Lectures on Hybrid Logic

NASSLLI'02, First North American Summer School in Logic, Language, and Information, 24–28 June 2002, Stanford

Patrick Blackburn	Maarten Marx
INRIA, Lorraine	University of Amsterdam
patrick@aplog.org	marx@science.uva.nl

This course introduces hybrid logic, a form of modal logic in which it is possible to name worlds (or times, or computational states, or situations, or nodes in parse trees, or people — indeed, whatever it is that the elements of Kripke Models are taken to represent). The course has two main goals. The first is to convey, as clearly as possible, the ideas and intuitions that have guided the development of hybrid logic. The second is to teach a concrete skill: tableau-based hybrid deduction. By the end of the course you will have ample evidence that modal logic can be useful in a wide range of circumstances, and that hybrid logic is a particularly simple way of doing modal logic.

Course Outline

The course consist of five lectures:

Lecture 1: From modal logic to hybrid logic

Lecture 2: Hybrid deduction

Lecture 3: The downarrow binder

Lecture 4: First-order hybrid logic

Lecture 5: The Priorean perspective

Each lecture is one hour long, and will be presented by Patrick Blackburn. The slides (developed by Patrick Blackburn and Maarten Marx) on which the course is based will be made available on the NASSLLI website, and on the hybrid logic homepage (`www.hylo.net`).

The course is relatively self-contained, and we attempt to make the material as accessible as possible to an interdisciplinary audience (that is, the course is not targeted solely at logicians). Nonetheless, we presuppose a certain level of logical literacy. Roughly speaking, to follow this course you should have a reasonable grasp of first-order logic and its semantics. Prior acquaintance with the basics of modal logic would be helpful, but is not essential.

This Reader

This reader consist of three papers:

- "Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto", by Patrick Blackburn. *Logic Journal of the IGPL*, 8(3), 339-625, 2000.
- "Tableaux for Quantified Hybrid Logic" by Patrick Blackburn and Maarten Marx, to appear in *Proceedings of Tableaux 2002*, Automated Reasoning with Analytic Tableaux and Related Methods, Copenhagen, Denmark, July 30th – August 1st, 2002.
- "Beyond Pure Axioms: Node Creating Rules in Hybrid Tableaux" by Patrick Blackburn and Balder ten Cate. Draft manuscript, 2002.

These papers are intended to act as background reading: they should help you understand the themes discussed in class better, and in some cases they develop these themes further, or in different directions.

The "Manifesto" provides an intuitive overview of the full range of (non first-order) hybrid logics, and explains the basics of hybrid tableaux for various hybrid language. It is central to the course, providing background to Lectures 1, 2, 3 and 5.

However the manifesto does not discuss first-order hybrid logic, the topic of Lecture 4. To fill this gap we have supplied "Tableaux for Quantified Hybrid Logic". We have deliberately chosen a paper in which the approach to first-order hybrid logic is slightly *different* from that which will be discussed in class. In Lecture 4 we present first-order hybrid logic under the varying-domain semantics. This paper, on the other hand, uses the (somewhat simpler) constant domain semantics. One of the pleasant aspects of hybrid logic is that either approach works smoothly, and the reader will find it interesting to compare the paper with the course slides. One other remark: modal logicians will find the completeness proof in this paper novel. Instead of using a traditional Henkin- or Hintikka-style argument, completeness is proved by a translation from first-order logic. Non logicians may find this part of the paper tough going.

Finally, the "Beyond Pure Axioms" paper will give you a taste of some recent work in hybrid deduction. Essentially, this paper shows that the results discussed in class about pure axioms can be extended to yield deduction systems for a far wider class of logics. Moreover, the paper also makes use of the strong Priorean-style hybrid language discussed in Lecture 5 to characterise these logics. In addition, the paper's appendix shows how to prove the completeness of a hybrid tableaux system using a traditional Hintikka-style argument. All in all, a nice way of rounding off various themes discussed in the course.

Acknowledgements

We would like to thank INRIA, France's national research organization in computer science, who provided financial support for this course as part of the INRIA-funded research collaboration between the Langue et Dialogue group (INRIA Lorraine, Nancy) and the Language and Inference Technology group (University of Amsterdam); see `http://www.loria.fr/projets/ledcalg`. The course material was developed during an INRIA-funded visit by Maarten Marx to Langue et Dialogue. We would also like to thank Carlos Areces, Torben Brauner, Balder ten Cate, and Eric Kow for helping out in various ways in the run-up to NASSLLI.

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

Patrick Blackburn
INRIA Lorraine, Nancy, France
patrick@aplog.org

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 (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:*

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

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

Definition 2 ((Kripke) models) *Such a language is interpreted on* models *(often called* Kripke models*). A model* $\mathcal{M}$ *(for a fixed choice of PROP and MOD) is a triple* $(W, \{R_\pi \mid \pi \in MOD\}, V)$*. Here* W *is a non-empty set (I'll call its elements states, or nodes), and each* R_π *is a binary relations on* W*. The pair* $(W, \{R_\pi \mid \pi \in MOD\})$ *is called the* frame *underlying* $\mathcal{M}$*, and* $\mathcal{M}$ *is said to be a model based on this frame.* V *(the* valuation*) is a function with domain PROP and range* $Pow(W)$*; it tells us at which states (if any) each propositional symbol is true.*

Definition 3 (Satisfaction and validity) *Interpretation is carried out using the Kripke satisfaction definition. Let* $\mathcal{M} = (W, \{R_\pi \mid \pi \in MOD\}, V)$ *and* $w \in W$*. Then:*

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

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

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

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

Remark 2 (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:

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

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 $\mathsf{ST}y$ are analogous to those given for $\mathsf{ST}x$; 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 (Temporal logic) *We'll be seeing a lot of the bimodal language with MOD =* $\{\mathsf{F}, \mathsf{P}\}$ *in this paper: the modality* F *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* F *looks forward along the flow of time, and* $\langle\mathsf{P}\rangle$ *backwards.*

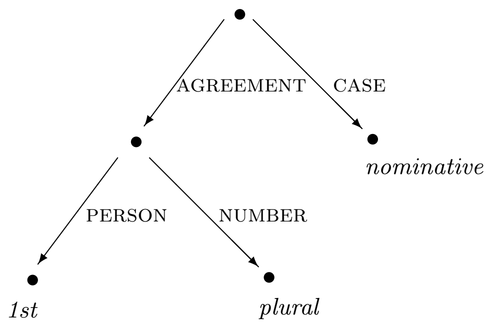
Consider the formula $\langle\mathsf{P}\rangle$Mia-unconscious*. This is true iff we can look back in time from the current state and see a state where Mia is unconscious. Similarly* FMia-unconscious *requires us to scan the states that lie in the future looking for one where Mia is unconscious. Thus 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* $\langle\mathsf{SUB}\rangle\mathsf{F}\langle\mathsf{SUP}\rangle p$ *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. 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 2 (Feature logic) *Feature structures are widely used in unification-based approaches to natural language. In essence, feature structures are multigraphs that represent lin-*

guistic information:



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

$$\begin{bmatrix} \text{AGREEMENT} & \begin{bmatrix} \text{PERSON} & \text{1st} \\ \text{NUMBER} & \text{plural} \end{bmatrix} \\ \text{CASE} & -\text{dative} \end{bmatrix}$$

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:

$$\begin{array}{ll} & \langle \text{AGREEMENT} \rangle (\langle \text{PERSON} \rangle \mathit{1st} \wedge \langle \text{NUMBER} \rangle \mathit{plural}) \\ \wedge & \langle \text{CASE} \rangle \neg \mathit{dative} \end{array}$$

Example 3 (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:*

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

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

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

*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 4 (Temporal logic) *Although the temporal language with modalities* F *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.*

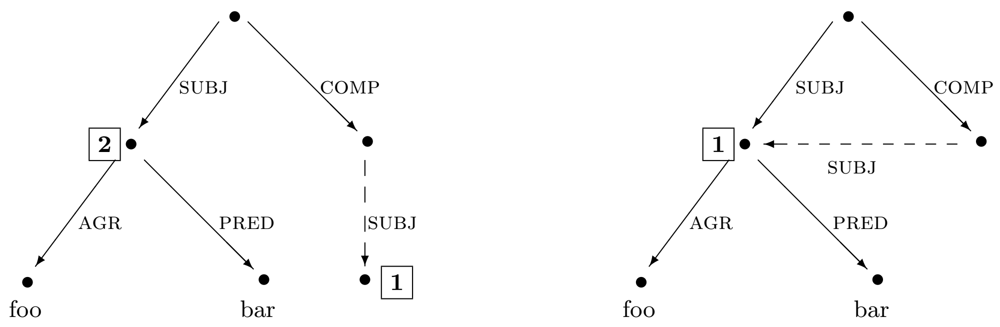
Similarly, while it's certainly possible to abstract elegant modal logics from the work of James Allen, such abstractions amputate a central feature of his work: reference to specific intervals. Allen's formalism includes the notation $\mathbf{Hold}(\mathsf{P}, i)$ *meaning "the property* P *holds at 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 F *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:* $\mathsf{FF}p \rightarrow \mathsf{F}p$ *does so. OK: but can we write down a formula valid on precisely the asymmetric frames (that is, frames* $(T, <)$ *such that* $\forall x \forall y(x < y \rightarrow y \not< x)$*)? Try what you like, you won't find any such formula: a central property of flows of time is invisible to modal representations.*

Example 5 (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:*

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

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:*



(i) *(ii)*

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

Example 6 (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 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 means it has 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 enabled them to get on with the topics that interested them with the minimum of syntactic fuss, but it meant that there were 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 genuinely *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 4 (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 following collection of formulas:*

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

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

Remark 3 (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 5 (Hybrid models, satisfaction, and validity) *A hybrid model is a triple $(W, \{R_\pi \mid \pi \in MOD\}, V)$ where $(W, \{R_\pi \mid \pi \in MOD\})$ is a frame and V is a* hybrid valuation. *A hybrid valuation is a function with domain PROP$\cup$NOM and range $Pow(W)$ such that for all nominals i, $V(i)$ is a singleton subset of W. We call the unique state in $V(i)$ the denotation of i. We interpret hybrid languages on hybrid models by adding the following two clauses to the Kripke satisfaction definition:*

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

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 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 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:*

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

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 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 7 (Temporal logic) *First, although* Vincent accidentally squeezed the trigger *can't be correctly represented in the ordinary temporal language in* F *and* $\langle \mathsf{P} \rangle$ *, it can be with the help of nominals:* $\langle\mathsf{P}\rangle(i\wedge\text{Vincent-accidentally-squeeze-the-trigger})$ *locates the trigger-squeezing not merely in the past, but at a specific temporal state there, namely 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: $i \rightarrow \neg\mathsf{FF}i$ defines asymmetry (that is, it is valid on all asymmetric frames and no others). More on this in Section 5.

Example 8 (Feature logic) *Nominals correspond to tags. Consider once more the problematic AVM:*

$$\begin{bmatrix} \textsc{subj} & \boxed{\mathbf{1}} \begin{bmatrix} \textsc{agr} & \text{foo} \\ \textsc{pred} & \text{bar} \end{bmatrix} \\ \textsc{comp} & [\textsc{subj}\ \ \boxed{\mathbf{1}}\,] \end{bmatrix}$$

This corresponds to the following L^N wff:

$$\begin{array}{ll} & \langle\textsc{subj}\rangle(i \wedge \langle\textsc{agr}\rangle\text{foo} \wedge \langle\textsc{pred}\rangle\text{bar}) \\ \wedge & \langle\textsc{comp}\rangle\langle\textsc{subj}\rangle i \end{array}$$

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 9 (Description Logic) *The TBoxes of the concept language $\mathcal{ALCO}$ (that is, $\mathcal{ALC}$ enriched with the $\mathcal{O}$ operator mentioned in Example 6) 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 : φ corresponds to the satisfaction statement $@_i\varphi$, and the specification (i, j) : 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 6 (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 and ψ is false at s. Note that both the $[\pi]$-rule and the $\neg\langle\pi\rangle$-rule take two input formulas, one of which (the minor premiss) is a formula of the form $@_s\langle\pi\rangle t$. For example, the $[\pi]$-rule says that if a pair of formulas of the form $@_s[\pi]\varphi$ and $@_s\langle\pi\rangle t$ can be found on some branch of the tableau, we are free to extend that branch by adding $@_t\varphi$ — a clear reflection of the Kripke semantics for $[\pi]$. Already first-order ideas are creeping into the system: this rule trades on the fact that hybrid logic is strong enough to make statements about state succession (using near-atomic satisfaction statements of the form $@_s\langle\pi\rangle t$).*

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

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

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

Remark 7 (The second group is a 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 10 (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 11 (A genuinely hybrid validity) *The previous example gives only the barest hint of what the system can do. Here's a more interesting example, which shows that hybrid reasoning not merely makes use of nominals and @, but also gets to grip with the logic of state identity and succession they embody.*

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

1	$\neg @_j (\langle \text{HOR} \rangle \langle \text{VERT} \rangle (i \wedge p) \wedge \langle \text{DIAG} \rangle i \to \langle \text{DIAG} \rangle p)$	
2	$@_j (\langle \text{HOR} \rangle \langle \text{VERT} \rangle (i \wedge p) \wedge \langle \text{DIAG} \rangle i)$	$1, \neg\to$
$2'$	$\neg @_j \langle \text{DIAG} \rangle p$	*Ditto*
3	$@_j \langle \text{HOR} \rangle \langle \text{VERT} \rangle (i \wedge p)$	$2, \wedge$
$3'$	$@_j \langle \text{DIAG} \rangle i$	*Ditto*
4	$@_j \langle \text{HOR} \rangle k$	$3, \langle \text{HOR} \rangle$
$4'$	$@_k \langle \text{VERT} \rangle (i \wedge p)$	*Ditto*
5	$@_k \langle \text{VERT} \rangle l$	$4', \langle \text{VERT} \rangle$
$5'$	$@_l (i \wedge p)$	*Ditto*
6	$@_l i$	$5', \wedge$
$6'$	$@_l p$	*Ditto*
7	$@_i l$	6, *Sym*
8	$@_i p$	$6', 7$, *Nom*
9	$\neg @_i p$	$2', 3', \neg\langle \text{DIAG} \rangle$
	✠ 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 8 (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* F *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 9).

