A Road-map on Complexity for Hybrid Logics

Carlos Areces[1], Patrick Blackburn[2], and Maarten Marx[3]

[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 PROP $= \{p, q, r, \ldots\}$ be a countable set of *propositional variables* and NOM $= \{i, j, k, \ldots\}$ a countable set of *nominals*, disjoint from PROP. We call ATOM = PROP $\cup$ 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 $a \in$ ATOM, and $i \in$ 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 : ATOM $\to$ Pow(M) is such that for all $i \in$ 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:

$$\begin{array}{ll}
\mathfrak{M}, m \Vdash a & \text{iff } m \in V(a),\ a \in \text{ATOM} \\
\mathfrak{M}, m \Vdash \neg\varphi & \text{iff } \mathfrak{M}, m \nVdash \varphi \\
\mathfrak{M}, m \Vdash \varphi \wedge \psi & \text{iff } \mathfrak{M}, m \Vdash \varphi \text{ and } \mathfrak{M}, m \Vdash \psi \\
\mathfrak{M}, m \Vdash \Box\varphi & \text{iff } \forall m'(Rmm' \Rightarrow \mathfrak{M}, m' \Vdash \varphi) \\
\mathfrak{M}, m \Vdash @_i\varphi & \text{iff } \mathfrak{M}, m' \Vdash \varphi, \text{ where } V(i) = \{m'\},\ i \in \text{NOM}.
\end{array}$$

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 $\mathsf{Holds}(t, \varphi)$-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 $\{X_0, \ldots, X_k\}$ 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_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 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. ∃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 ∃loise has a winning strategy iff ξ is satisfiable.

Now the right-to-left direction is clear: ∃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 ∃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 $\{X_0, \ldots, X_n\}$. 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 ∀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 ∀belard challenged this defect, so ∃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 $\mathfrak{M}, m \not\Vdash \Diamond\varphi$.

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 interrelated: 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 interrelated 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: $p^t = p$, $(\neg\varphi)^t = \neg\varphi^t$, $(\varphi \wedge \psi)^t = \varphi^t \wedge \psi^t$, $(\Diamond\varphi)^t = \mathsf{F}(\mathsf{P}i \wedge \varphi^t)$. 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 $i \wedge \mathsf{G}(\mathsf{P}i \to \varphi^t)$ is K-satisfiable.

For the left to right direction, let $\mathfrak{M} \models \varphi$, where $\mathfrak{M} = \langle M, R, V\rangle$ is a 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 $\Diamond$:

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

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

For the other direction, let $\mathfrak{M}, w \Vdash i \wedge \mathsf{G}(\mathsf{P}i \rightarrow \varphi^t)$, where $\mathfrak{M} = \langle M, R, V \rangle$ is a hybrid model. Define $\mathfrak{M}^*$ as follows: $M^* = \{m \in M \mid Rwm\}$, $R^* = R_{\restriction M^*}$, $V^* = V_{\restriction M^*}$. 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 $\Diamond$:

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

For all $m \in M^*$, Rwm holds, whence for all $m \in M^*$, $\mathfrak{M}, m \Vdash \mathsf{P}i$. So, since $\mathfrak{M}, w \Vdash \mathsf{G}(\mathsf{P}i \rightarrow \varphi^t)$, 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 $\mathsf{Holds}(t, \varphi)$-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:

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

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, PROP $= \{p, q, r, \ldots\}$ is a countable set of propositional variables, and NOM $= \{i, j, k, \ldots\}$ is a countable set of nominals. To this we add SVAR $= \{x_1, x_2, \ldots\}$ a countable set of *state variables*. We assume that PROP, NOM and SVAR are pairwise disjoint. We call SSYM $=$ NOM $\cup$ SVAR the set of *state symbols*, and ATOM $=$ PROP $\cup$ NOM $\cup$ 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 $a \in$ ATOM, $v \in$ SVAR and $s \in$ 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 : SVAR $\to M$. Given an assignment g, we define the assignment g^v_m by $g^v_m(v') = g(v')$ for $v' \neq v$ and $g^v_m(v) = m$. We say that g^v_m 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:

$$
\begin{array}{ll}
\mathfrak{M}, g, m \Vdash a & \text{iff } m \in [V,g](a),\ a \in \text{ATOM} \\
\mathfrak{M}, g, m \Vdash \neg\varphi & \text{iff } \mathfrak{M}, g, m \not\Vdash \varphi \\
\mathfrak{M}, g, m \Vdash \varphi \wedge \psi & \text{iff } \mathfrak{M}, g, m \Vdash \varphi \text{ and } \mathfrak{M}, g, m \Vdash \psi \\
\mathfrak{M}, g, m \Vdash \Box\varphi & \text{iff } \forall m'(Rmm' \Rightarrow \mathfrak{M}, g, m' \Vdash \varphi) \\
\mathfrak{M}, g, m \Vdash \downarrow v.\varphi & \text{iff } \mathfrak{M}, g^v_m, m \Vdash \varphi \\
\mathfrak{M}, g, m \Vdash @_s\varphi & \text{iff } \mathfrak{M}, g, m' \Vdash \varphi, \text{ where } [V,g](s) = \{m'\},\ s \in \text{SSYM}.
\end{array}
$$

