the nominal language are those of the uni-modal language plus four extra propositional variables 0, 1, 2, and 3. The translation $(\cdot)^t$ is defined as follows:

² $\mathfrak{M}^f$ is defined as follows. For each state $m \in \mathfrak{M}$, $|m|$ is the set of all states in $\mathfrak{M}$ that agree on all formulas in $SF(\xi)$, and $M^f = \{|m| \mid m \text{ is a state in } \mathfrak{M}\}$. R^f is the binary relation on M^f defined by $R^f |m| |m'|$ iff for all ψ , if $\Diamond\psi \in SF(\xi)$ and $\mathfrak{M}, m \not\Vdash \Diamond\psi$, then $\mathfrak{M}, m' \not\Vdash \psi$ and $\mathfrak{M}, m' \not\Vdash \Diamond\psi$. $V^f(a) = \{|m| \mid \mathfrak{M}, w \Vdash a\}$, for all atomic symbols $a \in SF(\xi)$, and V^f assigns arbitrary singletons to nominals not in $SF(\xi)$.

$$\begin{aligned} p^t &= p \\ (\neg\xi)^t &= \neg\xi^t \\ (\xi \wedge \theta)^t &= \xi^t \wedge \theta^t \\ (\Diamond\xi)^t &= F(1 \wedge P(2 \wedge F(3 \wedge P(0 \wedge P i \wedge \xi^t))))). \end{aligned}$$

Clearly $(\cdot)^t$ is a linear reduction. The intuition behind this translation is that we mimic one R -step in an ordinary Kripke frame by the zigzag transition shown in Figure 1 in a transitive hybrid frame.

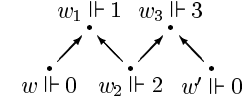


FIG. 1. Translation of diamond formulas.

The figure shows how an arrow from w to w' in the original model would be encoded in the transitive model, with intermediate stops at w_1 , w_2 and w_3 . The propositional symbol 0 will mark the elements of the original model, the others are auxiliary in the encoding. Note that on transitive frames, $FFp \rightarrow Fp$ and $PPp \rightarrow Pp$ are valid, but $FPFPp \rightarrow FPp$ is not.

CLAIM: For any uni-modal formula φ , φ is globally K-satisfiable iff $i \wedge F0 \wedge G(0 \rightarrow \varphi^t)$ is satisfiable in a hybrid model based on a transitive frame.

PROOF OF CLAIM.

[$\Rightarrow$]. Suppose $\mathfrak{M} \Vdash \varphi$, where $\mathfrak{M} = \langle M, R, V \rangle$ is a Kripke model. We now define a transitive hybrid model $\mathfrak{M}^{4s} = \langle M^{4s}, R^{4s}, V^{4s} \rangle$ as follows:

- $M^{4s} = (M \times \{0, 1, 2, 3\}) \cup \{s\}$, for $s \notin M \times \{0, 1, 2, 3\}$. For $x \in \{0, 1, 2, 3\}$, M^x is $\{(m, x) \mid m \in M\}$. In what follows, for all $m \in M$, we shall use the notation m_0, m_1, m_2 , and m_3 instead of $(m, 0)$, $(m, 1)$, $(m, 2)$, and $(m, 3)$. Points in M^0 will correspond to the points in the Kripke model $\mathfrak{M}$; points in M^1, M^2 and M^3 are a “coding space” in which we can construct the transitive relation we require. The point s is our *spypoint*.
- R^{4s} is defined as follows

$$R^{4s}ab \iff \begin{cases} a = x_0, b = x_3 & \text{for some } x \in M, \text{ or} \\ a = x_2, b = x_3 & \text{for some } x \in M, \text{ or} \\ a = x_2, b = x_1 & \text{for some } x \in M, \text{ or} \\ a = y_0, b = x_1 & \text{for some } y, x \in M \text{ \& } Ryx, \text{ or} \\ a = s & \end{cases}$$

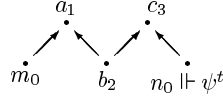
R^{4s} codes an R -transition from point y to x in $\mathfrak{M}$ as a forward step from y_0 to x_1 followed by a backward, forward, backward sequence from x_1 to x_3 . Note that this zig-zag sequence corresponds to the nesting pattern of the Fs and Ps in the translation clause for $\Diamond$. We leave the reader to verify that R^{4s} is a *transitive* relation. Note that the spypoint can see every point in the model.

- If p is a propositional variable not in $\{1, 2, 3, 4\}$, then $V^{4s}(p) = \{m_0 \mid m \in V(p)\}$; for $p \in \{1, 2, 3, 4\}$, $V^{4s}(p) = M^p$, and $V^{4s}(i) = \{s\}$.

It follows by induction that for all $m \in M$, for all uni-modal formulas ψ , $\mathfrak{M}, m \Vdash \psi$ iff $\mathfrak{M}^{4s}, m_0 \Vdash \psi^t$. The only interesting step is for $\Diamond$. That $\mathfrak{M}, m \Vdash \Diamond\psi$ implies $\mathfrak{M}^{4s}, m_0 \Vdash (\Diamond\psi)^t$ is straightforward. For the reverse implication, suppose that $\mathfrak{M}^{4s}, m_0 \Vdash (\Diamond\psi)^t$, that is, suppose

$$\mathfrak{M}^{4s}, m_0 \Vdash F(1 \wedge P(2 \wedge F(3 \wedge P(0 \wedge Pi \wedge \psi^t)))).$$

This implies that there exist $a_1 \in M^1$, $b_2 \in M^2$, $c_3 \in M^3$ and $n_0 \in M^0$ such that



But by the definition of R^{4s} it follows that $a_1 = n_1$, $b_2 = n_2$, and $c_3 = n_3$. Hence, as $R^{4s}m_0a_1$, it follows that Rmn . Moreover, as $\mathfrak{M}^{4s}, n_0 \Vdash \psi^t$, by the inductive hypothesis $\mathfrak{M}, n \Vdash \psi$. Thus $\mathfrak{M}, m \Vdash \Diamond\psi$.

From this equivalence it follows that $\mathfrak{M}^{4s}, s \Vdash i \wedge F0 \wedge G(0 \rightarrow \varphi^t)$, and we have proved the left to right direction of our claim.

[$\Leftarrow$]. Suppose $\mathfrak{M}, w \Vdash i \wedge F0 \wedge G(0 \rightarrow \varphi^t)$, where $\mathfrak{M} = \langle M, R, V \rangle$ is a transitive hybrid model. Define a Kripke model $\mathfrak{M}^\nabla$ as follows:

- $M^\nabla = \{m \in M \mid Rwm \text{ \& } \mathfrak{M}, m \Vdash 0\}$. Note that M^∇ is not empty, for $\mathfrak{M}, w \Vdash F0$.
- $R^\nabla = \{(m, n) \in M^\nabla \times M^\nabla \mid \exists a, b, c \in M(Rma \text{ \& } Rba \text{ \& } Rbc \text{ \& } Rnc \text{ \& } \mathfrak{M}, a \Vdash 1 \text{ \& } \mathfrak{M}, b \Vdash 2 \text{ \& } \mathfrak{M}, c \Vdash 3)\}$.
- $V^\nabla = V_{\upharpoonright M^\nabla}$.

It follows by induction that for all $m \in M^\nabla$, for all uni-modal formulas ψ , $\mathfrak{M}, m \Vdash \psi^t$ iff $\mathfrak{M}^\nabla, m \Vdash \psi$. Only the step for $\Diamond$ is interesting. That $\mathfrak{M}, m \Vdash \psi^t$ implies $\mathfrak{M}^\nabla, m \Vdash \psi$ is straightforward. For the converse, suppose $\mathfrak{M}^\nabla, m \Vdash \Diamond\psi$. This means:

$$\exists n \in M^\nabla (R^\nabla mn \text{ \& } \mathfrak{M}^\nabla, n \Vdash \psi).$$

As $n \in M$, we have that Rwn , hence by the definition of R^∇ we have that

$$\exists n, a, b, c \in M(Rma \text{ \& } Rba \text{ \& } Rbc \text{ \& } Rnc \text{ \& } Rwn \text{ \& } \mathfrak{M}^\nabla, n \Vdash \psi),$$

and moreover a, b, c satisfy 1, 2, and 3 respectively. By the inductive hypothesis $\mathfrak{M}, n \Vdash \psi^t$. Hence, as w denotes i and Rwn , and as all points in M^∇ satisfy 0, we have $\mathfrak{M}, n \Vdash 0 \wedge Pi \wedge \psi^t$. It follows that $\mathfrak{M}, m \Vdash (\Diamond\psi)^t$.

From this equivalence and $\mathfrak{M}, w \Vdash G(0 \rightarrow \varphi^t)$ it follows that $\mathfrak{M}^\nabla \Vdash \varphi$, and we have established the right to left direction of our claim. $\blacktriangleleft$

The theorem follows directly from the claim. $\blacksquare$

Clearly P played a crucial role in this proof — so if we want a PSPACE or lower satisfiability result we will need to add further restrictions “to the past” to tame temporal hybridization. We will learn how to do this in the following section.

Now for the upper bound. We will establish this by extending known results for Propositional Dynamic Logic with nominals. It is known that satisfiability for PDL enriched with both nominals and E is solvable in EXPTIME (see [PT91]). Moreover,

12 The Computational Complexity of Hybrid Temporal Logics

as De Giacomo observes (see [De 95]), his results on PDL-like description languages containing the $\mathcal{O}$ (“one-of”) operator show that the satisfiability problem for PDL with converse enriched with nominals is solvable in EXPTIME. Now, on *connected* frames — assuming a finite repertoire of atomic programs — the somewhere modality is definable in converse PDL. But to establish the upper bounds we want, we need to know that we can have access to both converse programs *and* E on *arbitrary* frames and still stay in EXPTIME. And in fact, we can. Once again, we make use of a spypoint argument, but this time to obtain an upper bound:

THEOREM 3.5

The satisfiability problem for Propositional Dynamic Logic with converse enriched with both nominals and E is solvable in EXPTIME.

PROOF. Let ξ be a formula in this language. Without loss of generality, we may assume that converse is only applied to atomic programs. We will transform ξ into a formula without occurrences of E and then use de Giacomo’s result. Let s be a nominal and σ be an atomic program not occurring in ξ . Define ξ^t by first recursively replacing every occurrence of $E\varphi$ in ξ by $\langle\sigma^{-1}\rangle(s \wedge \langle\sigma\rangle\varphi)$, obtaining a formula without occurrences of E . Now transform the programs occurring inside the modalities in ξ by replacing atomic programs p (converse programs p^{-1}) by $p; \langle\sigma^{-1}\rangle s$? (respectively $p^{-1}; \langle\sigma^{-1}\rangle s$?). We claim that ξ is satisfiable iff $\xi^t \wedge \langle\sigma^{-1}\rangle s$ is satisfiable. Since $\xi^t \wedge \langle\sigma^{-1}\rangle s$ is in the language covered by de Giacomo’s EXPTIME-algorithm, this claim proves the theorem.

The left to right direction of the claim is obvious: just add a new state to the model, make s true there, and let that state be σ -connected to all other states.

For the other direction, let $\mathfrak{M} = \langle M, R_i, V \rangle_{i \in I}$ satisfy $\xi^t \wedge \langle\sigma^{-1}\rangle s$ at w and $V(s) = \{s\}$. Let $\mathfrak{M}^-$ be the submodel of $\mathfrak{M}$ obtained by restricting the universe M to the set $M^- = \{x \in M \mid R_\sigma s x\}$. We claim that for all subformulas ψ of ξ , for all $x \in M^-$,

$$\mathfrak{M}^-, x \Vdash \psi \iff \mathfrak{M}, x \Vdash \psi.$$

As $w \in M^-$, this would provide us with the desired result. The proof of the claim goes by the usual double induction needed for inductive proofs in PDL. The proof is straightforward given the following observations:

- the translation of E formulas works because we restricted the model to successors of the spypoint s .
- $\mathfrak{M}^-$ is a *generated* submodel of $\mathfrak{M}$ for the new “atomic” programs $p; \langle\sigma^{-1}\rangle s$? and $p^{-1}; \langle\sigma^{-1}\rangle s$? occurring in ξ^t .

This finishes the proof of the theorem. $\blacksquare$

COROLLARY 3.6

1. The satisfiability problem for nominal tense logic with E over arbitrary frames is solvable in EXPTIME.
2. The satisfiability problem for nominal tense logic with E over transitive frames is solvable in EXPTIME.

PROOF. For the arbitrary frame case, we define a linear translation $(\cdot)^t$ of formulas of nominal tense logic (without $@$) into formulas of nominal PDL with converse as

follows: $(a)^t = a$ for all atoms a , $(\neg\xi)^t = \neg(\xi)^t$, $(\xi \wedge \theta)^t = (\xi)^t \wedge (\theta)^t$, $(F\xi)^t = \langle r \rangle (\xi)^t$, $(P\xi)^t = \langle r^{-1} \rangle (\xi)^t$, and $(E\xi)^t = E(\xi)^t$. Here r is a fixed atomic program. It is easy to see that φ is satisfiable iff $(\varphi)^t$ is satisfiable.

For the transitive frame case, we define a linear translation $(\cdot)^{tt}$ which is identical to $(\cdot)^t$ for atoms and booleans, but handles F and P as follows: $(F\xi)^{tt} = \langle r; r^* \rangle (\xi)^{tt}$, and $(P\xi)^{tt} = \langle r^{-1}; (r^{-1})^* \rangle (\xi)^{tt}$. It is easy to see that φ is satisfiable over a transitive frames iff $(\varphi)^{tt}$ is satisfiable. ■

Hence, temporal hybridization on transitive frames still moves us outside the complexity class of the original logic. But from now on things get better. Two main kinds of transitive structures are usually considered as standard representations of time: strict total orders for linear time, and transitive trees for branching time. In the next two sections we will prove that in both cases hybridization is tamed. With this we mean to say that the expansion with nominals and $@$ does *not* lead to an increase in complexity.

3.2 Linear time

For many applications, time is modeled as linear. Of course, choosing linearity leaves many interesting options open, such as whether density or discreteness holds, and whether or not initial and final points in time exist. But for our purposes, such choices are irrelevant: the complexity of the satisfiability problem for nominal tense logic *over any subclass* of the class of strict total orders (that is, frames that are irreflexive, transitive, and trichotomous ($\forall s, t \in W (sRt \vee s = t \vee tRs)$)) is the same as for Priorean tense logic. But given our remarks in Section 2 this should not come as a surprise: over strict total orders, F and P are strong enough to define D , as $D\varphi \leftrightarrow (F\varphi \vee P\varphi)$ is valid. Thus we can eliminate all occurrences of nominals (and $@$ and E) by simulating them using D ; in effect, we do the reverse of what we did in Theorem 2.1. Some care has to be taken to avoid a blow up in formula size during the elimination process, but it is easy to define an inductive replacement similar to that used in the proof of Theorem 2.1 that avoids such problems. Thus:

THEOREM 3.7

Let S be a subclass of the class of strict total orders. The complexity of the S -satisfiability problem is (up to a polynomial factor) the same for nominal tense logic (with or without $@$, E , or D) as it is for Priorean tense logic.

On many natural linear flows of time (for example, the class of all strict total orders, or the class containing only $\langle \mathbb{Q}, < \rangle$) the complexity of the satisfiability problem for the Priorean language is NP-complete [ON80, SC85], that is, no worse than propositional calculus, so nominal tense logic inherits these results.

Theorem 3.7 may seem a bit like cheating. We don't pay any computational cost, but this is because hybridization over strict total orders does not increase the expressive power at our disposal. This is true, but we think it misses the point. In many applications of temporal logic, reference to times is not an optional extra, it's fundamental. To model such problems naturally, we need formalisms which allow us to deal with temporal reference directly. Adding $@$ and nominals gives us what seems to be the correct level of abstraction needed. It is this level of abstraction (rather

than the lower level offered by D) that needs to be isolated and explored. We'll return to this point in Section 5, when we discuss interval-based logic.

3.3 Branching time

In discussions of temporal logic in philosophy, natural language semantics, and computer science, the following intuition plays an important role: while several possible futures may be allowed, the past has a linear structure. To put it another way, it is often assumed that time has a tree-like structure, with the branching occurring only towards the future, never towards the past. Call a directed graph $\langle T, < \rangle$ a *tree* if it is acyclic and connected, and every node has at most one predecessor. A *transitive tree* is the transitive closure of a tree. In this section we are interested in the satisfiability problem for nominal tense logic over such frames.

Note that the spypoint argument used to prove EXPTIME-hardness in Theorem 3.4 *will not work* for transitive trees: the encoding of transitivity in the model $\mathfrak{M}^{ts}$ made crucial use of branching towards the past (points in M^1 look back to points in M^0 and M^2). And in fact, if we demand that our frames are transitive *trees*, we tame P and drop back into PSPACE. The key intuition comes from inspecting the structure of the past in a transitive tree: if we generate the submodel by the converse of the accessibility relation from any point in a transitive tree, we obtain a strict linear order. Thanks to this property it will be easy to satisfy past formulas in small structures. Future formulas will require more work, particularly when they interact with nominals. Nonetheless, we will be able to show that all the required computations can be carried out in PSPACE.

NP-Complete Subfragments Before we prove the PSPACE-result for nominal tense logic on transitive trees, let's look at possible further reductions in complexity on this class of frames. Consider the formula $Fi \wedge G(Fi \rightarrow \varphi)$. It says that there is a state named i in the future and that φ holds at every state between now and the state named i . In other words, $Fi \wedge G(Fi \rightarrow \varphi)$ says the same thing as $\text{Until}(i, \varphi)$.

Consider the subfragment of nominal tense logic in which we only have future formulas of the form $\text{Until}(i, \varphi)$, for one fixed nominal i (the rest of the language remains unchanged). For this fragment every satisfiable formula can be satisfied in a model of size polynomial in the length of the formula, yielding NP-completeness. This is shown by a simple submodel generation argument. Let $\mathfrak{M}, m \models \varphi$. If $\mathfrak{M}, m \models Fi$, create a submodel by considering the i -named state plus all its predecessors; otherwise create a submodel by taking m plus all its predecessors. In both cases we obtain a linear model which still satisfies φ at m . Since the models are linear, we can now use standard techniques from temporal logic to create a polysize model for φ . Here again, we obtained a reduction in complexity by *guarding* the modalities: we only allowed formulas of the form $G(Fi \rightarrow \varphi)$.

THEOREM 3.8

Let $\mathcal{L}$ be the sublanguage of nominal tense logic in which every occurrence of a formula $F\psi$ is of the form $F(Fi \wedge \varphi)$, for some i, φ . Every $\mathcal{L}$ formula which is satisfiable on a transitive tree is satisfiable on a transitive tree of size polynomial in the length of the formula.

A simple inspection of the above argument shows that the restriction to just one

nominal as guard is not needed to obtain the result. By distributing out disjunctions the same result holds for formulas of the form $F(i_1 \vee \dots \vee i_k) \wedge \varphi$. These formulas are used in the proof of the next theorem to distinguish future formulas that need their witnessing state before any named state from ones which don't, as follows. Suppose in $\mathfrak{M}$ there are only finitely many states named by a nominal, say $i_1, \dots, i_k$. Let $\mathfrak{M}, m \Vdash F\varphi$. Then either $\mathfrak{M}, m \Vdash F(i_1 \vee \dots \vee i_k) \wedge \varphi$ or $\mathfrak{M}, m \Vdash \neg F(i_1 \vee \dots \vee i_k) \wedge \varphi$. If the latter is false, this means that φ has to be true between m and some named state.

THEOREM 3.9

The satisfiability problem for nominal tense logic over the class of transitive trees is PSPACE-complete.

PROOF. PSPACE-hardness follows from results in [Lad77]: the satisfiability problem for ordinary uni-modal logic over transitive trees is already PSPACE-hard. The real work is to prove the PSPACE upper bound.

Our argument will be similar to the one used in Theorem 3.3, but now we should construct the appropriate tree structure in the game played by $\forall$ belard and $\exists$ loise. Instead of Hintikka sets, the players will use sequences of Hintikka sets which will take the role of partial branches in the model. But let us set up the game; first we define the following notions.

For any formula ξ in the nominal tense language, define the closure set $Cl(\xi)$ as the smallest set containing ξ , closed under subformulas, single negation and under the following rule: if $i \in Cl(\xi)$ then $P(i) \in Cl(\xi)$.

Let ξ be fixed. A *thread* is a finite labelled frame $\langle T, <, l \rangle$ such that:

1. $<$ is a weak total order (i.e., $<$ is transitive and trichotomous),
2. $l : T \rightarrow Pow(Cl(\xi))$,
3. l labels with maximal consistent Hintikka sets over $Cl(\xi)$,
4. if $i \in l(x)$ then $Pi \notin l(x)$,
5. $|T| \leq |\{P\varphi \in Cl(\xi)\} \cup \{\varphi \in Cl(\xi)\}| + 1$,
6. (Future coherence) if $\varphi \notin l(x)$ and $x < y$, then $\{F\varphi, \varphi\} \cap l(y) = \emptyset$,
7. (Past saturation) $P\varphi \in l(y)$ iff for some $x, x < y$ and $\varphi \in l(x)$.

The size $|t|$ of a thread $t = \langle T, <, l \rangle$ is $|T|$, and we say that $t_1 = \langle T_1, <_1, l_1 \rangle$ and $t_2 = \langle T_2, <_2, l_2 \rangle$ *fit at* $x \in T_1 \cap T_2$ iff $\langle T_1, <_1, l_1 \rangle|_{\{s \in T_1 \mid s \leq_1 x\}} = \langle T_2, <_2, l_2 \rangle|_{\{s \in T_2 \mid s \leq_2 x\}}$. Two threads *fit* if there exists an x such that they fit at x .

We first take care of the trivial case when $|\xi| = 1$. Then ξ is either an atom or a logical constant, and satisfiability is trivial. From now on we assume $|\xi| > 1$.

We can now specify the game for $\forall$ belard and $\exists$ loise. She should set up the playing board by specifying a collection M_0 of threads such that:

1. for some thread $t = \langle T, <, l \rangle \in M_0$, for some $x \in T$, $\xi \in l(x)$,
2. each two threads in M_0 fit,
3. there are no more threads in M_0 than the number of nominals in $Cl(\xi)$ plus one.
4. for each nominal i in $SF(\xi)$ there is a thread $t = \langle T, <, l \rangle \in M_0$, and $x \in T$ such that $i \in l(x)$,
5. let i be a nominal in $SF(\xi)$, t_1, t_2 two threads in M_0 and $x_1 \in T_1, x_2 \in T_2$ be such that $i \in l_1(x_1)$ and $i \in l_2(x_2)$, then $x_1 = x_2$ and t_1 and t_2 fit at x_1 ,

The rationale behind M_0 is that — when the threads are fitted together — it can be seen as a transitive tree model in which no nominal occurs in the label of two distinct states.

If $\exists$ loise cannot set up the board she loses. Otherwise $\forall$ belard chooses one of the threads in M_0 , which will be the thread *in use* in the round. To start with, all the elements in the thread in use are *available*. In each round there will be a thread t in use, with a subset of its domain available for $\forall$ belard to pick from. Furthermore we will keep a table S containing for each $F\varphi$ formula in $Cl(\xi)$ a natural number $S(F\varphi)$. At the start, for all $F\varphi \in Cl(\xi)$, $S(F\varphi) = |\xi|$. We are now ready to specify the movements of the players. In each round

- $\forall$ belard points to an element x among the available elements in the thread t in use, and to a formula $F\varphi$ in $l(x)$. x should be a maximal element containing $F\varphi$. Formally, for all $y \in t$, $x < y$ implies $F\varphi, \varphi \notin l(y)$ and furthermore, there is no thread $t'' = \langle T'', <'', l'' \rangle$ on the board, fitting t at x such that either $F\varphi$ or φ are in $l''(y)$ for some $x <'' y$.
- The answer of $\exists$ loise depends on the value recorded for the formula chosen by $\forall$ belard.
 - If $S(F\varphi) > 1$, then she should play a thread $t' = \langle T', <', l' \rangle$ such that
 - t' fits t at x ,
 - there is $y \in T'$ such that $\varphi \in l'(y)$ and $x <' y$,
 - neither i nor Fi is in $l'(z)$ for $x < z, i \in \text{NOM}$,
 - $|\{z \in T' \mid x <' z\}| < S(F\varphi)$.

If she cannot present such a t' , she loses. Otherwise, t' becomes the thread in use and $\{z \in T' \mid x <' z\}$ the available Hintikka sets for $\forall$ belard. Furthermore $S(F\varphi)$ is updated to $|\{z \in T' \mid x <' z\}|$.

► If $S(F\varphi) = 1$, then $F\varphi$ has been played by $\forall$ belard before. $\exists$ loise should pick the Hintikka set e' containing φ that fixed the defect the last time $\forall$ belard challenged $F\varphi$. If $t' = \langle T_{\{z \in T \mid z \leq x\}} \cup \{n\}, < \cup \{(y, n) \mid y \in T\}, l \cup (n, e') \rangle$ for $n \notin T$ is a thread she wins. Otherwise she loses.

Notice that the values in S decrease as $\forall$ belard repeats choices of $F\varphi$ (which he cannot avoid as $Cl(\xi)$ is finite). This ensures termination of the game in at most $|\{F\varphi \in Cl(\xi)\}| * |\xi|$ steps.

To prove the theorem we should now establish that $\exists$ loise has a winning strategy in the game described above if and only if ξ is satisfiable; and then observe that in each game only a polynomial number of Hintikka sets are displayed on the board. (The extra information contained in the table S can obviously be coded using just linear space in the length of ξ .)

CLAIM: $\exists$ loise has a winning strategy for the ξ -game iff ξ is satisfiable in a model based in a transitive tree frame.

PROOF OF CLAIM.

[$\Rightarrow$] Suppose $\exists$ loise has a winning strategy in the ξ -game. We construct a model $\mathfrak{M} = \langle N, R, V \rangle$ as follows. Let M be the set of threads ever played by $\exists$ loise in her winning strategy. $\mathfrak{M}$ is obtained by “superposing” one by one any two threads in M at their fitting point. More precisely, given two threads t and t' there is a maximal

point x such that t and t' fit. Hence for any thread t we can identify a maximal point x such that t fits with some other thread at x . The elements above x are “unique” to t ; ensure uniqueness by renaming and let M' be the set of threads obtained in such a way. Notice that after such a renaming, for any element x of any thread in M' we can uniquely identify a Hintikka set corresponding to x , and for any two elements x, y in threads $t = \langle T, <, l \rangle$, $t' = \langle T', <', l' \rangle$ in M , if $x, y \in T \cap T'$ then $x < y$ iff $x <' y$.

Now let N be the union of all elements in threads in M' , R be the union of the $<$ relations, and let V assign to an atomic symbol a the elements x in N such that a is in the Hintikka set corresponding to x . If i is a nominal not in $SF(\xi)$ then $V(i)$ is any singleton set in N . Since during the setting up of the board nominals in M_0 are assigned to unique elements and Eloise never plays threads with new Hintikka sets containing nominals in her winning strategy, V is a hybrid valuation.

We claim that the truth lemma for elements in $CI(\xi)$ holds for $\mathfrak{M}$, i.e., let $\varphi \in CI(\xi)$, and s be any element in N , then $\mathfrak{M}, s \models \varphi$ iff $\varphi \in H(s)$, where for any x in N , $H(x)$ is the Hintikka set corresponding to x .

The interesting case of the truth lemma are the modalities. By condition 7 in the definition of threads, elements in M' are past saturated, and the structure of threads has been preserved in $\mathfrak{M}$, hence the truth lemma for $P\psi$ holds straightforwardly.

Suppose $F\psi \in H(s)$. But $s \in t$ for some t in M (disregarding renaming), and $\forall$ belard had a chance to chose each $F\psi$ formula in any Hintikka set in M , at least for s which were maximal for formula $F\psi$. If s is not maximal for $F\psi$, then it has a maximal successor s' containing $F\psi$, which could be challenged by $\forall$ belard. As Eloise has a winning strategy she should be able to answer the challenge and produce a thread t' fitting t at s' . By the condition of the game, t' will contain a Hintikka set s'' with ψ in its label set, such that $s' <' s''$. By the definition of $\mathfrak{M}$, transitivity, and IH, t' will contribute the needed witness into N .

For the converse we argue by contraposition. Suppose $F\psi \notin H(s)$ and let s' be such that $R(s, s')$. But then there is a thread $t = \langle T, <, l \rangle \in M$ such that $s, s' \in T$ and $s < s'$. By condition 6, on threads ψ does not belong to $H(s')$. By IH, $\mathfrak{M}, s' \not\models \varphi$. As s' was a generic successor of s we have $\mathfrak{M}, s \not\models F\varphi$.

The threads $t = \langle T, <, l \rangle$ used in the construction of $\mathfrak{M}$ might contain *clusters*, i.e., maximal (by inclusion) non empty sets S of T such that for all $x, y \in S$, $x < y$. But we can unravel them by a standard technique, preserving satisfaction. Notice that worlds labelled by nominals will never appear in clusters by items 4 and 7 in the definition of threads and hence they will not be duplicated by the unraveling. The unraveled model will be a transitive tree. This completes the left to right part of the claim.

[$\Leftarrow$] We should now prove that if ξ is satisfiable, then Eloise has a winning strategy. Let $\mathfrak{M}$ be a transitive tree model such that $\mathfrak{M}, w \models \xi$. And let $\mathfrak{M}^f = \langle M^f, R^f, V^f \rangle$ be a transitive filtration of $\mathfrak{M}$ under $CI(\xi)$. By properties of filtrations we know that $\mathfrak{M}^f$ is finite and R^f is transitive. Furthermore R^f has maximal (and minimal) elements, i.e., those $|m| \in M^f$ such that for all $|m'| \in M^f$, $R^f|m||m'| \Rightarrow R^f|m'| |m|$. Also, elements of M^f can be uniquely identified with Hintikka sets over $CI(\xi)$ (and we will treat them as such). But $|M^f|$ can still be exponential in $|\xi|$.

For each element $|m| \in M^f$ we can built a thread t as follows. For each $P\varphi$ in $|m|$ choose a minimal predecessor $|m'|$ of $|m|$ which satisfies φ . Order t by R^f . Since the filtration comes from a transitive tree, a standard argument shows that t is past

saturated and hence a thread. Furthermore $|t|$ is linear in $|\xi|$. We call such t a thread built from $|m|$ in M^f . Let φ be a formula satisfiable in $\mathfrak{M}^f$. Then a thread for φ is a thread t from some $|m|$, such that $\mathfrak{M}^f, |m| \models \varphi$, but we require in addition that $|m|$ is R^f -maximal satisfying φ , and that t is *long*, i.e., for all other threads t' built from $|m'|$ such that $\mathfrak{M}^f, |m'| \models \varphi$, $|t'| < |t|$.

Let t be a thread built from $|m|$ and let $|m'|$ be an element in t . Let $|m''|$ be an R^f -successor of $|m'|$. It is not difficult to prove that there is a thread t' built from $|m''|$ that fits t at $|m'|$. Furthermore t' is linear in $|\xi|$. From this fact,

(*) if t is a thread built from an element in $\mathfrak{M}^f$ and $|m|$ is any element in t such that $\mathfrak{M}^f, |m| \models F\varphi$. Then there is a thread for φ that fits t at $|m|$.

All this machinery is used to specify Eloise's answers to challenges made by $\forall$ belard. We first specify how Eloise can set up the board. Clearly she can find past saturated threads containing ξ and all the nominals and clearly they also fit. But she has to do something extra. Since later in the game she cannot use threads containing nominals in new points she sometimes has to insert extra witnessing points in these threads. Suppose she wants to include a thread containing $|m|$, and suppose $\mathfrak{M}^f, |m| \models F\varphi$ but *all* $|m'|$ such that $R^f|m||m'|$ and $\varphi \in m'$ lie before a nominal occurring in ξ . Then she should include a witnessing $|m'|$ already in step 0. Theorem 3.8 and the remarks just following it state precisely that she can include all these witnesses and still use only a polynomial number of states (the exact required bound is easily computed from the proof). With this setup it is guaranteed that in the case the antecedent of (*) above obtains, the needed fitting thread can be chosen not to contain any nominal or Fi formula.

With Eloise's groundwork taken care of, we now specify how Eloise answers the challenges of $\forall$ belard. Suppose that $\forall$ belard points to a defect formula $F\varphi$ in an element $|m|$ in t . If $S(F\varphi) > 1$, she answers with a thread t' provided by (*), again choosing t' to be maximal and long. As argued above she can answer with a thread in which neither i nor Fi appears in t' above the fitting point.

By obtaining threads from $\mathfrak{M}^f$, Eloise can always answer the moves of $\forall$ belard. What remains to be checked is the condition on decreasing lengths when $\forall$ belard plays a repeated formula. So assume that $\forall$ belard has in a previous round chosen $F\varphi \in |m| \in t_1$, and that Eloise answered with a long maximal thread t_2 . Furthermore he is now choosing $F\varphi \in |m'| \in t_3$. We know that Eloise can produce a long maximal thread t_4 to answer the challenge. But it is immediate that if the size of t_4 above $|m'|$ is greater than the size of t_2 above $|m|$ then t_2 was not a long maximal thread.

Suppose now that $S(F\varphi) = 1$ and let t_1 be the thread previously played by Eloise as an answer, with $\varphi \in |m_1|$. Again by (*) we could produce a thread t' fitting t at $|m|$ and containing φ in some element $|m'|$. Furthermore, because of the decreasing condition, we know that this time she can choose t' to be very short. Actually $|m|$ is at “one step” from the R^f -maximal cluster satisfying φ , and Eloise can choose any element of such a cluster, in particular $|m_1|$. $\blacktriangleleft$

As we already mentioned, the game stops in time quadratic in the size of ξ , furthermore all the information ever played in the board can be encoded in polynomial space. Hence the theorem follows. $\blacksquare$

What happens if we add E? Nothing — for we already have it: over transitive trees $E\varphi$ can be defined to be $\varphi \vee F\varphi \vee P\varphi \vee PF\varphi$.

COROLLARY 3.10

The transitive tree satisfiability problem for nominal tense logic with E is PSPACE-complete.

With this we close our analysis of the complexity of nominal tense logic. In the next two sections we discuss two other hybrid temporal logics for which we have obtained some preliminary results.

4 Nominal Until/Since logic

We first turn to two other operators often used in temporal logic: Until and Since. On any (Kripke or hybrid) model $\mathfrak{M} = \langle M, R, V \rangle$, they are defined as follows:

$$\begin{aligned} \mathfrak{M}, s \Vdash \text{Until}(\varphi, \psi) &\text{ iff } \exists t \in M (sRt \ \& \ \mathfrak{M}, t \Vdash \varphi \ \& \ \forall r \in M (sRrRt \Rightarrow \mathfrak{M}, r \Vdash \psi)) \\ \mathfrak{M}, s \Vdash \text{Since}(\varphi, \psi) &\text{ iff } \exists t \in M (tRs \ \& \ \mathfrak{M}, t \Vdash \varphi \ \& \ \forall r \in M (tRrRs \Rightarrow \mathfrak{M}, r \Vdash \psi)). \end{aligned}$$

Using Until we can define $F\varphi := \text{Until}(\varphi, \top)$ and similarly, using Since we can define $P\varphi := \text{Since}(\varphi, \top)$. It follows by our earlier results on nominal tense logic that enriching a language containing both Until and Since with even a single nominal results in an EXPTIME-hard satisfiability problem over both arbitrary and transitive frames. But it turns out that in this case, the introduction of a past operator is not to blame: already the pure future fragment is EXPTIME-hard. In what follows we will work with languages whose only primitive modality is Until, and treat F (and G) as defined symbols.

Now, it is well known that Priorean tense logic is not strong enough to define Until, even over linear frames, and the addition of nominals and @ does not help. On the other hand, unlike the difference operator, Until is not strong enough to simulate nominals (indeed, neither can Until and Since working together) on arbitrary structures. Thus enriching an Until-based language with nominals results in a genuine increase in expressive power; for example, many new classes of frames become definable. So: what is the complexity of nominal Until logic?

THEOREM 4.1

The satisfiability problem over arbitrary frames for a Until logic extended with a single nominal (and no @) is EXPTIME-hard.

PROOF. The proof strategy is the same as that used for Theorem 3.4: we shall reduce the EXPTIME-complete *global* satisfiability problem for ordinary uni-modal languages to the (local) satisfiability problem for a language containing the Until operator and at least one nominal (and no @) by means of a spypoint argument. First we define:

$$i\text{-spy} := i \wedge \neg Fi \wedge FF i \wedge G\neg \text{Until}(Fi, \neg i).$$

We shall use the following three observations in our proof. If *i-spy* is satisfied in a model $\langle M, R, V \rangle$ at a point *s* then:

1. $\neg Rss$.
2. $\exists m \in M (m \neq s \ \& \ Rsm \ \& \ Rms)$.
3. $\forall m, m' \in M (Rsm \ \& \ Rmm' \ \& \ Rm's \Rightarrow Rsm')$.

Observations 1 and 2 are obvious consequences of the first three conjuncts of *i-spy*. Observation 3 is the crucial one and requires comment. Any formula of the form $\neg \text{Until}(\xi, \theta)$ is true at a point *m* in a model iff for every successor *m'* of *m* at which ξ is true, there is a point *u* between *m* and *m'* at which θ is *false*. So if $\neg \text{Until}(Fi, \neg i)$ is true at *m*, this means that at every successor *m'* of *m* which precedes *s* (for that is what *Fi* asserts) there is a point *u* between *m* and *m'* at which $\neg i$ is false. But there is only one point in the entire model where $\neg i$ is false, namely *s*. In short $\neg \text{Until}(Fi, \neg i)$ guarantees that every successor *m'* of *m* which precedes *s* is also preceded by *s*. Prefixing G ensures that all successors *m* of *s* will have this property.

