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 multigraphs). 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) multimodal logic.

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

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

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 $(W, \{R_\pi \mid \pi \in \text{MOD}\}, V)$. 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 $(W, \{R_\pi \mid \pi \in \text{MOD}\})$ 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 $\mathcal{M} = (W, \{R_\pi \mid \pi \in \mathrm{MOD}\}, V)$ and $w \in W$. Then:

$$
\begin{array}{lll}
\mathcal{M}, w \Vdash p & \text{iff} & w \in V(p), \text{ where } p \in \mathrm{PROP} \\
\mathcal{M}, w \Vdash \neg\varphi & \text{iff} & \mathcal{M}, w \nVdash \varphi \\
\mathcal{M}, w \Vdash \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ and } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \varphi \vee \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ or } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \varphi \to \psi & \text{iff} & \mathcal{M}, w \nVdash \varphi \text{ or } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \langle\pi\rangle\varphi & \text{iff} & \exists w'(wR_\pi w' \ \& \ \mathcal{M}, w' \Vdash \varphi) \\
\mathcal{M}, w \Vdash [\pi]\varphi & \text{iff} & \forall w'(wR_\pi w' \ \Rightarrow \ \mathcal{M}, w' \Vdash \varphi).
\end{array}
$$

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 $\mathcal{M} = (W, \{R_\pi \mid \pi \in \mathrm{MOD}\}, V)$. The underlying frame $(W, \{R_\pi \mid \pi \in \mathrm{MOD}\})$ 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 $\mathcal{M} = (W, \{R_\pi \mid \pi \in \mathrm{MOD}\}, \{V(p) \mid p \in \mathrm{PROP}\})$.

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:

$$\begin{array}{lcl} ST_x(p) & = & \mathsf{P}(x),\ p \in \mathrm{PROP} \\ ST_x(\neg\varphi) & = & \neg ST_x(\varphi) \\ ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\psi) \\ ST_x(\langle\pi\rangle\varphi) & = & \exists y(x\mathsf{R}_\pi y \wedge ST_y(\varphi)) \\ ST_x([\pi]\varphi) & = & \forall y(x\mathsf{R}_\pi y \rightarrow ST_y(\varphi)). \end{array}$$

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 $\mathcal{M}, w \Vdash \varphi$ 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 $= \{\mathsf{F}, \mathsf{P}\}$ in this paper: the modality $\langle \mathsf{F} \rangle$ means "at some Future state", and $\langle \mathsf{P} \rangle$ 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 $\langle \mathsf{F} \rangle$ looks forward along the flow of time, and $\langle \mathsf{P} \rangle$ backwards.

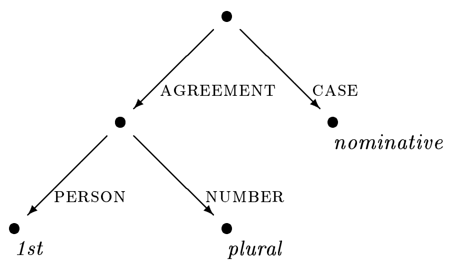
Consider the formula $\langle \mathsf{P} \rangle \mathsf{Mia\text{-}unconscious}$. This is true iff we can look back in time from the current state and see a state where Mia is unconscious. Similarly $\langle \mathsf{F} \rangle \mathsf{Mia\text{-}unconscious}$ requires us to scan the states that lie 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 $\langle \mathsf{SUB} \rangle$ (meaning "at some SUBinterval of the current state") and $\langle \mathsf{SUP} \rangle$ (meaning "at some SUPerinterval of the current state"). Then a formula of the form $\langle \mathsf{SUB} \rangle \langle \mathsf{F} \rangle \langle \mathsf{SUP} \rangle p$ 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 infor-

mation:



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

$$\begin{bmatrix} \textsc{agreement} & \begin{bmatrix} \textsc{person} & \textit{1st} \\ \textsc{number} & \textit{plural} \end{bmatrix} \\ \textsc{case} & -\textit{dative} \end{bmatrix}$$

