Hybrid Languages and Temporal Logic
(Full Version)

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. Technically, 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 three weaker local languages, all of which are capable of defining the *Until* operator; we provide simple minimal axiomatizations for all three systems, and show that in a wide range of temporally interesting cases, extended completeness results can be obtained automatically. Conceptually, we argue that the idea of sorted atomic symbols which underpins the hybrid enterprise can be developed much 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 $\mathrm{ML} + \forall$:

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

The x in this expression is a *state variable*, and all its occurrences are bound by the *binder* $\forall$. Syntactically, state variables are *formulas*: after all, the expression $x \to \neg\Diamond x$ is built using $\to$, $\neg$ and $\Diamond$ in the same way that $p \to \neg\Diamond p$ 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 $\mathrm{ML} + \forall$ 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 $\forall$ is not local; as our notation suggests, $\forall$ quantifies across *all* states. So, if we want a local hybrid language, $\mathrm{ML} + \forall$ 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 $\downarrow$. Now, $\downarrow$ does something simple and natural: it binds a variable to the *current* state. Unfortunately, while $\mathrm{ML}+\downarrow$ is a local language, it has two drawbacks. First, it is not expressive

[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 $\forall$ 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 $\forall$ 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 $\downarrow$ 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). Also relevant are Gargov and Goranko (1993), Blackburn (1993), and Blackburn (1994); these look at modal and tense logics enriched with nominals (in effect, the free variable fragments of hybrid languages).

enough for many applications (for example, we shall show that it is not strong enough to define the *Until* operator). Second, and in stark contrast to $\mathrm{ML} + \forall$ which has an elegant axiomatization, the only known axiomatization for $\mathrm{ML} + \downarrow$ (see Blackburn and Tzakova (1998)) makes use of a rather complex rule of proof called *COV*.

What are we to do? The paper explores this issue in depth. We develop three increasingly general answers, each of which builds on its predecessor. First we introduce a second local binder called $\Downarrow^1$. The $\Downarrow^1$ binder is a universal quantifier over *accessible* states. It is strong enough to define *Until*, and moreover — *as long as we are content to work with transitive models* — the minimal temporal logic of $\mathrm{ML} + \downarrow + \Downarrow^1$ (that is, the set of formulas valid on strict partial orders) has a straightforward axiomatization. This is pleasant, but leads in turn to a new question: how can we eliminate the need to assume transitivity?

Inspection of the completeness proof shows that the combination of transitivity and $\Downarrow^1$ is really a way of ensuring *communication* between parts of formulas (essentially it allows us to keep track of the way we instantiate binders at neighboring states). So the question becomes: how do we achieve better communication in hybrid languages? We give a preliminary answer by replacing the underlying forward-looking modal language by the bi-directional language of tense logic. As we shall show, the interaction of the forward and backward looking operators in tense logic gives us all the communication we require, and by adapting a method from Gabbay and Hodkinson (1990) we can axiomatize the formulas of $\mathrm{TL}{+}\downarrow$ that are valid on all models.

So it remains to put the pieces together: can we transfer the insight about communication back to forward-looking-only modal languages while maintaining locality? Yes, we can. We do so by introducing an operator called @ which retrieves the value stored by $\downarrow$. As we shall see, this retrieval operator is the missing link which allows us to give a complete treatment of local hybrid logics involving $\downarrow$ — not only can we handle the minimal logic of $\mathrm{ML}{+}\downarrow{+}@$ rather straightforwardly, we automatically get completeness results for a wide class of extended logics; this includes results for many frame properties that are not modally definable, such as discreteness.

These completeness results and the model constructions on which they are based are the *technical* core of the paper, but to close the papers we need to change gears — there is an important *conceptual* point that needs to 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.* To be concrete, for most of this paper we work with atomic formulas that allow us, in a sense, to 'name' states; that is, we will be combining termlike information with arbitrary information. But why stop there? For example, why not introduce atomic formulas that range over intervals, or paths, or events? After all, such entities are needed in many kinds of temporal reasoning. So, to conclude the paper we briefly discuss a hybrid language for working with paths, indicating both the promising and problematic aspects of the extension.

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. Two remarks are in order. We call the elements of S 'states' because this seems a reasonably neutral term that covers a wide range of temporal entities. The reader is free to think of 'states' as instants of time, as intervals, or as time-stamped events; for the purposes of the present paper, the choice between these interpretations is largely irrelevant. What about the binary relation R? As our interest is temporal logic, we will be particularly interested in *strictly partially ordered models*; that is, those models where R is both *transitive* and *irreflexive*. However it is important to be able to handle a wide range of binary relations (for example, in some applications we might want to reason about a step-by-step transition graph that gives rise to a flow of events, rather than the flow itself) so it is also useful to think about models in which no restrictions are placed on R.

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 $\mathcal{M} \models \varphi$. In what follows, we call the set of formulas that are valid on all strictly partially ordered models the *minimal temporal logic*, and the set of formulas valid on all models the *minimal logic*.

Perhaps the key intuition to note about the satisfaction definition is its locality: formulas are evaluated *inside* models at some particular state (called the *current state*), and the function of the $\Box$ and $\Diamond$ operators is to scan the states accessible from the current state via the precedence relation R. This locality intuition is arguably *the* central intuition underlying the 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 our temporal intuitions and language reflect this internal perspective. Accordingly, it seemed to him that modal approaches to temporal reasoning were likely to be the most revealing.[2]

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 ML-generated by s is the smallest submodel $\mathcal{M}^s$ of $\mathcal{M}$ that contains s and is R-closed. (That is, the submodel ML-generated by s contains just those states of $\mathcal{M}$ that are accessible from s by a finite number of transitions along R. Note that in strict partial orders, every state in the ML-generated submodel distinct from s is accessible from s in one step.) 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 the work that follows, we shall use preservation under generated submodels as one of our criterion for judging hybrid languages. We are interested in developing *local* hybrid 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 $\perp$) or to shed some Boolean expressivity.

But for many other applications, ML is too *weak*. For example, if we are interested in natural language semantics we need to deal with sentences that talk about the past, not just sentences that talk about the future. So for this application it is natural to add the modal operator H defined as follows:

$$\mathcal{M}, s \models H\varphi \quad \text{iff} \quad \forall s'(s'Rs \Rightarrow \ \mathcal{M}, s' \models \varphi)$$

That is, H means *at all past states*, or more mnemonically, it is always *has* been the case. We then define $P\varphi$ to be $\neg H \neg \varphi$; clearly P means *at some past state*. When ML is enriched in this way we obtain the language of Priorean Tense Logic, or TL. Incidentally, when these backward scanning operators are added to ML it is traditional to use the notation G for $\Box$ (read G as it is always *going* to be the case) and F for $\Diamond$ (read F as sometime in the *future*). Backward scanning operators may be unfamiliar to some readers; they are not widely used either in reasoning about program execution, or in the AI literature on planning. Nonetheless, TL is a widely studied — and indeed, very beautiful — language. It too is local: the submodel *TL-generated* by a state s consists of all states reachable from s by any finite sequence of forward *and backward* transitions along R, and satisfaction of TL formulas is preserved under the formation of such submodels. For the time being, apart from occasional asides we will pretty much ignore TL; but when we return to it in Section 5, it will provide an important clue to the logical structure of hybridization.

[2]Probably the best introduction to Prior's views is Prior (1967).

Other weaknesses of ML are more subtle. 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 one of our fundamental constraints on temporal precedence simply isn't reflected in this language. Moreover, consider the definition of 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 ML-generated submodels) and it has proved a valuable tool for temporal reasoning in computer science (indeed, computer scientists usually treat *Until* as the fundamental temporal modality).[3]

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. Incidentally, neither irreflexivity nor the *Until* operator are definable in TL either. The appropriate notion of bisimulation for

[3] The *Until* operator has a backward looking counterpart, namely the *Since* operator defined by

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

Satisfaction of *Since* formulas is preserved under TL-generated submodels. Like the backward looking Priorean operators, the *Since* operator is not as widely used as its forward-looking sibling.

the TL is simply that given above together with two backward-looking analogs of items 2 and 3, and it is easy to see that the counterexample just given work with TL-bisimulations as well.

Thus, ML (and indeed, TL) 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 are interested in finding hybrid languages in which the central ideas of modal and classical proof systems can be clearly combined. Indeed, we would like to exhibit a certain '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 (or indeed, TL) 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 kind which we call *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). In this section we introduce the $\forall$ and $\downarrow$ binders (which as far as we are aware are the only hybrid binders that have previously been discussed). In the following section we will introduce a new binder: $\Downarrow^1$.

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 hybrid 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. However, as experience with classical logic is a reliable guide, and as the relevant definitions may be found in Blackburn and Tzakova (1998), 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 terms: 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 $V : \mathrm{PROP} \cup \mathrm{NOM} \longrightarrow Pow(S)$. A model is called* standard *iff for all nominals $i \in \mathrm{NOM}$, $V(i)$ is a singleton subset of S.*

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

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

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 $\mathcal{M} \models \varphi$. For either of these languages (and indeed, for any of the languages we shall consider later) we call the set of formulas valid on all standard strictly partially ordered models the *minimal temporal logic*, and the set of formulas valid on all models the *minimal logic*. 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 $s \in S$, $(S, R, V), g, s \models \varphi$. Frame validity will be important in Section 6 when we examine extended completeness results for hybrid languages.

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* $g' \overset{x}{\sim} g$ *and* $g'(x) = g(y)$.

2. $\mathcal{M}, g, s \models \varphi[i/x]$ *iff* $\mathcal{M}, g', s \models \varphi$, *where* $g' \overset{x}{\sim} g$ *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 it we define $\exists x\varphi$ to be the dual binder $\neg\forall x\neg\varphi$, then:

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

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 + $\forall$ has a simple axiomatization that can be proved complete reasonably straightforwardly. We won't present the full axiomatization here (for that, see Blackburn and Tzakova (1998)), but for comparison with our later work on ML + $\downarrow$ + $\Downarrow^1$ it will be useful to note two of its components.

First, ML + $\forall$ validates two of the schemas standardly used to axiomatize first-order logic ($\mathbf{v}$ and $\mathbf{s}$ are used as metavariables over state variables and state symbols respectively):

Q1 $\quad \forall\mathbf{v}(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \forall\mathbf{v}\psi)$

Q2 $\quad \forall\mathbf{v}\varphi \rightarrow \varphi[\mathbf{s}/\mathbf{v}]$

(In *Q1*, φ cannot contain free occurrences of $\mathbf{v}$; and in *Q2*, $\mathbf{s}$ must be substitutable for $\mathbf{v}$ in φ.) As ML + $\forall$ also validates the classical rule of generalization (if φ is provable then so is $\forall x\varphi$) it is clear that it has a full classical core; this makes much of the completeness proof straightforward.

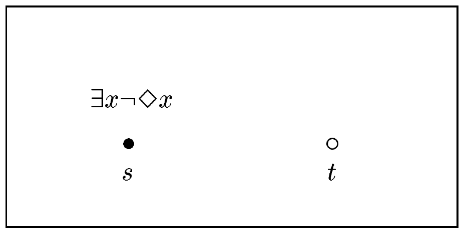
Second, ML + $\forall$ also validates the Barcan schema:

Barcan $\qquad \forall\mathbf{v}\Box\varphi \rightarrow \Box\forall\mathbf{v}\varphi$

Now, the *Barcan* schema is familiar from first-order modal logic (that is, the study of languages containing quantifiers that range over the individuals in some underlying collection of first-order models). In first-order modal logic, *Barcan* is a convenient (if rather dubious) principle. Its status is ML + $\forall$, however, is beyond dispute: all instances are valid in all standard models as the reader can easily check.

Barcan is important because it builds a robust bridge between the modalities and the binders. As it allows us to permute the relative scopes of $\Box$ and $\forall$, it is straightforward to combine the key techniques of modal and classical completeness proofs. From an axiomatic perspective, the validity of *Barcan* is what most sharply distinguishes $\mathrm{ML} + \forall$ from $\mathrm{ML} + \downarrow$.

All in all, $\mathrm{ML}+\forall$ is a lovely language. There's just one problem: it isn't local. To see that satisfaction 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 the point t, even though s is completely disconnected from it.

If you want a strong hybrid language and are not interested in maintaining locality, then $\mathrm{ML} + \forall$ is probably an excellent choice. Indeed, you may wish to consider working with a hybrid language even *less* local, namely $\mathrm{ML} + \forall$ enriched with the *universal modality* A.[4] 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 $\forall$ work together extremely smoothly, making elegant axiomatizations possible. But while such rich systems are interesting, they are far removed from the local temporal languages we wish to develop.

The $\downarrow$ binder

If one is interested in local hybrid languages, the $\downarrow$ binder is the most natural starting point. Quite simply, $\downarrow$ binds a variable to the current state; it creates a label for the here-and-now. