January 1998

Hybrid Languages and Temporal Logic

Patrick Blackburn
Universität des Saarlandes, patrick@coli.uni-sb.de

Miroslava Tzakova
Max-Planck-Institut für Informatik, tzakova@mpi-sb.mpg.de

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-98-16.

Hybrid Languages and Temporal Logic

Abstract
Hybridization is a method invented by Arthur Prior for extending the expressive power of modal languages. Although developed in interesting ways by Robert Bull, and by the Sofia school (notably, George Gargov, Valentin Goranko, Solomon Passy and Tinko Tinchev) the method remains little known. In our view this has deprived temporal logic of a valuable tool.

The aim of the paper is to explain why hybridization is useful in temporal logic. We make two major points, the first technical, the second conceptual. First, we show that hybridization gives rise to well-behaved logics that exhibit an interesting synergy between modal and classical ideas. This synergy, obvious for hybrid languages with full first-order expressive strength, is demonstrated for a weaker local language capable of defining the *Until* operator, we provide a minimal axiomatization, and show that in a wide range of temporally interesting cases extended completeness results can be obtained automatically. Second, we argue that the idea of sorted atomic symbols which underpins the hybrid enterprise can be developed further. To illustrate this, we discuss the advantages and disadvantages of a simple hybrid language which can quantify over paths.

Comments
University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-98-16.

Hybrid Languages and Temporal Logic

Patrick Blackburn
Computerlinguistik
Universität des Saarlandes
66041 Saarbrücken
Germany
patrick@coli.uni-sb.de

Miroslava Tzakova
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
Germany
tzakova@mpi-sb.mpg.de

In memory of George Gargov

Abstract

Hybridization is a method invented by Arthur Prior for extending the expressive power of modal languages. Although developed in interesting ways by Robert Bull, and by the Sofia school (notably, George Gargov, Valentin Goranko, Solomon Passy and Tinko Tinchev), the method remains little known. In our view this has deprived temporal logic of a valuable tool.

The aim of the paper is to explain why hybridization is useful in temporal logic. We make two major points, the first technical, the second conceptual. First, we show that hybridization gives rise to well-behaved logics that exhibit an interesting synergy between modal and classical ideas. This synergy, obvious for hybrid languages with full first-order expressive strength, is demonstrated for a weaker local language capable of defining the *Until* operator; we provide a minimal axiomatization, and show that in a wide range of temporally interesting cases extended completeness results can be obtained automatically. Second, we argue that the idea of sorted atomic symbols which underpins the hybrid enterprise can be developed further. To illustrate this, we discuss the advantages and disadvantages of a simple hybrid language which can quantify over paths.

1 Introduction

Arthur Prior proposed using modal languages for temporal reasoning more than 40 years ago, and since then the approach has become widespread in a variety of disciplines. Over this period, a wide range of (often very powerful) modalities has been used to reason about time. This is unsurprising. After all, different choices of temporal ontology (such as instants, intervals, and events) are relevant for different purposes, and (depending on the application) considerable expressive power may be needed to cope with the way information can be distributed across such structures. But inventing new modalities is not the only way of boosting modal expressivity. There is a largely overlooked alternative

called *hybridization*, and this paper explores its relevance for temporal logic.[1]

Hybridization is best introduced by example. Consider the following sentence from the language we call ML + ∀:

$$\forall x(x \rightarrow \neg\Diamond x).$$

The x in this expression is a *state variable*, and all its occurrences are bound by the *binder* ∀. Syntactically, state variables are *formulas*: after all, the expression $x \rightarrow \neg\Diamond x$ is built using $\rightarrow$, $\neg$ and $\Diamond$ in the same way that $p \rightarrow \neg\Diamond p$ is. Semantically, however, state variables are best thought of as *terms*. Our semantics will stipulate that state variables are satisfied at exactly one state in any model. In effect, state variables act as names; they *label* the unique state they are true at.

The use of 'formulas as terms' gives hybrid languages their unique flavor: they are formalisms which blend the operator-based perspective of modal logic with the classical idea of explicitly binding variables to states. Unsurprisingly, this combination offers increased expressive power. The above sentence, for example, is true at any irreflexive state in any model, and false at all reflexive ones. No ordinary modal formula has this property.

Now, the language ML + ∀ is not the only hybrid language, and for many purposes it is not the most natural one. One of the key intuitions underlying modal semantics is *locality*, and it is intuitively clear (we shall be precise later) that ∀ is not local; as our notation suggests, ∀ quantifies across *all* states. So, if we want a local hybrid language, ML + ∀ is not a suitable choice.

But what are the alternatives? To the best of our knowledge only one has been considered, namely the binder we here call ↓. Now, ↓ does something simple and natural: it binds a variable to the *current* state. Unfortunately, while ML+↓ is a local language, it has two drawbacks. First, it is not expressive enough for many applications (for example, we shall show that it is not strong

[1] The literature on hybrid languages consists of a handful of papers published over the last thirty years by researchers with very different interests. Confining ourselves to the main line of development, the idea can be traced back Prior (1967), and the posthumously published Prior and Fine (1977) contains some of Prior's unfinished papers on the subject together with an appendix by Kit Fine. Prior's concerns were largely philosophical; technical development seems to have started with Bull (1970). Bull investigated a hybrid temporal language containing the ∀ binder and the universal modality A, and introduced the idea of quantification over paths. In addition, he initiated the algebraic study of such systems. The paper never attracted the attention it deserved; in fact, apart from citations in the hybrid literature, the only mention we know of is from Burgess's survey of tense logic:

> *Other hybrids of a different sort — not easy to describe briefly — are treated in an interesting paper of Bull [1970].* (Burgess (1984, page 128)).

(This is probably the first use of 'hybrid' in connection with such languages.) The idea was independently invented by the Sofia School as a spin-off of their investigation of modal logic with names. The best guide to the Bulgarian tradition is the beautiful and ambitious Passy and Tinchev (1991), drafts of which were in circulation in the late 1980s. Hybridization is discussed in Chapter III and deals with Propositional Dynamic Logic enriched with both ∀ and the universal modality; see also Passy and Tinchev (1985) and the brief remarks at the end of Gargov, Passy and Tinchev (1987).

Recent papers on the subject include Goranko (1994) (probably the first published account of hybrid languages containing the ↓ binder), Blackburn and Seligman (1995), and Seligman (1997) (which investigates hybrid natural deduction and sequent calculi for applications in Situation Theory), and Blackburn and Tzakova (1998,1998a,1998b). Also relevant are Gargov and Goranko (1993), Blackburn (1993,1994); these look at modal and tense logics enriched with nominals (in effect, the free variable fragments of hybrid languages).

enough to define the *Until* operator). Second, in stark contrast to ML$+\forall$ which has an elegant axiomatization, axiomatizing ML$+\downarrow$ seems to require complex proof rules.

What are we to do? Here we show that introducing an operator @ which retrieves the value stored by $\downarrow$ solves these problems: it offers the expressivity we need, the minimal logic is elegant, and we automatically get completeness results for a wide class of interesting frame classes, many of which are not modally definable. All this without sacrificing locality.[2]

These results are the *technical* core of the paper, but to close our discussion we change gears — there is an important *conceptual* point to be made about hybridization and its relevance to temporal logic: hybridization is *not* simply about quantifying over states. Rather, *hybridization is about handling different types of information in a uniform way.* We illustrate this idea by discussing a simple hybrid language for quantifying over paths.

But we are jumping ahead. There is much to be done before we can usefully discuss such ideas, so let's call a halt to our introductory remarks and start developing the idea of hybridization systematically.

2 The basic modal language

One of the simplest languages for temporal reasoning is the propositional modal language that contains just two modalities: an operator $\Box$ (read as: *at all future states*) together with its dual operator $\Diamond$ (read as: *at some future state*). For most of this paper we will be working with various hybrid extension of this simple language (which we will call ML). The purpose of the present section is to fix notation and terminology, to remind the reader of various standard concepts (in particular, *generated submodels* and *bisimulations*), and to present a wish-list of properties for hybrid temporal languages.

Given a (countable) set of *propositional symbols* PROP $= \{p, q, r, \ldots\}$ the well-formed formulas of ML are defined as follows:

$$\text{WFF} := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box\varphi.$$

Other Boolean operators ($\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$, $\top$, and so on) are defined in the usual way, and we define $\Diamond\varphi$ to be $\neg\Box\neg\varphi$.

ML is interpreted on *models*. A model $\mathcal{M}$ is a triple (S, R, V) such that S is a non-empty set of *states*, and R is a binary relation on S (the *temporal precedence relation*); the pair (S, R) is called the *frame* underlying $\mathcal{M}$. The *valuation* V is a function with domain PROP and range $Pow(S)$; this tells us at which states (if any) each propositional symbol is true. Depending on the application, additional properties may be demanded of R: in temporal logic (various combinations of) such properties as transitivity, irreflexivity, density, discreteness, trichotomy, no-branching-to-the-right, and many others, are common. We shall deal with such demands later.

[2] Blackburn and Tzakova (1998a), an extended version of the present paper, examines two other local solutions in detail: (1) adding $\Downarrow^1$, a universal quantifier over *accessible* states, and (2) changing the underlying language from modal logic to tense logic. This version will be made available at `http://www.coli.uni-sb.de/~patrick/`. An earlier version which contains solution (1) is already available.

The satisfaction definition for ML is defined as follows. Let $\mathcal{M} = (S, R, V)$ and $s \in S$. Then:

$$\begin{array}{lll} \mathcal{M}, s \models p & \text{iff} & s \in V(p), \text{ where } p \in \text{PROP} \\ \mathcal{M}, s \models \neg\varphi & \text{iff} & \mathcal{M}, s \not\models \varphi \\ \mathcal{M}, s \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, s \models \varphi \ \& \ \mathcal{M}, s \models \psi \\ \mathcal{M}, s \models \Box\varphi & \text{iff} & \forall s'(sRs' \ \Rightarrow \ \mathcal{M}, s' \models \varphi). \end{array}$$

If $\mathcal{M}, s \models \varphi$ we say that φ is *satisfied* in $\mathcal{M}$ at s. If φ is true at all states in a model $\mathcal{M}$ we say it is *valid* in $\mathcal{M}$ and write $\mathcal{M} \models \varphi$.

Note the locality of the satisfaction definition: formulas are evaluated *inside* models at some particular state (called the *current state*), and the $\Box$ and $\Diamond$ operators scan the states accessible from the current state via the precedence relation R. This locality intuition is arguably *the* central intuition underlying modal approaches to temporal logic; it is certainly the intuition which prompted Arthur Prior to pioneer the "modal logic of time" (which he called *tense logic*). As he observed, we are situated *inside* the temporal flow, and many aspects of language (for example, the use of tense, and temporal indexicals such as *now*) reflect this internal perspective. Accordingly, he believed that modal analyses of temporal logic were likely to be most revealing.[3]

The locality of ML has an obvious mathematical consequence: satisfaction of ML formulas is *preserved under the formation of generated submodels*. To be more precise, given a model $\mathcal{M} = (S, R, V)$ and a state s of S, the submodel of $\mathcal{M}$ that is generated by s contains just those states of $\mathcal{M}$ that are accessible from s by a finite number of transitions along R. It follows by an easy induction that for all formulas φ:

$$\mathcal{M}, s \models \varphi \ \text{ iff } \ \mathcal{M}^s, s \models \varphi.$$

In what follows, we use preservation under generated submodels as a key criterion for judging hybrid *temporal* languages. We are interested in *local* temporal languages, and will reject hybrid extensions which lead to a loss of the generated submodel preservation results.

Now for a key question: does ML have the expressivity needed for temporal reasoning? There is no absolute answer: it depends on the application. For some applications, ML will often be too *strong*. For example, if one is interested in using modal languages to characterize various types of bisimulation invariance, it may be necessary to work with sublanguages of ML containing no propositional symbols (wffs would be built using the constant $\bot$) or to shed some Boolean expressivity.

But for many other applications, ML is too *weak*.[4] For a start, as has already been mentioned, no formula of ML is capable of distinguishing irreflexive from reflexive states in all models; this means that a fundamental constraint on temporal precedence simply isn't reflected. Moreover, consider the definition of

[3] The best introduction to Prior's views is Prior (1967).