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:

$$\begin{array}{ll} & \langle \textsc{agreement} \rangle (\langle \textsc{person} \rangle \textit{1st} \wedge \langle \textsc{number} \rangle \textit{plural}) \\ \wedge & \langle \textsc{case} \rangle \neg \textit{dative} \end{array}$$

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 $\mathcal{ALC}$ using the following expression:

$$\mathsf{killer} \sqcap \exists \textsc{employer}.\mathsf{gangster}$$

Here killer and $\mathsf{gangster}$ 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:

$$\mathsf{killer} \wedge \langle \textsc{employer} \rangle \mathsf{gangster}$$

Indeed, as Schild [49] pointed out, *any* $\mathcal{ALC}$ 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 Andréka, van Benthem, and Németi [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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$ 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, $\langle \mathsf{P} \rangle$Vincent-accidentally-squeeze-the-trigger, fails to capture this.

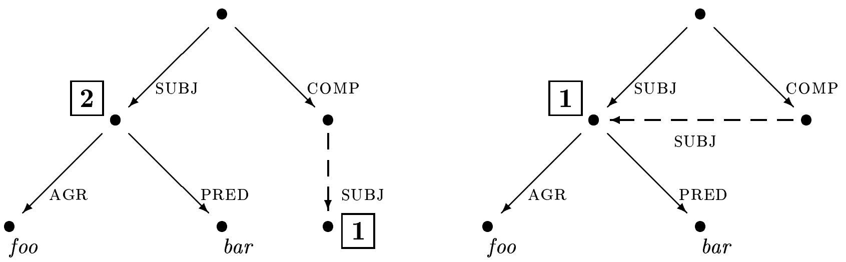
Similarly, while it's certainly possible to abstract elegant modal logics from the work of James Allen, such abstractions amputate a central feature of his work: *reference to specific intervals.* Allen's formalism includes the notation $\mathbf{Hold}(\mathsf{P}, i)$ meaning "the property P holds at 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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$. 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: $\langle \mathsf{F} \rangle \langle \mathsf{F} \rangle p \rightarrow \langle \mathsf{F} \rangle p$ 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:

$$\begin{bmatrix} \text{SUBJ} & \boxed{\mathbf{1}}\begin{bmatrix}\text{AGR} & \mathit{foo}\\ \text{PRED} & \mathit{bar}\end{bmatrix} \\ \text{COMP} & [\text{SUBJ}\ \boxed{\mathbf{1}}\,] \end{bmatrix}$$

The 'tag' $\boxed{\mathbf{1}}$ 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 $\boxed{\mathbf{1}}$. For this reason, the AVM is *not* satisfied on the left hand feature below (which otherwise gets everything right) but *is* satisfied by the right hand feature structure:



(ii) (i)

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 (Pumpkin, Honey-Bunny):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 $\mathcal{O}(\mathsf{Jules}, \ldots, \mathsf{Vincent})$ picks out one of the individuals Jules, ..., Vincent, and $\mathcal{O}(\mathsf{Mia})$ 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 it's 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:

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

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 $(W, \{R_\pi \mid \pi \in \mathrm{MOD}\}, V)$ where $(W, \{R_\pi \mid \pi \in \mathrm{MOD}\})$ 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:

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

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:

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

Clearly $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M} \models ST_x(\varphi)[w, V(i), \ldots, V(j)]$, where x, x_i, $\ldots$ 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 out 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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$, it can be with the help of nominals: $\langle \mathsf{P} \rangle (i \wedge \mathsf{Vincent\text{-}accidentally\text{-}squeeze\text{-}the\text{-}trigger})$ 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 $\mathbf{Hold}(i, \varphi)$ construct. More on this in Section 6.

Third, we also solve the deeper issue concerning definability: $i \rightarrow \neg\langle \mathsf{F} \rangle\langle \mathsf{F} \rangle i$ 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:

$$\begin{bmatrix} \text{SUBJ} & \boxed{1}\begin{bmatrix} \text{AGR} & \textit{foo} \\ \text{PRED} & \textit{bar} \end{bmatrix} \\ \text{COMP} & [\text{SUBJ} \ \ \boxed{1}\,] \end{bmatrix}$$

This corresponds to the following L^N wff:

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

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 $\mathcal{ALCO}$ (that is, $\mathcal{ALC}$ enriched with the $\mathcal{O}$ 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 $\mathcal{O}(i)$. Conversely, every $\mathcal{ALCO}$ expression of the form $\mathcal{O}(i, \ldots, j)$ corresponds to the formula $i \vee \cdots \vee j$.