Next, we define the following translation function $(\cdot)^t$ from ordinary uni-modal formulas to formulas in a language containing the Until operator and at least one nominal *i*:

$$\begin{aligned} p^t &= p \\ (\neg \xi)^t &= \neg \xi^t \\ (\xi \wedge \theta)^t &= \xi^t \wedge \theta^t \\ (\diamond \xi)^t &= F(\xi^t \wedge Fi). \end{aligned}$$

Clearly $(\cdot)^t$ is a linear reduction.

CLAIM: For any ordinary uni-modal formula φ , φ is globally satisfiable in a Kripke model iff *i-spy* $\wedge G(\varphi^t)$ is satisfiable in a hybrid model.

PROOF OF CLAIM.

[$\Leftarrow$]. Suppose that $\mathfrak{M}, s \Vdash i\text{-spy} \wedge G(\varphi^t)$, where $\mathfrak{M} = \langle M, R, V \rangle$ is a hybrid model. Clearly *s* must be the unique element in $V(i)$, and our observations concerning *i-spy* hold. Define $\mathfrak{M}^\nabla$ as follows

- $M^\nabla = \{m \in M \mid Rsm \ \& \ Rms\}$.
- $R^\nabla = R_{\upharpoonright M^\nabla}$.
- $V^\nabla = V_{\upharpoonright M^\nabla}$.

Note that by observation 2, $M^\nabla \neq \emptyset$. Next, note that for all $m \in M^\nabla$, for all ordinary uni-modal formulas ψ , $\mathfrak{M}, m \Vdash \psi^t$ iff $\mathfrak{M}^\nabla, m \Vdash \psi$. This follows by induction on the structure of φ . The interesting case is the step for $\diamond$:

$$\begin{aligned} &\mathfrak{M}, m \Vdash F(\psi^t \wedge Fi) \\ \iff &\exists m' \in M (Rmm' \ \& \ \mathfrak{M}, m' \Vdash \psi^t \ \& \ m' \neq s \ \& \ Rm's) \\ \iff &\exists m' \in M^\nabla (Rmm' \ \& \ \mathfrak{M}, m' \Vdash \psi^t \ \& \ m' \neq s \ \& \ Rm's) \\ \iff &\exists m' \in M^\nabla (R^\nabla mm' \ \& \ \mathfrak{M}^\nabla, m' \Vdash \psi) \\ \iff &\mathfrak{M}^\nabla, m \Vdash \diamond \psi. \end{aligned}$$

(The key step in the argument is showing that line 2 implies line 3; that is, that $m' \in M^\nabla$. This is where we appeal to observation 3.) Now, by assumption $\mathfrak{M}, s \Vdash i\text{-spy} \wedge G(\varphi^t)$. So for all $m \in M$ such that Rsm , we have $\mathfrak{M}, m \Vdash \varphi^t$. That is, for all $m \in M^\nabla$, $\mathfrak{M}, m \Vdash \varphi^t$. Hence by the equivalence just proved, for all $m \in M^\nabla$, $\mathfrak{M}^\nabla, m \Vdash \varphi$. In short, $\mathfrak{M}^\nabla \Vdash \varphi$, and we have proved the right to left direction of our claim.

[$\Rightarrow$]. Let $\mathfrak{M} \Vdash \varphi$, where $\mathfrak{M} = \langle M, R, V \rangle$ is an ordinary Kripke model. Let *s* be an element not belonging to *M*. We define a hybrid model $\mathfrak{M}^s$ as follows:

- $M^s = M \cup \{s\}$.
- $R^s = R \cup \{(s, m), (m, s) \mid m \in M\}$.
- $V^s = V \cup \{(i, \{s\})\}$.

(If there are nominals in the language besides i , we let V^s assign them arbitrary singletons sets; any such nominals are irrelevant to what follows.) We claim that for all $m \in M$, for all ψ , we have $\mathfrak{M}, m \Vdash \psi$ iff $\mathfrak{M}^s, m \Vdash \psi^t$. This follows by a simple induction. The only interesting step is for $\Diamond$:

$$\begin{aligned}
& \mathfrak{M}, m \Vdash \Diamond \psi \\
\iff & \exists m' \in M (Rmm' \ \& \ \mathfrak{M}, m' \Vdash \psi) \\
\iff & \exists m' \in M^s (R^s m m' \ \& \ \mathfrak{M}^s, m' \Vdash \psi^t \ \& \ m' \neq s \ \& \ R^s m' s) \\
\iff & \mathfrak{M}^s, m \Vdash F(\psi^t \wedge Fi) \\
\iff & \mathfrak{M}^s, m \Vdash (\Diamond \psi)^t.
\end{aligned}$$

(To see that the third line implies the second, note that if $m' \neq s$ and $R^s m' s$, then $m' \in M$.) Now by assumption, $\mathfrak{M} \Vdash \varphi$. Hence by the equivalence just proved, for all $m \in M$, $\mathfrak{M}^s, m \Vdash \varphi^t$. Hence $\mathfrak{M}^s, s \Vdash G(\varphi^t)$. We also need to show that $\mathfrak{M}^s, s \Vdash i\text{-spy}$. That the first three conjuncts of $i\text{-spy}$ hold at s is clear. $G\neg\text{Until}(Fi, \neg i)$, its last conjunction, holds because s precedes every point in the model except itself. $\blacktriangleleft$

The theorem follows directly from the claim. $\blacksquare$

Once again, however, if we give a temporal interpretation to Until we are more interested in satisfiability over transitive frames. Moreover, on transitive frames Until captures a much stronger sense of “betweenness” than it does over non-transitive frames. So: what is the complexity of nominal until logic over transitive frames? We don’t have an answer to this. We cannot directly use the zig-zag between past and future used in Theorem 3.4 (we don’t have a backward looking operator) and we don’t see any other way of polynomially encoding transitivity.

However we do have a related result. Suppose we replace Until by the operator Until^+ with the following satisfaction definition. Let $\mathfrak{M} = \langle M, R, V \rangle$ be any (hybrid or Kripke) model and let R^+ denote the transitive closure of R , then

$$\mathfrak{M}, x \Vdash \text{Until}^+(\varphi, \psi) \text{ iff } \exists y \in M (xRy \ \& \ \mathfrak{M}, y \Vdash \varphi \ \& \ \forall z \in M (xR^+ z R^+ y \Rightarrow \mathfrak{M}, z \Vdash \psi)).$$

That is, $\text{Until}^+(\varphi, \psi)$ holds at x if φ holds at some x -successor y and for all z which are in between x and y , ψ holds, where “in between” now is not restricted to immediate successors and predecessors. In other words Until^+ captures the same strong sense of “betweenness” on arbitrary frames that Until captures on transitive frames. Note that the satisfaction definition of Until^+ is not first-order. So: what is the complexity of this variant of nominal until logic? As before we have:

THEOREM 4.2

The satisfiability problem over arbitrary frames for a language containing Until^+ and at least one nominal (but no $@$) is EXPTIME-hard.

PROOF. We shall reduce the same EXPTIME-complete problem, using the same translation function $(\cdot)^t$, of Theorem 4.1 — but instead of a spypoint argument, we shall use what might be called a *sinkpoint* argument.

22 The Computational Complexity of Hybrid Temporal Logics

CLAIM: For any ordinary uni-modal formula φ , φ is globally satisfiable in a Kripke model iff $\varphi^t \wedge \text{Until}^+(i, \varphi^t) \wedge F(i \wedge \neg Fi)$ is satisfiable in a hybrid model.

PROOF OF CLAIM.

$\Rightarrow$. Let $\mathfrak{M} = \langle M, R, V \rangle \Vdash \varphi$. Without loss of generality we can assume that $\mathfrak{M}$ has a root w (simply take a submodel generated by any point w). Define $\mathfrak{M}^s$ as follows:

- $M^s = M \cup \{s\}$.
- $R^s = R \cup \{(x, s) \mid x \in M\}$.
- $V^s = V \cup \{(i, \{s\})\}$.

Note that s is a sinkpoint: instead of using s to spy on all the points in the model, we squeeze all the points we need between w and s . It is easy to show that $\mathfrak{M}^s, w \Vdash \varphi^t \wedge \text{Until}^+(i, \varphi^t) \wedge F(i \wedge \neg Fi)$.

$\Leftarrow$. Let $\mathfrak{M} = \langle M, R, V \rangle, w \Vdash \varphi^t \wedge \text{Until}^+(i, \varphi^t) \wedge F(i \wedge \neg Fi)$. Let $V(i) = \{s\}$. Define $\mathfrak{M}^\nabla$ as

- $M^\nabla = \{x \in M \mid x = w \text{ or } wR^+ xRs\}$.
- $R^\nabla = R \upharpoonright_{M^\nabla}$.
- $V^\nabla = V \upharpoonright_{M^\nabla}$.

Note that $w \in M^\nabla$. Obviously, for all $x \in M^\nabla$, $\mathfrak{M}^\nabla, x \Vdash \varphi$ if and only if $\mathfrak{M}, x \Vdash \varphi^t$. Thus $\mathfrak{M}^\nabla \Vdash \varphi$, because for all $x \in M^\nabla$, $\mathfrak{M}, x \Vdash \varphi^t$. $\blacktriangleleft$

The theorem follows directly from the claim. $\blacksquare$

For both, Theorem 4.1 and Theorem 4.2 matching EXPTIME upper bounds can be obtained by embedding into the loosely μ -guarded fragment [GW99], but we omit details.

Using ideas from Section 3.3, we can point to a subset of the language with an NP-complete satisfaction problem. Consider the formulas of the Until/Since language equivalent to formulas of the form $\text{Since}(i, \psi)$ or $\text{Until}(i, \psi)$. It is immediate that, in any structure, these formulas are equivalent to $Pi \wedge H(Pi \rightarrow \psi)$ and $Fi \wedge G(Fi \rightarrow \psi)$, respectively. We can use then Theorem 3.7 to prove that the complexity on linear frames drops from PSPACE-complete for the full language (see [SC85]) to NP-complete for the fragment. By Theorem 3.8 a similar drop in complexity occurs on the class of transitive trees.

5 Referential interval logic

We will now examine what happens when we add nominals and $@$ to the interval-based logic of Halpern and Shoham (see [HS86]). This logic arises by abstracting from James Allen’s system of temporal knowledge representation (see [All84]). As with our discussion of nominal tense logic over strict total orders, the complexity result we shall prove hinges on the observation that Halpern and Shoham’s language is powerful enough to define the difference operator D. However we are going to work with a more interesting type of hybrid language, namely one containing two distinct sorts of nominals. That is, we will be working in a three-sorted modal language.

The language of Halpern and Shoham builds formulas from a collection of atomic symbols ATOM using boolean operators and the primitive modalities $\langle A \rangle$, $\langle \bar{A} \rangle$, $\langle B \rangle$, $\langle \bar{B} \rangle$, $\langle E \rangle$ and $\langle \bar{E} \rangle$. As it is a traditional (unsorted) modal language, ATOM contains only the symbols in PROP, that is, propositional variables p, q, r , and so on. The language is interpreted on interval structures built over strict total orders. Let $\langle T, < \rangle$ be a strict total order and let $I(\langle T, < \rangle)$ be the set of all closed intervals $[t, t'] = \{s \in T : t \leq s \leq t'\}$ on $\langle T, < \rangle$. A Kripke model $\mathfrak{M}$ is a triple $\langle T, <, V \rangle$. Here V is a valuation, that is, a mapping $V : \text{ATOM} \rightarrow \text{Pow}(I(\langle T, < \rangle))$. The satisfaction definition is:

$\mathfrak{M}, [t_1, t_2] \Vdash a$	iff	$[t_1, t_2] \in V(a)$, for all atoms a
$\mathfrak{M}, [t_1, t_2] \Vdash \neg\varphi$	iff	$\mathfrak{M}, [t_1, t_2] \nVdash \varphi$
$\mathfrak{M}, [t_1, t_2] \Vdash \varphi \wedge \psi$	iff	$\mathfrak{M}, [t_1, t_2] \Vdash \varphi$ and $\mathfrak{M}, [t_1, t_2] \Vdash \psi$
$\mathfrak{M}, [t_1, t_2] \Vdash \langle A \rangle\varphi$	iff	$\exists t_3 : t_2 < t_3$ and $\mathfrak{M}, [t_2, t_3] \Vdash \varphi$
$\mathfrak{M}, [t_1, t_2] \Vdash \langle \bar{A} \rangle\varphi$	iff	$\exists t_3 : t_3 < t_1$ and $\mathfrak{M}, [t_3, t_1] \Vdash \varphi$
$\mathfrak{M}, [t_1, t_2] \Vdash \langle B \rangle\varphi$	iff	$\exists t_3 : t_3 < t_2$ and $t_1 \leq t_3$ and $\mathfrak{M}, [t_1, t_3] \Vdash \varphi$
$\mathfrak{M}, [t_1, t_2] \Vdash \langle \bar{B} \rangle\varphi$	iff	$\exists t_3 : t_2 < t_3$ and $\mathfrak{M}, [t_1, t_3] \Vdash \varphi$
$\mathfrak{M}, [t_1, t_2] \Vdash \langle E \rangle\varphi$	iff	$\exists t_3 : t_1 < t_3$ and $t_3 \leq t_2$ and $\mathfrak{M}, [t_3, t_2] \Vdash \varphi$
$\mathfrak{M}, [t_1, t_2] \Vdash \langle \bar{E} \rangle\varphi$	iff	$\exists t_3 : t_3 < t_1$ and $\mathfrak{M}, [t_3, t_2] \Vdash \varphi$.

Let's now sort this language to make it referential. We could introduce a “generic” nominal naming arbitrary intervals — but to make things more interesting, let's take a more fine grained approach. We will add two different sorts of nominals: *stretched interval nominals*, and *point interval nominals*. Stretched interval nominals will be used to name extended time periods, and point interval nominals will denote point-like intervals. So, let SNOM and PNOM be denumerably infinite sets that are mutually disjoint, and disjoint from PROP. We represent the elements of SNOM as e, d, c , and those of PNOM by i, j, k , and we redefine ATOM to be $\text{PROP} \cup \text{SNOM} \cup \text{PNOM}$. The only other change we will make is to add @; either type of nominal can appear as a subscript.

Let $S(\langle T, < \rangle) = \{[t, t'] \in I(\langle T, < \rangle) \mid t \neq t'\}$ be the set of stretched intervals. Let $P(\langle T, < \rangle) = \{[t, t'] \in I(\langle T, < \rangle) \mid t = t'\}$ be the set of point intervals. We now define a *hybrid valuation* to be a map $V : \text{ATOM} \rightarrow \text{Pow}(I(\langle T, < \rangle))$ assigning single subsets of $S(\langle T, < \rangle)$ to elements in SNOM and single subsets of $P(\langle T, < \rangle)$ to elements in PNOM. As usual, @ means “jump to the interval denoting the subscript and test for the truth of its argument there.” This three-sorted language can be thought of as the propositional fragment of James Allen's system of temporal knowledge representation [All84]. In particular, what Allen would write as $\text{Holds}(e, \varphi)$ corresponds to $@_e\varphi$ in the hybrid language.

We will make some further remarks on the connections with Allen's work shortly, but let's now look at the complexity of this system. In fact, sorting and adding @ has not made the satisfiability problem any worse. As we already said, the key point is that D is definable:

$$D\varphi := \bigvee_{\pi \in \{A, \bar{A}, B, \bar{B}, E, \bar{E}, L, \bar{L}, D, \bar{D}, O, \bar{O}\}} \langle \pi \rangle\varphi.$$

This definition makes use of the following defined operators:

$$\begin{aligned} \langle L \rangle\varphi &:= \langle A \rangle\langle A \rangle\varphi, & \langle D \rangle\varphi &:= \langle B \rangle\langle E \rangle\varphi, & \langle O \rangle\varphi &:= \langle E \rangle\langle \bar{B} \rangle\varphi, \\ \langle \bar{L} \rangle\varphi &:= \langle \bar{A} \rangle\langle \bar{A} \rangle\varphi, & \langle \bar{D} \rangle\varphi &:= \langle \bar{B} \rangle\langle \bar{E} \rangle\varphi, & \langle \bar{O} \rangle\varphi &:= \langle \bar{B} \rangle\langle \bar{E} \rangle\varphi. \end{aligned}$$

If you spell out what each of the disjunctions in the definition of D say, you will see that between them they cover 12 of the 13 possible relations between intervals in an interval structure $I(\langle T, < \rangle)$. The missing relation is equality, and if we add a disjunction for this we obtain E. That is, $E\varphi := \varphi \vee D\varphi$.

Now, if we had simply introduced a generic interval nominal, we could use D to simulate nominals and @ exactly as we did in our discussion of nominal tense logic over linear frames. But we have to draw a distinction between stretched and point nominals. We do so as follows. Observe that $\langle B \rangle\top$ is true on precisely the elements of $S(\langle T, < \rangle)$, while $\langle B \rangle\top$ is false on precisely the elements of $P(\langle T, < \rangle)$. So the formula $E(p \wedge \langle B \rangle\top) \wedge A(p \rightarrow \neg Dp)$ forces the propositional variable p to act like a stretched interval nominal, and $E(p \wedge \neg \langle B \rangle\top) \wedge A(p \rightarrow \neg Dp)$ forces it to act like a point interval nominal. Then, as we discussed in Section 3.2, by substituting ordinary propositional symbols for nominals, and using these definitions to make them behave appropriately, we can show that:

THEOREM 5.1

Let S be a subclass of the class of strict total orders. The complexity of the satisfiability problem over the interval structures built over the frames in S is the same for the three-sorted interval-based logic (with @) as it is for the logic of Halpern and Shoham.

But there is a difference between this result and our earlier result for nominal tense logic over linear frames: the satisfiability problem for Halpern and Shoham's logic is typically much harder. For example, as Halpern and Shoham show, validity is not recursively enumerable if we build our intervals over $\langle \mathbb{R}, < \rangle$, the real numbers in their usual order.

In our view, this suggests two main paths for further work. The first is to look at the complexity of weaker hybrid interval logics in which nominals and @ are added as *primitives*. The most striking feature of Allen's work is his use of Holds (without Holds, Allen's system is a fairly orthodox first-order theory of intervals). Moreover, the problem of solving constraints on temporal reference is one of the key tasks in temporal knowledge representation. But the system of Halpern and Shoham isn't built on the idea of reference — temporal reference is a side-effect of the definability of D. And this means that once we start dropping modalities in the hope of finding better behaved subfragments, we lose the ability to refer. It seems to us that it is much more in the spirit of Allen's work to add point interval nominals, stretched interval nominals, and @, right from the start, and then to see what can be achieved using a more restricted collection of operators (for example, the four interval operators discussed in [Ben83]).

A second approach is to *increase* the expressivity still further. In particular, it seems interesting to go to a richer hybrid language which allows explicit quantification over nominals. If this is done, the resulting system perspicuously captures the full version of Allen's system (Allen allows explicit quantification over interval terms). Hybrid languages that allow explicit quantification over nominals (and indeed, other sorts)

have long been studied; see in particular [Bul70] and [PT91]. Given that the logic of Halpern and Shoham is undecidable (or even highly undecidable) for many important classes of interval structure, we have nothing to lose on complexity theoretic grounds by adding explicit quantification over nominals. And we do have something to gain, namely expressivity: as is shown in [Ven90], the logic of Halpern and Shoham is not expressively complete over the class of dense linear orders, and in fact, neither is any interval logic based on a finite collection of interval tense operators.