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 closed tableau with $\neg @_i \varphi$ as its root (where i is a nominal not occurring in φ). Then:

Theorem 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 12 (Pure formulas and frame definability) *Consider the temporal language in* F *and* $\langle \mathsf{P} \rangle$*. Using pure formulas, we can define a number of properties relevant to temporal logic:*

$@_i\neg\mathsf{F}i$	$\forall x\neg(xR_Fx)$ *(Irreflexivity)*
$@_i\neg\mathsf{FF}i$	$\forall x\forall y(xR_Fy \to \neg yR_Fx)$ *(Asymmetry)*
$@_i[\mathsf{F}](\mathsf{F}i \to i)$	$\forall x\forall y(xR_Fy \wedge yR_Fx \to x = y)$ *(Antisymmetry)*
$\mathsf{FF}i \to \mathsf{F}i$	$\forall x\forall y\forall z(xR_Fy \wedge yR_Fz \to xR_Fz)$ *(Transitivity)*
$\mathsf{F}i \to \mathsf{FF}i$	$\forall x\forall y(xR_Fy \to \exists z(xR_Fz \wedge zR_Fy))$ *(Density)*
$@_i\mathsf{F}j \vee @_ij \vee @_j\mathsf{F}i$	$\forall x\forall y(xR_Fy \vee x = y \vee yR_Fz)$ *(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:

$@_i[\mathsf{F}]\langle\mathsf{P}\rangle i$	$\forall x\forall y(xR_Fy \to yR_Px)$
$@_i[\mathsf{P}]\mathsf{F}i$	$\forall x\forall y(xR_Py \to yR_Fx)$

It follows that the conjunction $@_i[\mathsf{F}]\langle\mathsf{P}\rangle i \wedge @_i[\mathsf{P}]\mathsf{F}i$ 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 F 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:

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

I mentioned in Example 4 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 9 (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. 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 2 *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 $\mathsf{H}+\mathsf{A}$ is complete with respect to F.*

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

Example 13 (An application in temporal logic) *Suppose we are working with the F 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 F and $\langle\mathsf{P}\rangle$ really are mutually converse, add the axioms $@_i[\mathsf{F}]\langle\mathsf{P}\rangle i$ and $@_i[\mathsf{P}]\mathsf{F}i$; we know from Example 12 that together these formulas define the converse property, and both are pure satisfaction statements. Now to guarantee transitivity. The pure formula $\mathsf{FF}i \to \mathsf{F}i$ defines this property. This is not a satisfaction statement, but $@_j\mathsf{FF}i \to \mathsf{F}i$ 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), $\langle \mathsf{P} \rangle \langle \mathsf{P} \rangle \varphi \to \langle \mathsf{P} \rangle \varphi$ is valid on the class of frames our axioms define. Thus, by Theorem 2, 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			$\neg @_i(\langle \mathsf{P} \rangle \langle \mathsf{P} \rangle \varphi \to \langle \mathsf{P} \rangle \varphi)$		
2			$@_i \langle \mathsf{P} \rangle \langle \mathsf{P} \rangle \varphi$		$1, \neg\!\to$
$2'$			$\neg @_i \langle \mathsf{P} \rangle \varphi$		*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			$@_j [\mathsf{P}] \mathsf{F} j$		*Axiom*
6			$@_k \mathsf{F} j$		$4, 5, [\mathsf{P}]$
7			$@_i [\mathsf{P}] \mathsf{F} i$		*Axiom*
8			$@_j \mathsf{F} i$		$3, 7, [\mathsf{P}]$
9			$@_k(\mathsf{FF} i \to \mathsf{F} i)$		*Axiom*
10		$\neg @_k \mathsf{FF} i$	|	$@_k \mathsf{F} i$	$9, \to$
11	$6, 10, \neg \mathsf{F}$	$\neg @_j \mathsf{F} i$		$@_k [\mathsf{F}] \langle \mathsf{P} \rangle k$	*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 2 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 10 (Related work) *Many of the same technical themes (including an essentially identical model construction technique) can be found in the Basin, Matthews, and Vigano [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 14 (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 F *and* $\langle \mathsf{P} \rangle$ *language can almost define $\mathbb{Z}$. As he notes, the formula*

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

(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 F *and* $\langle \mathsf{P} \rangle$*, transitivity, irreflexivity and trichotomy. Furthermore, the formulas* $\mathsf{F}\top$ *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 natural 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 15 (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 11 (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 ∃ and ∀ 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 ∃ and ∀).

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

Remark 12 (But even local binding is complex) *Be warned: ↓ may seem simple, but it's not. Even without @ (let alone ∀ or ∃, 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 ↓ and just a single diamond lacks the finite model property. Let SCID4 be the conjunction of the following formulas:

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

Note that these formulas are pure, and that ↓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 ↓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 13 (∀ and ∃ have classical tableau rules) *To cope with hybrid logic enriched with ∀ and ∃, 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, $\exists xx$ is valid (this says: it is always possible to bind a variable to the current state). We can prove it as follows:

1	$\neg@_i\exists xx$	
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 2 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 16 (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 17 (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*

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

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

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

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 $\mathsf{F}(\mathsf{yesterday} \land \varphi)$ *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 18 (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* F *and* ⟨P⟩*, even with the help of nominals and @.*

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

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

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 F and ⟨P⟩ 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 ∀ and ∃) 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 ∀ and ∃ made it easy to combine the modal canonical model construction with the first-order Henkin construction (and thus proved the earliest version of Theorem 2), 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 ∀ and ∃, 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 ↓, 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 ↓ 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 2 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`.

[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, Universita' 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.M. 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.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.Y. 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.N. Prior. *Past, Present and Future*. Oxford University Press, 1967.

[43] A.N. Prior. *Papers on Time and Tense*. University of Oxford Press, 1968.

[44] A.N. 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.

Tableaux for Quantified Hybrid Logic[1]

Patrick Blackburn
INRIA, Lorraine
patrick@aplog.org

Maarten Marx
University of Amsterdam
marx@science.uva.nl

Abstract

We present a (sound and complete) tableau calculus for Quantified Hybrid Logic (*QHL*). *QHL* is an extension of orthodox quantified modal logic: as well as the usual $\Box$ and $\Diamond$ modalities it contains names for (and variables over) states, operators $@_s$ for asserting that a formula holds at a named state, and a binder $\downarrow$ that binds a variable to the current state. The first-order component contains equality and rigid and non-rigid designators. As far as we are aware, ours is the first tableau system for *QHL*.

Completeness is established via a variant of the standard translation to first-order logic. More concretely, a valid *QHL*-sentence is translated into a valid first-order sentence in the correspondence language. As it is valid, there exists a first-order tableau proof for it. This tableau proof is then converted into a *QHL* tableau proof for the original sentence. In this way we recycle a well-known result (completeness of first-order logic) instead of a well-known proof.

The tableau calculus is highly flexible. We only present it for the constant domain semantics, but slight changes render it complete for varying, expanding or contracting domains. Moreover, completeness with respect to specific frame classes can be obtained simply by adding extra rules or axioms (this can be done for every first-order definable class of frames which is closed under and reflects generated subframes).

1 Introduction

Hybrid logic is an extension of modal logic in which it is possible to name states and to assert that a formula is true at a named state. Hybrid logic uses three fundamental tools to do this: nominals, satisfaction operators, and the $\downarrow$-binder. Nominals are special propositional symbols that are true at precisely one state in any model: nominals 'name' the unique state they are true at. A satisfaction operator has the form $@_s$ where s is a nominal. A formula of the form $@_s\phi$ asserts that ϕ is true at the state named by the nominal s. Finally, a formula of the form $\downarrow s.\phi$ binds all occurrences of the nominal s in ϕ to the current state of evaluation — that is, it makes s a name for the current state. (Actually, so that we don't have to worry about accidental binding in the course of tableau proofs, we shall distinguish between ordinary nominals, which cannot be bound, and 'state variables' which are essentially bindable nominals.)

Hybrid logic has a lengthy history (see the webpage `www.hylo.net` for further information), and over the years it has become clear that adding the hybrid apparatus of nominals (and state variables), satisfaction operators, and $\downarrow$ to modal logic often results in systems with better logical properties than the original. But most previous work on hybrid logic has examined the effects of hybridizing *propositional* modal logics. What about *quantified* (first-order) hybrid logic?

In fact, strong evidence already exists that quantified hybrid logic (*QHL*) is also better behaved logically than orthodox quantified modal logic. In [2], the only recent paper devoted to the topic, it is shown that a very general interpolation theorem holds in *QHL* (as is well known interpolation almost never holds in orthodox quantified modal logic [3]). The purpose of the present paper is to show that *QHL* is well behaved in another respect: just as in the propositional case, it is possible to define simple and intuitive tableau systems. We shall present a tableau system for *QHL* which handles equality, and rigid and non-rigid designators.

Our method for proving completeness is very simple and inspired by Jerry Seligman's paper [10]. Instead of redoing a proof we use existing results. Correspondence theory and

[1]Maarten Marx's work was supported by Netherlands Organization for Scientific Research (NWO, grant# 612.000.106). The paper was written during a visit by Maarten Marx to the Langue et Dialogue team, LORIA, Nancy, France. This work was carried out as part of the INRIA funded partnership between LIT (Language and Inference Technology, University of Amsterdam) and LED (Langue et Dialogue, LORIA, Nancy).

its notion of a standard translation $\mathsf{ST}(\cdot)$ places the model theory of (propositional and first-order) modal logic firmly into first-order logic [12, 13]. Our plan is the following. We prove completeness for our tableaux calculus by taking a proof P for $\mathsf{ST}\phi$ in a proven complete first-order calculus, and transform P into a proof P' for ϕ in our calculus. The tableaux system we use is by Fitting, in particular the one presented in [4]. This strategy works in hybrid logic because it has an equivalent expression for every subformula which might occur in a first-order proof of a translated formula. It is worth emphasizing that completeness could also be established using standard tehniques (in particular, use of Hintikka sets). However a long-standing claim of the hybrid logic community is that hybrid logic combines the best of modal and first-order logic. The completeness proof technique used here makes the links between first-order and hybrid proof theory particularly clear, which is why we have chosen it.

Outline of paper. The paper starts with a definition of first-order hybrid logic. Then we present the tableau system in three natural parts. The forth section is devoted to completeness issues. Again we split them up into three natural parts. This section ends with a very general completeness result. Finally we draw conclusions.

2 Quantified Hybrid Logic

We first define the syntax of *QHL*. We have a set NOM of nominals, a set SVAR of state variables, a set FVAR of first-order variables, a set CON of first-order constants, a set IC of unary function symbols, and predicates of any arity (note that predicates of nullary arity are simply propositional variables). The *terms* of the language are the constants from CON, the first-order variables from FVAR and the terms generated by the rule

$$\text{if } q \in \mathsf{IC} \text{ and } s \in \mathsf{NOM} \cup \mathsf{SVAR}, \text{ then } @_s q \text{ is a term.}$$

(For readers familiar with propositional hybrid logic, this notation may come as a surprise: we are combining a satisfaction operator with a term to make a new term. But as the semantics defined below will show, overloading the @ notation in this way is quite natural: $@_s q$ will be the value of the non-rigid term q at the world named by s.)

The *atomic formulas* are all symbols in NOM and SVAR together with the usual first-order atomic formulas generated from the predicate symbols and equality using the terms. *Complex formulas* are generated from these according to the rules

$$\neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \to \psi \mid \exists x\phi \mid \forall x\phi \mid \Diamond\phi \mid \Box\phi \mid @_n\phi \mid {\downarrow}w.\phi.$$

Here $x \in \mathsf{FVAR}$, $w \in \mathsf{SVAR}$ and $n \in \mathsf{NOM} \cup \mathsf{SVAR}$.

These formulas are interpreted in first-order modal models with constant domains. A *QHL* model is a structure $(W, R, D, I_{nom}, I_{con}, I_w)_{w\in W}$ such that

- (W, R) is a modal frame;
- I_{nom} is a function assigning members of W to nominals in NOM;
- I_{con} is a function assigning elements of D to constants in CON;
- for each $w \in W$, (D, I_w) is an ordinary first-order model.