[4] A very obvious weakness is that ML offers us no way of looking *backwards* along R; for that we need Prior's language of tense logic. However, while useful in natural language semantics, in many applications in AI and theoretical computer science, backward looking operators don't play a prominent role. Apart from occasional remarks we won't discuss tense logic here, but Blackburn and Tzakova (1998a), the extended version of the present paper, contains a full treatment.

the *Until* operator:

$$\mathcal{M}, s \models \mathit{Until}(\varphi, \psi) \quad \text{iff} \quad \exists s'(sRs' \ \& \ \mathcal{M}, s' \models \varphi \ \& \ \forall t(sRt \ \& \ tRs' \ \Rightarrow \ \mathcal{M}, t \models \psi))$$

This is an extremely natural *local* operator (note that formulas built using *Until* are preserved under the formation of generated submodels) and has proved a useful tool for temporal reasoning in computer science (indeed, computer scientists usually regard *Until* as the fundamental modality). However the *Until* operator is *not* definable in ML. As the non-definability of both *Until* and irreflexivity follows from the fact that ML formulas are preserved under *bisimulations*, and as we will later make use of special bisimulations called *quasi-injective bisimulations*, it will be useful to prove these non-definability results here.

A *bisimulation* between two models $\mathcal{M}_1 = (S_1, R_1, V_1)$ and $\mathcal{M}_2 = (S_2, R_2, V_2)$ is a non-empty binary relation Z between S_1 and S_2 such that:

1. For all states s_1 in S_1 and s_2 in S_2, if $s_1 Z s_2$ then s_1 and s_2 satisfy the same propositional symbols.

2. For all states s_1, $s_1{}'$ in S_1 and s_2 in S_2, if $s_1 R_1 s_1{}'$ and $s_1 Z s_2$ then there is a state $s_2{}'$ in S_2 such that $s_2 R_2 s_2{}'$ and $s_1{}' Z s_2{}'$.

3. For all states s_2, $s_2{}'$ in S_2, and s_1 in S_1, if $s_2 R_2 s_2{}'$ and $s_1 Z s_2$ then there is a state $s_1{}'$ in S_1 such that $s_1 R_1 s_1{}'$ and $s_1{}' Z s_2{}'$.

The fundamental result concerning bisimulations (which follows straightforwardly by induction on the structure of ML formulas) is that if Z is a bisimulation between models $\mathcal{M}_1$ and $\mathcal{M}_2$ and $s_1 Z s_2$ then s_1 and s_2 satisfy exactly the same ML formulas.

It follows that neither *Until* nor irreflexivity is definable — indeed the following counterexample (which we believe is due to Johan van Benthem) establishes both points simultaneously. Let $\mathcal{M}_1$ be an irreflexive model containing just two states s_1 and s_2, let $s_1 R s_2$ and $s_2 R s_1$, and suppose all propositional symbols are true at both states. Let $\mathcal{M}_2$ be an reflexive model containing just one state s, and suppose all propositional symbols are true at s. Clearly the relation Z which links both s_1 and s_2 to s and vice-versa is a bisimulation, hence all states in both models satisfy exactly the same ML formulas. So, as $\mathcal{M}_1$ is irreflexive and $\mathcal{M}_2$ reflexive, it follows that no ML formula succeeds in distinguishing irreflexive and reflexive states. Moreover, observe that $\mathit{Until}(\top, \bot)$ is false in $\mathcal{M}_1$ (at both s_1 and s_2) but true in $\mathcal{M}_2$. It follows that the *Until* operator cannot be expressed in ML.

Thus, ML has expressive weaknesses that are relevant to temporal reasoning, and one of the key goals of this paper will be to repair them by hybridization. But what should a hybrid temporal language look like? It is time to draw up a wish-list.

First, we would like our hybrid language to be *local*. Second, we would like our hybrid language to be expressive enough to detect irreflexivity and define *Until*. Third, we would like to find hybrid languages in which the central ideas of modal and classical proof systems can be clearly combined. Indeed, we would like to exhibit a synergy between modal and classical ideas; we want the whole, so to speak, to offer more than the sum of its parts. Let's now examine the two hybrid binders that have previously been studied and see how they measure up against these demands.

3 Two hybrid binders

Syntactically, hybridizing ML involves making two changes. First, we *sort* the atomic symbols; instead of having just one kind of atom (namely the symbols in PROP) we add a second sort called *state symbols*. For reasons we shall soon explain, it is convenient to divide state symbols into two subcategories: *state variables* and *nominals*. Second, we add *binders*. The binders will be used to bind state variables, but not nominals or propositional symbols.

Let PROP be as described before. Assume we have denumerably infinite set SVAR of state variables (whose elements we typically write as u, v, w, x, y and z), and a denumerably infinite set NOM of nominals (whose elements we typically write as i, j, k and l). We assume that PROP, SVAR and NOM are pairwise disjoint. We call SVAR $\cup$ NOM the set of *state symbols*, and PROP $\cup$ SVAR $\cup$ NOM the set of *atoms*. Choose B to be one of $\forall$ or $\downarrow$. We build the well-formed formulas of the hybrid language (over PROP, SVAR, NOM, and B) as follows:

$$\text{WFF } \varphi \ := \ a \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box\varphi \mid Bx\varphi.$$

Here $a \in$ ATOM, and $x \in$ SVAR. If B was chosen to be $\forall$, we obtain the language ML $+ \forall$, and if B was $\downarrow$ we get ML $+ \downarrow$. (Strictly speaking, different choices of PROP, SVAR and NOM give rise to different languages, but we ignore this whenever possible.)

A full discussion of the syntax of these languages would need to define such concepts as 'free', 'bound', 'substitutable for', and so on. But experience with classical logic is a reliable guide, and anyway the relevant definitions may be found in Blackburn and Tzakova (1998), so we'll simply remark that a *sentence* is a formula containing no free variables or nominals, and that we use the notation $\varphi[s/v]$ to denote the formula obtained by substituting the state symbol s for all free occurrences of the state variable v in φ.

As promised in the introduction, our hybrid languages use formulas as labels: in the semantics presented below, both state variables and nominals will be satisfied at exactly one state in any model. Now, the role of the state variables should be clear; but what is the point of having nominals? Simply this: it is convenient to have a supply of labels that cannot be bound by the binders; this simplifies some of the technicalities, for it saves us having to worry about accidental binding. In short, nominals are reminiscent of the 'parameters' used in classical proof theory.

Now for the semantics. The key idea is straightforward: we are going to insist that state symbols are interpreted by singleton subsets of models. We'll also need a smooth way to handle the fact that state variables may become bound, whereas this is not possible for nominals or propositional symbols. But there is an obvious way to do this: we'll let the state variables be handled by a separate assignment function in the manner familiar from classical logic.

Definition 1 (Standard models and assignments) *Let $\mathcal{L}$ be a hybrid language over* PROP, SVAR *and* NOM. *A model $\mathcal{M}$ for $\mathcal{L}$ is a triple (S, R, V) such that S is a non-empty set, R a binary relation on S, and V : PROP $\cup$ NOM $\longrightarrow$ $Pow(S)$. A model is called* standard *iff for all nominals $i \in$ NOM, $V(i)$ is a singleton subset of S.*

An assignment for $\mathcal{L}$ on $\mathcal{M}$ is a mapping g : SVAR $\longrightarrow$ $Pow(S)$. An assignment is called standard *iff for all state variables $x \in$ SVAR, $g(x)$ is a singleton*

subset of S. The notation $g' \stackrel{x}{\sim} g$ (g' is an x-variant of g) means that g' and g are standard assignments (on some model $\mathcal{M}$) such that g' agrees with g on all arguments save possibly x.

Let $\mathcal{M} = (S, R, V)$ be a standard model, and g a standard assignment. For any atom a, let $[V, g](a) = g(a)$ if a is a state variable, and $V(a)$ otherwise. Then interpretation of our hybrid languages is carried out using the following definition:

$$
\begin{array}{lll}
\mathcal{M}, g, s \models a & \text{iff} & s \in [V, g](a), \text{ where } a \in \text{ATOM} \\
\mathcal{M}, g, s \models \neg\varphi & \text{iff} & M, g, s \not\models \varphi \\
\mathcal{M}, g, s \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, g, s \models \varphi \ \& \ \mathcal{M}, g, s \models \psi \\
\mathcal{M}, g, s \models \Box\varphi & \text{iff} & \forall s'(sRs' \ \Rightarrow \ \mathcal{M}, g, s' \models \varphi). \\
\mathcal{M}, g, s \models \forall x\varphi & \text{iff} & \forall g'(g' \stackrel{x}{\sim} g \ \Rightarrow \ \mathcal{M}, g', s \models \varphi) \\
\mathcal{M}, g, s \models \downarrow x\varphi & \text{iff} & \mathcal{M}, g', s \models \varphi, \text{ where } g' \stackrel{x}{\sim} g \text{ and } g'(x) = \{s\}
\end{array}
$$

Let $\mathcal{M}$ be a standard model. We say that φ is *valid* on $\mathcal{M}$ iff for all standard assignments g on $\mathcal{M}$, and all states s in $\mathcal{M}$, $\mathcal{M}, g, s \models \varphi$, and if this is the case we write $\mathcal{M} \models \varphi$. We say that a formula φ is valid on a frame (S, R) (written $(S, R) \models \varphi$) iff for all standard valuations V and standard assignments g on (S, R), and all $s \in S$, $(S, R, V), g, s \models \varphi$.

Lemma 2 (Substitution lemma) *Let $\mathcal{M}$ be a standard model, let g be an assignment on $\mathcal{M}$, and let φ be a formula of any of the hybrid languages defined above. Then, for every state s in $\mathcal{M}$, if y is a variable that is substitutable for x in φ and i is a nominal then:*

1. *$\mathcal{M}, g, s \models \varphi[y/x]$ iff $\mathcal{M}, g', s \models \varphi$, where $g' \stackrel{x}{\sim} g$ and $g'(x) = g(y)$.*
2. *$\mathcal{M}, g, s \models \varphi[i/x]$ iff $\mathcal{M}, g', s \models \varphi$, where $g' \stackrel{x}{\sim} g$ and $g'(x) = V(i)$.*

Proof. By induction on the complexity of φ. $\dashv$

This concludes the preliminaries; it's time to take a closer look at the binders.

The $\forall$ binder

The $\forall$ binder is the stronger, more classical, of our binders: indeed it's just the familiar universal quantifier in a modal setting. Note that if we define $\exists x\varphi$ to be the dual binder $\neg\forall x\neg\varphi$, then:

$$\mathcal{M}, g, s \models \exists x\varphi \text{ iff } \ \exists g'(g' \stackrel{x}{\sim} g \ \& \ \mathcal{M}, g', s \models \varphi).$$

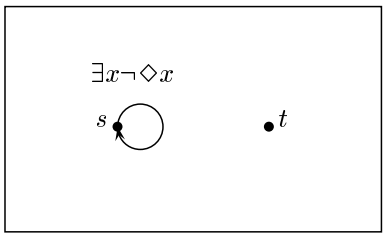
ML + $\forall$ is a powerful language. We saw in the introduction that it can distinguish irreflexive from reflexive states. Moreover it can define the *Until* operator:

$$\mathit{Until}(\varphi, \psi) \ := \ \exists y(\Diamond(y \wedge \varphi) \wedge \Box(\Diamond y \rightarrow \psi)).$$

This definition says: it is possible to bind the variable y to a successor state in such a way that (1) φ holds at the state labeled y, and (2) ψ holds at all successors of the current state that precede this y-labeled state. In addition,

the minimal temporal logic of ML + ∀ has a simple axiomatization that can be proved complete reasonably straightforwardly. All in all, it's a lovely language.

But there's a snag: it isn't local. To see that satisfaction of ML+∀ sentences need not be preserved under the formation of generated submodels, consider the following counterexample (taken from Blackburn and Seligman (1995)). Let $\mathcal{M}$ be the following two-element model where $S = \{s, t\}$, and $R = \{(s, s)\}$:



Then $\exists x \neg \Diamond x$ is true at s in $\mathcal{M}$, for we can assign the state t to x and $(s, t) \notin R$. However it is *not* true at s in the submodel $\mathcal{M}^s$ generated by s, for as $\mathcal{M}^s$ contains only the state s, all assignments assign s to x. As s is reflexive, $\neg \Diamond x$ will always be false. In short, $\exists$ detects t from s, even though t and s are completely disconnected.