6 Concluding remarks

In this paper we investigated the complexity of the satisfiability problem for a number of hybrid temporal logics. Our main results dealt with nominal tense logic, that is, Priorean tense logic enriched with nominals and possibly @, E, and D as well. The following table summarizes our results here:

Class of frames	[F, P]	[F, P, @, Nom]	[F, P, E], [F, P, E, @]	[F, P, E, Nom], [F, P, D]
All frames	PSPACE	EXPTIME	EXPTIME	EXPTIME
Transitive	PSPACE	EXPTIME	EXPTIME	EXPTIME
Strict Total Orders	NP	NP	NP	NP
Transitive trees	PSPACE	PSPACE	PSPACE	PSPACE

Some comments. Hybrid temporal logic has an EXPTIME-hard satisfiability problem over both arbitrary frames (Theorem 3.2) and transitive frames (Theorem 3.4). These results hold even without @, and only one nominal is needed to establish them. Matching EXPTIME upper bounds hold for nominal tense logic, over both arbitrary and transitive frames, even if E is added (Corollary 3.6). Over any class of strict total orders, the satisfiability problem for nominal tense logic is the same as that of Priorean tense logic (Theorem 3.7); for example, for the class of all strict total orders, nominal tense logic with E has an NP-complete satisfiability problem. Over trees, nominal tense logic (both with and without @) is PSPACE-complete (Theorem 3.9), even when extended with E or D (Corollary 3.10).

With respect to pure-future fragments, Theorem 3.1 and Theorem 3.3 show that hybridization does not damage the fundamental complexity results for modal logics.

We turned then to languages more expressive than tense logic. We obtained two EXPTIME characterizations for Until logics and briefly discussed a hybrid version of Halpern and Shoham interval logics. One general comment. It is striking how little is known about complexity results for these languages. Results from computer science for language containing Until typically deal with quite restricted (usually tree-like) classes of frames. As for interval logic, [Ben83] discusses a wide range of possible constraints on interval structure, but it seems that few of these have been investigated from a complexity theoretic perspective. Given the high expressive power of these logics, many decidable logics in this family will probably turn to be EXPTIME; still, a finer analysis of the trade off between complexity and expressive power in this area would be extremely useful.

To conclude the paper we note some consequences of our results for description logic. We have already mentioned in Section 1 that the description language $\mathcal{ALC}$ (without axioms) is a notational variant of multi-modal logic. Assertional axioms over $\mathcal{ALC}$ (that is, the formulas listed in the A-Box) can be formulated in a restricted fragment of multi-modal logic enriched with nominals and @, namely the fragment in

which nominals cannot be freely used in formulas, but can only occur as indices on the @ operator. The description logic $\mathcal{ALCO}$ moves closer towards full multi-modal logic enriched with nominals and @. $\mathcal{ALCO}$ allows the formation of concepts using the $\mathcal{O}$ (“one-of”) operator: an expression of the form $\mathcal{O}(i, \dots, k)$ corresponds to a disjunction of nominals $i \vee \dots \vee k$, and an expression of the form $\mathcal{O}(i)$ to the nominal i (see [BT98]). Thus the full language of multi-modal logic enriched with nominals and @ is essentially a version of $\mathcal{ALCO}$ in which no restrictions are made on the use of assertional statements.

Now, as was observed in [ABM99], Theorem 3.1 extends straightforwardly to the multi-modal case. Viewed from a description logic perspective, this tells us that dropping all restrictions on the use of A-box descriptions does not move us out of PSPACE (for the concept satisfiability problem). That is, we can work in a “one-level” language (let’s call it $\mathcal{ALCO@}$) in which A-box descriptions can be freely combined with the usual description logic machinery and $\mathcal{O}$, and remain in PSPACE.³

Compare this with what happens if we add converse roles. If we start with a language of $\mathcal{ALCO}$ that contains only one role symbol R we have a PSPACE-complete satisfiability problem, and this complexity is not modified no matter how many further ordinary role symbols we add. But Theorem 3.2 tells us that if we add the *converse* role symbol R^{-1} we immediately jump to EXPTIME.

One final remark: the spypoint argument used throughout the paper seems of interest in its own right. We have seen that the technique is useful in hybrid logic for establishing both upper and lower bounds, and we think it may be useful in description (and feature) logic too.

References

- [ABM99] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, number 1683 in LNCS, pages 307–321. Springer, 1999.
- [AdR99] C. Areces and M. de Rijke. Accounting for assertional information. Manuscript, 1999.
- [All84] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, July 1984.
- [AvBN98] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *J. of Philosophical Logic*, 27(3):217–274, 1998.
- [Ben83] J. van Benthem. *The logic of time*. Reidel, 1983.
- [Bla93] P. Blackburn. Nominal tense logic. *Notre Dame J. of Formal Logic*, 14:56–83, 1993.
- [Bla94] P. Blackburn. Tense, temporal reference, and tense logic. *J. of Semantics*, 11:83–101, 1994.
- [Bla98] P. Blackburn. Internalizing labeled deduction. Technical Report CLAUS-Report 102, Computerlinguistik, Universität des Saarlandes, 1998. <http://www.coli.uni-sb.de/cl/clus>.
- [BMV97] D. Basin, S. Matthews, and L. Viganò. Labelled propositional modal logics: theory and practice. *J. of Logic and Computation*, 7:685–717, 1997.
- [BS93] P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. *J. of Logic, Language and Information*, pages 129–169, 1993.
- [BS95] P. Blackburn and J. Seligman. Hybrid languages. *J. of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.
- [BT98] P. Blackburn and M. Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24:23–49, 1998.
- [BT99] P. Blackburn and M. Tzakova. Hybrid languages and temporal logics. *Logic J. of the IGPL*, 7(1):27–54, 1999.

³However the extension to $\mathcal{ALCO@}$ would have an impact on the optimization techniques for tableaux calculi developed in the description logic community.

- [Bul70] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.
- [Chl86] B. Chlebus. Domino-tiling games. *J. Comput. System Sci.*, 32(3):374–392, 1986.
- [De 95] G. De Giacomo. *Decidability of class-based knowledge representation formalisms*. PhD thesis, Università di Roma “La Sapienza”, 1995.
- [Dem99] S. Demri. Sequent calculi for nominal tense logics: a step towards mechanization? In N. Murray, editor, *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 140–154. Springer Verlag, 1999.
- [DG99] S. Demri and R. Goré. Cut-free display calculi for nominal tense logics. In N. Murray, editor, *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 155–170. Springer Verlag, 1999.
- [FL79] M. Fisher and R. Ladner. Propositional dynamic logic of regular programs. *J. of Computer and System Sciences*, 18(2):194–211, 1979.
- [Gab96] D. Gabbay. *Labelled Deductive Systems*. Clarendon Press, Oxford, 1996.
- [Gor96] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *J. of Logic, Language and Information*, 5(1):1–24, 1996.
- [GW99] E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *Proc. of 14th IEEE Symposium on Logic in Computer Science LICS’99, Trento*, 1999.
- [Has91] P. Hasle. Building a temporal logic for natural language understanding with the HOL-system. In C. Brown and G. Koch, editors, *Natural Language Understanding and Logic Programming, III*, pages 91–109. Elsevier Science Publishers, 1991.
- [HM92] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
- [HS86] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. In *Proc. of the First IEEE Symposium on Logic in Computer Science, Cambridge, Massachusetts*, pages 279–292. Computer Society Press, Washington, 1986.
- [Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. of Computing*, 6(3):467–480, 1977.
- [ON80] H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, 34:325–333, 1980.
- [Pri67] A. Prior. *Past, Present and Future*. Oxford University Press, 1967.
- [PT91] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93:263–332, 1991.
- [Rea94] M. Reape. A feature value logic. In C. Rupp, M. Rosner, and R. Johnson, editors, *Constraints, Language and Computation*, Synthese Language Library, pages 77–110. Academic Press, 1994.
- [Rij92] M. de Rijke. The modal logic of inequality. *J. of Symbolic Logic*, 57:566–584, 1992.
- [SC85] A. Sistla and E. Clarke. Complexity of propositional linear temporal logics. *J. of the ACM*, 32(3):733–749, 1985.
- [Sch91] K. Schild. A correspondence theory for terminological logics. In *Proc. of the 12th IJCAI*, pages 466–471, 1991.
- [Sel97] J. Seligman. The logic of correct description. In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.
- [Spa93a] E. Spaan. *Complexity of modal logics*. PhD thesis, ILLC. University of Amsterdam, 1993.
- [Spa93b] E. Spaan. The complexity of propositional tense logics. In *Diamonds and defaults (Amsterdam, 1990/1991)*, pages 287–307. Kluwer Acad. Publ., Dordrecht, 1993.
- [SS91] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.
- [Tza99] M. Tzakova. Tableaux calculi for hybrid logics. In N. Murray, editor, *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 278–292. Springer Verlag, 1999.
- [Ven90] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame J. Formal Logic*, 31(4):529–547, 1990.
- [Ven93] Y. Venema. Derivation rules as anti-axioms in modal logic. *J. of Symbolic Logic*, 58:1003–1034, 1993.

Model Checking Hybrid Logics (With an Application to Semistructured Data)

Massimo Franceschet*

Department of Sciences, University of Chieti-Pescara, Italy
Informatics Institute, University of Amsterdam, The Netherlands
E-mail: francesc@science.uva.nl

Maarten de Rijke

Informatics Institute, University of Amsterdam, The Netherlands
E-mail: mdr@science.uva.nl

May 15, 2004

Abstract

We investigate the complexity of the model checking problem for hybrid logics. We provide model checker algorithms for various hybrid fragments and we prove PSPACE-completeness for hybrid fragments including binders. We complement and motivate our complexity results with an application of model checking in hybrid logic to the problems of query and constraint evaluation for semistructured data.

1 Introduction

In *model checking* [21] we are given a formal model and a property and we have to check whether the model satisfies the property. The model is a labelled graph, sometimes called Kripke structure, and the property is a formula represented as a formula written in some logical language. We search the graph in order to *check* whether the formula is true in the *model*. As a technology, model checking has very strong links to (at least) two areas in computer science: verification and databases. In the first half of this paper, we focus on model checking algorithms for so-called hybrid logics; in the second half, we go on to show their relevance for reasoning about semistructured data.

Modal and temporal logics have been successfully used as specification languages in the model checking task [22]; they are algorithmically well-behaved and mathematically natural fragments of classical logics. However, something crucial is missing in propositional modal and temporal logics: they lack mechanisms for naming states, for accessing states by names, and for dynamically creating new names for states. In particular, traditional modal and temporal logics are able to express properties that satisfy the *tree model property*, that is, properties that are satisfiable if, and only if, they are satisfiable in a tree-like model. Are there extensions of modal and temporal logics violating the tree model property that are still computationally tractable? This is where *hybrid logics* come in. They allow one to refer to states in a truly modal framework, and, hence, they mix features from first-order logic and modal logic, whence the name *hybrid logic* [16]. In addition to ordinary propositional variables, hybrid languages provide a type of atomic formulas called *nominals*. Syntactically, nominals behave like ordinary propositional variables, but they have an important semantic property: nominals are true at exactly one state in any model.

Nominals are only the first ingredient that sets hybrid languages apart from traditional modal-like languages. Hybrid languages may also contain the *at operator* $@_i$ which gives *random access* to

*Corresponding author.

Language	Complexity
HL($@, \mathbf{F}, \mathbf{P}, \mathbf{A}$)	$\mathbf{O}(k \cdot (n + m))$
HL($@, \mathbf{U}, \mathbf{S}, \mathbf{A}$)	$\mathbf{O}(k \cdot n \cdot m)$
HDL($@, \mathbf{A}$)	$\mathbf{O}(k \cdot (n + m))$
HL($\downarrow, @$)	$\mathbf{O}(k \cdot n)$
HL($\downarrow, \mathbf{F}$)	PSPACE-complete
HL($\downarrow, \mathbf{A}$)	PSPACE-complete
HL($\downarrow, @, \mathbf{F}, \mathbf{A}$)	PSPACE-complete
HL _r ($\downarrow, @, \mathbf{F}, \mathbf{A}$)	$\mathbf{O}(k \cdot (n + m) \cdot n^r)$
HDL _r ($\downarrow, @, \mathbf{F}, \mathbf{A}$)	$\mathbf{O}(k \cdot (n + m) \cdot n^r)$
HL($\exists$)	PSPACE-complete
HL($\exists, @, \mathbf{F}, \mathbf{A}$)	PSPACE-complete
HL _r ($\exists, @, \mathbf{F}, \mathbf{A}$)	$\mathbf{O}(k \cdot (n + m) \cdot n^{2r})$
HDL _r ($\exists, @, \mathbf{F}, \mathbf{A}$)	$\mathbf{O}(k \cdot (n + m) \cdot n^{2r})$

Table 1: Complexity of model checking for hybrid logics.

the unique state named by i : $@_i p$ holds if, and only if, p holds at the state named by i . Moreover, hybrid languages may be extended with the *downarrow binder* $\downarrow x$ that assigns the variable name x to the current state of evaluation. The operator $@$ combines naturally with $\downarrow$: $\downarrow \text{ stores the current state of evaluation and } @ \text{ enables us to retrieve the information stored by shifting the point of evaluation in the model. While } \downarrow x \text{ stores the current state in } x, \text{ the binder } \downarrow x \text{ stores the 'label' of the current state in } x, \text{ that is, the set of propositions holding at the current state. Finally, the 'existential binder' } \exists x \text{ binds the variable name } x \text{ to a state in the model. As we will see in Section 5, the hybrid machinery has very natural uses in talking about data models for semi-structured data.}$

Model checking for hybrid languages has hardly been explored so far. In this paper we address this gap. Our approach is incremental: on top of well-known model checking results for Propositional Temporal Logic and Converse Propositional Dynamic Logic, we investigate the model checking problem for these languages extended with the hybrid machinery as well as with the universal modality $\mathbf{A}$. It turns out that the addition of nominals, the $@$ operator, and the universal modality $\mathbf{A}$ does not increase the complexity of the model checker. In contrast, an arbitrary use of hybrid binders in formulas is computationally dangerous. The model checking problem for any hybrid logic that freely mixes the hybrid binder $\downarrow$ or $\downarrow$ with temporal operators is PSPACE-complete, while $\exists$ is hard even without temporal operators. However, the model checker runs in exponential time with respect to the *nesting degree* of the binders in the formula. This means that we can still check in polynomial time long formulas, as soon as the nesting degree of the hybrid binders on the formula is bound. We summarize our complexity results in Table 1, where k is the length of the formula, n and m are the number of nodes and the number of edges of the graph structure, respectively, and r is the nesting degree of hybrid binders. Moreover, $\mathbf{F}$ is the Future temporal operator, $\mathbf{P}$ is Past, $\mathbf{U}$ is Until and $\mathbf{S}$ is Since. Finally, languages of the form HL($\cdot$) are hybrid extensions of Propositional Temporal Logic, while languages of the form HDL($\cdot$) are hybrid extensions of Converse Propositional Dynamic Logic.

In the second part of the paper we illustrate a general methodological point: since hybrid languages provide very natural modeling facilities, understanding the computational and algorithmic properties of hybrid is of great potential value. That is, while the complexity-theoretic results obtained in this paper are valuable results about hybrid logic in their own right, we believe they get additional value because of the fact that hybrid languages provide such natural modeling facilities, which makes the formal results of this paper fairly directly applicable. To back up these claims we apply model checking for hybrid logics to the problems of query and constraint evaluation for *semistructured data*: data with some structure but without a regular schema. We discuss a hierarchy of query languages for semistructured data corresponding to fragments of the

language Lorel [3], the query language in the Lore system [34], which was designed for managing semistructured data. In particular, the languages that we discuss offer regular expressions to navigate the query graph at arbitrary depths, as well as the possibility of comparing object identities and object values. We embed those query languages into different fragments of hybrid logics with different expressivity and establish a close relationship between the query processing problem for semistructured data and the global model checking problem for hybrid logics. Moreover, we describe languages to specify path constraints for semistructured data, including inclusion, inverse and functional path constraints. Path constraints are the generalization of relational integrity constraints for semistructured databases and they are useful to provide a loose schema to the otherwise unstructured database. Once again, we provide an embedding into hybrid logic, this time of the constraint language, and we underline the close connection between path constraint evaluation for semistructured data and model checking for hybrid logics.

The paper is organized as follows. In Section 2 we discuss previous work that is related to the current investigations. Section 3 introduces the basics of hybrid logics. In Section 4.1 we provide model checkers for different hybrid languages and we analyze their computational complexity, while in Section 4.2 we prove PSPACE-completeness of the model checking problem for hybrid fragments allowing binders. Readers mostly interested in the relation between model checking hybrid logic and semistructured data, can skip Section 4 and occasionally return to it for details of results used in Section 5, where we describe the application of model checking hybrid logic to semistructured data. We conclude the paper and outline future work in Section 6.

2 Related Work

The Hybrid Logic Manifesto [14] is probably the best entry point to the hybrid logic world. See [9] for a gentle introduction to the topic and [15, Chapter 7], for an introduction at the textbook level. Characterizations with respect to first-order correspondence theory, interpolation property and computational complexity (of the satisfiability problem) for hybrid modal logics have been studied in [12]. The complexity of the satisfiability problem for hybrid temporal logics with respect to different classes of frames (all frames, transitive frames, linear frames, transitive trees) has been investigated in [10, 11, 27]. As for implementations, a resolution-based theorem prover for hybrid logic with @ and ↓ has been implemented [13]. For a comprehensive entry point to the field see the hybrid logic home page [32].

Since the mid-1990s there has been a lot of work on the interface of computational logic and semistructured data, making use of a wide variety of logical tools and techniques. For instance, [18] uses simulations and morphisms, [4] concentrate on regular expressions and [20] make the connection with description logic. The relation between model checking and query processing has been extensively explored in the setting of *structured* data. We refer to [30] as a good entry-point to the area. The relation between model checking and query processing for *semistructured* data goes back at least to [6], where it was formulated in terms of suitable modal-like logics. Quaintarelli [37] embeds a fragment of the graphical query language G-Log into CTL, and she sketches a mapping for subsets of other semistructured query languages, like Lorel, GraphLog and UnQL. It is worth noticing that the fragments considered in [37] do not allow queries with joins. De Alfaro [8] proposes the use of model checking for detecting errors in the structure and connectivity of web pages. Finally, Miklau and Sucu [35] and Gottlob et al. [29] sketch an embedding of the forward looking fragment of XPath into CTL. As for the relation between model checking and path constraints evaluation for semistructured data, Alechina et al. [7] embed forward and backward path constraints into Converse Propositional Dynamic Logic. Calvanese et al. [20] use description logics for similar purposes, and Afanasiev et al. [5] turn to CTL and provide experimental results of the “query evaluation as model checking” perspective.