Furthermore, @ has a natural description logic interpretation. The ABox specification $i : \varphi$ corresponds to the satisfaction statement $@_i\varphi$, and the specification $(i, j) : \text{R}$ corresponds to $@_i\langle\text{R}\rangle j$. 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 $\mathcal{ALCO}$ 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:

$$\frac{@_s\neg\varphi}{\neg @_s\varphi}\ [\neg] \qquad\qquad \frac{\neg @_s\neg\varphi}{@_s\varphi}\ [\neg\neg]$$

$$\frac{@_s(\varphi\wedge\psi)}{\begin{array}{c}@_s\varphi\\ @_s\psi\end{array}}\ [\wedge] \qquad\qquad \frac{\neg @_s(\varphi\wedge\psi)}{\neg @_s\varphi \mid \neg @_s\psi}\ [\neg\wedge]$$

$$\frac{@_s @_t\varphi}{@_t\varphi}\ [@] \qquad\qquad \frac{\neg @_s @_t\varphi}{\neg @_t\varphi}\ [\neg @]$$

$$\frac{@_s\langle\pi\rangle\varphi}{\begin{array}{c}@_s\langle\pi\rangle a\\ @_a\varphi\end{array}}\ [\langle\pi\rangle] \qquad\qquad \frac{\neg @_s\langle\pi\rangle\varphi \quad @_s\langle\pi\rangle t}{\neg @_t\varphi}\ [\neg\langle\pi\rangle]$$

$$\frac{@_s[\pi]\varphi \quad @_s\langle\pi\rangle t}{@_t\varphi}\ [[\pi]] \qquad\qquad \frac{\neg @_s[\pi]\varphi}{\begin{array}{c}@_s\langle\pi\rangle a\\ \neg @_a\varphi\end{array}}\ [\neg[\pi]]$$

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 @:

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

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: $\langle\pi\rangle(p\vee q)\to\langle\pi\rangle p\vee\langle\pi\rangle q$ is valid (for any modality $\langle\pi\rangle$), hence this formula should be provable. Here's how to do it:

1	$\neg @_i(\langle\pi\rangle(p\vee q)\to\langle\pi\rangle p\vee\langle\pi\rangle q)$	
2	$@_i\langle\pi\rangle(p\vee q)$	$1,\neg\!\to$
$2'$	$\neg @_i(\langle\pi\rangle p\vee\langle\pi\rangle q)$	Ditto
3	$\neg @_i\langle\pi\rangle p$	$2',\neg\vee$
$3'$	$\neg @_i\langle\pi\rangle q$	Ditto
4	$@_i\langle\pi\rangle j$	$2,\langle\pi\rangle$
$4'$	$@_j(p\vee q)$	Ditto
5	$\neg @_j p$	$3,4,\neg\langle\pi\rangle$
6	$\neg @_j q$	$3',4,\neg\langle\pi\rangle$
7	$@_j p$ | $@_j q$	$4',\vee$
	✠ $5,7$ ✠ ✠ $6,7$ ✠	

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 $\langle\textsc{vert}\rangle$, $\langle\textsc{hor}\rangle$ and $\langle\textsc{diag}\rangle$ (for *vertical*, *horizontal* and *diagonal*) respectively. Now, $\langle\textsc{hor}\rangle\langle\textsc{vert}\rangle(i\wedge p)\wedge\langle\textsc{diag}\rangle i\to\langle\textsc{diag}\rangle p$ is valid (for there's only one state named i) and we can prove it as follows:

1	$\neg @_j(\langle\textsc{hor}\rangle\langle\textsc{vert}\rangle(i\wedge p)\wedge\langle\textsc{diag}\rangle i\to\langle\textsc{diag}\rangle p)$	
2	$@_j(\langle\textsc{hor}\rangle\langle\textsc{vert}\rangle(i\wedge p)\wedge\langle\textsc{diag}\rangle i)$	$1,\neg\!\to$
$2'$	$\neg @_j\langle\textsc{diag}\rangle p$	Ditto
3	$@_j\langle\textsc{hor}\rangle\langle\textsc{vert}\rangle(i\wedge p)$	$2,\wedge$
$3'$	$@_j\langle\textsc{diag}\rangle i$	Ditto
4	$@_j\langle\textsc{hor}\rangle k$	$3,\langle\textsc{hor}\rangle$
$4'$	$@_k\langle\textsc{vert}\rangle(i\wedge p)$	Ditto
5	$@_k\langle\textsc{vert}\rangle l$	$4',\langle\textsc{vert}\rangle$
$5'$	$@_l(i\wedge p)$	Ditto
6	$@_l i$	$5',\wedge$
$6'$	$@_l p$	Ditto
7	$@_i l$	$6,$ Sym
8	$@_i p$	$6',7,$ Nom
9	$\neg @_i p$	$2',3',\neg\langle\textsc{diag}\rangle$
	✠ $8,9$ ✠	

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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$ 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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$. Using pure formulas, we can define

a number of properties relevant to temporal logic:

$@_i\neg\langle\mathsf{F}\rangle i$	$\forall x\neg(xR_{\mathsf{F}}x)$ *(Irreflexivity)*
$@_i\neg\langle\mathsf{F}\rangle\langle\mathsf{F}\rangle i$	$\forall xy(xR_{\mathsf{F}}y \rightarrow \neg yR_{\mathsf{F}}x)$ *(Asymmetry)*
$@_i[\mathsf{F}](\langle\mathsf{F}\rangle i \rightarrow i)$	$\forall xy(xR_{\mathsf{F}}y \wedge yR_{\mathsf{F}}x \rightarrow x = y)$ *(Antisymmetry)*
$\langle\mathsf{F}\rangle\langle\mathsf{F}\rangle i \rightarrow \langle\mathsf{F}\rangle i$	$\forall xyz(xR_{\mathsf{F}}y \wedge yR_{\mathsf{F}}z \rightarrow xR_{\mathsf{F}}z)$ *(Transitivity)*
$\langle\mathsf{F}\rangle i \rightarrow \langle\mathsf{F}\rangle\langle\mathsf{F}\rangle i$	$\forall xy(xR_{\mathsf{F}}y \rightarrow \exists z(xR_{\mathsf{F}}z \wedge zR_{\mathsf{F}}y))$ *(Density)*
$@_i\langle\mathsf{F}\rangle j \vee @_i j \vee @_j\langle\mathsf{F}\rangle i$	$\forall xy(xR_{\mathsf{F}}y \vee x = y \vee yR_{\mathsf{F}}z)$ *(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:

$@_i[\mathsf{F}]\langle\mathsf{P}\rangle i$	$\forall xy(xR_{\mathsf{F}}y \rightarrow yR_{\mathsf{P}}x)$
$@_i[\mathsf{P}]\langle\mathsf{F}\rangle i$	$\forall xy(xR_{\mathsf{P}}y \rightarrow yR_{\mathsf{F}}x)$

It follows that the conjunction $@_i[\mathsf{F}]\langle\mathsf{P}\rangle i \wedge @_i[\mathsf{P}]\langle\mathsf{F}\rangle i$ 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 $\langle\mathsf{F}\rangle$ as looking forward along some binary relation $<$ (the "flow of time") and $\langle\mathsf{P}\rangle$ as looking backwards along the same relation. This enables us to define further temporally interesting properties:

$\langle\mathsf{P}\rangle\langle\mathsf{F}\rangle i$	$\forall xy\exists z(z < x \wedge z < y)$ *(Left-Directedness)*
$@_i(\langle\mathsf{F}\rangle\top \rightarrow \langle\mathsf{F}\rangle[\mathsf{P}][\mathsf{P}]\neg i)$	$\forall xy(x<y \rightarrow \exists z(x<z \wedge \neg\exists w(x<w<z)))$ *(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 $\mathsf{H}+\mathsf{A}$ be the tableau system that uses the formulas in A as axioms. That is, for any α in A, and any nominals $j, j_1, \ldots, j_n$ that occur on a branch of a tableau, we are free to add α or $\alpha[j_1/i_1, \ldots, j_n/i_n]$ to the end of that branch (here $i_1 \ldots, i_n$ are nominals in α, and $\alpha[j_1/i_1, \ldots, j_n/i_n]$ 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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$ 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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$ really are mutually converse, add the axioms $@_i[\mathsf{F}]\langle \mathsf{P} \rangle i$ and $@_i[\mathsf{P}]\langle \mathsf{F} \rangle i$; 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 $\langle \mathsf{F} \rangle\langle \mathsf{F} \rangle i \rightarrow \langle \mathsf{F} \rangle i$ defines this property. This is not a satisfaction statement, but $@_j(\langle \mathsf{F} \rangle\langle \mathsf{F} \rangle i \rightarrow \langle \mathsf{F} \rangle i)$ 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), $\langle \mathsf{P} \rangle\langle \mathsf{P} \rangle \varphi \rightarrow \langle \mathsf{P} \rangle \varphi$ 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			$\neg @_i(\langle \mathsf{P} \rangle\langle \mathsf{P} \rangle \varphi \rightarrow \langle \mathsf{P} \rangle \varphi)$		
2			$@_i\langle \mathsf{P} \rangle\langle \mathsf{P} \rangle \varphi$		$1, \neg\rightarrow$
2′			$\neg @_i\langle \mathsf{P} \rangle \varphi$		Ditto
3			$@_i\langle \mathsf{P} \rangle j$		$2, \langle \mathsf{P} \rangle$
3′			$@_j\langle \mathsf{P} \rangle \varphi$		Ditto
4			$@_j\langle \mathsf{P} \rangle k$		$3', \langle \mathsf{P} \rangle$
4′			$@_k \varphi$		Ditto
5			$@_j[\mathsf{P}]\langle \mathsf{F} \rangle j$		Axiom
6			$@_k\langle \mathsf{F} \rangle j$		$4, 5, [\mathsf{P}]$
7			$@_i[\mathsf{P}]\langle \mathsf{F} \rangle i$		Axiom
8			$@_j\langle \mathsf{F} \rangle i$		$3, 7, [\mathsf{P}]$
9			$@_k(\langle \mathsf{F} \rangle\langle \mathsf{F} \rangle i \rightarrow \langle \mathsf{F} \rangle i)$		Axiom
10		$\neg @_k\langle \mathsf{F} \rangle\langle \mathsf{F} \rangle i$	|	$@_k\langle \mathsf{F} \rangle i$	$9, \rightarrow$
11	$6, 10, \neg\langle \mathsf{F} \rangle$	$\neg @_j\langle \mathsf{F} \rangle i$		$@_k[\mathsf{F}]\langle \mathsf{P} \rangle k$	Axiom
12		✠ 8, 11 ✠		$@_i\langle \mathsf{P} \rangle k$	$10, 11, [\mathsf{F}]$
13				$\neg @_k \varphi$	$2', 12, \neg\langle \mathsf{P} \rangle$
				✠ 4′, 13 ✠	

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 Vigano'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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$ language can *almost* define $\mathbb{Z}$. As he notes, the formula

$$([\mathsf{P}]([\mathsf{P}]p \to p) \to (\langle \mathsf{P} \rangle [\mathsf{P}]p \to [\mathsf{P}]p)) \wedge ([\mathsf{F}]([\mathsf{F}]p \to p) \to (\langle \mathsf{F} \rangle [\mathsf{F}]p \to [\mathsf{F}]p))$$

(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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$, transitivity, irreflexivity and trichotomy. Furthermore, the formulas $\langle \mathsf{F} \rangle \top$ and $\langle \mathsf{P} \rangle \top$ 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 logic 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:

$$i \wedge \neg\langle \textsc{respect} \rangle i.$$

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:

$$\exists x(x \wedge \neg\langle \textsc{respect} \rangle x).$$

This sentence is true at precisely those 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:

$$\downarrow x.\neg\langle \textsc{respect} \rangle x.$$

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:

$$\mathsf{idiot} \wedge \downarrow x.\langle \textsc{admires} \rangle x.$$

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 ADMIRES relation.

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

$$\downarrow x.(\mathsf{smooth\text{-}talker} \wedge \forall y(\langle \textsc{talks-to} \rangle y \rightarrow \neg @_y \langle \textsc{trusts} \rangle x).$$

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:

$$
\begin{array}{lcl}
HT(x\mathsf{R}_\pi y) & = & @_x\langle\pi\rangle y \\
HT(\mathsf{P}x) & = & @_x p \\
HT(x = y) & = & @_x y \\
HT(\neg\varphi) & = & \neg HT(\varphi) \\
HT(\varphi\wedge\psi) & = & HT(\varphi)\wedge HT(\psi) \\
HT(\exists v\varphi) & = & \exists v HT(\varphi) \\
HT(\forall v\varphi) & = & \forall v HT(\varphi).
\end{array}
$$

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 *SCID4* be the conjunction of the following formulas:

$$
\begin{array}{ll}
S & x\wedge\neg\langle\mathsf{R}\rangle x\wedge\langle\mathsf{R}\rangle\neg x\wedge[\mathsf{R}]\langle\mathsf{R}\rangle x \\
C & [\mathsf{R}][\mathsf{R}]\downarrow y.(\neg x\to\langle\mathsf{R}\rangle(x\wedge\langle\mathsf{R}\rangle y)) \\
I & [\mathsf{R}]\downarrow y.\neg\langle\mathsf{R}\rangle y \\
D & [\mathsf{R}]\langle\mathsf{R}\rangle\neg x \\
4 & [\mathsf{R}]\downarrow y.\langle\mathsf{R}\rangle(x\wedge[\mathsf{R}](\langle\mathsf{R}\rangle(\neg x\wedge\langle\mathsf{R}\rangle y\to\langle\mathsf{R}\rangle y)))
\end{array}
$$

Note that these formulas are pure, and that $\downarrow x.SCID4$ 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 $\mathcal{N}^s, s \Vdash \downarrow x.SCID4$.

Obviously $\mathcal{N}^s$ is an infinite model. In fact *any* model $\mathcal{M} = (W, R, V)$ for $\downarrow x.SCID4$ is infinite. For suppose $\mathcal{M}, s \Vdash \downarrow x.SCID4$. 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 tableaux rules for existential and universal quantifiers:

$$\frac{\neg @_s \exists x \varphi}{\neg @_s \varphi[t/x]} \qquad \frac{@_s \exists x \varphi}{@_s \varphi[a/x]}$$

$$\frac{\neg @_s \forall x \varphi}{\neg @_s \varphi[a/x]} \qquad \frac{@_s \forall x \varphi}{@_s \varphi[t/x]}$$

(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, $\forall x(\varphi \to \psi) \to (\varphi \to \forall x \psi)$, where x does not occur free in φ) we can also prove intrinsically *modal* principles. For example, $\exists x x$ is valid (this says: it is always possible to bind a variable to the current state). We can prove it as follows:

1	$\neg @_i \exists x x$	
2	$\neg @_i i$	$1, \neg\exists$
3	$@_i i$	*Ref*
	✠ 2, 3 ✠	

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: $\mathbf{Hold}(\mathsf{P}, i)$ 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

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

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 representations like

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

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

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

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 $\langle \mathsf{F} \rangle (\mathsf{yesterday} \wedge \varphi)$ 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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$, 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:

$$\langle \textsc{every-fut} \rangle \varphi \ := \ \forall \rho (\rho \rightarrow \langle \mathsf{F} \rangle \exists x (x \wedge \rho \wedge \varphi)).$$

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 $\langle \mathsf{F} \rangle$ and $\langle \mathsf{P} \rangle$ 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 appeal 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/claus`. 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/claus`. 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/claus`.

[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. Zakharyaschev, 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, Universitá di Roma "La Sapienza", 1995.

[23] 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.

[24] 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.

[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. Skordev, 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, Nommer 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