To interpret formulas with free variables we use special two-sorted assignments. A *QHL assignment* is a function g from $\mathsf{SVAR} \cup \mathsf{FVAR}$ to $W \cup D$ which sends state variables to members of W and first-order variables to elements of D. Given a model and an assignment g, the interpretation of terms t, denoted by $\bar{t}$, is defined as

$$\begin{array}{lcll}
\bar{x} & = & g(x) & \text{for } x \text{ a variable} \\
\bar{c} & = & I_{con}(c) & \text{for } c \text{ a constant} \\
\overline{@_n q} & = & I_{\mathbf{n}}(q) & \text{for } q \text{ a non rigid designator,} \\
 & & \multicolumn{2}{l}{\text{and } \mathbf{n} \text{ is } I_{nom}(n) \text{ if } n \text{ a nominal, or } g(n) \text{ if } n \text{ a state variable.}}
\end{array}$$

Formulas are now interpreted as usual. With g^x_d we denote the assignment which is just like g except that $g(x) = d$. $\mathfrak{M}, g, s \Vdash \phi$ means that ϕ holds in model $\mathfrak{M}$ at state s under the assignment g. The inductive definition is

$$
\begin{array}{lcl}
\mathfrak{M}, g, s \Vdash P(t_1, \ldots, t_n) & \iff & \langle \bar{t}_1, \ldots, \bar{t}_n \rangle \in I_s(P) \\
\mathfrak{M}, g, s \Vdash t_i = t_j & \iff & \bar{t}_i = \bar{t}_j \\
\mathfrak{M}, g, s \Vdash n & \iff & I_{nom}(n) = s, \text{ for } n \text{ a nominal} \\
\mathfrak{M}, g, s \Vdash w & \iff & g(w) = s, \text{ for } w \text{ a state variable} \\
\mathfrak{M}, g, s \Vdash \neg\phi & \iff & \mathfrak{M}, g, s, \nVdash \phi \\
\mathfrak{M}, g, s \Vdash \phi \wedge \psi & \iff & \mathfrak{M}, g, s \Vdash \phi \text{ and } \mathfrak{M}, g, s \Vdash \psi \\
\mathfrak{M}, g, s \Vdash \phi \vee \psi & \iff & \mathfrak{M}, g, s \Vdash \phi \text{ or } \mathfrak{M}, g, s \Vdash \psi \\
\mathfrak{M}, g, s \Vdash \phi \to \psi & \iff & \mathfrak{M}, g, s \Vdash \phi \text{ implies } \mathfrak{M}, g, s \Vdash \psi \\
\mathfrak{M}, g, s \Vdash \exists x\phi & \iff & \mathfrak{M}, g^x_d, s \Vdash \phi, \text{ for some } d \in D \\
\mathfrak{M}, g, s \Vdash \forall x\phi & \iff & \mathfrak{M}, g^x_d, s \Vdash \phi, \text{ for all } d \in D \\
\mathfrak{M}, g, s \Vdash \Diamond\phi & \iff & \mathfrak{M}, g, t \Vdash \phi \text{ for some } t \in W \text{ such that } Rst \\
\mathfrak{M}, g, s \Vdash \Box\phi & \iff & \mathfrak{M}, g, t \Vdash \phi \text{ for all } t \in W \text{ such that } Rst \\
\mathfrak{M}, g, s \Vdash @_n\phi & \iff & \mathfrak{M}, g, I_{nom}(n) \Vdash \phi \text{ for } n \text{ a nominal} \\
\mathfrak{M}, g, s \Vdash @_w\phi & \iff & \mathfrak{M}, g, g(w) \Vdash \phi \text{ for } w \text{ a state variable} \\
\mathfrak{M}, g, s \Vdash {\downarrow} w.\phi & \iff & \mathfrak{M}, g^w_s, s \Vdash \phi.
\end{array}
$$

3 The tableau calculus

The tableau system can be divided into three natural pieces: **(A)** the propositional rules, the $\Diamond$ and $\Box$ rules and the rules for @; **(B)** the rule for $\downarrow$; **(C)** the rules for (first-order) quantification and equality. The blocks of rules taken separately form a complete calculus for the appropriate reducts. In particular:

1. **A** is complete for the propositional modal language expanded with nominals and @. (We name this system $\mathcal{HL}(@)$; in the literature it is often called the *basic hybrid language*.)

2. $\mathbf{A} \cup \mathbf{B}$ is complete for $\mathcal{HL}(@, \downarrow)$, the expansion of $\mathcal{HL}(@)$ with state variables and the $\downarrow$ binder;

3. $\mathbf{A} \cup \mathbf{B} \cup \mathbf{C}$ is complete for *QHL*.

Some terminology. As usual, a tableau branch is *closed* if it contains ϕ and $\neg\phi$, where ϕ is a formula. A tableau is closed if each branch is closed. A branch is *atomically closed* if it closes on an atom and its negation. A *(tableau) proof* of a hybrid sentence ϕ is a closed tableau beginning with $\neg @_s\phi$, where s is a nominal not occurring in ϕ.

3.1 Tableau for $\mathcal{HL}(@)$

A key feature of our tableau is that all modal formulas occurring in a proof are grounded to a named world by their label. (This same feature also occurs in labelled tableau for propositional modal logic [8, 7].)

Grounding to a named state is implemented in our system by ensuring that all formulas occurring in proofs are of the form $@_s\phi$ or $\neg @_s\phi$ for s a nominal. Thus the propositional rules become

Conjunctive rules

$$
\frac{@_s(\phi \wedge \psi)}{\begin{array}{c}@_s\phi \\ @_s\psi\end{array}} \qquad
\frac{\neg @_s(\phi \vee \psi)}{\begin{array}{c}\neg @_s\phi \\ \neg @_s\psi\end{array}} \qquad
\frac{\neg @_s(\phi \to \psi)}{\begin{array}{c}@_s\phi \\ \neg @_s\psi\end{array}}
$$

Disjunctive rules

$$
\frac{@_s(\phi \vee \psi)}{@_s\phi \mid @_s\psi} \qquad
\frac{\neg @_s(\phi \wedge \psi)}{\neg @_s\phi \mid \neg @_s\psi} \qquad
\frac{@_s(\phi \to \psi)}{\neg @_s\phi \mid @_s\psi}
$$

Negation rules

$$
\frac{\neg @_s\neg\phi}{@_s\phi} \qquad
\frac{@_s\neg\phi}{\neg @_s\phi}
$$

To these we add rules for diamond and box. In the diamond rules, t is a nominal which does not occur on the branch.

Diamond rules

$$\frac{@_s\Diamond\phi}{\begin{array}{c}@_s\Diamond t\\ @_t\phi\end{array}} \qquad \frac{\neg @_s\Box\phi}{\begin{array}{c}@_s\Diamond t\\ \neg @_t\phi\end{array}}$$

Box rules

$$\frac{@_s\Box\phi, \quad @_s\Diamond t}{@_t\phi} \qquad \frac{\neg @_s\Diamond\phi, \quad @_s\Diamond t}{\neg @_t\phi}$$

Finally the rules for @. There are two rewrite rules to delete nestings of @. Next, as $@_s t$ really means that s and t are equal, there are rules to handle equality. These three rules are direct analogues of the reflexivity and replacement rules in Fitting's first-order tableau system [4]. As we will use them often, we gave them separate names.

@ rules

$$\frac{@_s @_t\phi}{@_t\phi} \qquad \frac{\neg @_s @_t\phi}{\neg @_t\phi} \qquad \frac{[s \text{ on the branch}]}{@_s s}\,[\text{Ref}] \qquad \frac{@_s t \quad @_s\varphi}{@_t\varphi}\,[\text{Nom}] \qquad \frac{@_s t \quad @_r\Diamond s}{@_r\Diamond t}\,[\text{Bridge}]$$

The following rules can be derived: $\frac{@_s t}{@_t s}\,[\text{Sym}] \qquad \frac{@_s t \quad @_t r}{@_s r}\,[\text{Trans}] \qquad \frac{@_s t \quad @_t\varphi}{@_s\varphi}\,[\text{Nom}^{-1}]$

Example. Below we give a tableau proof for $(\Diamond p \wedge \Diamond\neg p) \to (\Box(q \to n) \to \Diamond\neg q)$. Here n is a nominal and p, q are propositional variables. The formula expresses that if a state has two successors, then if it has at most one q successor, it has at least one $\neg q$ successor. Note that this is not expressible in ordinary modal logic. In ordinary modal logic we cannot put an upper bound on the number of successors.

1. $\neg @_s(\Diamond p \wedge \Diamond\neg p \to (\Box(q \to n) \to \Diamond\neg q))$
2. $@_s(\Diamond p \wedge \Diamond\neg p)$
3. $\neg @_s(\Box(q \to n) \to \Diamond\neg q)$
4. $@_s\Diamond p$
5. $@_s\Diamond\neg p$
6. $@_s\Box(q \to n)$
7. $\neg @_s\Diamond\neg q$
8. $@_s\Diamond t$
9. $@_t p$
10. $@_s\Diamond r$
11. $@_r\neg p$
12. $@_t(q \to n)$

13.1 $\neg @_t q$	14. $@_t n$	
13.2 $\neg @_t\neg q$	15. $@_r(q \to n)$	
13.3 $@_t q$		
	16.1 $\neg @_r q$	17. $@_r n$
	16.2 $\neg @_r\neg q$	18. $@_n t$
	16.3 $@_r q$	19. $@_n r$
		20. $@_t r$
		21. $@_r p$
		22. $\neg @_r p$

In this, 2 and 3 are from 1 by a conjunctive rule; 4,5,6,7 are from 2 and 3 by conjunctive rules; 8,9,10,11 are from 4 and 5 by diamond rules; 12 is from 6 and 8 by box; 13.1 and 14 are from 12 by a disjunctive rule; 13.2 is from 7 and 8 by box; 13.3 is from 13.2 by a negation rule. The branch closes on 13.3 and 13.1. 15 is from 6 and 10 by box; 16.1 and 17 are from 15 by a disjunctive rule; 16.2 is from 10 and 7 by box; 16.3 is from 16.2 by a negation rule. The branch closes on 16.1 and 16.3. 18 is from 14 by the derived Sym rule; 19 is from 17 by Sym; 20 is from 18 and 19 by Nom; 21 is from 20 and 9 by the Nom rule; 22 is from 11 by a negation rule. The final branch closes on 21 and 22.

3.2 Tableau for $\mathcal{HL}(\downarrow, @)$

To obtain a complete tableau system for the expansion of $\mathcal{HL}(@)$ with variables over states and the binder $\downarrow$, we only need to add the following two rewrite rules to the rules for $\mathcal{HL}(@)$:

Downarrow rules

$$\frac{@_s{\downarrow}w.\phi}{@_s\phi[s/w]} \qquad \frac{\neg @_s{\downarrow}w.\phi}{\neg @_s\phi[s/w]}$$

Here $[s/w]$ means substitute s for all free occurrences of w in ϕ. Because s is always a nominal, whence cannot be quantified over, we do not have to worry about accidental bindings. As an example the reader can try to prove the validities ${\downarrow}w.\Diamond w \rightarrow (p \rightarrow \Diamond p)$ and ${\downarrow}w.\Box\Diamond w \rightarrow (\Diamond\Box p \rightarrow p)$.

3.3 Tableau for *QHL*

A complete tableau system for quantified hybrid logic consists of the $\mathcal{HL}(\downarrow, @)$ system, plus the (adjusted) rules for the quantifiers and equality from Fitting's system (see [4]) for first-order logic with equality, plus two rules relating equalities across worlds. In the existential rules, c is a parameter which is new to the branch. As parameters are never quantified over, the substitution $[c/x]$ is free for the formula $\phi(x)$. In the universal rules, t is any grounded term on the branch (thus either a first-order constant, a parameter or a grounded definite description). A grounded definite description is a term $@_n q$ for n a nominal and q a non-rigid designator from IC.

Existential rules

$$\frac{@_s\exists x\phi(x)}{@_s\phi(c)} \qquad \frac{\neg @_s\forall x\phi(x)}{\neg @_s\phi(c)}$$

Universal rules

$$\frac{@_s\forall x\phi(x)}{@_s\phi(t)} \qquad \frac{\neg @_s\exists x\phi(x)}{\neg @_s\phi(t)}$$

Besides Fitting's [4] Reflexivity (Ref) and Replacement (RR) rules, there are three extra rules for equality. The first (called DD) states that if n and m denote the same state, then $@_n q$ and $@_m q$ denote the same individual. The second and third (both called @=) embody that equality is a rigid predicate: if two terms are the same in one world, they are the same in every world. Because these two rules peel the leading $@_n$ off equalities, reflexivity and replacement can be kept in the old format.

QHL Equality rules

$$\frac{}{t = t}[\mathsf{Ref}] \qquad \frac{t = u, \quad \phi(t)}{\phi[u]}[\mathsf{RR}], \qquad \frac{@_n m}{@_n q = @_m q}[\mathsf{DD}] \qquad \frac{@_n(t_i = t_j)}{t_i = t_j}[@=] \qquad \frac{\neg @_n(t_i = t_j)}{\neg(t_i = t_j)}[@=]$$

In the Replacement rule, $\phi[u]$ denotes $\phi(t)$ with some of the occurrences of t replaced by u.

Example. The most interesting examples deal with equality and rigid and non-rigid designators. Consider the sentence *Caroline is Miss America.* When formalising this let c be a rigid designator denoting Caroline and q a non-rigid designator denoting Miss America. Then ${\downarrow}x.(c = @_x q)$ means *Caroline is the present Miss America.* It is true in a state w if $I_{con}(c) = I_w(q)$. This formula has the following relation with the $\Box$ operator:

$$\not\models ({\downarrow}w.c = @_w q) \quad \rightarrow \quad \Box{\downarrow}w.c = @_w q \tag{1}$$

$$\models ({\downarrow}w.c = @_w q) \quad \rightarrow \quad {\downarrow}w.\Box c = @_w q. \tag{2}$$

A falsifying model for the sentence in (1) is given by two worlds n and m, with Rnm, and a domain $\{a, b\}$ with the interpretation $I_{con}(c) = I_n(q) = a$ and $I_m(q) = b$. Then (1) fails at world n. When downarrow has wide scope in the consequent, the formula becomes true. Here is the tableau proof:

1. $\neg @_n((\downarrow w.c = @_w q) \to \downarrow w.\Box(c = @_w q))$
2. $@_n \downarrow w.c = @_w q$
3. $\neg @_n \downarrow w.\Box(c = @_w q)$
4. $@_n(c = @_n q)$
5. $\neg @_n \Box(c = @_n q)$
6. $@_n \Diamond m$
7. $\neg @_m(c = @_n q)$
8. $c = @_n q$
9. $\neg(c = @_n q)$.

In this, 2 and 3 are from 1 by a conjunctive rule; 4 and 5 are from 2 and 3 by a downarrow rule, respectively; 6 and 7 are from 5 by a diamond rule; 8 and 9 are from 4 and 7 by an @= rule, respectively.

4 Soundness and Completeness

The argument to establish soundness follows the familiar pattern: show that satisfiability is preserved by each tableau rule application. This is easy to check and left to the reader. Completeness will be established using the standard translation and a complete first-order inference system. We use the system that is closest to the one presented here: the tableau calculus for first-order logic with equality from Fitting [4] with the reflexivity and replacement rules (restricted to atoms). The main line of the argument is the following. We need to establish that every valid *QHL* sentence has a *QHL* tableau proof. The standard translation preserves validity, thus a *QHL* sentence ϕ is valid if and only if the first-order sentence $\mathsf{ST}\phi$ is valid. For valid $\mathsf{ST}\phi$, there exists a closed first-order tableau proof T starting with $\neg ST\phi$. Our task is to transform this closed first-order proof T starting with $\neg\mathsf{ST}\phi$ into a closed *QHL* tableau proof T' starting with $\neg\phi$.

Most of the work concerns the modalities and the @ operator, because with these the standard translation creates the largest change in syntactic structure. For this reason we present the completeness proof for the simplest logic $\mathcal{HL}(@)$ separately. After that, the rest will be easy.

Before we can continue we have to settle two things. We change Fitting's first-order tableau rules a little bit in order to better cope with translations of modal formulas. Besides that we have to use a modified translation. We start with the former.