3 Hybrid Logics

Temporal logics [25] (TL, for short) may be viewed as fragments of classical logics [15]. It extends propositional logic by adding the well-known temporal operators future **F**, past **P**, until **U** and since **S**. Let $\text{PROP} = \{p, q, \dots\}$ be a set of propositional variables. The syntax of temporal logic is as follows:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \mathbf{F}\phi \mid \mathbf{P}\phi \mid \phi \mathbf{U}\phi \mid \phi \mathbf{S}\phi.$$

Shorthands are $\perp = \neg\top$, $\alpha \vee \beta = \neg(\neg\alpha \wedge \neg\beta)$, $\alpha \rightarrow \beta = \neg\alpha \vee \beta$, and $\alpha \leftrightarrow \beta = (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. The dual of **P** is $\mathbf{H}\alpha = \neg\mathbf{P}\neg\alpha$, and the dual of **F** is $\mathbf{G}\alpha = \neg\mathbf{F}\neg\alpha$.

Temporal logic is interpreted over *Kripke structures* of the form $\langle M, R, V \rangle$, where M is a set of states (or worlds, points, nodes), R is a binary relation on M called the accessibility (or reachability) relation, and V is a valuation function from PROP to the powerset of M . Let $\mathcal{M}$ be a Kripke structure and $m \in M$. The semantics of temporal logic is as follows:

$$\begin{aligned} \mathcal{M}, m &\models \top && \text{iff } m \in V(p), p \in \text{PROP} \\ \mathcal{M}, m &\models p && \text{iff } \mathcal{M}, m \models \phi \\ \mathcal{M}, m &\models \neg\phi && \text{iff } \mathcal{M}, m \not\models \phi \\ \mathcal{M}, m &\models \phi \wedge \psi && \text{iff } \mathcal{M}, m \models \phi \text{ and } \mathcal{M}, m \models \psi \\ \mathcal{M}, m &\models \mathbf{F}\phi && \text{iff } \exists m' (Rmm' \wedge \mathcal{M}, m' \models \phi) \\ \mathcal{M}, m &\models \mathbf{P}\phi && \text{iff } \exists m' (Rm'm \wedge \mathcal{M}, m' \models \phi) \\ \mathcal{M}, m &\models \psi \mathbf{U}\phi && \text{iff } \exists m' (Rmm' \wedge \mathcal{M}, m' \models \phi \wedge \\ &&& \forall m'' (Rmm'' \wedge Rm'm'' \rightarrow \mathcal{M}, m'' \models \psi)) \\ \mathcal{M}, m &\models \psi \mathbf{S}\phi && \text{iff } \exists m' (Rm'm \wedge \mathcal{M}, m' \models \phi \wedge \\ &&& \forall m'' (Rm'm'' \wedge Rm'm'' \rightarrow \mathcal{M}, m'' \models \psi)) \end{aligned}$$

Sometimes, TL also includes the transitive closure operators $\mathbf{F}^+$, $\mathbf{P}^+$, $\mathbf{U}^+$ and $\mathbf{S}^+$, interpreted over the transitive closure R^+ of the accessibility relation, as well as the universal modality **A**. The semantics of the universal modality is as follows: $\mathcal{M}, m \models \mathbf{A}\phi$ iff for all m it holds that $\mathcal{M}, m \models \phi$. The dual of the universal modality is the existential modality $\mathbf{E}\alpha = \neg\mathbf{A}\neg\alpha$.

Hybrid logic (HL, for short) extends temporal logic by introducing devices for naming states and for accessing states by names. Let $\text{NOM} = \{i, j, \dots\}$ be a set of nominals and $\text{VVAR} = \{x, y, \dots\}$ a set of state variables. The syntax of hybrid logic is as follows:

$$\phi ::= \text{TL} \mid i \mid x \mid @_i\phi \mid \downarrow x\phi \mid \exists x\phi,$$

with $i \in \text{NOM}$, $x \in \text{VVAR}$, $t \in \text{NOM} \cup \text{VVAR}$. The operators in $\{\@_i, \downarrow, \exists\}$ are called *hybrid operators*. We call $\text{WSYM} = \text{NOM} \cup \text{VVAR}$ the set of *state symbols*, $\text{ALET} = \text{PROP} \cup \text{NOM}$ the set of *atomic letters*, and $\text{ATOM} = \text{PROP} \cup \text{NOM} \cup \text{VVAR}$ the set of *atoms*. We use $x = y$ for $@_x y$ and $x \neq y$ for $@_x \neg y$.

Hybrid logic is interpreted over *hybrid Kripke structures*, that is, Kripke structures $\langle M, R, V \rangle$ where the valuation function V is such that $V(i)$ is a singleton subset of M for all nominals $i \in \text{NOM}$. To give meaning to the formulas, we also need the notion of *assignment*. An assignment g is a mapping $g : \text{VVAR} \rightarrow M$. Given an assignment g , we define g_m^x by $g_m^x(x) = m$ and $g_m^y(y) = g(y)$ for $x \neq y$. For any atom a , let $[V, g](a) = \{g(a)\}$ if a is a state variable, and $V(a)$ otherwise.

The semantics of hybrid logic is as follows (the semantics for Boolean and temporal operators is as for temporal logic). Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid Kripke structure, $m \in M$, and g an assignment. Then,

$$\begin{aligned} \mathcal{M}, g, m &\models a && \text{iff } m \in [V, g](a), a \in \text{ATOM} \\ \mathcal{M}, g, m &\models @_i\phi && \text{iff } \mathcal{M}, g, m' \models \phi, \text{ where } [V, g](i) = \{m'\}, t \in \text{WSYM} \\ \mathcal{M}, g, m &\models \downarrow x\phi && \text{iff } \mathcal{M}, g_m^x, m \models \phi \\ \mathcal{M}, g, m &\models \exists x\phi && \text{iff there is } m' \in M \text{ such that } \mathcal{M}, g_m^{x'}, m \models \phi \end{aligned}$$

In words, the @ operator $@_i$ shifts evaluation to the state named by i , where t is a nominal or a variable. The downward binder $\downarrow x$ binds the state variable x to the *current* state (where evaluation is being performed), while the existential binder $\exists x$ binds the state variable x to *some* state

in the model; $\downarrow$ and $\exists$ do not shift evaluation away from the current state. We use $\text{HL}(O_1, \dots, O_n)$ to denote the hybrid language with hybrid and temporal operators $O_1, \dots, O_n$.

It is worth noticing that the until operator $\mathbf{U}$ can be written in hybrid logic as follows:

$$\alpha \mathbf{U} \beta = \downarrow x. \mathbf{F}(\beta \wedge \mathbf{H}(\mathbf{P}x \rightarrow \alpha)),$$

and the since operator $\mathbf{S}$ as follows:

$$\alpha \mathbf{S} \beta = \downarrow x. \mathbf{P}(\beta \wedge \mathbf{G}(\mathbf{F}x \rightarrow \alpha)).$$

Moreover, the past operator is $\mathbf{P}\alpha = \downarrow x. \exists y. @_y(\mathbf{F}x \wedge \alpha)$. Finally, the downarrow binder $\downarrow$ is a particular case of the existential binder: $\downarrow x. \alpha = \exists x. (x \wedge \alpha)$, while $\exists$ can be simulated by $\downarrow$ and $\mathbf{E}$ as follows: $\exists x. \alpha = \downarrow y. \mathbf{E}[x. \mathbf{E}(y \wedge \alpha)]$.

In addition to the hybrid languages listed so far, we will also discuss hybrid *dynamic* languages; these will prove to be especially useful in Section 5. Specifically, we consider a hybridization of converse propositional dynamic logic (CPDL) [31]. CPDL adds two operators to propositional logic, namely $\langle e \rangle \alpha$ and $\langle e \rangle^{-1} \alpha$, where e is a regular expression on a set of labels Σ and α is a CPDL formula.

CPDL is interpreted over *Labelled Transitions Systems* (LTSs), Kripke structures in which both the nodes and the edges are labelled. In particular, the edges are labelled with symbols in Σ . Each regular expression e on the set of edge labels Σ identifies a binary relation R_e on the set of states of the model. The relation R_e is recursively defined in terms of the structure of e . More precisely, R_e contains all the edges labelled with l , $R_{e_1 e_2} = R_{e_1} \circ R_{e_2}$, $R_{e_1 + e_2} = R_{e_1} \cup R_{e_2}$, and $R_{e^*} = (R_e)^*$. A state s is *reachable* from a state r through the regular expression e if $\langle r, s \rangle \in R_e$. Given a LTS $\mathcal{M}$ and a state s in $\mathcal{M}$, we have that $\langle e \rangle \alpha$ is true in $\mathcal{M}$ at s if there exists a state s' reachable from s through e such that α is true in $\mathcal{M}$ at s' . Moreover, $\langle e \rangle^{-1} \alpha$ is true in $\mathcal{M}$ at s if there exists a state s' such that s is reachable from s' through e and α is true in $\mathcal{M}$ at s' .

Hybrid dynamic logic is simply the *hybridization* of CPDL. We write $\text{HDL}(O_1, \dots, O_n)$ to denote the extension of CPDL with nominals, hybrid operators and possibly the universal modality in $O_1, \dots, O_n$. For instance, $\text{HDL}(@, \downarrow, \mathbf{A})$ is CPDL with nominals, $@$, $\downarrow$ and $\mathbf{A}$ operators.

We conclude this section by introducing a new hybrid binder $\Downarrow$. We have separated the introduction of this binder because usually it is not included in hybrid languages. However, $\Downarrow$ will come in handy in Section 5. Intuitively, while $\downarrow$ stores the current state into a variable, $\Downarrow$ stores the *label* of the current state, that is, the set of propositions that hold at the current state. Let $\text{WWAR}' = \{v, w, \dots\}$ be a set of variables such that WWAR and WWAR' are disjoint sets. The new variables in WWAR' serve as containers for sets of propositions. We add to the hybrid language the formulas $\Downarrow v. \alpha$, v , and $v = w$, and the new shortcut $v \neq w$ for $\neg(v = w)$, where $v, w \in \text{WWAR}'$, with the following meaning. Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid Kripke structure, $m \in M$, and g an assignment extended to WWAR' ; that is, g is a mapping that associates each variable in WWAR to a state in M and each variable in WWAR' to a subset of PROP . We denote by V^{-1} the function from M to the powerset of PROP such that $p \in V^{-1}(m)$ iff $m \in V(p)$. Then, $\mathcal{M}, g, m \models \Downarrow v. \alpha$ iff $\mathcal{M}, g[V^{-1}(m)/v] \models \alpha$. Moreover, $\mathcal{M}, g, m \models v$ iff $g(v) = V^{-1}(m)$ and $\mathcal{M}, g, m \models v = w$ iff $g(v) = g(w)$. In words, the binder $\Downarrow v$ binds the variable v to the label of the current state, while $v = w$ compares the labels stored in v and w .

4 Model Checking for Hybrid Logics

In this section, we investigate the global model checking problem for various hybrid languages. A hybrid Kripke structure $\mathcal{M} = \langle M, R, V \rangle$ is *finite* if M is finite. The *global model checking problem* for hybrid logic is as follows: Given a finite hybrid Kripke structure $\mathcal{M}$, an assignment g , and a hybrid formula ϕ , is there a state $m \in M$ such that $\mathcal{M}, g, m \models \phi$?

```

Procedure  $\text{MC}_{\mathbf{F}}(\mathcal{M}, g, \alpha)$ 
for  $w \in L(\alpha)$  do
  for  $v \in R^{-1}(w)$  do
     $L(\mathbf{F}\alpha, w) \leftarrow 1$ 
  end for
end for

Procedure  $\text{MC}_{\mathbf{A}}(\mathcal{M}, g, \alpha)$ 
if  $L(\alpha) = M$  then
  for  $v \in M$  do
     $L(\mathbf{A}\alpha, v) \leftarrow 1$ 
  end for
end if

Procedure  $\text{MC}_{\mathbf{U}}(\mathcal{M}, g, t, \alpha)$ 
let  $\{w\} = \{v, g(t)\}$ 
if  $L(\alpha, w) = 1$  then
  for  $v \in M$  do
     $L(@\alpha, v) \leftarrow 1$ 
  end for
end if

Procedure  $\text{MC}_{\mathbf{U}}(\mathcal{M}, g, \alpha, \beta)$ 
for  $w \in M$  do
   $\text{unmark}(w)$ 
end for
for  $w \in L(\beta)$  do
  for  $v \in R^{-1}(w)$  do
    if  $L(\alpha, w) = 0$  then
      for  $u \in R^{-1}(v)$  do
         $\text{mark}(u)$ 
      end for
    end if
  end for
  for  $v \in R^{-1}(w) \cup R^{-2}(w)$  do
    if  $\text{marked}(v)$  then
       $\text{unmark}(v)$ 
    else
       $L(\alpha \mathbf{U} \beta, v) \leftarrow 1$ 
    end if
  end for
end for

```

Figure 1: MCLITE subprocedures.

4.1 Model Checkers

In this section we provide global model checkers for different hybrid logics and we analyze their worst-case behavior. We start by describing a model checker called MCLITE for the language $\text{HL}(@, \mathbf{F}, \mathbf{P}, \mathbf{U}, \mathbf{S}, \mathbf{A})$. It receives a hybrid model $\mathcal{M} = \langle M, R, V \rangle$, an assignment g , and a hybrid formula ϕ in the considered language and, after termination, every state in the model is labelled with the subformulas of ϕ that hold at that state. The algorithm uses a *bottom-up strategy*: it examines the subformulas of ϕ in increasing order of length, until ϕ itself has been checked.

We need some auxiliary notation. Let R be an accessibility relation; then R^{-} is the inverse of R : $R^{-}vu$ if, and only if, Ruv . For $n \geq 1$, let $R^n(w)$ be the set of states that are reachable from w in n R -steps, and $R^{-n}(w)$ be the set of states that are reachable from w in n R^{-} -steps. The states belonging to $R^1(w)$ are called *successors* of w , while those belonging to $R^{-1}(w)$ are called *predecessors* of w . Given a model $\mathcal{M} = \langle M, R, V \rangle$, we denote by $\mathcal{M}^{-}$ the model $\langle M, R^{-}, V \rangle$. The *length* of a formula ϕ , denoted by $|\phi|$, is the number of operators (boolean, temporal and hybrid) of ϕ plus the number of atoms (propositions, nominals and variables) of ϕ . Let $\text{sub}(\phi)$ be the set of subformulas of ϕ . Notice that $|\text{sub}(\phi)| = |\phi|$.

The model checker MCLITE updates a table L of size $|\phi| \times |M|$ whose elements are bits. Initially, $L(\alpha, w) = 1$ if, and only if, α is an atomic letter in $\text{sub}(\phi)$ such that $w \in V(\alpha)$. When MCLITE terminates, $L(\alpha, w) = 1$ if, and only if, $\mathcal{M}, g, w \models \alpha$ for every $\alpha \in \text{sub}(\phi)$. Given $\alpha \in \text{sub}(\phi)$ and $w \in M$, we denote by $L(\alpha)$ the set of states $v \in M$ such that $L(\alpha, v) = 1$ and by $L(w)$ the set of formulas $\beta \in \text{sub}(\phi)$ such that $L(\beta, w) = 1$. MCLITE uses subroutines $\text{MC}_{\mathbf{F}}$, $\text{MC}_{\mathbf{U}}$, $\text{MC}_{\mathbf{A}}$, and $\text{MC}_{\mathbf{S}}$ in order to check subformulas of the form $\mathbf{F}\alpha$, $\alpha \mathbf{U} \beta$, $\mathbf{A}\alpha$, and $@\alpha$, respectively. The pseudocode for these procedures is in Figure 1.

Proposition 4.1 (Correctness of subroutines) Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid model, g an assignment, α and β hybrid formulas, and t a state symbol. Let L be a table such that, for every $w \in M$, $L(\alpha, w) = 1$ (respectively, $L(\beta, w) = 1$) if, and only if, $\mathcal{M}, w \models \alpha$ (respectively, $\mathcal{M}, w \models \beta$). Then,

1. after termination of $\text{MC}_{\mathbf{F}}(\mathcal{M}, g, \alpha)$, we have that, for every state $w \in M$, $L(\mathbf{F}\alpha, w) = 1$ if, and only if, $\mathcal{M}, g, w \models \mathbf{F}\alpha$;

2. after termination of $\text{MCA}_{\mathbf{A}}(\mathcal{M}, g, \alpha)$, we have that, for every state $w \in M$, $L(\mathbf{A}\alpha, w) = 1$ if, and only if, $M, g, w \models \mathbf{A}\alpha$;
3. after termination of $\text{MC}_{\mathbf{S}}(\mathcal{M}, g, t, \alpha)$, we have that, for every $w \in M$, $L(\alpha_t, w) = 1$ if, and only if, $M, g, w \models \alpha_t$; and
4. after termination of $\text{MC}_{\mathbf{U}}(\mathcal{M}, g, \alpha, \beta)$, we have that, for every $w \in M$, $L(\alpha\mathbf{U}\beta, w) = 1$ if, and only if, $M, g, w \models \alpha\mathbf{U}\beta$.

Proof. The proofs for cases 1, 2, and 3 are easy. We only show the more involved case 4. The procedure $\text{MC}_{\mathbf{U}}$ works as follows. First, all nodes are set to unmarked. Then, for each w that is labelled with β , the first inner for loop marks all the nodes u such that there exists a node v which is not labelled with α and Rwv . The second inner for loop labels with $\alpha\mathbf{U}\beta$ the remaining unmarked nodes. The set of states labelled with $\alpha\mathbf{U}\beta$ grows monotonically as computation proceeds. Indeed, a node that is not labelled with $\alpha\mathbf{U}\beta$ during the iteration for some w may be labelled with $\alpha\mathbf{U}\beta$ during a later iteration for some w' . Let $w \in L(\beta)$. We claim that:

Claim 4.2 After termination of the main for loop dedicated to w in the procedure $\text{MC}_{\mathbf{U}}$, for every $v \in R^{-1}(w) \cup R^{-2}(w)$, we have that v is labelled with $\alpha\mathbf{U}\beta$ if, and only if, every successor of v that is a predecessor of some $w \in L(\beta)$ is labelled with α , which means that $M, v \models \alpha\mathbf{U}\beta$.

It follows that, after termination of the procedure $\text{MC}_{\mathbf{U}}$, for every $v \in M$, v is labelled with $\alpha\mathbf{U}\beta$ if, and only if, every successor of v that is a predecessor of some $w \in L(\beta)$ is labelled with α , which means that $M, v \models \alpha\mathbf{U}\beta$.

We prove the claim. For the left to right direction, suppose that v is labelled with $\alpha\mathbf{U}\beta$. Then, v is unmarked before the second inner for loop. We infer that every successor of v that is a predecessor of w is labelled with α . Indeed, suppose there exists a successor z of v such that z precedes w and z is not labelled with α . Because of the first inner for loop, the predecessors of z are marked. Since v is a predecessor of z , v is marked as well, which contradicts the fact that v is unmarked. For the right to left direction, suppose every successor of v that is a predecessor of w is labelled with α . Then, after the first inner for loop, v is unmarked. Indeed, suppose v is marked. Then, there exists some successor z of v that is a predecessor of w and is not labelled with α . This contradicts the fact that every successor of v that is a predecessor of w is labelled with α . Hence, v is unmarked before entering the second inner for loop and, hence, v is labelled with $\alpha\mathbf{U}\beta$ at the end of it. ■

What is the complexity of the subroutines? Let $\mathcal{M} = \langle M, R, V \rangle$ be a finite hybrid model, with $n = |M|$ and $m = |R|$. The procedure $\text{MC}_{\mathbf{F}}$ runs in $O(n+m)$, and both $\text{MC}_{\mathbf{S}}$ and MCA run in $O(n)$. As for $\text{MC}_{\mathbf{U}}$, for every $w \in L(\beta)$, the procedure pays two backward visits to the states reachable in 2 steps. Each visit costs $O(m)$. Since there are $O(n)$ states in $L(\beta)$, $\text{MC}_{\mathbf{U}}$ runs in $O(n \cdot m)$ time.

A model checker MCLITE for the language $\text{HL}(\mathbf{A}, \mathbf{F}, \mathbf{P}, \mathbf{U}, \mathbf{S}, \mathbf{A})$ can be easily programmed by taking advantage of subroutines $\text{MC}_{\mathbf{F}}$, MCA , $\text{MC}_{\mathbf{S}}$, and $\text{MC}_{\mathbf{U}}$. MCLITE works bottom-up checking all the subformulas of the input formula in increasing length order. When a formula starting with one of $\mathbf{A}$, $\mathbf{F}$, $\mathbf{U}$, $\mathbf{A}$ needs to be checked, the corresponding procedure is invoked. Past temporal operators $\mathbf{P}$ and $\mathbf{S}$ are handled by feeding the subroutines $\text{MC}_{\mathbf{F}}$ and $\text{MC}_{\mathbf{U}}$, respectively, with the reversed model $\mathcal{M}^*$. Finally, Boolean connectives are treated as usual. The correctness of MCLITE follows from the correctness of its subroutines (Proposition 4.1) and the semantics of the past operators $\mathbf{P}$ and $\mathbf{S}$.

Theorem 4.3 (Correctness of MCLITE) Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid model, g an assignment, and ϕ a formula in $\text{HL}(\mathbf{A}, \mathbf{F}, \mathbf{P}, \mathbf{U}, \mathbf{S}, \mathbf{A})$. Then, after termination of $\text{MCLITE}(\mathcal{M}, g, \phi)$, we have that, for every $w \in M$ and for every $\alpha \in \text{sub}(\phi)$, it holds that $L(\alpha, w) = 1$ if, and only if, $M, g, w \models \alpha$.

Now for the computational complexity.

Theorem 4.4 (Complexity of MCLITE) Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid model such that $n = |M|$ and $m = |R|$. Let g be an assignment, and ϕ a hybrid formula of length k . Then,

- if ϕ belongs to $\text{HL}(\mathbf{A}, \mathbf{F}, \mathbf{P}, \mathbf{U}, \mathbf{S}, \mathbf{A})$, then the model checker $\text{MCLITE}(\mathcal{M}, g, \phi)$ terminates in time $O(k \cdot n \cdot m)$;
- if ϕ belongs to $\text{HL}(\mathbf{A}, \mathbf{F}, \mathbf{P}, \mathbf{A})$, then the model checker $\text{MCLITE}(\mathcal{M}, g, \phi)$ terminates in time $O(k \cdot (n+m))$; and
- if ϕ belongs to $\text{HL}(\mathbf{A}, \mathbf{A})$, then the model checker $\text{MCLITE}(\mathcal{M}, g, \phi)$ terminates in time $O(k \cdot n)$.

Proof. There are $|\text{sub}(\phi)| = |\phi| = k$ subformulas to check. The complexity of each check depends on the form of the subformula ψ . If ψ is atomic, then it is checked in constant time. If its main operator is either Boolean or $\mathbf{A}$, then ψ is checked in $O(n)$. If the main operator of ψ is either $\mathbf{F}$ or $\mathbf{P}$, then ψ is checked in $O(n+m)$. Finally, if the main operator of ψ is either $\mathbf{U}$ or $\mathbf{S}$, then ψ is checked in $O(n \cdot m)$. ■

We can extend MCLITE to cope with transitive closure operators $\mathbf{F}^+$ and $\mathbf{U}^+$, as well as with their past counterparts, without increasing the complexity. The idea is to replace the visit to the predecessors of w with a backward depth-first visit of the nodes that can reach w . As an alternative, one may first compute the transitive closure of the accessibility relation, and then use the model checker MCLITE on the transitive model.

We now move to hybrid languages with binders. The bad news is that the efficient bottom-up strategy used in MCLITE does not work for such languages. Why? Consider the formula $\mathbf{G}\downarrow x.\mathbf{F}x$. It says that every successor of the current point is reflexive. If we try to check this formula in a bottom-up fashion, that is, from simple formulas to more complex ones, we initially have to check the subformula x . However, at this stage, we do not have enough information to check x , and hence we cannot label the states of the model with x . Instead, we can proceed as follows: we first check $\downarrow x.\mathbf{F}x$ with a procedure yet to be developed, then we check $\mathbf{G}\downarrow x.\mathbf{F}x$, that is $\neg\mathbf{F}\neg\downarrow x.\mathbf{F}x$, taking advantage of the subprocedure $\text{MC}_{\mathbf{F}}$ of MCLITE . In order to check $\downarrow x.\mathbf{F}x$ for each state w , we first assign w to x , and then we check $\mathbf{F}x$ at w under the new assignment for x . The latter again can be done using $\text{MC}_{\mathbf{F}}$. The sketched strategy, which combines top-down and bottom-up reasoning, has been implemented in a recursive model checker MCFULL for the full hybrid language $\text{HL}(\text{HO} \cup \text{TO})$, where $\text{HO} = \{\mathbf{A}, \downarrow, \exists\}$, and $\text{TO} = \{\mathbf{F}, \mathbf{P}, \mathbf{U}, \mathbf{S}, \mathbf{A}\}$. The auxiliary procedures $\text{Check}_{\mathbf{F}}$, $\text{Check}_{\downarrow}$, and $\text{Check}_{\exists}$. The procedures for the other operators are similar to $\text{Check}_{\mathbf{F}}$. Whenever $\text{Check}_{\mathbf{F}}$, with $* \in \text{HO} \cup \text{TO}$, handle the cases of subformulas with main operator $*$. Whenever possible, these procedures re-use the subprocedures $\text{MC}_{\mathbf{F}}$, with $* \in \{\mathbf{A}\} \cup \text{TO}$, of MCLITE . Moreover, they use the new subroutine $\text{Clear}(L, x)$ to reset all the values of $L(\alpha)$, for any α containing the variable x . Boolean operators are treated as usual. In Figure 2 we show the pseudocode for $\text{Check}_{\mathbf{F}}$, $\text{Check}_{\downarrow}$, and $\text{Check}_{\exists}$. The procedures for the other operators are similar to $\text{Check}_{\mathbf{F}}$.

The correctness of MCFULL follows from the correctness of MCLITE and the semantics of the hybrid binders.

Theorem 4.5 (Correctness of MCFULL) Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid model, g an assignment, and ϕ a hybrid formula. Then, after termination of $\text{MCFULL}(\mathcal{M}, g, \phi)$, we have that:

- for every $w \in M$, $L(\phi, w) = 1$ if, and only if, $M, g, w \models \phi$; and
- for every $w \in M$ and every sentence $\alpha \in \text{sub}(\phi)$, $L(\alpha, w) = 1$ if, and only if, $M, g, w \models \alpha$.

Now for the computational complexity.

Theorem 4.6 (Complexity of MCFULL) Let $\mathcal{M} = \langle M, R, V \rangle$ be a hybrid model such that $n = |M|$ and $m = |R|$, and g an assignment. Let ϕ be a hybrid formula in $\text{HL}(\exists, \downarrow, \mathbf{S})$, and let r_1 and r_2 be the nesting degree of $\downarrow$ and $\exists$, respectively, in ϕ . Let C_S be the model checking complexity for $\text{HL}(\mathbf{S})$. Then, $\text{MCFULL}(\mathcal{M}, g, \phi)$ terminates in time $O(C_S \cdot n^{(r_1+2 \cdot r_2)})$ and it uses polynomial space.

Proof. We have that:

```

Procedure  $\text{Check}_{\mathbf{F}}(\mathcal{M}, g, \alpha)$ 
 $\text{MCFULL}(\mathcal{M}, g, \alpha)$ 
 $\text{MC}_{\mathbf{F}}(\mathcal{M}, g, \alpha)$ 

Procedure  $\text{Check}_{\mathbf{I}}(\mathcal{M}, g, \alpha)$ 
for  $w \in M$  do
   $g(x) \leftarrow w$ 
   $\text{MCFULL}(\mathcal{M}, g, \alpha)$ 
  if  $w \in L(\alpha)$  then
     $\text{Clear}(L, x)$ 
     $L(\exists x, \alpha, w) \leftarrow 1$ 
  else
     $\text{Clear}(L, x)$ 
  end if
end for

Procedure  $\text{Check}_{\exists}(\mathcal{M}, g, \alpha)$ 
for  $w \in M$  do
  for  $v \in M$  do
     $g(x) \leftarrow v$ 
     $\text{MCFULL}(\mathcal{M}, g, \alpha)$ 
    if  $w \in L(\alpha)$  then
       $\text{Clear}(L, x)$ 
       $L(\exists x, \alpha, w) \leftarrow 1$ 
    else
       $\text{Clear}(L, x)$ 
    end if
  end for
end for

```

Figure 2: MCFULL subprocedures.

- the procedure Check_* runs in time $C_\alpha + C_*$, where C_* is the cost of MC_* and C_α is the cost to check α ;
- the procedure $\text{Check}_{\mathbf{I}}$ runs in time $n \cdot C_\alpha$; and
- the procedure $\text{Check}_{\exists}$ runs in time $n^2 \cdot C_\alpha$.

Thus, the overall worst-case time complexity is $O(C_S \cdot n^{(r_1 + 2r_2)})$. Since the height of the recursion stack for MCFULL is at most $|\phi|$, we have that MCFULL uses polynomial space. ■

The model checker MCFULL can easily be extended to cope with formulas involving the $\Downarrow$ binder and the comparison of state labels. A procedure similar to $\text{Check}_{\mathbf{I}}$ can be used to check formulas of the form $\Downarrow v, \alpha$. The only difference is that we have to bind the variable v to the *label* of the current state, and not to the current state itself. The computational complexity remains the same. Moreover, the formula v , for $v \in \text{WVAR}'$, can be verified by comparing the label of the current state and the label stored in v , while the formula $v = w$, for $v, w \in \text{WVAR}$, can be checked by comparing the labels stored in v and w .

Furthermore, the schema of the algorithm MCFULL can be viewed as a general model checker for the hybridization of *any* temporal logic. Given a temporal logic $\mathbf{T}$, the hybridization of $\mathbf{T}$ is the hybrid logic obtained from the language of $\mathbf{T}$ by adding the hybrid machinery. Suppose that, for each temporal operator $\mathbf{O}$ of arity k in $\mathbf{T}$, we can exploit a procedure $\text{Check}_{\mathbf{O}}$ that, given a model and k formulas $\alpha_1, \dots, \alpha_k$, labels each state m of the model with the formula $\mathbf{O}(\alpha_1, \dots, \alpha_k)$ if, and only if, the formula is true at m . A model checker for the hybridization of $\mathbf{T}$ can be synthesized from these model checking procedures following the example of MCFULL. For instance, a model checker for $\text{HDL}(\@, \downarrow, \Psi, \mathbf{A})$ can be programmed by taking advantage of a CPDL model checker. It is worth remarking that model checking for CPDL can be done in linear time with respect to the length of the formula and the size of the model [23]. Hence, we have the following:

Theorem 4.7 *Model checking for $\text{HDL}(\@, \downarrow, \Psi, \mathbf{A})$ has linear-time complexity, while model checking for $\text{HDL}(\@, \downarrow, \Psi, \mathbf{A})$ can be done in exponential time in the nesting degree of hybrid binders and in polynomial space.*

4.2 Lower Bounds

The natural question is: can we do better than the upper bounds given in Section 4.1? It is well-known that model checking for first-order logic is PSPACE-complete. Since $\text{HL}(\exists, \@, \mathbf{F})$ is

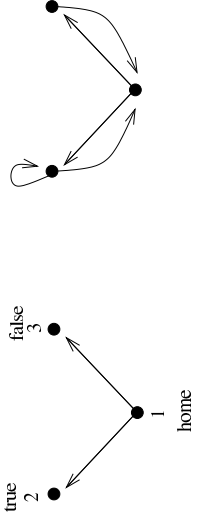


Figure 3: Embedding QBF into model checking for hybrid logics.

as expressive as first-order logic [16], model checking for $\text{HL}(\exists, \@, \mathbf{F})$ is PSPACE-complete as well. However, what about fragments of $\text{HL}(\exists, \@, \mathbf{F})$? The question is particularly interesting for the fragment $\text{HL}(\downarrow, \@, \mathbf{F})$, since it corresponds to the *bounded fragment* of the first-order correspondence language [10]. Unfortunately, the model checking problem for $\text{HL}(\downarrow, \@, \mathbf{F})$ and hence for the bounded fragment is still PSPACE-hard, even without $\@$, nominals and propositions.

Theorem 4.8 *Model checking for the pure nominal-free fragment of $\text{HL}(\downarrow, \mathbf{F})$ is PSPACE-complete.*

Proof. PSPACE-membership follows from Theorem 4.6. To prove PSPACE-hardness, we embed Quantified Boolean Formulas (QBF) [36] into the model checking problem for the pure nominal-free fragment of $\text{HL}(\downarrow, \mathbf{F})$. We proceed in two steps. We first embed QBF into the model checking problem for $\text{HL}(\downarrow, \@, \mathbf{F})$. Then, we remove the $\@$ operator and atomic letters.

Recall that an instance of QBF has the form $\Psi = Q_1x_1, \dots, Q_nx_n, \alpha(x_1, \dots, x_n)$, where $Q_i \in \{\exists, \forall\}$, and $\alpha(x_1, \dots, x_n)$ is a Boolean formula using variables $x_1, \dots, x_n$. Let $\text{NOM} = \{\text{true}, \text{false}, \text{home}\}$ and let $\mathcal{M}$ be the model depicted in Figure 3, left-hand side. Let ϕ_Ψ be the $\text{HL}(\downarrow, \@, \mathbf{F})$ -formula obtained from Ψ by replacing every occurrence of $\exists x$ by $\@_{\text{home}}\mathbf{F}\downarrow x$, every occurrence of $\forall x$ by $\@_{\text{home}}\mathbf{G}\downarrow x$, every occurrence of x by $\@_x\text{true}$, and every occurrence of $\neg x$ by $\@_x\text{false}$. We have that Ψ is true if, and only if, $\mathcal{M}, 1 \models \phi_\Psi$.

We now remove $\@$ and atomic letters. To remove $\@$, we have to find a way to “come back home” after $\mathbf{F}\downarrow x$ has fixed the variable x . We can add to the previous model two more edges, one from 2 to 1, and the other from 3 to 1, and use the $\mathbf{F}$ operator to come home. Since we don’t have atomic letters, we have to distinguish in some structural way between state 2 (denoting true) and state 3 (denoting false). We can add, for instance, a reflexive edge leaving 2. The resulting frame is depicted in Figure 3 (right-hand side). Without loss of generality, we assume that negation in $\alpha(x_1, \dots, x_n)$ is applied only to variables. Let τ be the following translation:

$$\begin{aligned}
 \tau(x) &= \mathbf{F}(x \wedge \downarrow y, \mathbf{F}y) \\
 \tau(\neg x) &= \mathbf{F}(x \wedge \downarrow y, \mathbf{G}\neg y) \\
 \tau(\alpha_1 \wedge \alpha_2) &= \tau(\alpha_1) \wedge \tau(\alpha_2) \\
 \tau(\alpha_1 \vee \alpha_2) &= \tau(\alpha_1) \vee \tau(\alpha_2) \\
 \tau(\exists x, \alpha) &= \mathbf{F}\downarrow x, \mathbf{F}(\downarrow y, \mathbf{G}\neg y \wedge \tau(\alpha)) \\
 \tau(\forall x, \alpha) &= \mathbf{G}\downarrow x, \mathbf{F}(\downarrow y, \mathbf{G}\neg y \wedge \tau(\alpha))
 \end{aligned}$$

We leave it to the reader to check that Ψ is true if and only if $\mathcal{M}, 1 \models \tau(\Psi)$. ■

Theorem 4.8 still holds if we replace $\mathbf{F}$ by $\mathbf{F}^+$:

Theorem 4.9 *Model checking for the pure nominal-free fragments of $\text{HL}(\downarrow, \mathbf{F}^+)$ is PSPACE-complete.*

Proof. PSPACE-membership follows from Theorem 4.6. To prove PSPACE-hardness, we embed QBF into the model checking problem for the pure nominal-free fragments of $\text{HL}(\downarrow, \mathbf{F}^+)$. Let $\Psi = Q_1x_1, \dots, Q_nx_n, \alpha(x_1, \dots, x_n)$, where $Q_i \in \{\exists, \forall\}$, and $\alpha(x_1, \dots, x_n)$ is a Boolean formula

using variables $x_1, \dots, x_n$. Let $\mathcal{M}$ be the unlabelled model with frame $\{\{1, 2\}, \{(1, 2), (2, 1)\}\}$. Let ϕ_Ψ be the $\text{HL}(\downarrow, \odot, \mathbf{F})$ -formula obtained from Ψ by replacing every occurrence of $\exists x$ by $\mathbf{F}^+ \downarrow x$ and every occurrence of $\forall x$ by $\mathbf{G}^+ \downarrow x$. Then Ψ is true if, and only if, $\mathcal{M}, 1 \models \phi_\Psi$. ■

If we replace $\downarrow$ by $\exists$, hardness holds even without temporal operators.

Theorem 4.10 *Model checking for the pure nominal-free fragment of $\text{HL}(\exists)$ is PSPACE-complete.*

Proof. PSPACE-membership follows from Theorem 4.6. To prove PSPACE-hardness, we embed QBF into the model checking problem for the pure nominal-free fragments of $\text{HL}(\exists)$. Let $\Psi = Q_1 x_1 \dots Q_n x_n \alpha(x_1, \dots, x_n)$, where $Q_i \in \{\exists, \forall\}$, and $\alpha(x_1, \dots, x_n)$ is a Boolean formula using variables $x_1, \dots, x_n$. Note that Ψ is a pure nominal-free formula in $\text{HL}(\exists)$. Let $\mathcal{M}$ be the unlabelled model based on the frame $\langle \{1, 2\}, \emptyset \rangle$. Then Ψ is true if, and only if, $\mathcal{M}, 1 \models \Psi$. ■

Since $\exists x \alpha$ is equivalent to $\downarrow y. \mathbf{E}(\downarrow x. \mathbf{E}(y \wedge \alpha))$, we obtain the following:

Corollary 4.11 *Model checking for the pure nominal-free fragment of $\text{HL}(\downarrow, \mathbf{A})$ is PSPACE-complete.*

Next, we show that replacing $\downarrow$ by $\downarrow$ does not change the model checking complexity.

Theorem 4.12 *Model checking for $\text{HL}(\downarrow, \mathbf{F})$ is PSPACE-complete.*

Proof. The PSPACE upper bound comes from Section 4.1. To prove PSPACE-hardness, we embed the model checking problem for $\text{HL}(\downarrow, \mathbf{F})$ into the same problem for $\text{HL}(\downarrow, \mathbf{F})$. Consider a hybrid model $\mathcal{M} = \langle M, R, V \rangle$ and a formula α in $\text{HL}(\downarrow, \mathbf{F})$. We construct a new hybrid model $\mathcal{M}' = \langle M', R', V' \rangle$, where, for each $m \in M$, there exists one fresh nominal i_m such that $V'(i_m) = \{m\}$. Moreover, let α' be the formula in $\text{HL}(\downarrow, \mathbf{F})$ that is obtained from α by replacing each instance of $\downarrow$ by $\downarrow$ and each instance of x by v_x , where $x \in \text{WVAR}$ and $v_x \in \text{WVAR}'$. Then, we have that $\mathcal{M}, g, m \models \alpha$ if, and only if, $\mathcal{M}', g, m \models \alpha'$. ■

Since hybrid dynamic logic extends hybrid logic, and because of Theorem 4.7, we have the following:

Theorem 4.13 *Model checking for the pure nominal-free fragment of $\text{HDL}(\downarrow)$ is PSPACE-complete.*

5 Hybrid Logic in Action

On the previous pages we have discussed, and obtained, complexity results for model checking a variety of hybrid logics. What's the point? In this section we describe an application of hybrid logic model checking to the task of evaluation of queries and constraints on semistructured data. We will encounter a number of query and constraint formats as well as reasoning tasks, all of which correspond to model checking in some specific hybrid language.

5.1 Semistructured data

The growth of the World Wide Web has given us a vast, largely accessible database. The Extensible Markup Language (XML) [24] is a textual representation of information that was designed to represent the hierarchical content of documents. It has been proposed as a data model for semistructured databases since it is able to naturally represent missing or duplicated data as well as deeply nested information [1]. As an example, Figure 4 contains the XML representation of a simple bibliography file.

A very natural representation for semistructured data is a labelled graph. In this paper, we will represent semistructured data as a graph in which a node corresponds to an object and an edge corresponds to an object attribute. Edges are labelled with attribute names, while leaf nodes are labelled with object values (internal nodes are not labelled). The data graph corresponding to the XML example in Figure 4 is depicted in Figure 4.

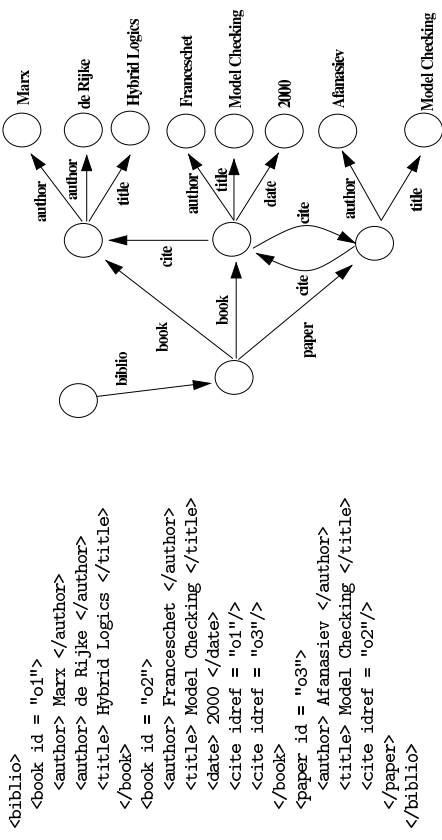


Figure 4: (Left) An XML representation of a bibliography file. (Right) A graphical representation of the same file.

5.2 Query Processing via Hybrid Logic Model Checking

It is possible to perform significant query processing on semistructured data via model checking for hybrid languages. Quite a number of query languages for semistructured data have been proposed in the literature. However, many of them are similar in spirit and sometimes even in syntax [1]. For this reason, we have chosen one of them as a model: Lorel [3]. Lorel is the query language in the Lore system [34], which was designed at Stanford University specifically for managing semistructured data. We discuss three fragments of Lorel, that differ in their expressive power; each will be embedded in a suitable hybrid logic. As a consequence, we are able to process a query in Lorel by taking advantage of a model checker for hybrid logics. Moreover, with the aid of the well-understood hybrid logic framework, it will be easy to investigate and compare the expressive power of various fragments of the Lorel query language.

All fragments that we consider allow regular expressions for navigating the data graph. Some offer the possibility of comparing object identities, which allows us to implement queries with joins. We will focus on *monadic queries* only: queries that return a set of objects of the database, or, equivalently, a set of nodes of the graph representation of the database. The reason for this restriction is that we want to embed the query processing problem into the global model checking problem, and the output of a global model checker is a set of nodes. We will consider 3 fragments of Lorel forming an inclusion chain: $\mathcal{L}_{\text{qry}}^1 \subset \mathcal{L}_{\text{qry}}^2 \subset \mathcal{L}_{\text{qry}}^3$. The most expressive language $\mathcal{L}_{\text{qry}}^3$ captures a large subfragment of Lorel. Additional features of Lorel that are not expressible in $\mathcal{L}_{\text{qry}}^3$ will be discussed in Section 5.2.4.

5.2.1 Fragment 1: $\mathcal{L}_{\text{qry}}^1$

We start by defining the query language $\mathcal{L}_{\text{qry}}^1$. A query in $\mathcal{L}_{\text{qry}}^1$ has the following schema Q_1 :

```
(Q1) select x
      from rexp x
      where x.fexp,
```

where **rexp** is a regular expression on edge labels (i.e., attribute names), and **fexp** is a so-called filter expression. Intuitively, Q_1 retrieves all nodes reachable from the root through **rexp** satisfying

the filter **fexp**. The variable **X** is used as a container for these nodes. We call the variable **X** in the 'select' clause of the query the *focus* of the query; the condition **rexp X** in the 'from' clause of the query the *selection expression*; and the condition **X.fexp** in the 'where' clause of the query the *filter expression*. A filter expression has the following syntax:

fexp = **true** | **rexp** | **rexp :: a** | **fexp and fexp** | **fexp or fexp** | **not fexp**,

where **a** is an object value. Given a set of nodes **X**, the filter **X.rexp** selects a node *v* in **X** if there exists at least one reachable node from *v* through **rexp**. The filter **X.rexp::a** adds an additional constraint: it filters a node *v* in **X** if there exists at least one reachable node from *v* through **rexp** that is labelled with *a*. For instance, with reference to the our example depicted in Figure 4, consider the following query:

```
select X
from biblio.book X
where X.(author::Franceschet and date::2000)
```

This query selects all books written in 2000 such that Franceschet is one of the authors. In our example, book *o2* is retrieved. Moreover, the query

```
select X
from biblio._ X
where X.abstract
```

retrieves all entries for which an abstract has been provided. In our example, no entry is retrieved.

The simple query language $\mathcal{L}_{\text{qry}}^1$ can be embedded into hybrid dynamic logic with only one nominal *root* for the data graph root and no hybrid operators. The embedding τ_1 is as follows. Let ω be an embedding from filter expressions to hybrid formulas:

$$\begin{aligned} \omega(\text{true}) &= \top \\ \omega(\text{rexp}) &= \langle \text{rexp} \rangle \top \\ \omega(\text{rexp} :: a) &= \langle \text{rexp} \rangle a \\ \omega(\text{rexp}_1 \text{ and } \text{rexp}_2) &= \omega(\text{rexp}_1) \wedge \omega(\text{rexp}_2) \\ \omega(\text{rexp}_1 \text{ or } \text{rexp}_2) &= \omega(\text{rexp}_1) \vee \omega(\text{rexp}_2) \\ \omega(\text{not rexp}) &= \neg \omega(\text{rexp}) \end{aligned}$$

To the query schema Q_1 we associate the hybrid formula $\tau_1(Q_1) = \langle \text{rexp} \rangle^{-1} \text{root} \wedge \omega(\text{fexp})$. Notice how the selection expression is translated 'backward,' while the filter expression is translated 'forward.'

Next, we clarify the relation between query processing for semistructured data and model checking for hybrid logic. Define the *answer set* of a monadic query *q* with respect to a semistructured database *D* as the set of nodes retrieved by *q* belonging to the data graph G_D associated to *D*. Each node in the answer set corresponds to an element in the corresponding XML representation of *D*. Moreover, given a hybrid formula α and a hybrid model $\mathcal{M}$, let the *truth set* of α with respect to $\mathcal{M}$ be the set of nodes of $\mathcal{M}$ at which α is true. The following result relates the query processing problem for semistructured data to the model checking problem for hybrid dynamic logic.

Proposition 5.1 *Let q be a query in $\mathcal{L}_{\text{qry}}^1$ and D be a semistructured database. Then, the answer set of q with respect to D corresponds to the truth set of $\tau_1(q)$ with respect to G_D .*

Moreover, since the translation τ_1 goes inside a fragment of HDL($@, \mathbf{A}$), by virtue of Theorem 4.7, we have the following corollary.

Corollary 5.2 *Query processing for $\mathcal{L}_{\text{qry}}^1$ can be done in linear time.*

5.2.2 Fragment 2: $\mathcal{L}_{\text{qry}}^2$

The expressive power of $\mathcal{L}_{\text{qry}}^1$ can be extended by splitting the selection expression in more than one chunk. Consider the following query that selects the authors of all entries with title *Model Checking*:

```
select Y
from biblio._ X, X.author Y
where X.title::Model Checking
```

This query is not definable in $\mathcal{L}_{\text{qry}}^1$ but it can be embedded into the basic hybrid dynamic logic as $\langle \text{author} \rangle^{-1} \langle (\text{biblio} \cdot _)^{-1} \text{root} \wedge (\text{title}) \text{Model Checking} \rangle$. We define the query schema Q_2 as follows:

```
(Q2) select X1
from rexp1 X1, X1.rexp2 X2, ..., Xn-1.rexpn Xn
where fexp1, ..., fexpm
```

Each filter expression **fexp_j** has the form **X.fexp**, where **X** is a variable and **fexp** in a filter expression as in Q_1 . Intuitively, the schema Q_2 binds the variable X_1 to the nodes reachable from the root through the regular expression **rexp₁**, it binds the variable X_2 to the nodes reachable from a node in X_1 through the regular expression **rexp₂**, and so on. The filter expression **X.fexp** filters the nodes placed in the variable **X** according to the Boolean filter **fexp**. Finally, the nodes contained in the focus X_1 are selected.

A well-formed query *q* is defined as follows. Let *X* be the focus of *q*, *S* the selection sequence of *q*, *F* the filter sequence of *q*, and *V* the variables in *S*. The query *q* is *well-formed* if (i) $X \in V$, and (ii) all variables used in the filter sequence *F* are in *V*. Let $\mathcal{L}_{\text{qry}}^2$ be the query language containing all the well-formed queries according to the schema Q_2 . Notice that $\mathcal{L}_{\text{qry}}^1 \subset \mathcal{L}_{\text{qry}}^2$.

We now develop an embedding τ_2 of queries in $\mathcal{L}_{\text{qry}}^2$ into the basic hybrid dynamic logic with only one nominal *root* for the root of the data graph. Consider the query schema Q_2 . For each variable X_j in Q_2 , let **fexp_j** be the filter expression associated with the variable X_j , or **fexp_j** = **true** if no filter expression has been explicitly associated with X_j . We will use the following two auxiliary functions ν and ν^{-1} mapping a variable in Q_2 into a hybrid formula.

$$\begin{aligned} \nu(X_j) &= \begin{cases} \omega(\text{fexp}_j) \wedge \langle \text{rexp}_{j+1} \rangle \nu(X_{j+1}) & \text{if } j < n \\ \omega(\text{fexp}_j) & \text{if } j = n \end{cases} \\ \nu^{-1}(X_j) &= \begin{cases} \omega(\text{fexp}_j) \wedge \langle \text{rexp}_j \rangle^{-1} \nu^{-1}(X_{j-1}) & \text{if } j > 1 \\ \omega(\text{fexp}_j) \wedge \langle \text{rexp}_j \rangle^{-1} \text{root} & \text{if } j = 1 \end{cases} \end{aligned}$$

Now, let X_1 be the focus of Q_2 , that is, X_1 is the variable in the select clause of Q_2 . The translation τ_2 of Q_2 into hybrid logic is as follows:

$$\tau_2(Q) = \begin{cases} \nu(X_1) \wedge \langle \text{rexp}_1 \rangle^{-1} \nu^{-1}(X_{i-1}) & \text{if } i > 1 \\ \nu(X_1) \wedge \langle \text{rexp}_1 \rangle^{-1} \text{root} & \text{if } i = 1 \end{cases}$$

Notice that $\tau_2(\mathcal{L}_{\text{qry}}^1) = \tau_1(\mathcal{L}_{\text{qry}}^1)$.

Proposition 5.3 *Let q be a query in $\mathcal{L}_{\text{qry}}^2$ and D be a semistructured database. Then, the answer set of q with respect to D corresponds to the truth set of $\tau_1(q)$ with respect to G_D .*

Since the codomain of the translation τ_2 is a fragment of HDL($@, \mathbf{A}$), by virtue of Theorem 4.7, we have the following corollary:

Corollary 5.4 *Query processing for $\mathcal{L}_{\text{qry}}^2$ can be done in linear time.*

5.2.3 Fragment 3: $\mathcal{L}_{\text{qry}}^3$

So far, we have not used the full power of hybrid logic. In particular, the hybrid binder $\downarrow$ has not been exploited in the query translation. However, consider the following query that selects papers with at least two authors:

```
select X
from biblio.paper X, X.author Y, X.author Z
where Y  $\neq$  Z
```

This query can be embedded in hybrid logic, but the binder $\downarrow$ needs to be invoked:

$$\downarrow x.(\text{biblio.paper})^{-1}\text{root} \wedge (\text{author})\downarrow y. @_x(\text{author})\downarrow z. y \neq z.$$

Moreover, the following query selects all the self-reference papers, that is, all papers whose authors cite themselves:

```
select X
from biblio.paper X, X.cite Y
where X = Y
```

It corresponds to the hybrid formula $\downarrow x.(\text{biblio.paper})^{-1}\text{root} \wedge (\text{cite})\downarrow y.x = y$. The following query retrieves all papers p such that there exists a paper q reachable from p through a path of cite edges of positive length that cites back to p :

```
select X
from biblio.paper X, X.cite.cite* Y
where X = Y
```

This query maps to the hybrid formula: $\downarrow x.(\text{biblio.paper})^{-1}\text{root} \wedge (\text{cite.cite}^*)\downarrow y.x = y$.

We define a query schema Q_3 that allows the above queries as follows:

```
(Q3) select Xi
from sexp1, sexp2, ..., sexpn
where fexp1, fexp2, ..., fexpm
```

where each sexp_j is a selection expression and each fexp_j is a filter expression. A selection expression is either $\text{rexp } X$ or $X.\text{rexp } Y$, where rexp is a regular expression and X and Y are variables. A filter expression is either $X.\text{fexp}$, or $X = Y$, or $X \neq Y$, where X and Y are variables and fexp is a filter expression as in Q_1 .

However, not every selection sequence is legal. For instance, $X.a Y.X.b Y$ is not legal for two reasons. First, it does not specify the content of X . Second, the content of Y is ambiguous. A well-formed query q is defined as follows. Let X be the focus of q , S the selection sequence of q , F the filter sequence of q , and V the variables in S . We construct the edge-labelled directed graph T_S with nodes in $V \cup \{\cdot\}$. There is an edge $(\cdot, X)$ labelled with rexp if $\text{rexp } X$ is in S , and there is an edge (X, Y) labelled with rexp if $X.\text{rexp } Y$ is in S . We say that S is *legal* if T_S is a tree rooted at $\cdot$. That is (i) each node in T_S is reachable from $\cdot$, (ii) the root $\cdot$ of T_S has no predecessor and all the other nodes in T_S have exactly one predecessor. The query q is *well-formed* if (i) $X \in V$, (ii) S is legal, and (iii) all the variables used in the filter sequence F are in V . Let $\mathcal{L}_{\text{qry}}^3$ be the query language containing all the well-formed queries according to the schema Q_3 . Notice that $\mathcal{L}_{\text{qry}}^3 \subset \mathcal{L}_{\text{qry}}^2$.

We now develop an embedding τ_3 of queries in $\mathcal{L}_{\text{qry}}^3$ into the hybrid dynamic logic with nominals, $@$ and $\downarrow$. For the sake of simplicity, we describe the embedding with an example. Consider the following abstract query q :

```
select Z
from a X, X.b Y, X.c Z, Z.d W
where X.e.Y.f.Z.g.W.h.Y = W
```

Notice that q is well-formed. The corresponding hybrid formula is as follows:

$$\downarrow z.(g)^{\top} \wedge \langle d \rangle \downarrow w.(h)^{\top} \wedge \\ @_z(c)^{-1} \downarrow x.(e)^{\top} \wedge \langle b \rangle \downarrow y.(f)^{\top} \wedge @_x(a)^{-1}\text{root} \wedge \\ y = w$$

The formula has been deliberately divided into three lines. The first line constraints the focus of the query (node Z) and its subtree (node W). The second line predicates over the unique path from the parent of the focus to the root of the tree as well as over all the subtrees rooted at children of nodes on this path not belonging to the path (nodes X , Y , and $\cdot$). The third line captures the identity checking constraints. This technique can be generalized to arbitrary trees. Notice that the use of the binder $\downarrow$ is necessary to encode the constraint that compares object identities. Indeed, if we remove the filter $Y = W$ from the above query, the latter can be encoded without $\downarrow$ as follows:

$$\langle g \rangle^{\top} \wedge \langle d \rangle \langle h \rangle^{\top} \wedge \\ \langle c \rangle^{-1} \langle e \rangle^{\top} \wedge \langle b \rangle \langle f \rangle^{\top} \wedge \langle a \rangle^{-1}\text{root}$$

Proposition 5.5 *Let q be a query in $\mathcal{L}_{\text{qry}}^3$ and D be a semistructured database. Then, the answer set of q with respect to D corresponds to the truth set of $\tau_3(q)$ with respect to G_D .*

Since the the translation τ_3 embeds into HDL($@, \downarrow, \downarrow$, $\mathbf{A}$), by virtue of Theorem 4.7, we have the following corollary:

Corollary 5.6 *Query processing for $\mathcal{L}_{\text{qry}}^3$ can be done in exponential time and polynomial space.*

5.2.4 Additional Features of Lorel

Lorel includes additional queries that are not immediately expressible in our language. For instance, a query in Lorel may use regular expressions on object values, as in the following:

```
select X
from biblio.paper X, X.title Y
where matches("*.D[d]data.*", Y)
```

The query selects all the papers with a title containing either the word **Data** or **data**. Regular expressions on object values can be incorporated into our model checking framework as follows. Given a query q featuring the regular expression r and a database D , the data graph representing D is preprocessed and all the leaf nodes with a value matching r are labelled with a fresh symbol reg.r . Moreover, each instance of $\text{matches}("*.D[d]data.*", Y)$ in q is replaced by $Y.e::\text{reg.r}$. The query q is then translated into hybrid logic as shown above and the model checking is applied.

Lorel queries may also allow variables containing object values, instead of object identifiers. Consider the following query that retrieves all papers with two different authors with the same first name (we assume that the **author** element has a subelement **name** which in turn has an atomic subelement **first**):

```
select X
from biblio.paper X, X.author Y, Y.name.first N,
X.author Z, Z.name.first M
where Y  $\neq$  Z, N = M
```

The constraint $Y \neq Z$ compares object identities, while $N = M$ compares object values, since Y and Z contain internal nodes, while N and M contain leaf nodes. This query can be implemented in hybrid logic by using the hybrid binder $\downarrow x$ that binds the current node value to the variable x :

$$(\text{biblio.paper})^{-1}\text{root} \wedge \downarrow x.(\text{author})\downarrow y.(\text{name.first})\downarrow n. \\ @_x(\text{author})\downarrow z.(\text{name.first})\downarrow m.y \neq z \wedge n = m$$

3. for each author a and for each publication p in a 's publications list, a should be contained in p 's authors list; and
4. for each publication p and for each author a in p 's authors list, p should be contained in a 's publications list.

The data in Figure 5 satisfies the above constraints. Moreover, we could require that the attribute `code` is a *key* for the element `publication`, meaning that any two different publications should have different code values. Less restrictively, we could ask that the attribute `code` *functionally determines* the authors of a publication, meaning that any two different publications with the same code values should have the same set of authors.

More generally, let r , p and q be regular expressions on attribute names. Let τ be an attribute name and S be a set of attribute names. A τ node is a node reachable through an edge labelled with τ . We consider the following types of constraints on semistructured data:

- *key constraints*, denoted $\tau[S] \rightarrow \tau$, saying that, for all τ nodes x and y , if x and y agree on the values of nodes reachable through attributes in S , then x and y are the same node;
- *path functional constraints*, denoted $\tau p \rightarrow \tau q$, saying that, for all τ nodes x and y , if x and y agree on the values of nodes reachable through p , then they should agree on the values of nodes reachable through q as well;
- *circular constraints*:
 - *forward* version, denoted $p \rightrightarrows q$, saying that all nodes reachable from the root through p are also reachable from the root through q ;
 - *backward* version, denoted $p \leftrightharpoons q$, saying that all nodes reachable from the root through p can reach back to the root through q ;
- *lollipop constraints*:
 - *forward* version, denoted $r \rightarrow p \rightrightarrows q$, saying that, for each node x reachable from the root through r it holds that all nodes that are reachable from x through p are also reachable from x through q ;
 - *backward* version, denoted $r \rightarrow p \leftrightharpoons q$, saying that, for each node x reachable from the root through r it holds that all nodes that are reachable from x through p can reach back to x through q .

Notice that forward circular constraints are special cases of forward lollipop constraints in which r is empty; similarly for backward constraints. Forward circular constraints are also called *inclusion path constraints*, and backward lollipop constraints are sometimes referred to as *inverse path constraints* [1, 26]. Notice also that, whenever S is a singleton $\{a\}$, the key constraint $\tau\{a\} \rightarrow \tau$ corresponds to the functional constraint $\tau.a \rightarrow \tau$.

Constraints (1) and (2) in our example above are forward circular constraints. In particular, (1) is a forward circular constraint in which we have $p = \text{*publication.written-by}$ and $q = \text{*author}$; and (2) is a forward circular constraint with $p = \text{*author.has-written}$ and $q = \text{*publication}$. Moreover, constraints (3) and (4) above are backward lollipop constraints. In particular, (3) is a backward lollipop constraint with $r = \text{*publication}$, $p = \text{written-by}$ and $q = \text{has-written}$, and (2) is a backward lollipop constraint with $r = \text{*author}$, $p = \text{has-written}$ and $q = \text{written-by}$. The key constraint `publication[code]  $\rightarrow$  publication` states that the attribute code of the element `publication` is a key for the element `publication`. Finally, the functional path constraint `publication.code  $\rightarrow$  publication.written-by.name` asks that the code of the element `publication` determines the set of authors of the publication.

We will show how such constraints can be expressed in hybrid dynamic languages. To this end, we define a constraint language $\mathcal{L}_{\text{con}}$ containing any Boolean combination of constraints as

```

:
<author id = "a1">
  <name> Marx </author>
  <has-written idref = "p1"/>
  <has-written idref = "p2"/>
</author>
<author id = "a2">
  <name> de Bijke </author>
  <has-written idref = "p1"/>
</author>
<publication id = "p1">
  <title> Hybrid Logics </title>
  <code> MdeR03 </code>
  <year> 2003 </year>
  <written-by idref = "a1"/>
  <written-by idref = "a2"/>
</publication>
<publication id = "p2">
  <title> Computational Complexity </title>
  <code> MOO </code>
  <year> 2000 </year>
  <written-by idref = "a1"/>
</publication>
:

```

Figure 5: Authors and publications in XML.

Finally, Lorel includes label and path variables as well. These variables can be used to combine schema and data information. These features are beyond the expressive power of hybrid logic with nominals for states. *Path nominals*, i.e., nominals interpreted over paths, and corresponding path binders, have been introduced in hybrid logics [17]. However, at present model checking results for hybrid logics with path nominals and path binders are non-existent.

5.3 Constraint Evaluation via Hybrid Model Checking

So far we have considered hybrid logic model checking as a mechanism for evaluating queries (fragments of) Lorel. We now change tack and consider constraint evaluation for semistructured data. Recall that integrity constraints on structured data are conditions that restrict the possible populations of the database. They are important to maintain the integrity of data as well as for query optimization [2]. *Path constraints* are generalizations of integrity constraints in the context of semistructured data [1, 26]. They are navigation conditions imposing conditions on nodes at arbitrary depths in the data graph. They include functional, inclusion and inverse path constraints [26].

Consider an example of a semistructured database containing information about authors and publications. Each author has a corresponding list of publications and each publication has a corresponding list of authors. An example in XML is shown in Figure 5. It is reasonable to impose the constraints such as the following on this type of data:

1. for each author a , all the publications in a 's publications list should be contained in the database;
2. for each publication p , all the authors in p 's authors list should be contained in the database;

introduced above. Formally, a constraint c in $\mathcal{L}_{\text{con}}$ has the following syntax:

$$\begin{aligned} c &= \text{atom} \mid c \wedge c \mid c \vee c \mid \neg c \\ \text{atom} &= \tau[S] \rightarrow \tau \mid \tau.p \rightarrow \tau.q \mid p \rightarrow q \mid p \rightarrow q \mid r \rightarrow p \rightarrow q \mid r \rightarrow p \rightarrow q, \end{aligned}$$

where τ is an attribute name, S is a set of attribute names, and r, p, q are regular expressions on attribute names.

Next we show how to express the constraints in $\mathcal{L}_{\text{con}}$ within hybrid dynamic logic. We will define a mapping σ from the constraint language $\mathcal{L}_{\text{con}}$ to HDL($\odot, \downarrow, \uparrow$). Let root be a nominal for the root of the data graph. The easiest constraints to encode are the circular ones. Their encodings do not require the use of hybrid binders:

$$\begin{aligned} \sigma(p \rightarrow q) &= \odot_{\text{root}}[p](q)^{-1} \cdot \text{root} \\ \sigma(p \rightarrow q) &= \odot_{\text{root}}[p](q)/\text{root} \end{aligned}$$

We encode lollipop constraints. Their encodings require only one nesting of the hybrid binder $\downarrow$:

$$\begin{aligned} \sigma(r \rightarrow p \rightarrow q) &= \odot_{\text{root}}[r](\downarrow x.[p](q)^{-1} \cdot x) \\ \sigma(r \rightarrow p \rightarrow q) &= \odot_{\text{root}}[r](\downarrow x.[p](q)/x) \end{aligned}$$

Key and functional constraints are more involved, since they involve both the comparison of node identities and the comparison of node values. The translation of the key constraint $\tau[S] \rightarrow \tau$ states that, for all different τ nodes x and y , if x and y agree on the values of nodes reachable through attributes in S , then x and y are the same node:

$$\begin{aligned} \sigma(\tau[S] \rightarrow \tau) &= \mathbf{A}_{\downarrow x} \mathbf{A}_{\downarrow y} (\odot_x(\tau)^{-1} \uparrow \wedge \odot_y(\tau)^{-1} \wedge x \neq y) \rightarrow \\ &\quad \left(\bigwedge_{a \in S} \odot_x[a](\downarrow v_1. \odot_y(a)(\downarrow v_1.v_1) = w_1 \wedge \right. \\ &\quad \left. \odot_y[a](\downarrow v_2. \odot_x(a)(\downarrow v_2.v_2) = w_2) \rightarrow x = y \right) \end{aligned}$$

Finally, the translation of the path functional constraints $\tau.p \rightarrow \tau.q$ states that, for all different τ nodes x and y , if x and y agree on the values of nodes reachable through p , then they should agree on the values of nodes reachable through q as well:

$$\begin{aligned} \sigma(\tau.p \rightarrow \tau.q) &= \mathbf{A}_{\downarrow x} \mathbf{A}_{\downarrow y} (\odot_x(\tau)^{-1} \uparrow \wedge \odot_y(\tau)^{-1} \wedge x \neq y) \rightarrow \\ &\quad (\odot_x[p](\downarrow v_1. \odot_y(p)(\downarrow v_1.v_1) = w_1 \wedge \\ &\quad \odot_y[p](\downarrow v_2. \odot_x(p)(\downarrow v_2.v_2) = w_2) \rightarrow \\ &\quad (\odot_x[q](\downarrow v_1. \odot_y(q)(\downarrow v_1.v_1) = w_1 \wedge \\ &\quad \odot_y[q](\downarrow v_2. \odot_x(q)(\downarrow v_2.v_2) = w_2)) \end{aligned}$$

We conclude the description of our translation by stipulating that σ distributes over the Boolean operators. In σ it is convenient to replace the universal modality $\mathbf{A}$ with $\odot_{\text{root}}[\ast]$. Notice that in this way each translated constraint is a formula starting with $\odot_{\text{root}}$. Given a semistructured database D and an integrity constraint $c \in \mathcal{L}_{\text{con}}$, it is possible to check whether D satisfies c as follows. The database D is represented as a rooted graph G_D and the constraint c is translated into the hybrid formula $\sigma(c)$. Then, the formula $\sigma(c)$ is checked on G_D by using a hybrid model checker (in fact, it is sufficient to check the formula at the root of the graph). If the outcome of the model checker is the empty set of nodes, then D does not satisfy c , otherwise it does.

Proposition 5.7 *Let c be an integrity constraint in $\mathcal{L}_{\text{con}}$ and D be a semistructured database. Then, D satisfies c if, and only if, the truth set of $\sigma(c)$ with respect to G_D is non-empty.*

Notice that the translation σ embeds (any Boolean combination of) circular constraints into HDL($\odot$), lollipop constraints into the fragment of HDL($\odot, \downarrow$) in which $\downarrow$ is nested only once, and key and functional constraints into the fragment of HDL($\odot, \downarrow, \uparrow, \mathbf{A}$) in which the hybrid binders are nested a fixed number of times. As a consequence, by virtue of Theorem 4.7, we have the following corollary.

Corollary 5.8

- The constraint evaluation problem for circular constraints can be solved in linear time both in the length of the constraint and in the size of the database;
- The constraint evaluation problem for lollipop constraints can be solved in linear time in the length of the constraint and in quadratic time in the size of the database;
- The constraint evaluation problem for key and functional constraints can be solved in linear time in the length of the constraint and in polynomial time in the size of the database.

6 Conclusion and Work for the Future

We investigated the model checking problem for hybrid logics. We gave model checkers for a large number of fragments of hybrid and hybrid dynamic logic. We formally proved computational complexity lower bounds for the model checking problem for hybrid logics with binders. We found that the addition of nominals and the $\odot$ operator does not increase the complexity of the model checking task. In contrast, whenever hybrid binders are present in the language, the running time of the resulting model checker is exponential in the nesting level of the binders. We cannot do better, since we proved that the model checking problem for hybrid logics with binders is PSPACE-complete.

We applied our findings to the problems of query and constraint evaluation for semistructured data. We identified significant fragments of well-known query and constraint languages for semistructured data that can be efficiently embedded into hybrid languages. These embeddings allowed us to solve query and constraint evaluation problems via model checking for hybrid logics.

In [28] we used hybrid logics as a specification language for *mobile systems*. Roughly speaking, a mobile system is a program that exploits a wide area computational infrastructure like the World-Wide-Web. It consists of several spatially distributed and dynamically connected components that may go mobile among locations. From a technical point of view, two main ingredients of mobility can be isolated: the notion of *location* and the change of spatial configurations of locations over time. The notion of location can be naturally expressed in hybrid logics with nominals to give names to locations, and with the $\odot$ operator to access a location through its name. We believe that interesting mobile requirements may be encoded in hybrid logics and that the model checking algorithms presented in this paper may be used to verify whether the mobile system enjoys its specification; the details remain to be examined, though.

An important practical question remains open: does the model checking-based approach we proposed in this paper work in practice? The only way to find an answer to this question is to implement a prototype model checker of hybrid languages, run some tests and compare the experimental results with alternative query processors and path constraint evaluators. This is our goal for the future. The following issues should be taken into account: (i) semistructured databases are often redundant. In particular, the skeletal information contained in XML documents is often repeated [19]. Our aim is to investigate compact representations of semistructured databases. (ii) Usually, a query only accesses a limited portion of the semistructured database [33]. Our goal is to study a systematic way to avoid the processing of database regions that are irrelevant for the query, and to integrate this projection technique with the database compression. (iii) Large semistructured databases need to be stored on secondary memory. For this, relational databases have been proposed (e.g., in [38]). However, it is not clear how the above compression and projection techniques match with the storage of a semistructured databases into a relational database.

Acknowledgments

Massimo Franceschet was supported by a grant from the Netherlands Organization for Scientific Research (NWO) under project number 612.000.207. Maarten de Rijke was supported by grants

from NWO, under project numbers 365-20-005, 612.069.006, 612.000.106, 220-80-001, 612.000.207, and 612.066.302.

References

- [1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 2000.
- [2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, Reading, Mass., 1995.
- [3] S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. Wiener, and J. Widom. The Lorel Query Language for Semistructured Data. *International Journal on Digital Libraries*, 1(1):68–88, 1997.
- [4] S. Abiteboul and V. Vianu. Regular path queries with constraints. *Journal of Computer and System Sciences*, 58(3):428–452, 1999.
- [5] L. Afanasiev, M. Franceschet, M. Marx, and M. de Rijke. CTL Model Checking for Processing Simple XPath Queries. In *Proceedings Temporal Representation and Reasoning (TIME 2004)*, 2004.
- [6] N. Alechina and M. de Rijke. Describing and querying semistructured data: Some expressiveness results. In S.M. Embury, N.J. Fiddian, W.A. Gray, and A.C. Jones, editors, *Advances in Databases*, LNCS, Springer, 1998.
- [7] N. Alechina, S. Deniri, and M. de Rijke. A modal perspective on path constraints. *Journal of Logic and Computation*, 13(6):939–956, 2003.
- [8] L. De Alfaro. Model checking the World Wide Web. In *Proceedings of Conference on Computer Aided Verification (CAV)*, volume 2102 of *LNCS*, pages 337–349, 2001.
- [9] C. Areces and P. Blackburn. Bringing them all together. *Journal of Logic and Computation*, 11(5):657–669, 2001. Special Issue on Hybrid Logics. Areces, C. and Blackburn, P. (eds.).
- [10] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, volume 1683 of *LNCS*, pages 307–321. Springer, 1999.
- [11] C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.
- [12] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation, and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.
- [13] C. Areces and J. Heguiabehere. HyLoRes 1.0: Direct resolution for hybrid logics. In Andrei Voronkov, editor, *Automated Deduction – CADE-18*, volume 2392 of *LNCS*, pages 156–160. Springer-Verlag, July 27–30 2002.
- [14] P. Blackburn. Representation, reasoning, and relational structures: A hybrid logic manifesto. *Logic Journal of the IGPL*, 8(3):339–365, 2000.
- [15] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [16] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyashev, editors, *Advances in Modal Logic, Volume 1*, pages 41–62. CSLI Publications, 1998.
- [17] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.
- [18] P. Buneman, W. Fau, and S. Weinstein. Path constraints on semistructured and structured data. In *Proceedings of the Symposium on Principles of Database Systems (PODS, 1998)*.
- [19] P. Buneman, M. Grohe, and C. Koch. Path queries on compressed XML. In *Proceedings of the International Conference on Very Large Data Bases (VLBD)*, 2003.

- [20] D. Calvane, G. De Giacomo, and M. Lenzerini. Representing and reasoning on XML documents: A description logic approach. *Journal of Logic and Computation*, 9(3):295–318, 1999.
- [21] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.
- [22] E. M. Clarke and H. Schlingloff. Model checking. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 24, pages 1635–1790. Elsevier Science, 2001.
- [23] R. Cleaveland and B. U. Steffen. A linear-time model checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2:121–147, 1993.
- [24] World Wide Web Consortium. Extensible markup language (XML). Available at <http://www.w3.org/XML>, 1998.
- [25] E.A. Emerson. Temporal and modal logic. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 995–1072. Elsevier Science Publishers B.V., 1990.
- [26] W. Fan and J. Siméon. Integrity constraints for XML. *Journal of Computer and System Sciences*, 66(1):254–291, 2003.
- [27] M. Franceschet, M. de Rijke, and B. H. Schlingloff. Hybrid logics on linear structures: expressivity and complexity. In *Proceedings of the International Symposium on Temporal Representation and Reasoning - International Conference on Temporal Logic*. IEEE Computer Society Press, 2003.
- [28] M. Franceschet, A. Montanari, and M. de Rijke. Model checking for combined logics with an application to mobile systems. *Automated Software Engineering*, 2004. Special issue on Distributed and Mobile Software Engineering. To appear.
- [29] G. Gottlob and C. Koch. Monadic Queries over Tree-Structured Data. In *Logic in Computer Science*, pages 189–202. Los Alamitos: CA, USA, July 22–25 2002. IEEE Computer Society.
- [30] J. Y. Halpern, R. Harper, N. Immerman, P. G. Kolaitis, M. Vardi, and V. Vianu. On the unusual effectiveness of logic in computer science. *The Bulletin of Symbolic Logic*, 7(2):213–236, 2001.
- [31] D. Harel, J. Tiurny, and D. Kozen. *Dynamic Logic*. MIT Press, 2000.
- [32] HyLo: The Hybrid Logic home page. URL: <http://www.hylo.net>.
- [33] A. Marian and J. Siméon. Projecting XML documents. In *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pages 213–224. Morgan Kaufmann Publishers, 2003.
- [34] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A database management system for semistructured data. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 26(3):54–66, 1997.
- [35] G. Miklau and D. Suciu. Containment and equivalence for and XPath fragment. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*, pages 65–76, 2002.
- [36] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [37] E. Quintarelli. *Model-Checking Based Data Retrieval: an application to semistructured and temporal data*. PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, 2000.
- [38] J. Shannugundaram, H. Gang, K. Tufte, C. Zhang, D. J. DeWitt, and J. Naughton. Relational databases for querying XML documents: Limitations and opportunities. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 302–314. Morgan Kaufmann, 1999.