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 $p_j \in$ 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 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 $\mathfrak{M} \Vdash ST_x(\varphi)[g^x_m]$.*

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:

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

where x_i 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:

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

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, $\mathfrak{M} \Vdash \varphi[g]$ 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:

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 $@_s(\Box\downarrow y.\Box\Box\downarrow x_1.@_y\Box\Box\downarrow x_2.@_y\Box\Box\downarrow x_3.@_y\Box\Box\downarrow x_4.(\bigvee_{1\leq i\neq j\leq 4} @_{x_i}x_j))$.

What does *Grid* express? Suppose it is satisfied in a model $\mathfrak{M}$ on a frame (W, R). 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.

$$\begin{array}{ll}
& \mathfrak{M}, g, s \Vdash \Box\downarrow y.\Box\downarrow x_1.@_y\Box\downarrow x_2.@_{x_1}x_2 \\
\iff & (\forall w : sRw) : \mathfrak{M}, g^y_w, w \Vdash \Box\downarrow x_1.@_y\Box\downarrow x_2.@_{x_1}x_2 \\
\iff & (\forall u : wRu) : \mathfrak{M}, (g^y_w)^{x_1}_u, u \Vdash @_y\Box\downarrow x_2.@_{x_1}x_2 \\
\iff & \mathfrak{M}, (g^y_w)^{x_1}_u, w \Vdash \Box\downarrow x_2.@_{x_1}x_2 \\
\iff & (\forall v : wRv) : \mathfrak{M}, ((g^y_w)^{x_1}_u)^{x_2}_v, v \Vdash .@_{x_1}x_2 \\
\iff & (\forall w : sRw)(\forall u : wRu)(\forall v : wRv) : u = v.
\end{array}$$

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 $\mathfrak{M} = (W, R, V)$ 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 $\mathfrak{M}, s \Vdash \downarrow s.(\neg\Diamond s \wedge \Box\Box \downarrow x.\Diamond(s \wedge \Diamond x) \wedge \Box\Diamond s)$. 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 $u \in M$, $\mathfrak{M}_s, g, u \Vdash @_x\varphi$ iff $\mathfrak{M}_s, g, u \Vdash @_s(\varphi \vee \Diamond(x \wedge \varphi))$.

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 $\{p_1, \ldots, p_n\}$ of propositional variables. The spypoint model of $\mathfrak{M}$ (notation $\mathsf{Spy}[\mathfrak{M}]$) is the structure $\langle M', R', V'\rangle$ in which

i. $M' = M \cup \{s\} \cup \{w_{p_1}, \ldots, w_{p_n}\}$, for $s, w_{p_1}, \ldots, w_{p_n} \notin M$

ii. $R' = R \cup \{(s, x), (x, s) \mid x \in M' \backslash \{s\}\} \cup \{(x, w_{p_i}) \mid x \in M \text{ and } x \in V(p_i)\}$

iii. $V' = \emptyset$.

Let $\{s, x_{p_1}, \ldots, x_{p_n}\}$ 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 $\neg s \wedge \neg x_{p_1} \wedge \ldots \wedge \neg x_{p_n}$. Note that when evaluated under the spypoint assignment, the denotation of $\mathbf{m}$ in $\mathsf{Spy}[\mathfrak{M}]$ is precisely M.

$\mathsf{Spy}[\mathfrak{M}]$ 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,*

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

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 Propostion 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,*

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

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 $\{p_1, \ldots, p_n\}$ 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

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

where

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

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 $\mathsf{Spy}[\mathfrak{M}], s \Vdash F$. The first conjunct of F is true in $\mathsf{Spy}[\mathfrak{M}]$ 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 $\mathsf{Spy}[\mathfrak{M}], g, s \Vdash DIS$. By Propositions 4 and 5, also $\mathsf{Spy}[\mathfrak{M}], g, s \Vdash VAL \wedge FR$.

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

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

Note that $\mathsf{Spy}[\mathfrak{M}']$ 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 dissallow disjuctions (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.

[BGG97] 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. Zakharyaschev, 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.