If you want a strong hybrid language and are not interested in maintaining locality, then ML + ∀ is probably an excellent choice. Indeed, you may wish to consider working with a hybrid language even *less* local, namely ML + ∀ enriched with the *universal modality* A.[5] The universal modality has the following satisfaction definition: $\mathcal{M}, s \models A\varphi$ iff $\mathcal{M}, s' \models \varphi$ for all states $s' \in \mathcal{M}$. It is not hard to see that adding the universal modality yields a hybrid language with first-order expressive power (Prior knew this result, and formulated it in a number of ways). Moreover, the A and ∀ work together extremely smoothly, making elegant axiomatizations possible (see Bull (1970)). But while such rich systems are interesting, they are far removed from the local temporal languages we wish to develop.

The ↓ binder

If one is interested in local hybrid languages, the ↓ binder is the most natural starting point. Quite simply, ↓ binds a variable to the current state; it creates a label for the here-and-now. Let's look at it more closely.[6]

First, note that ↓ is self-dual; that is, at any state, in any standard model, under any standard assignment, $\downarrow x\varphi$ is satisfied if and only if $\neg\downarrow x\neg\varphi$ is satisfied too. To put it another way, we are free to regard ↓ as either a "universal quantifier over the current state" or as an "existential quantifier over the current state"; as there is exactly one current state, these amount to the same thing.

Next, note that $\downarrow x\varphi$ is definable in ML+∀; we can define it either as $\forall x(x \rightarrow \varphi)$ or $\exists x(x \wedge \varphi)$, thus ML + ↓ is a fragment of ML + ∀. It's quite an interesting

[5] Virtually the entire literature on hybrid languages is devoted to such systems. For example, both Bull (1970) and Passy and Tinchev (1991) make use of ∀ and A.

[6] Incidentally, while ↓ is a relative newcomer to hybrid languages (Goranko (1994) seems to be the first published account) essentially the same binder has been introduced to a number of different non-hybrid languages for a wide variety of purposes; see for example Richards *et al* (1989), Cresswell (1990), and Sellink (1994). Labeling the here-and-now seems to be an important operation.

fragment. For a start, sentences of ML$+\downarrow$ *are* preserved under the formation of generated submodels. (We leave the simple proof to the reader. Essentially it boils down to the observation that the only states that $\downarrow$ can bind to variables in the course of evaluation must be states in the generated submodel. For example, in the previous diagram, if we evaluate a sentence at s, the only state that we can bind to any variable is s itself; ML$+\downarrow$ cannot detect t, which is what we want.) Moreover, adding the $\downarrow$ binder boosts the expressive power of ML in temporally interesting ways. In particular, note that the sentence

$$\downarrow x \Box \neg x$$

is true in a model at a state s iff s is irreflexive.

Unfortunately, ML $+\downarrow$ has two drawbacks. First, there is no obvious way to provide a complete axiomatization without resorting to a fairly complex rule of proof.[7] Second, for many purposes it simply isn't expressive enough. Let's examine this second problem more closely.

Although adding $\downarrow$ increases the expressive power, *Until* still isn't definable. To see why, we make use of the *quasi-injective bisimulations* introduced in Blackburn and Seligman (1997). Let us say that states s and s' in a model $\mathcal{M} = (S, R, V)$ are *mutually inaccessible* iff s is not in the submodel generated by s' and s' is not in the submodel generated by s. We then define:

Definition 3 (Quasi-injective bisimulations) *Let Z be a bisimulation between $\mathcal{M}_1$ and $\mathcal{M}_2$; Z is a* quasi-injective bisimulation *iff:*

1. *For all states s_1, $s_1{}'$ in $\mathcal{M}_1$, and s_2 in $\mathcal{M}_2$, if $s_1 Z s_2$ and $s_1{}' Z s_2$, and $s_1 \neq s_1{}'$ then s_1 and $s_1{}'$ are mutually inaccessible, and*

2. *For all states s_2, $s_2{}'$ in $\mathcal{M}_2$, and s_1 in $\mathcal{M}_1$, if $s_1 Z s_2$ and $s_1 Z s_2{}'$, and $s_2 \neq s_2{}'$ then s_2 and $s_2{}'$ are mutually inaccessible.*

Now, ML $+\downarrow$ sentences need not be preserved under arbitrary bisimulations (the fact that $\downarrow x \Box \neg x$ picks out irreflexive states shows this), but Blackburn and Seligman show that they *are* preserved under quasi-injective bisimulations. That is:

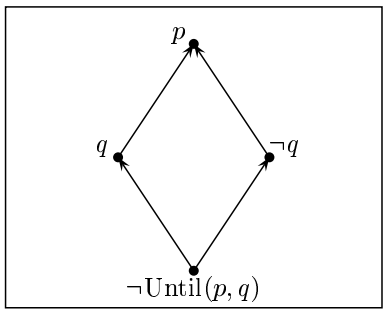
Proposition 1 *Let Z be a quasi-injective bisimulation between models $\mathcal{M}_1$ and $\mathcal{M}_2$, and let s_1 and s_2 be states in $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively such that $s_1 Z s_2$. Then for all sentences of* ML $+\downarrow$, *$\mathcal{M}_1, s_1 \models \varphi$ iff $\mathcal{M}_2, s_2 \models \varphi$.*

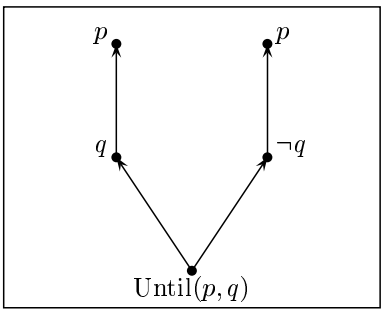
We can use this result to show that no sentence of ML $+\downarrow$ defines the *Until* operator. To be more specific, let p and q be propositional symbols. Then, even over strictly partially ordered models, there is no sentence $\varphi^{U(p,q)}$ of ML $+\downarrow$

[7] Blackburn and Tzakova (1997) axiomatize the set of valid ML $+\downarrow$ by making use of the *COV* rule (see Gargov, Passy and Tinchev (1987), Gargov and Goranko (1993)). Unfortunately, the *COV* rule is rather complex: it employs arbitrarily deep nestings of modalities.

The only other work on axiomatic systems for $\downarrow$, we know of are Goranko (1994) and Goranko (1996a). However Goranko's investigations have little bearing on the concerns of the present paper, for Goranko investigates a language containing both the universal modality and $\downarrow$. Note that the $\forall$ binder is definable in this language by $\forall x \varphi := \downarrow y A \downarrow x A(y \rightarrow \varphi)$, thus Goranko's language has full first-order expressive power.

that is satisfied in a model $\mathcal{M}$ at a state s iff *Until*(p, q) is satisfied in $\mathcal{M}$ at s. To see this consider the following two models:





(In both models, the relation we are interested in is the transitive closure of the relation indicated by the arrows, thus both models are strict partial orders.) Note that *Until*(p, q) is false in the left-hand model at the root node, and true in the right-hand model at the root. Hence if some sentence $\varphi^{U(p,q)}$ of ML $+ \downarrow$ expressed *Until*(p, q), it would be false at the root of the left-hand model, and true at the root of the right-hand side one. But this is impossible, for the obvious 'unraveling' relation between the two models is a quasi-injective bisimulation.

Summing up, previously studied hybrid systems don't meet our three wish-list criteria. The $\forall$ binder is interesting and elegant — but to adopt it is to abandon locality. The $\downarrow$ binder is far more promising — binding to the current state is such an intrinsically modal idea that it deserves further attention. But can we overcome its expressive weakness? And are there natural ways to avoid dependence on complex rules of proof? The answer is "Yes". As we shall now show, we can do this by adding a retrieval operator @ to match the action of $\downarrow$; for two further solutions, consult the extended version of this paper.

4 The @ operator

Suppose we were given a brand new web-browser to test, and we discovered it had the following limitation: although it allowed us to bookmark URLs, it *didn't* allow us to jump to these locations by clicking on the stored bookmark. Frankly, we wouldn't dream of working with such a browser; we'd demand that this shortcoming be fixed right away.

ML+$\downarrow$ is rather like this (hopefully non-existent) browser: $\Diamond$ pushes us through cyberspace, and $\downarrow$ allows us to label the states we visit on our travels — but ML+$\downarrow$ doesn't offer us a general mechanism for jumping to the states we label. Let's put this right. We shall allow ourselves to construct formulas of the form $@_s\varphi$. To evaluate such a formula we will jump to the state s labels and see whether φ holds there; in effect, @ will enable us to use the values $\downarrow$ has so carefully stored for us.

Let's make this precise. If s is a state symbol and φ is a formula then $@_s\varphi$ is a formula. It is possible to think of @ as a binary modality whose first argument is a state symbol and whose second argument is a formula — but as will soon become clear, it is more natural to view the composite symbol $@_s$ as a unary modal operator. If we add all these state-symbol-indexed unary modalities to ML+$\downarrow$, we obtain ML+$\downarrow$+@. Most syntactic aspects of ML+$\downarrow$+@ are obvious,

though the following point is worth stressing: @ does *not* bind variables. Only the $\downarrow$ binder does that.

Now for the semantics. Let $\mathcal{M} = (S, R, V)$ be a standard model, let g be a standard assignment on $\mathcal{M}$, and let $\mathsf{Den}(s)$ be the denotation of the state symbol s (that is, $\mathsf{Den}(s)$ is $g(s)$ if s is a state variable, and $V(s)$ if s is a nominal). Then:

$$\mathcal{M}, g, t \models @_s\varphi \text{ iff } \mathcal{M}, g, \mathsf{Den}(s) \models \varphi.$$

As promised, $@_s$ jumps to the denotation of s and evaluates its argument there.

Sentences of ML+$\downarrow$+@ are preserved under generated submodels. After all, in a *sentence*, the only occurrences of @ will be of the form $@_y$, where y is a state variable bound by some occurrence of $\downarrow$, and as $\downarrow$ binds locally, the result follows. Second, @ can define *Until*.[8] As we have already seen, *Until* is not definable in ML+$\downarrow$, but it certainly *is* in ML+$\downarrow$+@:

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

Note how this works: we label the current state with x, use $\Diamond$ to move to an accessible state, which we label y, and then use @ to jump us back to x. We then use the modalities to insist that (1) φ holds at the state labeled y, and (2) ψ holds at all successors of the current state that precede this y-labeled state. Note the similarities (and differences) with our earlier $\exists$-based definition of *Until*.[9]

As this example shows, $\downarrow$ and @ make a great team; they communicate smoothly and their cooperation gives rise to an axiomatization called $\mathcal{H}[\downarrow, @](K)$. This axiomatization is an extension of the minimal modal logic K. Recall that K is the smallest set of formulas containing all propositional tautologies, and all instances of $\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$, that is closed under *modus ponens* (if φ and $\varphi \rightarrow \psi$ are both provable, then so is ψ) and *necessitation* (if φ is provable then so is $\Box\varphi$). To the axioms and rules of proof of K we add axioms and rules governing both $\downarrow$ and @. Let's deal with $\downarrow$ first. First, we have all instances of the following schemas:[10]

[8] A lot more could be said about @, and we can't say it all here. But two things should be said. First, the reader has almost certainly seen something like @ in non-hybrid languages: for example it's Prior's $T(s, \varphi)$ construct in *third grade tense logic*, it's the Holds(s, φ) operator introduced by Allen (1984) for temporal representation in AI, and it is the characteristic operator of the *Topological Logic* of Rescher and Urquhart (1971). Note that the @ operator supports a variety of natural interpretations: for example, computationally it can be viewed as a `goto` instruction.

But one perspective is particularly relevant here: @ *can be viewed as a restricted version of the universal modality*. First, note that $@_s\varphi$ can be defined as either $A(s \rightarrow \varphi)$ or $E(s \wedge \varphi)$, where E is the dual if A. In short, @ allows *limited* access to the power of A, and the limitation results in a generated submodel for sentences. But as we shall see below, @ has enough power to support elegant proof theories.

[9] Note that the prenex block $\downarrow x \Diamond \downarrow y @_x$ defines an existential quantifier over states reachable in 1 R-step: $\downarrow^1_y \varphi := \downarrow x \Diamond \downarrow y @_x \varphi$; this binder is discussed in detail in the extended version of the paper. Similarly, we can define an existential quantifier over states accessible in 2 R-steps: $\downarrow^2_y \varphi := \downarrow x \Diamond \Diamond \downarrow y @_x \varphi$. Indeed, for any natural number n we can define an existential quantifier over states accessible in n R-steps. Note that we also have simple definitions of the universal quantifiers over states reachable in n R-steps: for example, $\Downarrow^2_y \varphi := \downarrow x \Box\Box \downarrow y @_x \varphi$. It is easy to see that $\downarrow^n$ and $\Downarrow^n$ are dual binders, for any natural number n.

[10] These axioms were used as part of the *COV*-based axiomatization of Blackburn and Tzakova (1998). In Blackburn and Tzakova (1998a), the extended version of the present paper, these axioms are discussed further, and analogs of *Q1–Q3* are given for the $\Downarrow^1$ binder mentioned in the previous footnote.

Q1	$\downarrow \mathbf{v}(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \downarrow \mathbf{v}\psi)$
Q2	$\downarrow \mathbf{v}\varphi \rightarrow (\mathbf{s} \rightarrow \varphi[\mathbf{s}/\mathbf{v}])$
Q3	$\downarrow \mathbf{v}(\mathbf{v} \rightarrow \varphi) \rightarrow \downarrow \mathbf{v}\varphi$
Self-Dual	$\downarrow \mathbf{v}\varphi \leftrightarrow \neg \downarrow \mathbf{v} \neg \varphi$

(Here $\mathbf{v}$ is a metavariable over state variables, $\mathbf{s}$ a metavariables over state symbols, and φ and ψ metavariables over arbitrary wffs. In *Q1*, φ cannot contain free occurrences of $\mathbf{v}$; in *Q2*, $\mathbf{s}$ must be substitutable for $\mathbf{v}$ in φ.)

Q1 and *Q2* are obvious analogs of familiar first-order axiom schemas. The major difference is that the present version of *Q2* only lets us substitute state symbols for binders when the obvious locality condition is fulfilled: $\mathbf{s}$ must be true in the current state. This restriction motivates the introduction of *Q3*, which allows us to eliminate bound occurrences of state variables in antecedent position. In addition to these axioms we have the rule of *state variable localization*; that is, if φ is provable then so $\downarrow x\varphi$. Summing up: $\downarrow$ supports a *local* form of classical reasoning. But in spite of the locality restriction, the axioms just introduced are strong enough to support many classical principles such as α-conversion. As an illustration (for full details, see the extended version) we show:

Lemma 4 (Normality) *For all formulas φ and ψ we have: $\vdash \downarrow x(\varphi \rightarrow \psi) \rightarrow (\downarrow x\varphi \rightarrow \downarrow x\psi)$.*

Proof. Note that $\downarrow x(\varphi \rightarrow \psi) \rightarrow (x \rightarrow (\varphi \rightarrow \psi))$ is an instance of *Q2*, as is $\downarrow x\varphi \rightarrow (x \rightarrow \varphi)$. Hence $\vdash (\downarrow x(\varphi \rightarrow \psi) \wedge \downarrow x\varphi)) \rightarrow (x \rightarrow \psi)$. Use localization to prefix this formula with $\downarrow x$, and then *Q1* to distribute $\downarrow x$ over the main implication to get $\vdash (\downarrow x(\varphi \rightarrow \psi) \wedge \downarrow x\varphi) \rightarrow \downarrow x(x \rightarrow \psi)$. Note that $\downarrow x(x \rightarrow \psi) \rightarrow \downarrow x\psi$ is an instance of *Q3*, so we can simplify the consequent and so obtain the result. (Using *Q3* in this way to simplify the conditionals produced by applications of *Q2* is typical of $\mathcal{H}[\downarrow, @](K)$ proofs.) $\dashv$

Let's turn to @. For every state symbol s, we have the rule of $@_s$-*necessitation* (if φ is provable then so is $@_s\varphi$). In addition we have the rules *Paste-0* and *Paste-1*; these will be introduced below. In addition, we have all instances of the following schemas. These fall naturally into three groups. The first identifies the basic logic of @.

K	$@_{\mathbf{s}}(\varphi \rightarrow \psi) \rightarrow (@_{\mathbf{s}}\varphi \rightarrow @_{\mathbf{s}}\psi)$
Self-Dual	$@_{\mathbf{s}}\varphi \leftrightarrow \neg @_{\mathbf{s}} \neg \varphi$
Introduction	$\mathbf{s} \wedge \varphi \rightarrow @_{\mathbf{s}}\varphi$

Note that *K* is simply the familiar modal distribution schema; hence as we have the rule of $@_s$-necessitation, $@_s$ is a normal modal operator. Obviously *Self-Dual* states that $@_s$ is self-dual; but note that, viewed in more traditional modal terms, it tells us that $@_s$ is a modality whose transition relation is a *function*: one direction is the modal determinism axiom, while the other is the characteristic axiom of deontic logic. Given the jump-to-the-labeled-state

interpretation of @, this is exactly what we would expect. *Introduction* tells us how to introduce information under the scope of the @ operator. Actually, it also tells us how to get hold of such information, for if we replace φ by $\neg\varphi$, contrapose, and make use of *Self-Dual*, we obtain $(\mathbf{s} \wedge @_{\mathbf{s}}\varphi) \rightarrow \varphi$; we call this is *Elimination* schema.

The next group is a *modal theory of labeling* (or to put it another way, a *modal theory of state equality*).

Label $\quad @_{\mathbf{s}}\mathbf{s}$

Nom $\quad @_{\mathbf{s}}\mathbf{t} \rightarrow (@_{\mathbf{t}}\varphi \rightarrow @_{\mathbf{s}}\varphi)$

Swap $\quad @_{\mathbf{s}}\mathbf{t} \leftrightarrow @_{\mathbf{t}}\mathbf{s}$

Scope $\quad @_{\mathbf{t}}@_{\mathbf{s}}\varphi \leftrightarrow @_{\mathbf{s}}\varphi$

The final group tells us how @ and $\Diamond$ interact:

Back $\quad \Diamond @_{\mathbf{s}}\varphi \rightarrow @_{\mathbf{s}}\varphi$

Bridge $\quad \Diamond\mathbf{s} \wedge @_{\mathbf{s}}\varphi \rightarrow \Diamond\varphi$

And (apart from the *Paste* rules) that's $\mathcal{H}[\downarrow, @](K)$. We leave the soundness proof to the reader, and turn straight to the issue of completeness. Essentially we're going to adapt the modal canonical model method to our new language (we assume the usual notions of consistency, Maximal Consistent Sets (MCSs) and so on; see the extended version for further details).

Definition 5 (Canonical Models) *For a countable language $\mathcal{L}$, the* canonical model $\mathcal{M}^c$ *is (S^c, R^c, V^c), where S^c is the set of all $\mathcal{L}$-MCSs; R^c is the binary relation on S^c defined by $\Gamma R^c \Delta$ iff $\Box\varphi \in \Gamma$ implies $\varphi \in \Delta$, for all $\mathcal{L}$-formulas φ; and V^c is the valuation defined by $V^c(a) = \{\Gamma \in S^c \mid a \in \Gamma\}$, where a is a proposition symbol or nominal.*

We begin by proving a key lemma *without* the help of the yet-to-be-introduced *Paste* rules. Let us say that an MCS is *labeled* if and only if it contains a state symbol; if a state symbol belongs to an MCS we call it a *label* for that MCS.

Lemma 6 *Let Γ be a labeled MCS, and for all state symbols s, let Δ_s be $\{\varphi \mid @_s\varphi \in \Gamma\}$. Then:*

1. *For every state symbol s, Δ_s is a labeled MCS that contains s.*
2. *For all state symbols s and t, $@_s\varphi \in \Delta_t$ iff $@_s\varphi \in \Gamma$.*
3. *There is a state symbol s such that $\Gamma = \Delta_s$.*
4. *For all state symbols s, $\Delta_s = \{\varphi \mid @_s\varphi \in \Delta_s\}$.*
5. *For all state symbols s and t, if $s \in \Delta_t$ then $\Delta_t = \Delta_s$.*

Proof. Clause 1. First, for every state symbol s we have the *Label* axiom $@_s s$, hence $s \in \Delta_s$. Next, Δ_s is consistent. For assume for the sake of a contradiction that it is not. Then there are $\delta_1, ..., \delta_n \in \Delta_s$ such that $\vdash \neg(\delta_1 \wedge ... \wedge \delta_n)$. By $@_s$-necessitation, $\vdash @_s \neg(\delta_1 \wedge ... \wedge \delta_n)$, hence $@_s \neg(\delta_1 \wedge ... \wedge \delta_n)$ is in Γ, and thus by *Self-Dual* $\neg @_s(\varphi_1 \wedge ... \wedge \varphi_n)$ is in Γ too. On the other hand, as $\delta_1, ..., \delta_n \in \Delta_s$, we have $@_s\delta_1, ..., @_s\delta_n \in \Gamma$. By simple modal argumentation (all we need is the fact that $@_s$ is a normal modality) it follows that $@_s(\delta_1 \wedge ... \wedge \delta_n) \in \Gamma$ as well, contradicting the consistency of Γ. We conclude that Δ_s must be consistent after all.

It remains to show that Δ_s is maximal. So assume it is not. Then there is a formula χ such that neither χ nor $\neg\chi$ is in Δ_s. But then both $\neg @_s\chi$ and $\neg @_s\neg\chi$ belong to Γ, and this is impossible: if $\neg @_s\chi \in \Gamma$, then by self duality $@_s\neg\chi \in \Gamma$ as well, and we contradict the consistency of Γ. So Δ_s is maximal.

Clause 2. We have $@_s\varphi \in \Delta_t$ iff $@_t@_s\varphi \in \Gamma$. By *Scope*, $@_t@_s\varphi \in \Gamma$ iff $@_s\varphi \in \Gamma$. (We call this the *@-agreement property*; though simple, it plays an important role in our completeness proof.)

Clause 3. By assumption, Γ contains at least one state symbol; let us call it s. If we can show that $\Gamma = \Delta_s$, we will have the result. But this is easy. Suppose $\varphi \in \Gamma$. Then as $s \in \Gamma$, by *Introduction* $@_s\varphi \in \Gamma$, and hence $\varphi \in \Delta_s$. Conversely, if $\varphi \in \Delta_s$, then $@_s\varphi \in \Gamma$. Hence, as $s \in \Gamma$, by *Elimination* we have $\varphi \in \Gamma$.

Clause 4. Use *Introduction* and *Elimination*, much as in the previous clause.

Clause 5. Let Δ_t be such that $s \in \Delta_t$; we shall show that $\Delta_t = \Delta_s$. First observe that since $s \in \Delta_t$, we have that $@_t s \in \Gamma$. Hence, by *Swap*, $@_s t \in \Gamma$ too. But now the result is more-or-less immediate. First, $\Delta_t \subseteq \Delta_s$. For if $\varphi \in \Delta_t$, then $@_t\varphi \in \Gamma$. Hence, as $@_s t \in \Gamma$, it follows by *Nom* that $@_s\varphi \in \Gamma$, and hence that $\varphi \in \Delta_s$ as required. A similar *Nom*-based argument shows that $\Delta_s \subseteq \Delta_t$. $\dashv$

This lemma gives us a lot — in essence it says that the subscripted @ operators in any labeled MCS index a well-behaved collection of labeled MCSs. Now, thinking ahead to the Truth Lemma we will have to prove, it should be clear why we want to work with *labeled* MCSs: with the help of *Q2*, we can use these labels to instantiate state variables bound by $\downarrow$, and hence establish the inductive step for $\downarrow$. Thus the Δ_s are plausible model-building material; nonetheless, they don't yet have all the properties we want.

First there's a small wrinkle: we would like the MCSs we use to be labeled by a *nominal*, not just a free variable; this isn't crucial, but it saves having to worry about about accidental binding. But note that even if Γ itself contains a nominal (say i), we have no guarantee that all the Δ_s do too: for example, Γ may contain $@_x\neg j$ for all nominals j, in which case Δ_x won't contain any nominals at all, though of course it will contain x.

And there's a second, far more serious, problem. Suppose we take the collection of Δ_s yielded by a labeled MCS as the building blocks of our model. Doing this means we have thrown away MCSs; we will be working in a *submodel* of the canonical model. How do we know that a modal style Existence Lemma holds for this submodel? That is, how can prove the clause of the Truth Lemma for the modalities? Bluntly, there is no obvious way to do this.

The *Paste* rules enable us to fix both problems. Here they are:

$$\frac{\vdash @_{\mathbf{s}}(\mathbf{t} \wedge \varphi) \rightarrow \theta}{\vdash @_{\mathbf{s}}\varphi \rightarrow \theta} \qquad\qquad \frac{\vdash @_{\mathbf{s}}\Diamond(\mathbf{t} \wedge \varphi) \rightarrow \theta}{\vdash @_{\mathbf{s}}\Diamond\varphi \rightarrow \theta}$$

The rule on the left is called *Paste-0*, the rule on the right *Paste-1*. In both, t must be a state symbol distinct from s that does not occur in φ or θ.

The key rule is *Paste-1*. Read contrapositively (that is, read from bottom to top) it tells is that pasting a brand new state symbol under the scope of $\Diamond$ is a consistency preserving operation — for if we *can't* derive a contradiction (that is, θ) without the new nominal, then we *can't* derive the contradiction after we have pasted. We shall leave the reader to ponder the simpler *Paste-0* rule (essentially it says that giving a brand new name to a labeled state isn't going to cause any problems) and prove the Extended Lindenbaum's Lemma we need.[11]

Definition 7 (Pasted MCSs) *An MCS* Γ *is* 0-pasted *iff* $@_s\varphi \in \Gamma$ *implies that for some nominal* i, $@_s(i \wedge \varphi) \in \Gamma$. *It is* 1-pasted *iff* $@_s\Diamond\varphi \in \Gamma$ *implies that for some nominal* i, $@_s\Diamond(i \wedge \varphi) \in \Gamma$. *We say that* Γ *is pasted iff it is both 0-pasted and 1-pasted.*

Lemma 8 (Extended Lindenbaum's Lemma) *Let* $\mathcal{L}$ *and* $\mathcal{L}^+$ *be countable languages such that* $\mathcal{L}^+$ *is* $\mathcal{L}$ *enriched with a countably infinite set of new nominals. Then every consistent set of* $\mathcal{L}$*-formulas can be extended to a pasted* $\mathcal{L}^+$*-MCS that is labeled by a nominal.*

Proof. Enumerate the new nominals. Given a consistent set of $\mathcal{L}$-formulas Φ, define Φ_j to be $\Phi \cup \{j\}$, where j is the first new nominal. Φ_j is consistent. For suppose not. Then for some conjunction of formulas δ from Φ we have that $\vdash j \rightarrow \neg\delta$; as j is from the new-nominal enumeration, it does not occur in δ. Let P be a proof of $\vdash j \rightarrow \neg\delta$ and let x be any state variable that does not appear in this proof. Then replacing every occurrence of j in P by x yields a proof of $\vdash x \rightarrow \neg\delta$. Localization then yields $\vdash \downarrow x(x \rightarrow \neg\delta)$. By *Q3*, $\vdash \downarrow x\neg\delta$. Now vacuous occurrences of the $\downarrow$ binder are eliminable in $\mathcal{H}[\downarrow, @](K)$ (for $\vdash \neg\varphi \rightarrow \neg\varphi$, so for any variable x not occurring in φ, localization and *Q1* yield $\vdash \neg\varphi \rightarrow \downarrow x\neg\varphi$, whereupon contraposition and the self duality of $\downarrow$ yield the result). Hence $\vdash \neg\delta$, which contradicts the consistency of Φ. Thus Φ_j is consistent after all.

We now paste. Enumerate all the formulas of $\mathcal{L}^+$, define Θ^0 to be Φ_j, and suppose we have defined Θ^m, where $m \geq 0$. Let φ_{m+1} be the $m+1$-th formula in

[11] The extended version of this paper discusses the admissibility of these rules. A semantic argument is given which strongly suggests that *Paste-0* isn't a genuine enrichment of the system, though at the time of writing this hadn't been backed up by a syntactic proof. The admissibility of *Paste-1* is posed as an open problem.

But while interesting, to focus exclusively on the admissibility of *Paste-1* over an *axiomatic* basis is to miss the true significance of this rule: *Paste-1* is actually the most natural part of $\mathcal{H}[\downarrow, @](K)$ — it's the other components that should be eliminated! This is the strategy adopted in Blackburn and Seligman (1998). Drawing on ideas from Seligman (1997) an @-based *sequent* system is presented and the idea underlying *Paste-1* finds its true home.

Incidentally, *Paste-1* is closely related to a rule introduced by Gabbay and Hodkinson (1990) for *Until-Since* logic. The Gabbay and Hodkinson method is discussed in detail in the extended version of the paper, and *Paste-1* is introduced as, so to speak, an @-based implementation of their idea that bypasses the need to work with arbitrary sequences of tense operators.

our enumeration. We define Θ^{m+1} as follows. If $\Theta^{m+1} \cup \{\varphi_{m+1}\}$ is inconsistent, then $\Theta^{m+1} = \Theta^m$. Otherwise:

1. $\Theta^{m+1} = \Theta^m \cup \{\varphi_{m+1}\}$ if φ_{m+1} is not of the form $@_v v$ or $@_s \Diamond \varphi$. Here s is a state symbol, and v is a state variable.

2. $\Theta^{m+1} = \Theta^m \cup \{\varphi_{m+1}\} \cup \{@_v(k \wedge v)\}$, if φ_{m+1} is of the form $@_v v$. Here k is the next new nominal that does not occur in Θ^m.

3. $\Theta^{m+1} = \Theta^m \cup \{\varphi_{m+1}\} \cup \{@_s \Diamond (k \wedge \varphi)\}$, if φ_{m+1} is of the form $@_s \Diamond \varphi$. Here k is the next new nominal that does not occur in Θ^m or $@_s \Diamond \varphi$.

Let $\Theta = \bigcup_{n \geq 0} \Theta^n$. It is clear that this set is labeled by a nominal, maximal, and 1-pasted. Furthermore, it must be consistent, for the only non-trivial aspects of the expansion are those defined by items 2 and 3, and *Paste-0* and *Paste-1* respectively guarantee that these are consistency preserving.

So it only remains to check that Θ is 0-pasted; because of the rather limited way item 2 uses *Paste-0* this may not be entirely obvious. First, note that by basic modal reasoning $\vdash @_s \theta \wedge @_s \psi \rightarrow @_s(\theta \wedge \psi)$. So suppose $@_s \varphi \in \Sigma$. If s is a nominal, say i, then because $@_i i$ is an axiom, $@_i(i \wedge \varphi) \in \Sigma$ as required. On the other hand, if s is a variable, say x, then because of the pasting process carried out in item 2, for some nominal i we have that $@_x(i \wedge x) \in \Theta$. As $@_s$ is a normal modal operator, $@_x i \in \Theta$, so $@_x(i \wedge \varphi) \in \Sigma$. We conclude that Θ is the required $\mathcal{L}^+$-MCS. $\dashv$

We're now ready to prove the completeness of $\mathcal{H}[\downarrow, @](K)$ — in fact we have everything we need to prove the completeness of many of its extensions as well.

Definition 9 (Labeled models and natural assignments) *Let Γ be a pasted MCS labeled by a nominal. For all state symbols s, let Δ_s be $\{\varphi \mid @_s \varphi \in \Gamma\}$, and define S to be $\{\Delta_s \mid s$ is a state symbol$\}$. Then we define $\mathcal{M}$, the labeled model yielded by Γ, to be (S, R, V), where R and V are the restrictions of R^c (the canonical relation) and V^c (the canonical valuation) to S. We define the natural assignment $g : SVAR \longrightarrow S$ by $g(x) = \{s \in S \mid x \in s\}$.*

Such labeled models have all the structure we want. For a start, by Clause 3 of Lemma 6, $\Gamma \in S$, and by Clause 5, V is a *standard* valuation and g is a *standard* assignment. Further, all states in the model contain nominals (because Γ is 0-pasted), and hence are well-behaved as far as $\downarrow$ is concerned. Moreover, we know from Lemma 6 that $\mathcal{M}$ is extremely well-behaved with respect to $@$. So it only remains to ensure that such models are well-behaved with respect to the modalities; that is, we want an Existence Lemma. This, of course, is where 1-pasting comes in:

Lemma 10 (Existence Lemma) *Let $\mathcal{M} = (S, R, V)$ be the labeled model yielded by a pasted set Γ that is labeled by some nominal. Suppose $\Theta \in S$ and $\Diamond \varphi \in \Theta$. Then there is a $\Phi \in \mathcal{M}$ such that $\Theta R \Phi$ and $\varphi \in \Phi$.*

Proof. As $\Theta \in S$, for some nominal i we have that $\Theta = \Delta_i$; hence as $\Diamond \varphi \in \Theta$, $@_i \Diamond \varphi \in \Gamma$. But Γ is pasted (and hence 1-pasted) so for some nominal k, $@_i \Diamond (k \wedge \varphi) \in \Gamma$, and so $\Diamond(k \wedge \varphi) \in \Delta_i$. If we could show that (1) $\Delta_i R \Delta_k$, and (2) $\varphi \in \Delta_k$, then Δ_k would be a suitable choice of Φ. And in fact *Bridge*

and *Back*, aided by the @-agreement property of our model (that is, item 2 of Lemma 6) will let us establish this.

For (1), we need to show that for any $\psi \in \Delta_k$, we have that $\Diamond\psi \in \Delta_i$. So suppose $\psi \in \Delta_k$. This means that $@_k\psi \in \Gamma$. By @-agreement, $@_k\psi \in \Delta_i$. But $\Diamond k \in \Delta_i$. Hence, by *Bridge*, $\Diamond\psi \in \Delta_i$ as required.

For (2), we know that $\Diamond(k \wedge \varphi) \in \Delta_i$. But $\vdash k \wedge \varphi \rightarrow @_k\varphi$ (this is an instance of *Introduction*), hence $\Diamond@_k\varphi \in \Delta_i$. But then, by *Back*, $@_k\varphi \in \Delta_i$. By @-agreement, $@_k\varphi \in \Gamma$. Hence $\varphi \in \Delta_k$ as required. $\dashv$

Lemma 11 (Truth Lemma) *Let Θ be an MCS in $\mathcal{M}$. For all formulas φ, $\varphi \in \Theta$ iff $\mathcal{M}, \Theta \models \varphi$.*

Proof. By induction: the atomic, boolean, and modal steps are standard (we use the Existence Lemma just proved for the latter).

So suppose $\downarrow x\psi \in \Delta$. Since Δ contains a nominal (say i), by *Q2* $\psi[i/x] \in \Delta$. By the inductive hypothesis $\mathcal{M}, g, \Delta \models \psi[i/x]$. Thus, $\mathcal{M}, g, \Delta \models i \wedge \psi[i/x]$, and by the contrapositive of the *Q2* axiom, $\mathcal{M}, g, \Delta \models \downarrow x\psi$. For the other direction assume $\mathcal{M}, g, \Delta \models \downarrow x\psi$. That is, $\mathcal{M}, g', \Delta \models \psi$, where $g' \stackrel{x}{\sim} g$ such that $g'(x) = \{\Delta\}$. Now Δ contains a nominal, say i, so by the Substitution Lemma, $\mathcal{M}, g, \Delta \models \psi[i/x]$, hence by the inductive hypothesis $\psi[i/x] \in \Delta$. So, by the contrapositive of the *Q2* axiom, $\downarrow x\psi$ is in Δ as required.

The argument for @ runs as follows: $\mathcal{M}, \Theta \models @_s\psi$ iff $\mathcal{M}, \Delta_{\mathbf{s}} \models \psi$ (for by Clause 5 of Lemma 6, Δ_s is the only MCS containing s, and hence, by the the atomic case of the present lemma, the only state in $\mathcal{M}$ where s is true) iff $\psi \in \Delta_s$ (inductive hypothesis) iff $@_s\psi \in \Delta_s$ (using the fact that $s \in \Delta_s$ together with *Introduction* for the left-to-right direction and *Elimination* for the right-to-left direction) iff $@_{\mathbf{s}}\psi \in \Theta$ (by the @-agreement property for the MCSs in S). Thus all cases have been proved, and the Truth Lemma follows by induction. $\dashv$

Theorem 12 (Completeness) *Every $\mathcal{H}[\downarrow, @](K)$-consistent set of formulas in a countable language $\mathcal{L}$ is satisfiable in a countable standard model with respect to a standard assignment function. Moreover, every $\mathcal{H}[\downarrow, @](K)$-consistent set of* sentences *in $\mathcal{L}$ is satisfiable in a countable* connected *standard model*

Proof. The first is proved in the expected way: given a $\mathcal{H}[\downarrow, @](K)$-consistent set of formulas Σ, use the Extended Lindenbaum Lemma to expand it to a pasted set Σ^+ labeled by some nominal in a countable language $\mathcal{L}^+$. By the Truth Lemma just proved, the labeled model and natural assignment that Σ^+ gives rise to satisfy Σ at Σ^+. This model need not be connected, but the submodel generated by Σ^+ is, and all *sentences* in Σ^+ are true in this submodel. $\dashv$

But there's no need to stop here — one of the nicest things about hybrid languages is the ease with which general completeness results for richer logics can be proved.[12] Moreover, such results typically link completeness and *frame-definability* in a very straightforward way.

[12] Historically, this has been a major motivation for exploring hybrid languages. Bull (1970) points out (see page 285), that all state-symbol-based extensions of the basic logic are complete, and a neat argument to the same effect is given at the end of Gargov, Passy and Tinchev (1987). Passy and Tinchev (1991) push matters further; like the earlier Passy and Tinchev (1985), this paper takes PDL as the underlying modal language and explores what happens beyond the first-order barrier. The present paper applies similar arguments to weaker local languages.

A formula is said to *define* some property of frames (for example, transitivity) iff it is valid on precisely the frames with that property (recall from Section 2 that a formula is valid on a frame iff it is impossible to falsify it at any state in that frame, no matter which valuation or assignment is used). The sort of results we are after have roughly the following form: for any formula φ from some specified syntactic class, if φ defines a property P, then using it as an additional axiom *guarantees* completeness with respect to the class of frames with property P. For ordinary modal languages, the Sahlqvist Theorems are the best known result of this type (see Sahlqvist (1975)); as we shall see, analogous results for hybrid languages come far more easily. We shall give two. The idea underlying both is the same: stop thinking in terms of propositional variables, and start thinking in terms of state symbols.

We say that a formula of ML+$\downarrow$+@ is *pure* iff it contains no propositional variables; our first result concerns pure *sentences*. As the following examples show, pure sentences are remarkably expressive; each sentence defines the property listed to its right. All these properties are relevant to temporal reasoning, and (with the exception of transitivity and density) none are definable in ordinary modal logic:

$\downarrow x \Box \neg x$	*Irreflexivity*
$\downarrow x \Box \Box \neg x$	*Asymmetry*
$\downarrow x \Box (\Diamond x \rightarrow x)$	*Antisymmetry*
$\downarrow x \Box \downarrow y @_x \Diamond \Diamond y$	*Density*
$\downarrow x \Box \Box \downarrow y @_x \Diamond y$	*Transitivity*
$\downarrow x \Diamond \downarrow y @_x (\Box \Box \neg y \wedge \Box \downarrow z @_y (z \vee \Diamond z))$	*Discreteness*

The last three expressions can be simplified using $\Downarrow^n$ notation.[13]

Let us say that a *pure sentential axiomatic extension* of $\mathcal{H}[\downarrow, @](K)$ is any system obtained by adding as axioms a set of pure sentences of ML+$\downarrow$+@.

Theorem 13 (Extended Completeness I) *Let* `Pure` *be a set of pure sentences of ML+$\downarrow$+@, and let $\mathcal{P}$ be the pure sentential axiomatic extension of $\mathcal{H}[\downarrow, @](K)$ obtained by adding all sentences in* `Pure` *as axioms. Then every $\mathcal{P}$-consistent set of formulas in a countable language $\mathcal{L}$ is satisfiable in a countable standard model, based on a frame that validates every axiom in* `Pure`*, with respect to a standard assignment function. Moreover, every consistent set of sentences in $\mathcal{L}$ is satisfiable in a countable connected standard model based on a frame that validates* `Pure`*.*

[13] This notation was introduced in Footnote 9. The definition of density can be rewritten as $\Downarrow^1_y \Diamond \Diamond y$ ("every state y that can be reached in one step can be reached in two steps"), the definition of transitivity is $\Downarrow^2_y \Diamond y$ ("every state y that can be reached in two steps can be reached in one step"), while discreteness simplifies to $\downarrow^1_y (\Box \Box \neg y \wedge \Downarrow^1_z @_y (z \vee \Diamond z))$ ("there is a successor state y, that is *not* 2-step reachable, from which any successor state z is 0- or 1-step reachable").

Proof. An easy corollary of Theorem 12: given a $\mathcal{P}$-consistent set of formulas Σ, build a satisfying model by expanding Σ to a set Σ^+ in a countable language $\mathcal{L}^+$, and forming the labeled model $\mathcal{M} = (S, R, V)$ and the natural assignment g. Now, the labeled model is built of MCSs, and each axiom in `Pure` belongs to every $\mathcal{P}$-MCS, thus by the Truth Lemma, $\mathcal{M}, g \models$ `Pure`. But as `Pure` contains only *sentences*, the choice of assignment is irrelevant, hence $\mathcal{M} \models$ `Pure`. Moreover, as `Pure` contains only *pure* sentences, the choice of valuation is also irrelevant, and $(S, R) \models$ `Pure`. This proves the first claim. Finally, if Σ contains only sentences, we obtain a connected model by restricting our attention to the submodel generated by Σ^+; the underlying subframe validates `Pure`. $\dashv$

As a simple application, note that we obtain the logic of *strictly partially ordered* frames (which many writers, for example van Benthem (1983), would regard as the minimal *temporal* logic) by adding as axioms $\downarrow x \Box \neg x$ and $\Downarrow^2_y \Diamond y$; the previous theorem guarantees that the labeled model validates these axioms, hence as they define irreflexivity and transitivity respectively, the labeled model will have these properties.

This is pleasant, but let's push things further. Theorem 13 requires us to use sentences as axioms. However it can be more natural to use *pure schemas*. Consider, for example, the schema $\Diamond\Diamond\mathbf{s} \rightarrow \Diamond\mathbf{s}$. Any instance of this schema defines transitivity, and it is easy to verify that including all instances as axioms guarantees a transitive labeled model. Similarly, any instance of the schema

$$\Diamond\mathbf{s} \wedge \Diamond\mathbf{t} \rightarrow [\Diamond(\mathbf{s} \wedge \Diamond\mathbf{t}) \vee \Diamond(\mathbf{s} \wedge \mathbf{t}) \vee \Diamond(\mathbf{t} \wedge \Diamond\mathbf{s})]$$

defines the no-branching-to-the-right property, and including all instances as axioms guarantees a labeled model with this property. Both transitivity and no-branching-to-the-right are definable using pure sentences,[14] but the use of schemas can offer more. A simple example is the schema $\Diamond\mathbf{s}$; any instance of this defines the class of frames (S, R) such that $R = S \times S$, and its inclusion as an axiom schema imposes this property on labeled models.[15]

A *pure schematic extension* of $\mathcal{H}[\downarrow, @](K)$ is any system obtained by adding all ML+$\downarrow$+@ instances of a set of pure schemas of ML+$\downarrow$+@ as axioms to $\mathcal{H}[\downarrow, @](K)$.

Theorem 14 (Extended Completeness II) *Let* `Schemas` *be a set of pure schemas of ML+$\downarrow$+@, and let $\mathcal{S}$ be the pure schematic extension of $\mathcal{H}[\downarrow, @](K)$ obtained by adding all instances of the schemas in* `Schemas` *as axioms. Then every $\mathcal{S}$-consistent set of sentences in a countable language $\mathcal{L}$ is satisfiable in a countable standard model, based on a frame that validates all these axioms, with respect to a standard assignment function. Moreover, every consistent set of sentences in $\mathcal{L}$ is satisfiable in a countable connected standard model based on a frame that validates all these axioms.*

Proof. See the extended version of this paper. $\dashv$

[14] The pure sentence $\Downarrow^1_y \Downarrow^1_z (\Diamond y \wedge \Diamond z \rightarrow [\Diamond(y \wedge \Diamond z) \vee \Diamond(y \wedge z) \vee \Diamond(z \wedge \Diamond y)])$ defines no-branching-to-the-right.

[15] We don't know many temporally relevant examples in ML+$\downarrow$+@ that require the use of schemas, but examples are easy to find in *tense* logic enriched with $\downarrow$. For example, the schema $P\mathbf{s} \vee \mathbf{s} \vee F\mathbf{s}$ guarantees *trichotomy* (that is, $\forall xy(xRy \vee x = y \vee yRx)$), while $PF\mathbf{s}$ guarantees us *left-directedness* (that is, $\forall xy \exists z(zRx \wedge zRy)$).

What sort of coverage do Theorems 13 and 14 offer? For a start, note that all our examples of frame properties definable by pure sentences or (instances of) pure schemas were *first-order*. This is no accident: a simple extension of the *Standard Translation* for the basic modal language shows that *every* pure formula of ML+↓+@ defines a first-order condition on frames. The Standard Translation for the basic modal language is defined as follows:

$$\begin{array}{lcl} ST_x(p) & = & Px, \text{ for all propositional symbols } p \\ ST_x(\neg\varphi) & = & \neg ST_x(\varphi) \\ ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\varphi) \\ ST_x(\Box\varphi) & = & \forall y(xRy \rightarrow ST_y(\varphi)) \end{array}$$

(In the first clause, P is a monadic second-order predicate variable; each propositional symbol corresponds uniquely to such a symbol.) Following Blackburn and Seligman (1998), we extend this translation to ML+↓+@ as follows: we assume that the first-order variables we have available consist of all the usual state variables, plus a distinct variable x_i for each nominal i and define:

$$\begin{array}{lcl} ST_x(y) & = & x = y, \text{ for all state variables } y \\ ST_x(i) & = & x = x_i, \text{ for all nominals } i \\ ST_x(\downarrow x\varphi) & = & \exists y(x = y \wedge ST_x(\varphi)) \\ ST_x(@_y\varphi) & = & ST_y(\varphi) \end{array}$$

Suppose φ is a formula of ML+↓+@; we suppose that φ has been α-converted so that it contains no occurrences of the variable x (we reserve this variable to denote the current state). It is easy to see that $ST_x(\varphi)$ will contain at least one free variable (namely x). It is also easy to see that this extended version of ST preserves satisfaction. That is for any ML+↓+@ formula φ, any standard model $\mathcal{M} = (S, R, V)$, any standard assignment g, and any $s \in S$:

$$\mathcal{M}, s, g \models \varphi \text{ iff } \mathcal{M} \models ST_x(\varphi)[s, \mathbf{g(z)}, \mathbf{V(i)}, \mathbf{V(p)}].$$

The notation on the right means: assign s to the free variable x, assign the unique element of $g(z)$ to z if z occurs free in the translation, assign the unique element of $V(i)$ to x_i if x_i occurs free in the translation, and assign $V(p)$ to P if P is a monadic predicate variable that occurs free in the translation. Now we can see why it pays to be pure: if φ contains no propositional variables, then the previous expression simplifies to

$$\mathcal{M}, g, s \models \varphi \text{ iff } \mathcal{M} \models ST_x(\varphi)[s, \mathbf{g(z)}, \mathbf{V(i)}].$$

We are now firmly in the world of first order logic. But let's carry on. We have:

$$\mathcal{M}, g \models \varphi \text{ iff } \mathcal{M} \models \forall x ST_x(\varphi)[\mathbf{g(z)}, \mathbf{V(i)}],$$

and hence:

$$(S, R) \models \varphi \text{ iff } (S, R) \models \forall z_1 \cdots \forall z_n \forall x ST_x(\varphi).$$

On the righthand side we have simply universally quantified over all the free-variables in $\forall x ST_x(\varphi)$. In short, the frame property any pure formula defines can be calculated by applying the standard translation and forming the universal closure. Thus Theorem 13 and 14 bear a certain family resemblance to the

Sahlqvist Theorems: all these results cover first-order properties which can be effectively calculated from the relevant axioms.