In order to save on inductive proofs and definitions, we assume fror now on that the *QHL* language contains as primitive logical operators only $\neg, \wedge, \Box, @_s, \downarrow w.$ and $\forall v$. Clearly this is without loss of generality because the other operators can be defined in terms of these.

4.1 Tableau rules for relativized quantifiers

The translation of a box modality yields a relativized universal formula of the form $\forall x(A(x) \to C(x))$, with $A(x)$ an atom. For these relativized universals, a more efficient tableau rule exists than the combination of universal and $\to$ rule together. In fact it is nothing but Modes Ponens. For t a closed term,

Modes Ponens (MP)	
$\dfrac{A(t), \quad \forall x(A(x) \to \phi(x))}{\phi(t)}$	$\dfrac{A(t), \quad \neg\exists x(A(x) \wedge \phi(x))}{\neg\phi(t)}$

We change Fitting's calculus such that on universals relativized by an atom the normal universal rules cannot be applied, but MP can. This is easily seen to be complete (cf also [11]. We can make a further reduction in complexity in the case the antecedent is an equality. Then the statement just expresses a substitution. We also add the following rules to Fitting's calculus and make the proviso that universal and existential rules are never applied to quantified sentences relativized by an equality.

Substitution Rules			
$\dfrac{\forall x(x = t \to \phi(x))}{\phi(t)}$	$\dfrac{\exists x(x = t \wedge \phi(x))}{\phi(t)}$	$\dfrac{\neg\forall x(x = t \to \phi(x))}{\neg\phi(t)}$	$\dfrac{\neg\exists x(x = t \wedge \phi(x))}{\neg\phi(t)}$

4.2 Translation using predicate abstraction

Unfortunately, the usual standard translation does not square well with the intention to change one proof into another. Though it is truth preserving, it does not preserve syntactic structure. Because we want to transform a proof for the translation of ϕ into a proof for ϕ, we need to translate backwards as well. It is crucial that applying the backwards translation to the translation of ϕ yields ϕ again. This is simply not obtainable by the standard translation or obvious variants.

An example might explain why not. We can read $@_s(p \wedge q)$ as saying that state s has the property $p \wedge q$. As we want to translate proposition letters to one place predicates, in first-order logic we can only say then that s has property p and s has property q. This is of course logically equivalent, but syntactically different. We would like to have machinery which can turn formulas into predicates, so that we can speak about the property "p and q". The lambda calculus provides precisely this: $\langle \lambda x.(Px \wedge Qx)\rangle$ denotes the property of being P and Q. The formula $\langle \lambda x.(Px \wedge Qx)\rangle(s)$ serves then as an excellent proxy for $@_s(p \wedge q)$.

We add predicate abstraction to the language. This will only be defined for variables ranging over individuals. Thus we only add a piece of syntactic sugar. The expressive power of the language remains the same, it is just first-order logic. For a thorough introduction to real predicate abstraction in modal logic we refer to [6].

Suppose ϕ is a first-order formula and x a first-order variable. Then $\langle \lambda x.\phi\rangle$ is a predicate abstract. Its free variable occurrences are the free variable occurrences of ϕ except for x. Predicate abstracts behave as unary predicate symbols; new atomic formulas from predicate abstracts $\langle \lambda x.\phi\rangle$ can be made by the rule

if t is a term, then $\langle \lambda x.\phi\rangle(t)$ is a formula.

Examples are $\langle \lambda x.Px\rangle(t)$ and $\langle \lambda x.Px \wedge Qx\rangle(s)$. The new formulas get their meaning by performing *β-reduction:*

the β-reduction of $\langle \lambda x.\phi\rangle(t)$ is $\phi[t/x]$.

The meaning of $\langle \lambda x.\phi\rangle(t)$ is simply the meaning of $\phi[t/x]$. This shows that the expressive power remains the same. Our convention is that in λ expressions, the . takes wide scope, thus $\langle \lambda x.\phi \wedge \psi\rangle = \langle \lambda x.(\phi \wedge \psi)\rangle$.

In order to handle predicate abstracts in tableau proofs, we need only add two very simple rules to Fitting's system. The rules just implement β-reduction. Here they are

Abstract rules	
$\dfrac{\langle \lambda x.\phi\rangle(t)}{\phi[t/x]}$	$\dfrac{\neg\langle \lambda x.\phi\rangle(t)}{\neg\phi[t/x]}$

Fitting's tableau system with the two abstract rules added is a complete inference system for the expansion of first-order logic with λ abstraction with variables ranging over individuals [5].

We are ready to define the new standard translation AT for the propositional hybrid language, together with its inverse AT^-. In a certain sense, this translation can be traced back to the paper [9] in which McCarthy and Hayes introduce the situation calculus. $AT_y(\phi)$ and $AT^-_y(\phi)$ are defined in the same way but with x and y interchanged, e.g., $AT_y(p) := Py$ and $AT_y(\Box\phi) := \langle \lambda y.\forall x(Ryx \to AT_x(\phi))\rangle(y)$.

$$
\begin{array}{lcl}
AT_x(p) & := & Px \\
AT_x(n) & := & x = n \\
AT_x(\neg\phi) & := & \langle \lambda x.\neg AT_x(\phi)\rangle(x) \\
AT_x(\phi \wedge \psi) & := & \langle \lambda x.AT_x(\phi) \wedge AT_x(\psi)\rangle(x) \\
AT_x(\Box\phi) & := & \langle \lambda x.\forall y(Rxy \to AT_y(\phi))\rangle(x) \\
AT_x(@_n\phi) & := & \langle \lambda x.\forall x(x = n \to AT_x(\phi))\rangle(x)
\end{array}
$$

$$
\begin{array}{lcl}
AT^-_x(Px) & := & p \\
AT^-_x(x = n) & := & n \\
AT^-_x(\langle \lambda x.\neg\phi\rangle(x)) & := & \neg AT^-_x(\phi) \\
AT^-_x(\langle \lambda x.\phi \wedge \psi\rangle(x)) & := & AT^-_x(\phi) \wedge AT^-_x(\psi) \\
AT^-_x(\langle \lambda x.\forall y(Rxy \to \phi)\rangle(x)) & := & \Box AT^-_y(\phi) \\
AT^-_x(\langle \lambda x.\forall x(x = n \to \phi)\rangle(x)) & := & @_n AT^-_x(\phi)
\end{array}
$$

The following properties of AT and AT^- hold, for every $\mathcal{HL}(@)$ formula ϕ,

$$AT_x(\phi) \text{ is always a formula of the form } \langle\lambda x.\psi\rangle(x) \text{ or } Px \text{ or } x = n. \quad (3)$$

$$AT_x^-(AT_x(\phi)) = \phi, \text{ and similarly when } x \text{ is replaced by } y. \quad (4)$$

$$\phi \text{ is } \mathcal{HL}(@) \text{ valid iff } AT_x(\phi) \text{ is first-order valid.} \quad (5)$$

(3) follows from the definition. (4) is proved by induction on the complexity of the $\mathcal{HL}(@)$ formula. (5) is immediate by performing β-reduction and the well-known result on the standard translation.

4.3 Completeness for $\mathcal{HL}(@)$

We now specify an algorithm for turning a closed Fitting tableau for the formula $AT_x(\phi)[c/x]$ (where c is a parameter) into a closed $\mathcal{HL}(@)$ tableau for $@_c\phi$. Some terminology will be useful. A literal is a grounded formula of the form

$$P(t) \mid t = u \mid Rtu \mid \langle\lambda x.\phi\rangle(t) \mid \langle\lambda y.\phi\rangle(t), \text{ or its negation.}$$

Define the following translation $(\cdot)^*$ from positive literals to $\mathcal{HL}(@)$ sentences

$$\begin{array}{rcl} P(t)^* & := & @_t p \\ (t = u)^* & := & @_t u \\ (Rtu)^* & := & @_t \Diamond u \\ (\langle\lambda x.\phi\rangle(t))^* & := & @_t AT_x^-(\langle\lambda x.\phi\rangle(x)) \\ (\langle\lambda y.\phi\rangle(t))^* & := & @_t AT_y^-(\langle\lambda y.\phi\rangle(y)). \end{array}$$

For negative literals $(\neg\phi)$, we set $(\neg\phi)^* = \neg\phi^*$.

We recapitulate: AT translates a hybrid formula into a first-order formula and AT^- translates them backwards. The translation $(\cdot)^*$ translates literals occurring in a first-order tableau proof to hybrid formulas. Note that these literals may contain parameters introduced in the proof. The crucial connection between the forward and backward translations is that they preserve syntactic structure: for ϕ a hybrid formula and t a nominal or parameter,

$$(AT_x(\phi)[t/x])^* = @_t\phi \text{ and } (\neg AT_x(\phi)[t/x])^* = \neg @_t\phi. \quad (6)$$

Property (6) follows immediately from the definition of $(\cdot)^*$ and (4).

We are ready to specify the algorithm. Let T be a closed Fitting tableau for the formula $AT_x(\phi)[c/x]$. Without loss of generality we may assume that T is atomically closed. Let T' simply be T with all literals replaced by their $(\cdot)^*$ translation and all other formulas removed.

Claim T' is $\mathcal{HL}(@)$ tableau proof for ϕ.

We first observe that T' starts with $\neg @_c\phi$. This is because T starts with the literal $\neg AT_x(\phi)[c/x]$ whose $*$ translation is $\neg @_c\phi$ by (6).

Secondly, every branch in T' closes. This is because T branches close on literals, which we all move over to T', keeping the negation signs in place.

Finally we show that T' is a correct $\mathcal{HL}(@)$ tableau. We prove this by showing that every application of rules to a literal l in T can be matched by an application of a (derived) rule to l^* in T'. More precisely, for every literal l in T, for all literals l_1, l_2 produced from l by applying rules, the literals l_1^*, l_2^* can be obtained from l^* by applying a (derived) rule in T'.

We do a case-analysis according to the structure of the literals. By the translation we know that every literal which is a λ-formula has the form $\langle\lambda z.AT_z(\psi)\rangle(t)$, for z either x or y, and ψ an $\mathcal{HL}(@)$ formula. After performing β-reduction we obtain $AT_z(\psi)[t/z]$ whose $(\cdot)^*$ translation is $@_t\psi$ by (6). The case analysis is presented for the connectives in Table 2. This table is read as follows. On the left are first-order proofs with small annotations indicating which rule is applied on what to obtain the result. On the right are the $\mathcal{HL}(@)$ proofs which derive the $(\cdot)^*$ translated results from the $(\cdot)^*$ translated premises, again annotated.

First Order proof	Corresponding $\mathcal{HL}(@)$ proof
$\dfrac{t = u, P(t)}{P(u)}$ [RR]	$\dfrac{@_t u, @_t p}{@_u p}$ [Nom]
$\dfrac{t = u, v = t}{v = u}$ [RR]	$\dfrac{@_t u, @_v t}{@_v u}$ [Nom$^-$1]
$\dfrac{t = u, vRt}{vRu}$ [RR]	$\dfrac{@_t u, @_v \Diamond t}{@_v \Diamond u}$ [Bridge].

Table 1: Corresponding replacement proofs

Besides the rules for the connectives we must give analogues to every possible application of the Replacement Rule on grounded atoms occurring as subformulas of a translated hybrid modal formula in the hybrid tableau system. The possible instantiations of these grounded atoms in which t is replaced are

$$t = v, v = t, vRt, tRv, P(t) \text{ and } \langle \lambda x.\phi \rangle(t),$$

where $\langle \lambda x.\phi \rangle(x)$ is $AT_x(\phi')$ for some ϕ'.

In Table 1 the application of the replacement rule is given on the left while the corresponding proof on the AT^- images of the formulas is on the right. The cases for $P(t)$, $t = v$, Rtv and $\langle \lambda x.\phi \rangle(t)$ are all by applications of Nom. We only show the case for $P(t)$.

This finishes the proof of the claim and yields

Theorem 3 *The $\mathcal{HL}(@)$ tableau calculus is complete.*

4.4 Completeness for $\mathcal{HL}(\downarrow, @)$

Theorem 4 *The tableau system for $\mathcal{HL}(\downarrow, @)$ is complete.*

With all the groundwork done, the proof is very easy. We have to extend the translation to incorporate the variables and downarrow formulas. We assume that x and y are new variables. The translation and its inverse for the state variables and downarrow is simply

$$\begin{array}{rcl} AT_x(w) & := & x = w \\ AT^-_x(x = w) & := & w \\ AT_x(\downarrow w.\phi) & := & \langle \lambda x.\forall w(w = x \rightarrow AT_x(\phi))\rangle(x) \\ AT^-_x(\langle \lambda x.\forall w(w = x \rightarrow \phi)\rangle(x)) & := & \downarrow w.AT^-_x(\phi). \end{array}$$

In a straightforward way, the properties (3)-(6) still hold. Then the completeness proof amounts to showing that Fitting's rules applied to translations of downarrow formulas can be transformed to applications of the downarrow rules. On these translations only substitutions can be applied. This case is similar to the @ case, so we do not spell it out.

4.5 Completeness for *QHL*

Theorem 5 *The tableau system for QHL is complete.*

Again the proof is simple after we made the needed straightforward adjustments. The translation and its inverse for the full *QHL* language is obtained by adding the following rules to the ones already existing:

$$\begin{array}{rcl} AT_x(P(t_1, \ldots, t_k)) & := & P'(x, t_1, \ldots, t_k) \\ AT_x(t_i = t_j) & := & \langle \lambda x.t_i = t_j \rangle(x) \\ AT_x(\forall v \phi) & := & \langle \lambda x.\forall v AT_x(\phi)\rangle(x) \end{array}$$

$$\begin{array}{rcl} AT^-_x(\langle \lambda x.P'(x, t_1, \ldots, t_k)\rangle(x)) & := & P(t_1, \ldots, t_k) \\ AT^-_x(\langle \lambda x.t_i = t_j \rangle(x)) & := & t_i = t_j \\ AT^-_x(\langle \lambda x.\forall v \phi \rangle(x)) & := & \forall v AT^-_x(\phi) \end{array}$$

Case	FO tableau			$\mathcal{HL}(@)$ tableau		
$\neg$, pos	(1)	$\langle \lambda x.\neg AT_x(\phi)\rangle(t)$		(1)	$@_t\neg\phi$	
	(2)	$\neg AT_x(\phi)[t/x]$	(1), λ	(2)	$\neg @_t\phi$	(1), Neg
$\neg$ neg	(1)	$\neg\langle \lambda x.\neg AT_x(\phi)\rangle(t)$		(1)	$\neg @_t\neg\phi$	
	(2)	$\neg\neg AT_x(\phi)[t/x]$	(1), $\neg\lambda$			
	(3)	$AT_x(\phi)[t/x]$	(2), $\neg\neg$	(2)	$@_t\phi$	(1), Neg
$\wedge$ pos	(1)	$\langle \lambda x.AT_x(\phi) \wedge AT_x(\psi)\rangle(t)$		(1)	$@_t(\phi \wedge \psi)$	
	(2)	$AT_x(\phi)[t/x] \wedge AT_x(\psi)[t/x]$				
	(3)	$AT_x(\phi)[t/x]$	(2), Con	(2)	$@_t\phi$	(1), Con
	(4)	$AT_x(\psi)[t/x]$	(2), Con	(3)	$@_t\psi$	(1), Con
$\wedge$ neg	(1)	$\neg\langle \lambda x.AT_x(\phi) \wedge AT_x(\psi)\rangle(t)$		(1)	$\neg @_t(\phi \wedge \psi)$	
	(2)	$\neg[AT_x(\phi)[t/x] \wedge AT_x(\psi)[t/x]]$				
	(3)	$\neg AT_x(\phi)[t/x] \mid \neg AT_x(\psi)[t/x]$	(2), Dis	(2)	$\neg @_t\phi \mid \neg @_t\psi$,	(1), Dis
@ pos	(1)	$\langle \lambda x.\forall x(x = n \rightarrow AT_x(\phi))\rangle(t)$		(1)	$@_t @_n\phi$	
	(2)	$\forall x(x = n \rightarrow AT_x(\phi))$				
	(3)	$AT_x(\phi)[n/x]$	(2), Sub	(2)	$@_n\phi$	(1), @
@ neg	(1)	$\neg\langle \lambda x.\forall x(x = n \rightarrow AT_x(\phi))\rangle(t)$		(1)	$\neg @_t @_n\phi$	
	(2)	$\neg\forall x(x = n \rightarrow AT_x(\phi))$				
	(3)	$\neg AT_x(\phi)[n/x]$	(2), Sub	(2)	$\neg @_n\phi$	(1), @
$\Box$ pos	(1)	$\langle \lambda x.\forall y(Rxy \rightarrow AT_y(\phi))\rangle(t)$		(1)	$@_t\Box\phi$	
	(2)	Rtn		(2)	$@_t n$	
	(3)	$\forall y(Rty \rightarrow AT_y(\phi))$				
	(4)	$AT_y(\phi)[n/y]$	(2), (3) MP	(3)	$@_n\phi$	(1),(2), $\Box$
$\Box$ neg	(1)	$\neg\langle \lambda x.\forall y(Rxy \rightarrow AT_y(\phi))\rangle(t)$		(1)	$\neg @_t\Box\phi$	
	(2)	$\neg(\forall y(Rty \rightarrow AT_y(\phi)))$				
	(3)	Rtc	(2), Exi	(2)	$@_t\Diamond c$	(1), $\Diamond$
	(4)	$AT_y(\phi)[c/y]$	(2), Exi	(3)	$@_c\phi$	(1), $\Diamond$

Table 2: Corresponding proof rules

The translation $(\cdot)^*$ is extended for the new literals as follows:

$$\begin{array}{lll} P'(s,t_1,\ldots,t_k)^* & := & @_s P(t_1,\ldots,t_n) \\ (t_i = t_j)^* & := & t_i = t_j. \end{array}$$

We don't translate the *QHL* terms $@_s q$ but just view them as the first-order terms $q(a)$. Again, properties (3)–(6) still hold. (The first-order tableau calculus has to respect the two sorts of course. For example, $\forall x P'(s,x)$ does not yield the not correctly typed $P'(s,s)$ by universal instantiation.) The atomic hybrid formula $t_i = t_j$ is translated as $\langle \lambda x. t_i = t_j \rangle (x)$. This is done to have a syntactic analogue of $@_s(t_i = t_j)$. In a first-order proof, β-reduction can be applied to $\langle \lambda x. t_i = t_j \rangle (s)$ or its negation, yielding $t_i = t_j$ and $\neg t_i = t_j$, respectively. This proof step corresponds to an application of one of the $@=$ rules on the $(\cdot)^*$ translations in a *QHL* tableau.

It is immediate that the quantifier rules can be mimicked in *QHL* tableaux (provided they respect the sorts).

For the application of replacement, there are now terms $@_n q$ for q a non-rigid designator and n a nominal. The replacement rule can then with the premise $n = m$ replace $@_n q$ by $@_m q$ in any atom. But $n = m$ back-translates to $@_n m$ and from that the *QHL* equality rule DD yields $@_n q = @_m q$. Now replacement in *QHL* with this premise on the same atom yields the same result.

Thus all first-order rules have a corresponding *QHL* analogue and we are done.

4.6 Completeness for specific frame classes

We only considered the (quantified) hybrid logic of the class of all frames. Here we establish completeness for every elementary first-order definable class of frames which is closed under and reflects generated subframes. A class of frames is closed under generated subframes if all generated subframes of its members are in the class. A class reflects generated subframes if whenever $\mathcal{F}$ is in the class and $\mathcal{F}$ is a generated subframe of $\mathcal{F}'$, then also $\mathcal{F}'$ is in the class. Note that this implies that the class is closed under disjoint unions. Closure under and reflection of generated subframes is a requirement which reflects the local evaluation of modal formulas.

We recall from [1], that every such elementary class of frames is definable by a first-order sentence $\forall y \gamma(y)$, in which $\gamma(y)$ is equivalent to a pure hybrid $\mathcal{HL}(@,\downarrow)$ sentence γ' (i.e., without propositional variables nor nominals). As AT preserves meaning we may without loss of generality assume that $\gamma(y) = AT_y(\gamma')$.

Let such a class K be defined by $\forall y \gamma(y)$. Then a *QHL* sentence ϕ is valid on K iff $\forall y \gamma(y) \to AT_x(\phi)[c/x]$ for c a new parameter is first-order valid. In that case, there is a first-order tableau proof starting with

$$\begin{array}{ll} 1. & \forall y \gamma(y) \\ 2. & \neg AT_x(\phi)[c/x]. \end{array}$$

Whence the proof will develop almost as for $AT_x(\phi)[c/x]$ except that for any state parameter or nominal s, $\gamma(s)$ may be introduced on the branch. This insight leads to the following rule to be added to the *QHL* tableau system:

$$\frac{}{@_s \gamma'} \quad \text{for } s \text{ on the branch.}$$

Now every time a $\gamma(s)$ is added to the branch in the first-order proof, we apply the new rule on s in the *QHL* proof. Because of the assumption on the form of γ, translating $\gamma(s)$ by $(\cdot)^*$ yields $@_s \gamma'$. Thus we have shown

Theorem 6 *Let γ a pure nominal free hybrid sentence which axiomatizes the class K. Then adding the above rule to the QHL tableau calculus yields completeness for the quantified hybrid logic of the class K.*

5 Conclusions

The positive effects of hybridization in propositional logic extend well to the first-order case. In fact, one could argue that the need for hybridization is felt much stronger in first-order modal logic. The field is plagued with failures of desirable properties, and consequently more difficult and obscure than its propositional counterpart. Here we have presented an extremely general completeness theorem (Theorem 6) covering virtually all modally interesting elementary frame classes. In a companion paper we have shown that the presented calculus can be used to construct interpolants. Interpolation is one of the properties which fail in many quantified modal logics. This theorem also extends to all frame classes from Theorem 6. These very general results indicate that the additions to the syntax are natural and extremely useful.

The paper contained two important results. First of all, the proof method for showing completeness. The standard translation was used in a non-trivial way to transfer a first-order result into the modal setting. In the hybrid language, this was particularly easy, as it contains such first-order proof-elements as parameters. We wonder if the translation using λ-abstraction could also be used to obtain completeness of traditional tableau systems. We think this is an important research direction. Too many proofs are repeated over and over with tiny changes in modal logic. Maybe hybridization is needed to change modal logic into a field in which results are recycled instead of proofs. It's worth the price.

The second important result is our treatment of definite descriptions like *Miss America*.In *QHL* it is not possible to write intensional terms as in Montague's IL. The hidden variables in intensional terms cause many technical problems and make IL mathematically complicated. The use of @ to ground non-rigid designators to states is a simple remedy.

References

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

[2] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation lemma in quantified modal logic. Report PP–2001–19, ILLC. To appear in Annals of Pure and Applied Logic.

[3] K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 44(2):201–206, 1979.

[4] M. Fitting. *First Order Logic and Automated Theorem Proving (second edition)*. Springer Verlag, 1996.

[5] M. Fitting. *Types, Tableaus, and Gödel's God*. Unpublished Draft, 1999.

[6] M. Fitting and R. Mendelsohn. *First–Order Modal Logic*. Kluwer, 1998.

[7] D. Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996.

[8] R. Goré. Tableau methods for modal and temporal logics. In M D'Agostino,et al editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.

[9] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[10] J. Seligman. Internalization: The case of hybrid logic. *Journal of Logic and Computation*, 11(5):671–689, 2001.

[11] R. Smullyan. *First Order Logic*. Springer–Verlag, 1968.

[12] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.

[13] J. van Benthem. Correspondence theory. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 167–248. Reidel, Dordrecht, 1984.

Beyond Pure Axioms:

Node Creating Rules in Hybrid Tableaux[2]

(DRAFT VERSION)

Patrick Blackburn
INRIA Lorraine
615, rue du Jardin Botanique
54602 Villers les Nancy Cedex, France
Email: patrick@aplog.org

Balder ten Cate
ILLC, Universiteit van Amsterdam
Nieuwe Doelenstraat 15
1012CP Amsterdam, Netherlands
Email: b.ten.cate@hum.uva.nl

Abstract

We present a method of extending the tableau calculus for the basic hybrid language which automatically yields completeness results for many frame classes that cannot be defined by means of pure axioms (for example, Church-Rosser frames). The extended calculus makes use of *node-creating rules*. These rules trade on the idea of using nominals to perform skolemization on formulas of the *strong hybrid language*. Alternatively, viewing them from a Hilbert-style perspective, such rules can be viewed as a systematic generalization of Gabbay's rules for the undefinable. Our tableau complexness result covers all frame classes definable by *pure nominal-free universal existential sentences* of the strong hybrid language. This properly includes all frame classes definable by universal existential first-order sentences.

1 Basic Hybrid Logic

Basic hybrid logic is the result of extending modal logic with *nominals* and the *@-operator*. Suppose we are given a set σ of modalities, and two (countably infinite) disjoint sets PROP (whose elements are typically written p, q, and r, possibly subscripted, and called proposition letters) and NOM (whose elements are typically written i, j, k, and l, possibly subscripted, and called nominals). Then the basic hybrid language over σ, PROP and NOM is defined as follows:

$$\phi ::= p \mid i \mid \neg\phi \mid \phi \wedge \psi \mid \triangle(\phi_1, \ldots \phi_n) \mid @_i\phi.$$

Here p is a proposition letter, i is a nominal, and $\triangle$ is an n-ary modality (an element of σ). Thus, except for the clauses for i and $@_i\phi$, this is the standard definition of a modal language with arbitrary arity modalities (see, for example, Definition 1.12 in [6]). We follow the usual convention of writing $\Diamond\phi$ rather than $\triangle(\phi)$ when working with unary modalities.

What do the clauses for i and $@_i\phi$ give us? Nominals are special proposition letters that are true at precisely one node in any model: they 'name', or 'label', the unique node they are true at. The @ operator allows us to assert that a formula is true at a named node: $@_i\phi$ says that ϕ is true at the node named by the nominal i. In short, by hybridizing the modal language we make it referential: it can now talk about individual nodes in Kripke models.

Let's be precise. A *model* for the basic hybrid language over σ, PROP, and NOM, is a Kripke model $\mathfrak{M} = (W, (R^{\triangle})_{\triangle\in\sigma}, V)$, such that the valuation V assigns *singleton* subsets of W to nominals; such valuations are sometimes called *hybrid valuations*. Apart from this restriction, everything is standard: W is a non-empty set of nodes, and for all $\triangle$ in σ, if $\triangle$ is an n-ary modality, then $R^{\triangle}$ is an $n+1$-ary relation. Following standard terminology we call the pair $(W, (R^{\triangle})_{\triangle\in\sigma})$ the *frame* underlying the model.

[2]This paper was written during a visit by Balder ten Cate to the Langue et Dialogue team, LORIA, Nancy, France. This visit was carried out as part of the INRIA funded partnership between LIT (Language and Inference Technology, University of Amsterdam) and LED (Langue et Dialogue, LORIA, Nancy).

Given such a hybrid model $\mathfrak{M}$, we interpret our language as follows:

$$
\begin{array}{lll}
\mathfrak{M}, w \models a & \text{iff} & w \in V(a), \text{ where } a \in \text{PROP} \cup \text{NOM} \\
\mathfrak{M}, w \models \neg\phi & \text{iff} & \mathfrak{M}, w \not\models \phi \\
\mathfrak{M}, w \models \phi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \phi \text{ and } \mathfrak{M}, w \models \psi \\
\mathfrak{M}, w \models \triangle(\phi_1, \dots, \phi_n) & \text{iff} & \text{there are } v_1, \dots, v_n \in W \text{ such that} \\
 & & (w, v_1, \dots, v_n) \in R^{\triangle} \text{ and } \mathfrak{M}, v_1 \models \phi_1 \dots \mathfrak{M}, v_n \models \phi_n \\
\mathfrak{M}, w \models @_i\phi & \text{iff} & \mathfrak{M}, v \models \phi, \text{ where } V(i) = \{v\}
\end{array}
$$

Readers unfamiliar with arbitrary arity modalities should note that in the unary case the clause for modalities simplifies down to the more familiar:

$$\mathfrak{M}, w \models \Diamond\phi \quad \text{iff} \quad \text{there is a } v \in W \text{ such that } (w, v) \in R^{\Diamond} \text{ and } \mathfrak{M}, v \models \phi.$$

If $\mathfrak{M}, w \models \phi$ then we say that ϕ is *satisfied* in $\mathfrak{M}$ at w. For any frame $\mathfrak{F}$, if ϕ is satisfied in every model $(\mathfrak{F}, V)$ at every w in $\mathfrak{F}$ no matter which (hybrid) valuation V we choose, then we say that ϕ is *valid on* $\mathfrak{F}$. A formula is *valid* if it is valid on every frame. A formula is *valid on a class of frames* F if it is valid on every frame in F.

As promised, the hybridized language is referential. That nominals name is hardwired into the definition of valuations, and the clause for $@_i\phi$ says "evaluate ϕ at the node that i names". Notice that $@_ij$ says that the nominals i and j name the same node, that $@_i\Diamond j$ means that the node named i has the node named j as an $R^{\Diamond}$-successor, and that $@_i \triangle (j_1, \dots, j_n)$ means that the $n+1$ nodes named i, j_1,. . . ,j_n, stand in the $R^{\triangle}$ relation.

It's worth mentioning that the language we have just defined is a very simple hybrid language: far stronger hybrid languages have been studied, for example languages in which it is possible to bind nominals using the classical quantifiers $\forall$ and $\exists$ (see, for example, [9, 13, 8]). But the syntactic simplicity of the basic hybrid language is attractive. Moreover, its syntactic simplicity pays off in terms of computational simplicity: the satisfaction problem of the basic hybrid language (over the class of all models) is decidable in PSPACE (this is proved for the unary case in [1]; the proof extends to the arbitrary arity case). That is, the satisfaction problem for the hybridized language is (up to a polynomial) no more complex than the satisfaction problem for the underlying modal language.

2 Hybrid Tableaux

Hybridization gives us precisely the tools needed to define natural proof systems. This is because the basic hybrid apparatus of @-operators and nominals allows us to reason about what happens at particular nodes, and to extract information from under the scope of the modalities.

In Table 3 we give a sound and complete tableau calculus for basic hybrid logic (the calculus is an arbitrary arity modality version of the calculus of [3] that incorporates improvements to the equality rules introduced in [7]). To prove a formula ϕ, proceed as follows: chose a nominal, say i, that is not in ϕ, and start applying tableaux rules to $\neg@_i\phi$. That is, assert that it is possible to falsify ϕ in some model at a node named i, and use the tableau rules to try and build a falsifying model. If it turns out that this is *not* possible (that is, if the tableau closes) we have proved ϕ.

It is clear from Table 3 that it is the hybrid machinery that propels this calculus. For a start, all the rules are @-driven: we use @ to reason about what must hold at named nodes. And, given the semantics of the @ operator, the import of most of these rules should be clear. For example, the $\neg$ and $\wedge$ rules dismantle the Boolean connectives in the obvious way, and the @ rule lets us drop outermost occurrences of @. The first equality rule says that for any nominal i on a branch we can conclude that i is true at the node named i, which is obviously true. The second equality rule says that from $@_ij$ ("the nominals i and j name the same node") and $@_i\phi$ ("ϕ is true at the node named i") we can conclude $@_j\phi$ ("ϕ is true at the node named j").

But the hybrid apparatus has a second, deeper, role to play. This becomes apparent when we consider the $\triangle$ rules, and in particular the left-hand rule for $\triangle$. It is probably easier to see what is going on in the unary case: Table 4 displays the unary form of all three rules containing occurrences of $\triangle$ (that is, the two $\triangle$ rules and the third equality rule).

Table 3: Tableau calculus for the basic hybrid logic

$\neg$	$\dfrac{@_i\neg\phi}{\neg @_i\phi}$	$\dfrac{\neg @_i\neg\phi}{@_i\phi}$
$\wedge$	$\dfrac{@_i(\phi\wedge\psi)}{\begin{array}{c}@_i\phi\\ @_i\psi\end{array}}$	$\dfrac{\neg @_i(\phi\wedge\psi)}{\neg @_i\phi \mid \neg @_i\psi}$
$\triangle$	$\dfrac{@_i \triangle (\phi_1,\ldots,\phi_n)}{\begin{array}{c}@_i \triangle (k_1,\ldots,k_n)\\ @_{k_1}\phi_1\\ \vdots\\ @_{k_n}\phi_n\end{array}}$	$\dfrac{\neg @_i \triangle (\phi_1,\ldots,\phi_n) \quad @_i \triangle (k_1,\ldots,k_n)}{\neg @_{k_1}\phi_1 \mid \ldots \mid \neg @_{k_n}\phi_n}$
$@$	$\dfrac{@_i @_j\phi}{@_j\phi}$	$\dfrac{\neg @_i @_j\phi}{\neg @_j\phi}$
Equality	$\dfrac{(i \text{ occurs on branch})}{@_i i}$	$\dfrac{@_i j \quad @_i\phi}{@_j\phi}$
	$\dfrac{@_i \triangle (k_1,\ldots,k_n) \quad @_{k_1}k'_1 \quad \ldots \quad @_{k_n}k'_n}{@_i \triangle (k'_1,\ldots,k'_n)}$	
Closure	A branch is closed if it contains a formula and its negation. A tableau is closed if all of its branches are closed.	

Table 4: Unary form of the rules containing $\triangle$

$\Diamond$	$\dfrac{@_i\Diamond\phi}{\begin{array}{c}@_i\Diamond k\\ @_k\phi\end{array}}$	$\dfrac{\neg @_i\Diamond\phi \quad @_i\Diamond k}{\neg @_k\phi_1}$
Equality	$\dfrac{@_i\Diamond k \quad @_k k'}{@_i\Diamond k'}$	

It's the left-hand $\Diamond$ rule that is crucial. We are given the assertion $@_i\Diamond\phi$ ("at the node named i, $\Diamond\phi$ is true"). From this we conclude two things: $@_i\Diamond k$ ("the node named i has at least one successor, which we shall call k") and $@_k\phi$ ("at the node named k, ϕ holds"). That is, we have used the nominal k to decompose the the original information into two simpler parts. Where does the nominal k come from? Nowhere: it's brand new. That is, it is a nominal that hasn't previously occurred on the tableau branch. It is a newly created name for the ϕ-witnessing R-successor node to i that must exist if the original assertion is true.

The general form this rule takes in the left-hand $\triangle$ rule should now be clear: distinct new nominal k_1, …, k_n are introduced to extract information from under the scope of the n-place modality. Incidentally, the convention that nominals only occurring below the bar of a tableau rule are new (and syntactically distinct) is a convention we shall use in Sections 4 and 5 of the paper when we discuss node creating rules.

We won't discuss the right-hand $\triangle$ rule, nor the third equality rule; the unary case should make clear what they do. Neither rule introduces new nominals.

3 Adding Pure Axioms

If the first benefit of hybridization is that it is straightforward to define natural proof systems, the second is this: once a base system has been defined, it is easy to extend it to a complete system for many important classes of frames. All we have to do is add *pure axioms.*

A *pure formula* is a formula that contains no propositional letters. Many properties of frames can be defined using pure formulas. For example, transitivity of R is defined by

$$@_i(\Diamond\Diamond j \rightarrow \Diamond j).$$

(This formula is valid on every transitive frame and falsifiable on every non-transitive frame.) Similarly, irreflexivity of R is defined by

$$@_i\neg\Diamond i,$$

and trichotomy of R (that is, $\forall xy(xRy \vee x = y \vee yRx)$ is definable by

$$@_i\Diamond j \vee @_i j \vee @_j\Diamond i.$$

As these examples show, the frame-defining powers of pure formulas are different from those of orthodox modal formulas (that is, formulas built out of ordinary proposition letters): transitivity is definable using an orthodox modal formula, but irreflexivity and trichotomy are not.

But for present purposes the key fact about pure formulas is this: when used as axioms (over a suitable base proof system) they are guaranteed to be complete with respect to the class of frames they define (for a detailed proof, see Chapter 7.3 of [6]). For example, adding the pure axiom for transitivity given above to a (suitable) base proof system yields a system complete for transitive frames, and adding the pure axiom for irreflexivity yields a system complete for irreflexive frames. These results are cumulative: adding both axioms yields a system complete for strict partial orders. Many suitable base proof systems are known: for example, see Chapter 7.3 of [6] and [5] for Hilbert-style approaches, and Seligman [15] for sequent calculi. And, once a small technical point has been observed, the tableau calculus just presented is suitable too.

The small technical point is this: the tableaux rules only take as input (and produce as output) @-prefixed formulas. This means that the pure axioms used with tableaux must be @-prefixed too, but some natural axioms (such as the trichotomy defining formula given above) are not of this form. However this is only an apparent restriction: if a pure formula ϕ defines a class of frames F, then $@_k\phi$ defines F as well, where k is any nominal not occurring in ϕ. Thus any pure axiom can be put a form suitable for tableau processing simply by prefixing a new nominal. For example

$$@_k(@_i\Diamond j \vee @_i j \vee @_j\Diamond i)$$

is a suitable tableaux axiom for trichotomy. With this observed, we have the following completeness result:

Theorem 7 *Let A be a finite collection of @-prefixed pure formulas. Let TA be the tableau system given above extended with the following rule: at any stage in the tableau, we are free to choose a formula from A, instantiate it with nominals occurring on some branch B, and then add the result to the end of branch B. Then TA is complete with respect to the class of frames defined by $\bigwedge_{\phi\in A}\phi$.*

PROOF: The result for unary modalities is proved in [3]. The extension to arbitrary arity modalities is routine. □

Note the restriction on instantiations: we only introduce instances of the axioms which make use of 'old' nominals. In fact, this is the restriction we shall remove when we introduce node creating rules in the following section. But before doing this, let's look at the theorem just stated, and indeed at the whole idea of frame-definability using pure formulas, from a rather different perspective.

As we mentioned earlier, there are far stronger hybrid languages than the basic hybrid language, including hybrid languages in which we can bind nominals with the classical quantifiers $\forall$ and $\exists$. In what follows we call the extension of a basic hybrid language with such quantifiers a *strong hybrid language*. Precise definitions of the syntax and semantics of strong hybrid languages can be found in [8], but the reader will probably find the examples given below of these languages in action clear enough.

Strong hybrid languages give us full first-order expressive power over frames (this has been known since Arthur Prior's pioneering work on hybrid languages in the late 1960s, such as some of the papers in [14]; for a more recent discussion, see [8]). This means that strong hybrid languages are vastly more powerful than the basic languages of the present paper, and have undecidable satisfaction problems. But it spite of these differences, strong languages throw useful light on what is going on when we use pure axioms.

As we have already remarked, the following pure formula of the basic hybrid defines transitivity:

$$@_i(\Diamond\Diamond j \rightarrow \Diamond j).$$

Using a strong hybrid language, this can be re-expressed as follows:

$$\forall x \forall y @_x(\Diamond\Diamond y \rightarrow \Diamond y).$$

That is, we have substituted variables for nominals, and taken the universal closure. It should be clear that the basic hybrid formula we started with is *valid* on a frame $\mathfrak{F}$ iff the strong sentence is *true* on the frame $\mathfrak{F}$: the $\forall$ quantifier captures the effect of trying out all possible assignments to the nominals/variables. To give another example, the basic formula

$$@_i \neg\Diamond i$$

which defines irreflexivity can also be re-expressed as

$$\forall x @_x \neg\Diamond x.$$

Indeed, the process is completely general. Call a sentence of a strong hybrid language of the form $\forall x_1 \ldots x_n \phi$, where ϕ does not contain any quantifiers, propositional letters, or nominals, a PUNF-sentence (this stands for *pure, universal, nominal-free sentence*). The previous examples show that transitivity and irreflexivity can be expressed by PUNF-sentences, and in fact we have the following:

Proposition 1

- *For any pure formula ϕ of the basic hybrid language there is a* PUNF-*sentence $\phi^\forall$ of the strong hybrid language such that for any frame $\mathfrak{F}$, ϕ is valid on $\mathfrak{F}$ iff $\phi^\forall$ is true on $\mathfrak{F}$.*
- *For any* PUNF-*sentence ϕ of the strong hybrid language there is a pure formula ϕ^B of the basic hybrid language such that for any frame $\mathfrak{F}$, ϕ is true in $\mathfrak{F}$ iff ϕ^B is valid on $\mathfrak{F}$.*

PROOF: Item 1 is a generalization of the above examples: suppose we are given a pure formula $\phi(i_1, \ldots, i_n)$, where $i_i, \ldots, i_n$ are all the nominals in ϕ, then the required $\phi^\forall$ is $\forall x_1 \cdots \forall x_n \phi([i_1 \leftarrow x_1, \ldots, i_n \leftarrow x_n])$.

As for item 2, any PUNF-sentence ϕ of the strong hybrid language must have the form $\forall x_1 \cdots \forall x_n \phi(x_1, \ldots, x_n)$ where ϕ contains no quantifiers, proposition letters, or nominals. Hence the required ϕ^B is $\phi([x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n])$. □

This gives us a new perspective on pure axioms. For a start it tells us that the frame classes expressible by pure axioms are simply the frame classes definable by PUNF-sentences. Moreover, it gives us an alternative way of thinking about Theorem 7:

Theorem 8 *Let S be a finite collection of* PUNF-*sentences, all of which have an @-prefixed matrix. Let TS be the tableau system given above extended with the following rule: at any stage in the tableau, we are free to choose a formula from S, perform universal instantiation on it using nominals present on branch B, and add the resulting basic hybrid formula to the end of branch B. Then TS is complete with respect to the class of frames defined by $\bigwedge_{\phi \in S} \phi$.*

PROOF: Simply a reformulation of Theorem 7. It's worth stressing that we are only using the strong hybrid language in the background, as a way of recording extra axioms. The formulas actually used on the tableau are, just as before, basic formulas: in concrete terms this reformulation changes nothing.

We insist that the matrices of the PUNF-sentences be @-prefixed to ensure that the result of universal instantiation can be processed by the tableau. As we have remarked, this restriction does not decrease the frame-defining powers at our disposal. □

While the perspectival shift from basic to strong hybrid languages is not particularly deep, it will be useful. We are about to introduce node creating rules, which will give us complete tableau for many more frame classes. Which classes? As we shall eventually see, any frame class definable by a strong sentence of the form $\forall x_1 \ldots x_n \exists y_1 \ldots y_m \phi$ (where ϕ does not contain any quantifiers, propositional letters, or nominals). In essence, we are going to show how to move from the universal fragment of the strong hybrid language, to the universal-existential fragment.