There are a host of related questions worth pursuing. For example, we have seen many examples of first-order properties which are not modally definable but which are definable using pure formulas; can *all* modally definable first-order conditions be captured in this way? And if not, can all Sahlqvist definable properties be so captured?[16]

5 Working with other sorts

Our technical work is done, but our conceptual work is not. The reader may have gained the impression that hybridization is simply the business of quantifying over states in a modal setting. But while that's part of the story, and an important part too, we believe that a more general idea deserves to be made explicit.

Our preceding work rested on a simple idea: combining two forms of information in a uniform way. Our languages dealt with arbitrary information (via the propositional symbols) and labeling information (via the state symbols) and yet we drew no distinction between terms and formulas; both types of information were handled *propositionally*. Now the natural question is: if this works for state-label information, why shouldn't it work for other types of information as well? For example, in some applications we might want to work with intervals, or events, or paths, or some combination of these entities — so why not introduce special atomic symbols that label such entities and allow ourselves to bind them? In short, why not attempt hybridization in more ambitious ways?[17]

Intriguingly, there are at least two ways of doing this. The first involves little change to the work of previous sections. For example, working with intervals in a modal logic standardly means working with richer frames, perhaps frames of the form $(S, <, \sqsubseteq)$. Here S is thought of as a set of intervals, $<$ as the precedence relation on intervals, and $\sqsubseteq$ as the inclusion relation on intervals.[18]

[16] There are first-order properties which are modally definable but not Sahlqvist definable, which can be defined by pure sentences. For example, transitivity + atomicity ($\forall x \exists y(xRy \wedge \forall z(yRz \rightarrow z = y))$) is definable by the conjunction of the modal transitivity axiom ($\Diamond\Diamond p \rightarrow \Diamond p$) and the McKinsey formula ($\Box\Diamond p \rightarrow \Diamond\Box p$), but no Sahlqvist formula defines this condition. Incidentally, McKinsey does not define atomicity, and in fact, no ordinary modal formula does so; only transitivity + atomicity is modally definable. But the following pure sentence defines atomicity: $\Diamond\downarrow y \Box y$. We have already seen that transitivity is definable by a pure sentence.

[17] In suggesting this we are merely echoing Arthur Prior, for this idea was an important — perhaps the dominant — theme in his later work; the key reference here is the posthumous Prior and Fine (1977), which consists of draft chapters of a book, together with papers, and an invaluable appendix by Kit Fine which attempts to systematically reconstruct Prior's views. Prior attached immense philosophical weight to this project; in his view it showed that that possible worlds were not needed to analyze modal notions; and indeed, that times were not needed to analyze temporal expressions. Only (suitably sorted) propositions (and properties) mattered.

Prior's philosophical position is interesting: it is strongly information oriented, has natural affinities with frameworks such as Property Theory and Situation Semantics, and deserves further exploration. Nonetheless, here we prefer to adopt a neutral perspective on the philosophical significance of hybrid languages: for present purposes, they are simply an elegant tool for talking about structures locally, and adding further sorts is simply an interesting technical idea.

[18] Various constraints would be imposed to make this interpretation plausible. Typically we would demand that $(S, <)$ be a strict partial order, that $(S, \sqsubseteq)$ be partial order, and that $<$

Or perhaps we'd prefer working with frames bearing the 14 relations demanded in Allen (1984). Either way, the fundamental point is that we are enriching our notion of what a state is by locating it in a richer web of relations. This mode of enrichment is obviously compatible with the methods discussed earlier; for example, it is straightforward to work with Allen-style intervals using $\downarrow$ and @.[19] Such an approach naturally leads to multi-sorted systems. For example, if we wanted to work with atomic interval structures, it would be natural to have a sort which labeled arbitrary intervals, and a subsort which labeled *atomic* intervals (see Blackburn (1992)).

But there is another way of developing multi-sorted hybrid languages. This hinges on the following observation: some entities can be thought of as structured sets of states. For example, an interval is the set of all states between two end points.[20] Why not add atomic symbols that range over such sets? After all, we already have propositional symbols ranging over arbitrary subsets, and state-symbols ranging over singleton subsets — so why not symbols that range over convex sets too? This is arguably a useful idea (see Blackburn (1990,1992,1993)) and it is certainly simple to handle logically.[21] But to illustrate the structured-set approach to sorting in more detail we want to discuss not intervals but *paths*, because this example not only provides a nice illustration of the potential of sorting for temporal logic, it also makes clear that even simple-looking extensions can give rise to non-trivial problems.

Many applications of temporal logic demand the use of *paths*, or *courses of history*. For example, for philosophical purposes it is natural to model the idea that the future is unknown by using tree-like models of time that branch into alternative futures, and in computer science it is standard to reason about unravelings of non-deterministic transition systems. On the face of it, these applications only seem to demand that we work with new classes of tree-like models, and clearly we can do that with the tools we already have. But this is only half the story. As well as new models, we are faced with new expressive demands, and these will lead us to new territory.

For example, in natural language semantics we would like to have a future tense operator $\mathcal{F}$ such that $\mathcal{F}\varphi$ is true precisely when φ holds somewhere in every possible future (that is, when φ holds at least once on every path through the current state). However we can't define $\mathcal{F}$ in any of our hybrid languages; even abandoning locality and working with ML+$\forall$+A doesn't help. As a second example consider *fairness*. In computer science applications we may want to insist that a process is activated infinitely often along every possible computation

and $\sqsubseteq$ interacted appropriately (for example, we'd want $\forall stt'((s \sqsubseteq t \wedge t < t') \rightarrow \neg s \sqsubseteq t')$); see van Benthem (1983) for further discussion.

[19] The 'straightforward' is justified: many of the frame properties required are expressible by pure sentences or schemas, hence completeness will often be automatic. For example, $\downarrow x[\sqsubseteq]G\downarrow y@_x\neg Fy$ regulates the interaction of $<$ and $\sqsubseteq$ (here $[\sqsubseteq]$ means "at all super-intervals"). As a second example, we have already noted that atomicity (which we may want for $\sqsubseteq$) is enforceable using a pure sentence (see Footnote 15). It would be interesting to compare an $\downarrow$- and @-based treatment with Yde Venema's two-dimensional analysis (see Venema (1990)).

[20] Of course, one might want to distinguish between various types of intervals, such as open and closed, but we won't do so here.

[21] Readers familiar with the representation theorems for abstract interval structures in terms of point-based structures proved in van Benthem (1983) will (rightly) suspect that in many cases this structured-set approach to hybrid interval logic will turn out to be equivalent to the additional-relations approach. Incidentally, this 'duality' between the additional-relations and the structured-set approaches may be relevant for paths too.

path; but our state symbols won't help us define a fairness operator. Thus we have a genuine expressivity shortcoming on our hands. Let's try to fix it by hybridization.[22]

The basic strategy for dealing with paths in hybrid languages should be clear. First we add a third sort, the sort of *path symbols* (presumably we want to keep the state symbols, though this of course is optional). As with state symbols, path symbols should be divided into two subcategories, namely path variables (which will be open to binding) and path nominals (which will not). So we choose PVAR to be a countably infinite set of path variables (whose elements we typically write as ρ and ρ') and PNOM to be a countably infinite set of path nominals (whose elements we typically write as τ and τ'), and of course we choose these sets to be disjoint from each other and from PROP, SVAR, and NOM. We define the set of atoms of our enriched language to be $\mathrm{PROP} \cup \mathrm{SVAR} \cup \mathrm{NOM} \cup \mathrm{PVAR} \cup \mathrm{PNOM}$.

The second step is to add a binder. We shall add a binder called $\Downarrow^\pi$, thus forming the language $\mathrm{ML}+\downarrow+@+\Downarrow^\pi$. As the notation is meant to suggest, $\Downarrow^\pi$ is a universal quantifier over paths through the current state (that is, 'local paths'). The wffs of this language are defined in the expected way, as are such concepts as free and bound path variables, so let's proceed straight to the semantics.

We shall work with strictly partially ordered trees (S, R), and adopt Bull's definition of a path: a path π in (S, R) is a linearly ordered subset of S that is maximal among the linearly ordered subsets of S. That is, paths are convex subsets of S that contain the root node and are closed under R-successorship. We denote the set of paths in (S, R) by $\Pi(S, R)$. If $\pi \in \Pi(S, R)$ and $s \in \pi$ then we say that π passes through s. Obviously $\Pi(S, R)$ is never empty, and at least one path passes through every state.

Definition 15 (Standard models and assignments) *Let $ML + \downarrow + @ + \Downarrow^\pi$ be a hybrid language built over* PROP, SVAR, NOM, PVAR *and* PNOM. *A model $\mathcal{M}$ for this language is a triple (S, R, V) such that (S, R) is a strictly partially ordered tree, and $V : \mathrm{PROP} \cup \mathrm{NOM} \cup \mathrm{PNOM} \longrightarrow Pow(S)$. A model is called* standard *iff for all nominals $i \in \mathrm{NOM}$, $V(i)$ is a singleton subset of S, and for all path nominals $\tau \in \mathrm{PNOM}$, $V(\tau) \in \Pi(S, R)$.*

An assignment on $\mathcal{M}$ is a mapping $g : \mathrm{SVAR} \cup \mathrm{PVAR} \longrightarrow Pow(S)$. An assignment is called standard *iff for all state variables $x \in \mathrm{SVAR}$, $g(x)$ is a singleton subset of S, and for all path variables $\rho \in \mathrm{PNOM}$, $V(\rho) \in \Pi(S, R)$.*

Now to interpret the language. The atomic clause is automatically taken care of by our $[V, g]$ notation, and the clauses for the Booleans and modalities are unchanged. So it only remains to interpret $\Downarrow^\pi$:

$$\mathcal{M}, g, s \models \Downarrow^\pi_\rho \varphi \quad \text{iff} \quad \mathcal{M}, g', s \models \varphi, \text{ for all } g' \overset{\rho}{\sim} g \text{ such that } s \in g'(\rho).$$

That is, $\Downarrow^\pi$ is a universal quantifier over local paths; the dual binder $\downarrow^\pi_\rho \varphi$ is an existential quantifier over local paths.

[22] We are not the first to do this. Motivated by Prior's arguments, Robert Bull added a universal quantifier over paths to $\mathrm{TL}+\forall+A$ in his classic 1970 paper; thus, far from being the new kid on the block, hybridization is actually one of the oldest approaches to path-based reasoning we know of. A recent paper by Goranko on hybrid languages strong enough to embed CTL^* (see Goranko (1996b)) is worth noting; Goranko's language doesn't contain path binders, but it does contain path nominals.

It is easy to see that sentences of this language are preserved under generated submodels. Moreover, the expressivity has clearly been boosted. For example, we can now define the $\mathcal{F}$ operator:

$$\mathcal{F}\varphi \ := \ \Downarrow^{\pi}_{\rho}\Diamond(\rho \wedge \varphi).$$

It is also straightforward to define a fairness operator:

$$\mathit{Fair}(\varphi) \ := \ \Downarrow^{\pi}_{\rho}(\Diamond(\rho \wedge \varphi) \wedge \Box((\rho \wedge \varphi) \rightarrow \Diamond(\rho \wedge \varphi)).$$

At any state s in a standard model, $\mathit{Fair}(\varphi)$ is true at a state s iff φ is true infinitely often along every path through s.

Moreover, familiar-looking principles of hybrid reasoning extend to our new binder. For example, the rule of *path variable localization* (if φ is provable then so is $\Downarrow^{\pi}_{\rho}\varphi$, for any path variable ρ) preserves validity, and all instances of the following three schemas are valid:

Q1	$\Downarrow^{\pi}_{\boldsymbol{\rho}}(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \Downarrow^{\pi}_{\boldsymbol{\rho}}\psi)$
Q2	$\Downarrow^{\pi}_{\boldsymbol{\rho}}\varphi \rightarrow (\mathbf{p} \rightarrow \varphi[\mathbf{p}/\boldsymbol{\rho}])$
Q3	$\Downarrow^{\pi}_{\boldsymbol{\rho}}(\boldsymbol{\rho} \rightarrow \psi) \rightarrow \Downarrow^{\pi}_{\boldsymbol{\rho}}\psi$
Local-Path	$\downarrow^{\pi}_{\boldsymbol{\rho}}\boldsymbol{\rho}$

(Here $\boldsymbol{\rho}$ and $\mathbf{p}$ are used as metavariables across path variables and path symbols respectively. In *Q1*, $\boldsymbol{\rho}$ must not be free in φ; and in *Q2*, $\mathbf{p}$ must be substitutable for $\boldsymbol{\rho}$ in φ.) In short, the basic quantificational powers of $\Downarrow^{\pi}$ described by *Q1–Q3* are analogous to those of $\downarrow$, and *Local-Path* is analogous to the validity $\downarrow x x$.

Moreover, we have a *Barcan* analog:[23]

Barcan_{π}	$\Downarrow^{\pi}_{\boldsymbol{\rho}}\Box\varphi \rightarrow \Box\Downarrow^{\pi}_{\boldsymbol{\rho}}\varphi$

The contraposed and dualised form $\Diamond\downarrow^{\pi}_{\boldsymbol{\rho}}\varphi \rightarrow \downarrow^{\pi}_{\boldsymbol{\rho}}\Diamond\varphi$ is perhaps easier to grasp. Essentially this says: "if we can select a suitable path at a successor state, then we can select a suitable path at the current state"; it is a path existence principle.

Our language also supports schemas that reflect path geometry (we use $\mathbf{p}$ as a metavariable over path symbols and $\mathbf{s}$ and $\mathbf{t}$ as metavariable over state nominals):

P1	$\Diamond\mathbf{p} \rightarrow \mathbf{p}$
P2	$\mathbf{p} \wedge \Diamond\top \rightarrow \Diamond\mathbf{p}$
P3	$\Diamond(\mathbf{s} \wedge \mathbf{p}) \wedge \Diamond(\mathbf{t} \wedge \mathbf{p}) \rightarrow \Diamond(\mathbf{s} \wedge \Diamond\mathbf{t}) \vee \Diamond(\mathbf{s} \wedge \mathbf{t}) \vee \Diamond(\mathbf{t} \wedge \Diamond\mathbf{s})$

[23] The significance of this may not be apparent to readers of this short version. Roughly speaking, in hybrid languages the validity of Barcan analogs is often a sign that the logic will be well-behaved. For further discussion, see Blackburn and Tzakova (1998a), the extended version of the present paper.

Clearly *P1* reflects convexity, *P2* reflects R-maximality under successorship, and *P3* reflects linearity; note the way the state and path symbols cooperate here. Summing up, in many ways $\mathrm{ML} + \downarrow + @ + \Downarrow^{\pi}$ is a pleasant language.

That's the good part — let's turn to the bad. It seems that proving completeness results for $\Downarrow^{\pi}$ will require new ideas; the labeled model method used in the previous section does not automatically give us completeness results for the new binder, or at least, not with respect to the standard semantics defined above. What's the problem? It's simple, but deadly: although the labeled model construction will guarantee that all *states* are labeled, we don't have any guarantee that all *paths* will be labeled by some path symbol.[24]

This is not easy to fix. What are we to do? Robert Bull makes an interesting remark. He comments (see his Footnote 5 on page 292) that although not every path is the interpretation of some path symbol, his model:

> *. . . does provide enough paths V(u) to give a reasonable interpretation.*

With this remark, Bull hints at a line of work that has subsequently become common in path-based temporal logic. All reasonably expressive path-based logics we know of (for example, Ockhamist logic or CTL*) face similar difficulties regarding completeness. A standard response to the problem is to prove completeness with respect to some suitably liberalized notion of model, for example models containing 'bundles' of paths (see Zanardo (1996)); such approaches have affinities with the use of generalized models in second-order logic, or general frames in modal logic. We believe it would be interesting to explore this landscape using hybrid path languages, and suspect that the labeled model construction may be useful in such investigations.

But what of the standard semantics defined above? This may call for a more brutal line of approach: the use of infinitary rules. Intuitively what is needed is an infinitary extension of the *Local-Path* schema. From *Local-Path* we can deduce that there is a path through the current state; what we also need is a principle that ensures that given a sequence of states (one of which is the current state) that satisfies the convexity, R-maximality, and linearity principles, then there is a path nominal that is true at all the states in this sequence. Infinitary rules are unpalatable — but a clean infinitary approach *may* provide a framework which can (at least, in some cases of interest) be suitably finitized; however we must admit that at present we don't know how realistic the prospects of success here are.

And that's a taste of the joys and sorrows of hybrid path languages. We have only scratched the surface of a vast topic, but we hope we have said enough to indicate why we find this terrain worthy of further exploration. Moreover, we hope we have given the reader a taste of the variety of options hybridization offers to the study of rich temporal ontologies.

[24] Incidentally, we're *not* claiming that adding the axioms and rules mentioned above to $\mathcal{H}[\downarrow, @](K)$ yields a system complete with respect to the standard semantics — it's obvious that it doesn't. Rather, the point is that even after we plug up all the obvious gaps with suitable axioms, we'll still face a tough problem. For further discussion, see the extended version.

6 Concluding remarks

We have argued that the hybridization technique introduced by Arthur Prior and developed by Robert Bull and the Sofia School is a natural tool for temporal logic. Our argument had both a technical and conceptual side.

Our technical results showed that hybridization is compatible with a temporally natural *locality* assumption, namely that temporal operators and binders should only be able to work with temporally accessible states. We showed that ML+$\downarrow$+@, a local language in which *Until* was definable, had an elegant minimal logic and that many temporally interesting extended completeness results could be obtained automatically. In our view, this language meets the three criteria listed at the end of Section 2; in particular, we feel it exhibits a genuine synergy of modal and classical ideas.

It's only fair to warn the reader that we pay a price for this synergy: $\mathcal{H}[\downarrow, @](K)$ lacks the finite model and is undecidable, and the same is true of the logic of strict partial orders.[25] Of course, the logics of many interesting frame classes are decidable (for example, the logics of various classes of trees can be proved decidable using Rabin-style arguments; see Blackburn and Seligman (1998)), nonetheless the fact remains that binding variables to states tilts the underlying computational properties firmly in the classical direction.

But we believe this is a price worth paying. Labeled deductive systems (Fitting (1983), Gabbay (1992)) have proved an important technique for automating modal inference — but labels are usually regarded as a convenient (if somewhat *ad-hoc*) metalinguistic tool. Labels are far more important than that; indeed, if Prior is right, they are fundamental to the entire modal enterprise. Hybrid languages *internalize* the notion of label in the object language, and this internalization can be motivated on grounds that are completely independent of the desire for deductive felicity. Nonetheless, as the use of the *Paste-1* rule already indicates (see Footnote 11) deductive felicity is there for the taking: Seligman (1997) discusses natural deduction and sequent-based methods for global hybrid languages containing both $\forall$ and @, and Blackburn and Seligman (1998a) shows that these methods can be adapted even to weak (decidable) languages that contain no binders at all. In our view the deductive and conceptual clarity offered by internalized labels is more than ample compensation for the undecidability results just noted.

Our main conceptual argument in favor of hybridization is essentially a secular version of Prior's vision of abstract entities as propositions. That is, we feel that regardless of whether there is an interesting metaphysical sense in which arbitrary information types *should* be thought of propositionally, freely combining different sorts of information in one modal algebra is a natural way of modeling temporal reasoning over rich ontologies.

Acknowledgments

We would like to thank the referees for their comments on the first version of this paper, and Natasha Kurtonina for her comments on later versions. Patrick

[25] For proofs of the latter, see Blackburn and Seligman (1998). Proofs of the former are easy to come by using the 'spy point' technique introduced in Blackburn and Seligman (1995).

Blackburn would like to thank Aravind Joshi and the staff of the IRCS, University of Pennsylvania, for their hospitality while the final version of this paper was being prepared.

References

[1] J. Allen. Towards a general theory of knowledge and action. *Artificial Intelligence*, 23:123–154, 1984.

[2] J. van Benthem. *The Logic of Time*. Reidel, 1983.

[3] P. Blackburn *Nominal Tense Logic and Other Sorted Intensional Frameworks*. PhD Thesis, Centre for Cognitive Science, University of Edinburgh, 1990.

[4] P. Blackburn. Fine grained theories of time. In M. Aurnague, A. Borillo, M. Borillo and M. Bras, editors, *Semantics of Time, Space and Movement. Working Papers of the 4th International Workshop TSM'92, Château de Bonas, 1992*, pages 327–348, 1992.

[5] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 14:56–83, 1993.

[6] P. Blackburn. Tense, temporal reference, and tense logic. *Journal of Semantics*, 11:83–101, 1994.

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

[8] 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.

[9] P. Blackburn and J. Seligman. Internalizing labeled deduction. Manuscript, 1998a. Will be made available at `http://www.coli.uni-sb.de/~patrick/`.

[10] P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 4:625–650, 1998.

[11] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic (extended version). Manuscript, 1998a. Will be made available at http://www.coli.uni-sb.de/~patrick/. An earlier version, containing a completeness proof for ML+$\downarrow$+$\Downarrow^1$, is available as Max-Planck-Institut für Informatik report MPI-I-98-2-006, February, 1998.

[12] P. Blackburn and M. Tzakova. Hybridizing concept languages. Manuscript, 1998b.

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

[14] J. Burgess. Basic tense Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 135–165. D. Reidel, 1984.

[15] M. Cresswell. *Entities and Indices.* Kluwer, 1990.

[16] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics.* Reidel, 1983.

[17] D. Gabbay. *Labeled Deductive Systems.* Oxford University Press, 1992.

[18] D. Gabbay and I. Hodkinson. An axiomatization of the temporal logic with since and until over the real numbers. *Journal of Logic and Computation*, 1:229–259, 1990.

[19] G. Gargov, S. Passy and T.Tinchev. Modal environment for boolean speculations (preliminary report). In D. Skordev, editor, *Mathematical Logic and its Applications. Proc. of the Summer School and Conference dedicated to the 80th Anniversary of Kurt Gödel, Druzhba, 1986*, pages 253–263. Plenum Press, 1987.

[20] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22(6):607–636, 1993.

[21] V. Goranko. Temporal logic with reference pointers. In D. Gabbay and H. J. Ohlbach, editors, *Proceedings of the 1st International Conference on Temporal Logic*, volume 827 of *LNAI*, pages 133–148. Springer, 1994.

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

[23] V. Goranko. An interpretation of computational tree logics into temporal logics with reference pointers. Technical Report 2/96, Verslagreeks van die Department Wiskunde, RAU, Department of Mathematics, Rand Afrikaans University, Johannesburg, South Africa, 1996b.

[24] 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.

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

[26] A. Prior. *Past, Present and Future.* Oxford University Press, Oxford, 1967.

[27] A. Prior and K. Fine. *Worlds, Times, and Selves.* University of Massachusetts Press, 1977.

[28] N. Rescher and A. Urquhart. *Temporal Logic.* Springer Verlag, 1971.

[29] B. Richards, I. Bethke, J. van der Does, and J. Oberlander. *Temporal Representation and Inference.* Academic Press, New York, 1989.

[30] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973*, pages 110–143. North-Holland, 1975.

[31] J. Seligman. The logic of correct descriptions. In M. de Rijke, editor, *Advances in Intensional Logic*. Applied Logic Series, Kluwer, 1997.

[32] M.P.A. Sellink. Verifying modal formulas over I/O-automata by means of type theory. Logic group preprint series, Department of Philosophy, Utrecht University, 1994.

[33] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31:529–547, 1990.

[34] A. Zanardo. Branching-time logic with quantification over branches: the point of view of modal logic. *Journal of Symbolic Logic*, 61:1–39, 1996.