4 Node Creating Rules for Geach Formulas

Many important frame classes cannot be captured using pure axioms. One example is the class of Church-Rosser frames, that is, frames satisfying the property

$$\forall wvu \exists t (wRv \land wRu \rightarrow vRt \land uRt).$$

Another is the class of *right-directed* frames, that is, frames satisfying the (stronger) property

$$\forall vu \exists t (vRt \land uRt).$$

We shall now show how to capture the logic of such frame classes in the setting of a tableau calculus. We do so in two steps. In this section we discuss the Church-Rosser property, and go on to define tableau rules capable of handling any class of frames definable by a Geach formula. In the following section we discuss right-directedness, and go on to show that we can handle any condition expressible in the (pure, nominal-free) universal-existential fragment of the strong hybrid language.

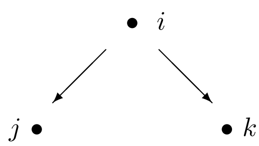
A little experimentation will quickly convince the reader that there is no pure axiom defining Church-Rosser: there is simply no way to get a handle on the convergence node (the t in the above first-order definition). However if we think more dynamically, and in particular, if we think in terms of tableau rules that create new nodes, we see that there is a very natural way of getting exactly what we want:

$$\frac{@_i \Diamond j \quad @_i \Diamond k}{\begin{array}{c} @_j \Diamond l \\ @_k \Diamond l \end{array}}$$

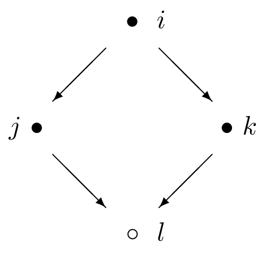
Remember our notational convention for tableau expansion rules: any nominal that occurs only below the bar is newly introduced. Therefore the above expansion rule should be read as follows:

> *Whenever $@_i \Diamond j$ and $@_i \Diamond k$ occur on a branch, we are allowed to create a new nominal l and add $@_j \Diamond l$ and $@_k \Diamond l$ to the branch.*

That is, suppose that we already have the following configuration of nodes:



Then we are allowed to introduce a brand new nominal l that acts as a name for the required convergence node:



Two remarks. First, this rule amounts to “using nominals as skolem constants”: in effect we are using nominals to eliminate the existential quantifier $\exists t$ in the first-order definition given above. Second, readers familiar with completeness proofs for hybrid tableau will realize that this rule is not only sound, but complete as well. For by systematically applying this rule, we saturate any branch of a tableau with convergence points. And this guarantees that any countermodel to the input that we find will be a countermodel based on a frame with the Church-Rosser property.

Does this strategy generalize to other frame properties? In fact, it generalizes straightforwardly to all frame properties that can be characterized by *Geach formulas* (see [12]). These are orthodox modal axioms of the following form (where $m, n, s, t \geq 0$, and p is an ordinary propositional letter):

$$\Diamond^m \Box^n p \rightarrow \Box^s \Diamond^t p.$$

Geach axioms cover many well-known properties, among which are transitivity ($m = 2, t = 1, n = s = 0$) and Church-Rosser ($m = n = s = t = 1$).

Now, every Geach axiom corresponds to a first-order frame property, namely the following:

$$\forall xyz \exists u (xR^m y \wedge xR^s z \rightarrow yR^n u \wedge zR^t u).$$

Here, as is customary, we have used R^n as a shorthand for a sequence of n R-transitions. In other words, the above formula is shorthand for the following formula:

$$\begin{array}{l} \forall xyza_1 \ldots a_{m-1} b_1 \ldots b_{s-1} \exists u c_1 \ldots c_{n-1} d_1 \ldots d_{t-1} \\ \quad ((xRa_1 \wedge \ldots \wedge a_{m-1} Ry) \wedge (xRb_1 \wedge \ldots \wedge b_{s-1} Rz) \rightarrow \\ \qquad (yRc_1 \wedge \ldots \wedge c_{n-1} Ru) \wedge (zRd_1 \wedge \ldots \wedge d_{t-1} Ru)). \end{array}$$

And now it is easy to define the required tableaux rules. All we need to do is 'walk along' this formula from left to right, writing down the transition relations in hybrid notation. If we do this (using i instead of x, j instead of y, k instead of z, and l instead of u) we obtain the following tableaux expansion rule:

$$\frac{@_i \Diamond a_1 \quad \ldots \quad @_{a_{m-1}} \Diamond j \qquad @_i \Diamond b_1 \quad \ldots \quad @_{b_{s-1}} \Diamond k}{\begin{array}{c} @_j \Diamond c_1 \\ \vdots \\ @_{c_{n-1}} \Diamond l \\ @_j \Diamond d_1 \\ \vdots \\ @_{d_{t-1}} \Diamond l \end{array}}$$

Let's look at a couple of examples. If we apply this schema to the Church-Rosser property (that is, if we instantiate m, n, s and t to 1), then this gives us exactly the expansion rule we have just discussed. It's also interesting to see the rule for transitivity produced by this schema:

$$\frac{@_i \Diamond j \quad @_j \Diamond k}{@_i \Diamond k}$$

Now, this rule does not create new nodes (after all, there's no existential quantifier in the definition of transitivity to be skolemized away). Moreover, as transitivity can be characterized by a pure formula (namely $@_k(\Diamond\Diamond i \rightarrow \Diamond i)$), it doesn't offer us anything new as far as frame classes are concerned. Nonetheless, from a computational perspective, this rule is better than the rule for transitivity of the previous section, which simply dumps the defining pure formula on the tableau branch. The new rule, so to speak, sits and waits for the antecedent to be fulfilled, and then fires to build the required R-transition between i and k.

Two remarks. First, our approach to the Geach rules is similar to work by Basin, Matthews and Viganò [2] in the labelled deduction tradition. These authors don't work with hybrid logic, rather they work with an orthodox modal language and a labelling algebra interacting through a fixed interface. Nonetheless, their use of skolemization has close affinities with the rules just discussed, and their approach can handle all Geach conditions. Second, we remark that the schema for Geach formulas could be generalized to languages containing several distinct unary modalities: all that needs to be done is to add appropriate indices to the diamonds in the above rule (to indicate which binary relations interpret the various diamonds). But we shall leave this to the reader, for a broader generalization awaits us.

5 Node Creating Rules and the Universal-Existential Fragment

We now have a way of obtaining complete tableaux systems for frame properties such as the Church-Rosser property. However, the problem with the *right-directedness* property has still not been solved, since right-directedness is not equivalent to a Geach formula. In this section we will show how to handle this condition, and then go on to show that we can handle any condition definable in the (pure, nominal-free) universal-existential fragment of the strong hybrid language.

Right-directness is not definable by a Geach formula (in fact, no orthodox modal formula whatsoever defines this condition, as the class of right-directed frames is not closed under disjoint-union). There *is* a formula of the basic hybrid language that defines the class of right-directed frames, namely $@_i\Box p \rightarrow @_j\Diamond p$. Unfortunately, as this is not a pure formula, we cannot apply Theorem 7 to obtain completeness automatically. However, a little lateral thinking shows that it is possible to handle this frame condition naturally, using a more general kind of node creating rule.

First observe that right-directness can be defined by a very simple sentence of the *strong* hybrid language, namely

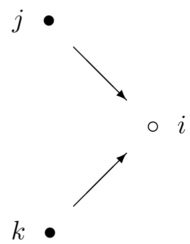
$$\forall xy\exists z(@_x\Diamond z \wedge @_y\Diamond z).$$

And now the crucial point: this strong formula provides a 'recipe' for what we should do in a tableau proof, namely this: instantiate the universally bound variables with old nominals, and the existentially bound variables with new nominals. In short, it suggests the following tableau rule:

$$\frac{(j,\, k \text{ occur on the branch})}{@_j\Diamond i \wedge @_k\Diamond i}$$

This rule provides exactly what we want. It says: suppose we have two nodes named j and k:

Then we are free to create a new node, glue it in place to the right of these nodes, and give it a name, say i.



As in the Church-Rosser example of the previous section, this new rule essentially boils down to "skolemization using nominals"; the difference is we are now thinking directly in terms of skolemizing *strong* hybrid formulas.

Of course, in one sense we have taken a step backwards. This rule is not as nice as those for Geach conditions: we are back to simply placing instances of a complex formula on the branch of a tableaux. But we gain something important in return: generality.

Call a sentence of a strong hybrid language of the form

$$\forall x_1, \cdots x_n \exists y_1 \cdots y_m \phi$$

where ϕ does not contain any quantifiers, propositional letters, or nominals, a PUENF-sentence (this stands for pure, *universal existential*, nominal-free sentence). Then for any such sentence we have the following tableaux rule:

$$\frac{(i_1, \ldots, i_n \text{ occur on the branch})}{\phi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow k_1, \ldots, y_n \leftarrow k_m]}$$

Here we make use of the convention mentioned in Section 2: the $k_1, \ldots, k_m$ are *new, distinct*, nominals. And now we have:

Theorem 9 *Let S be a finite collection of* PUENF*-sentences, all of which have an @-prefixed matrix. Let TS be the tableau system given above extended with the following rule: at any stage in the tableau, we are free to choose a formula from S, perform the tableaux rule associated with this formula using nominals present on branch B together with the needed new nominals; we add the basic hybrid formula so obtained to the end of branch B. Then TS is complete with respect to the class of frames defined by $\bigwedge_{\phi \in S} \phi$.*

PROOF: A proof is given in Appendix A. It uses the same idea that was used in [3] to prove Theorem 7 (or equivalently, Theorem 8). The fact that we are now instantiating PUENF-sentences rather that PUNF-sentences does not change the heart of the argument. □

Notice that, in constrast to our previous result concerning Geach formulas, Theorem 9 applies to modalities of arbitrary arity.

6 Conclusion

A fundamental result of hybrid logic is that it is possible to define basic proof calculi in such a way that any extension with pure axioms is automatically complete. While general, this result has limitations: it does not cover all Geach formulas, let alone conditions like right-directedness. In this paper we have shown that it straightforward to overcome these limitations. The key is to make use of node creating rules. These enable us to provide complete proof systems in the basic hybrid language that cover any frame condition definable by a PUENF-sentence. To conclude the paper we briefly note another perspective on node creating rules, and make a conjecture.

Node creating rules are not restricted to tableaux proof systems: they make perfectly good sense in Hilbert-style proof systems, where they become special rules of proof. For example, when working with a Hilbert system, the Church-Rosser tableau expansion rule takes the following form:

$$\frac{\vdash (@_i\Diamond j \wedge @_i\Diamond k \rightarrow @_j\Diamond l \wedge @_k\Diamond l) \rightarrow \theta \text{ and } l \notin \theta \text{ and } l \neq i, j, k}{\vdash \theta}$$

Likewise, the right-directedness expansion rule becomes the following Hilbert-style proof rule:

$$\frac{\vdash (@_j\Diamond i \wedge @_k\Diamond i) \rightarrow \theta \text{ and } i \notin \theta \text{ and } i \neq j, k}{\vdash \theta}$$

And in general, given any PUENF-sentence $\forall x_1, \cdots x_n \exists y_1 \cdots y_m \phi \in A$ the corresponding Hilbert-style rule is:

$$\frac{\vdash \phi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow k_1, \ldots, y_m \leftarrow k_m] \rightarrow \theta}{\vdash \theta}$$

where $k_1, \ldots, k_m$ are distinct, unequal to $i_1, \ldots i_n$, and don't occur in θ.

Two remarks. First, proving completeness in the Hilbert system case is rather similar to the proof in the tableaux case. The crucial work takes place in an extended version of the Lindenbaum Lemma (that is the step in the completeness proof where a consistent set is transformed into a maximal consistent set; see, for example, Lemma 7.25 of [6]). Essentially we add the required new nominals when performing the inductive Lindenbaum construction; the new rules guarantee the consistency of these additions. Full details are presented in [5].

Second, the fact that node creating rules make sense in the Hilbert setting should not be a surprise, since they are essentially something familiar to modal logicians: *rules for the undefinable*. Gabbay [10] introduced the *irreflexivity rule*, an additional rule of proof for orthodox modal language that makes it possible to directly construct irreflexive models, and Venema [16] presents a far-reaching generalization to logics containing the difference operator. This is not the place to make detailed comparisons, but we make two remarks. First, because the notion of reference to nodes is primitive in hybrid logic, the hybrid rules are arguably more natural (though this is partly a matter of taste). Second, in the hybrid logical case, there is a particularly simple answer to the question "But where do such rules come from?": they reflect the skolemization possibilities offered by PUENF-formulas.

We close with a conjecture. In one sense, we know how many frame classes are covered by this result, namely precisely the frame classes definable by PUENF-formulas. But obviously it would be nice to back this up with a characterization in terms of traditional correspondence theory. We have such a result: in [4] we show that the frame conditions by PUENF-formulas are exactly the universal-existential closures of *strongly bounded* first-order formulas. This is a large class of formulas, and we conjecture it covers every Sahlqvist-definable frame class. If this could be proved it would nicely supplement the result of Goranko and Vakarelov [11], which states that in *reversive* hybrid languages (that is, languages where the set of modalities is closed under converses) every Sahlqvist formula is equivalent to a pure formula. Incidentally, note that the sets of PUENF-definable and Sahlqvist-definable frame conditions cannot be identical, since many PUENF-definable (indeed PUNF-definable) conditions such as irreflexivity, antisymmetry, and discreteness are not definable by any orthodox modal formula at all.

References

[1] 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.

[2] David Basin, Seán Matthews, and Luca Viganò. Labelled propositional modal logics: Theory and practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.

[3] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:137–168, 2000.

[4] P. Blackburn and B. ten Cate. Correspondence theory for hybrid logic, 2002. Manuscript, INRIA Lorraine and ILLC, University of Amsterdam.

[5] P. Blackburn and B. ten Cate. Hybrid axiomatics: the propositional case, 2002. Manuscript, INRIA Lorraine and ILLC, University of Amsterdam.

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

[7] P. Blackburn and M. Marx. Tableaux for quantified hybrid logic. In C. Areces and M. de Rijke, editors, *Methods for Modalities 2, Workshop Proceedings, November 29-30, 2001*. ILLC Amsterdam, 2001.

[8] 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.

[9] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[10] D.M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. Reidel, Dordrecht, 1981.

[11] Valentin Goranko and Dmiter Vakarelov. Sahlqvist formulas in hybrid polyadic modal logics. *Journal of Logic and Computation*, 11(5):737–754, 2001. Special issue on hybrid logic.

[12] G. Hughes and M.J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[13] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93:263–332, 1991.

[14] A.N. Prior. *Papers on Time and Tense*. University of Oxford Press, 1968.

[15] J. Seligman. Internalization: The case of hybrid logics. *Journal of Logic and Computation*, 11(5):671–689, 2001. Special issue on hybrid logic.

[16] Y. Venema. Derivation rules as anti-axioms in modal logic. *Journal of Symbolic Logic*, 58:1003–1034, 1993.

A Completeness

.

What follows is an adaptation of the completeness theorem for hybrid tableaux presented in [3]. Let S be a finite set of PUENF-sentences, all of which have an @-prefixed matrix. With TS we denote the tableau system of Table 1, extended with the following rule:

> Given any PUENF-sentence $\forall x_1 \ldots x_n \exists y_1 \ldots y_m \phi \in S$ and any sequence of nominals $i_1 \ldots i_n$ occurring on the branch. Pick new nominals $j_1 \ldots j_m$ that do not occur on the branch, and add the formula $\phi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow j_1, \ldots y_m \leftarrow j_m]$ to the end of the branch.

We are about to prove Theorem 9, which states that TS is a sound and complete tableau calculus for the class of frames satisfying $\bigwedge_{\phi \in S} \phi$. First, we will state soundness.

Theorem 10 (Soundness) *Suppose a branch Σ (i.e., a set of satisfaction statements) is satisfiable. Then after any application of a tableau expansion rule, one of the outcomes is satisfiable.*

PROOF: The proof is straightforward, and proceeds by a case distinction. □

Before we proceed with completeness, we need to make one observation. Notice that at all times during the tableau construction process, we are only concerned with @-prefixed formulas or negations of @-prefixed formulas. Let us call such formulas *satisfaction statements*.

Definition 6 *A formula ϕ is a* satisfaction statement *if it is of the form $@_i\psi$ or $\neg @_i\psi$.*

The truth of a satisfaction statement in a model is not dependent on the point of evaluation. Therefore, instead of saying that a satisfaction statement ϕ is true (false) in a model $\mathfrak{M}$ at a point w we will simply say that $\mathfrak{M}$ satisfies (does not satisfy) ϕ. The same holds for any element of S since we explicitly required that they have an @-prefixed matrix. Therefore, we will also speak of a PUENF-sentence $\phi \in S$ being true or false in a model, without reference to a specific point.

We will prove completeness along standard lines, starting with the definition of a *Hintikka set*.

Definition 7 *Let S be any set of PUENF-sentences, all of which have an @-prefixed matrix. A set of satisfaction statements Σ is a* Hintikka set *with respect to S if it satisfies the following conditions.*

1. *For all formulas ϕ and nominals i, if $@_i\phi \in \Sigma$ then $\neg @_i\phi \notin \Sigma$*
2. *Σ is closed under the application of the tableau rules. That is, let R be any tableau expansion rule from Table 3 and let σ be any mapping of nominals occurring in the premises of R to nominals occurring in Σ. If the premises of R occur in Σ under the substitution σ, then one of the possible outcomes of R occurs in Σ under a substitution extending σ.*
3. *For every PUENF formula $\forall x_1 \ldots x_n \exists y_1 \ldots y_m \phi \in S$ and every sequence of nominals $i_1 \ldots i_n$ occurring in Σ, there is a sequence of nominals $e_1 \ldots e_m$ such that $\phi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow j_1, \ldots y_m \leftarrow j_m] \in \Sigma$.*

We will call a set of satisfaction statements Σ a Hintikka set if it is a Hintikka set with respect to some collection of PUENF sentences S. Given a Hintikka set Σ, we denote by $\text{NOM}(\Sigma)$ the set of all nominals occurring in Σ. Furthermore, for any nominal $i \in \text{NOM}(\sigma)$, by $|i|_\Sigma$ we denote the set of nominals true at i according to Σ, i.e., $|i|_\Sigma = \{j \in \text{NOM}(\Sigma) \mid @_i j \in \Sigma\}$.

Lemma 1 (Model induced by a Hintikka set) *Given any Hintikka set Σ, there is a unique model $\mathfrak{M}_\Sigma = (W, R, V)$ such that the following hold: (we will call $\mathfrak{M}_\Sigma$ the* model induced by Σ*)*

1. $W = \{|k|_\Sigma \mid k \in \text{NOM}(\Sigma)\}$
2. $(|i|_\Sigma, |k_1|_\Sigma, \ldots, |k_n|_\Sigma) \in R$ *iff* $@_i \triangle (k_1, \ldots, k_n) \in \Sigma$
3. $|i|_\Sigma \in V(p)$ *iff* $@_i p \in \Sigma$
4. $|i|_\Sigma \in V(j)$ *iff* $@_i j \in \Sigma$

PROOF: Uniqueness of the model satisfying these conditions is straightforward to prove. As for existence, we reason from the fact that Σ is closed under the three Equality tableau rules of Table 3. The two left-most Equality rules ensure that $|i|_\Sigma$ defines an equivalence class. The second Equality rule ensures that condition 3 and 4 are met: suppose that $@_i i' \in \Sigma$. Then it immediately follows that $@_i p \in \Sigma$ iff $@_{i'} p \in \Sigma$, and that $@_i j \in \Sigma$ iff $@_{i'} j \in \Sigma$. Finally, that condition 2 is met is guaranteed by the second Equality rule in combination with the third Equality rule: suppose $@_i i' \in \Sigma$ and $@_{k_m} k'_m \in \Sigma$ $(1 \leq m \leq n)$. Then it follows from closure under these two rules that $@_i \triangle (k_1, \ldots, k_n) \in \Sigma$ iff $@_{i'} \triangle (k'_1, \ldots, k'_n) \in \Sigma$. □

Lemma 2 (Truth lemma) *Let Σ be a Hintikka set and let $\mathfrak{M}_\Sigma$ be the model induced by Σ. Then for all nominals i and formulas ϕ, the following hold.*

1. *If $@_i\phi \in \Sigma$ then $\mathfrak{M}_\Sigma, |i|_\Sigma \models \phi$*
2. *If $\neg @_i\phi \in \Sigma$ then $\mathfrak{M}_\Sigma, |i|_\Sigma \not\models \phi$*

Proof: By induction. The base case is taken care of by conditions 3 and 4 of Lemma 1. For the induction step, we proceed as follows. Let ϕ be any complex formula. Then one of the following must be the case (for ease of notation, we drop the subscript Σ).

- ϕ is of the form $\neg\psi$.

  Suppose $@_i\phi \in \Sigma$. Then $@_i\neg\psi \in \Sigma$. By closure under the tableau rules for negation, $\neg @_i\psi \in \Sigma$. By induction hypothesis, $\mathfrak{M}, |i| \not\models \psi$, and therefore, $\mathfrak{M}, |i| \models \neg\psi$.

  Suppose $\neg @_i\phi \in \Sigma$. Then $\neg @_i\neg\psi \in \Sigma$. By closure under the tableau rules for negation, $@_i\psi \in \Sigma$. By induction hypothesis, $\mathfrak{M}, |i| \models \psi$, and therefore, $\mathfrak{M}, |i| \not\models \neg\psi$.

- ϕ is of the form $\psi_1 \wedge \psi_2$.

  Suppose $@_i\phi \in \Sigma$. Then $@_i\psi_1 \wedge \psi_2 \in \Sigma$. By closure under the tableau rules for conjunction, $@_i\psi_1 \in \Sigma$ and $@_i\psi_2 \in \Sigma$. By induction hypothesis, $\mathfrak{M}, |i| \models \psi_1$ and $\mathfrak{M}, |i| \models \psi_2$. Therefore, $\mathfrak{M}, |i| \models \psi_1 \wedge \psi_2$.

  Suppose $\neg @_i\phi \in \Sigma$. Then $\neg @_i\psi_1 \wedge \psi_2 \in \Sigma$. By closure under the tableau rules for negation, either $\neg @_i\psi_1 \in \Sigma$ or $\neg @_i\psi_2 \in \Sigma$. By induction hypothesis, either $\mathfrak{M}, |i| \not\models \psi_1$ or $\mathfrak{M}, |i| \not\models \psi_2$. Therefore, $\mathfrak{M}, |i| \not\models \psi_1 \wedge \psi_2$.

- ϕ is of the form $\triangle(\phi_1, \ldots, \phi_n)$.

  Suppose $@_i\phi \in \Sigma$. Then $@_i \triangle (\phi_1, \ldots, \phi_n) \in \Sigma$. By closure under the tableau rules for the modalities, there are nominals $k_1 \ldots k_n$ such that $@_i \triangle (k_1, \ldots, k_n) \in \Sigma$ and $@_{k_m}\phi_m \in \Sigma$ $(1 \leq m \leq n)$. By condition 2 of Lemma 1, $|i|_\Sigma, |k_1|_\Sigma, \ldots, |k_n|_\Sigma) \in R$. By the induction hypothesis, $\mathfrak{M}, |k_m| \models \phi_m$ $(1 \leq m \leq n)$. Therefore, $\mathfrak{M}, |i| \models \triangle(\phi_1, \ldots, \phi_n)$.

  Suppose $\neg @_i\phi \in \Sigma$. Then $\neg @_i \triangle (\phi_1, \ldots, \phi_n) \in \Sigma$. We need to show that $\mathfrak{M}, |i| \not\models \triangle(\phi_1, \ldots, \phi_n)$. Suppose $(|i|, |k_1|, \ldots, |k_n|) \in R$. By condition 2 of Lemma 1, $@_i \triangle (k_1, \ldots, k_n) \in \Sigma$. By closure under the tableau rules for the modalities, we know that $\neg @_{k_m}\phi_m \in \Sigma$ for some $m \leq n$. By induction hypothesis, we can infer that $\mathfrak{M}, |k_m| \not\models \phi_m$. Since we didn't make any assumptions on $k_1, \ldots, k_n$, we can conclude that $\mathfrak{M}, |i| \not\models \triangle(\phi_1, \ldots, \phi_n)$.

- ϕ is of the form $@_j\psi$.

  Suppose $@_i\phi \in \Sigma$. Then $@_i@_j\psi \in \Sigma$. By closure under the tableau rules for @, we can infer that $@_j\psi \in \Sigma$. By induction hypothesis, $\mathfrak{M}, |j| \models \psi$. Using the fact that the denotation of j in $\mathfrak{M}$ is $|j|$, we conclude that $\mathfrak{M}, |i| \models @_j\psi$

  Suppose $\neg @_i\phi \in \Sigma$. Then $\neg @_i@_j\psi \in \Sigma$. By closure under the tableau rules for @, we can infer that $\neg @_j\psi \in \Sigma$. By induction hypothesis, $\mathfrak{M}, |j| \not\models \psi$. Using the fact that the denotation of j in $\mathfrak{M}$ is $|j|$, we conclude that $\mathfrak{M}, |i| \not\models @_j\psi$

□

Lemma 3 (Frame lemma) *Suppose Σ is a Hintikka set with respect to a set of* PUENF*-sentences S, all of which have an @-prefixed matrix. Let $\mathfrak{M}_\Sigma$ be the model induced by Σ. Then $\mathfrak{M}_\Sigma \models \phi$ for all (strong hybrid) formulas $\phi \in S$.*

Proof: Suppose $\forall x_1 \ldots x_n \exists y_1 \ldots y_m @_z\psi \in S$. Let w be any world, and g any assignment. We need to show that $\mathfrak{M}_\Sigma, v, g \models @_z\phi$. That is, we need to show that for every sequence of points $w_1, \ldots, w_n$, there are points $v_1, \ldots, v_m$ such that $\mathfrak{M}_\Sigma, v, g[x_1/w_1, \ldots, x_n/w_n, y_1/v_1, \ldots, y_m/v_m] \models @_z\psi$.

Let $w_1, \ldots, w_n$ be any sequence of points. Since $\mathfrak{M}_\Sigma$ is the model induced by Σ, all worlds are in fact equivalence classes of nominals. Let $w_1 = |i_1|_\Sigma, \ldots, w_n = |i_n|_\Sigma$. By the third clause of Definition 7, we know that there are nominals $j_1, \ldots j_m$ such that $@_z\psi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow j_1, \ldots, y_m \leftarrow j_m] \in \Sigma$. By the Truth Lemma, it follows that $\mathfrak{M}_\Sigma \models @_z\psi[x_1 \leftarrow i_1, \ldots, x_n \leftarrow i_n, y_1 \leftarrow j_1, \ldots, y_m \leftarrow j_m]$. From this, together with the fact that every nominal is a member of its denotation, we can conclude that $\mathfrak{M}_\Sigma, [x_1 \leftarrow w_1, \ldots, x_n \leftarrow w_n, y_1 \leftarrow |j_1|_\Sigma, \ldots, y_m \leftarrow |j_n|_\Sigma] \models @_z\psi$. □

Theorem 11 (Completeness) *Let S be a set of* PUENF*-sentences, all of which have an @-prefixed matrix. The tableau system TS is complete with respect to the class of frames defined by the strong hybrid formula $\bigwedge_{\phi \in S} \phi$.*

Proof: Given a set of satisfaction statements Σ and a fully developed (possibly infinite) open TS tableau starting with Σ. Since the tableau is open, there must be a fully developed branch that doesn't contain an inconsistent pair of formulas. Let Γ be the set of all formulas occurring on this branch. Γ is a Hintikka set and $\Sigma \subseteq \Gamma$. Let $\mathfrak{M}_\Gamma$ be the model induced by Γ. By the Truth Lemma, $\mathfrak{M}_\Gamma, |i| \models \Sigma$ for all nominals i. By the Frame Lemma, $\mathfrak{M}_\Gamma$ satisfies all strong hybrid formulas of S. Therefore, Σ is satisfiable in a model that satisfies $\bigwedge_{\phi \in S} \phi$. □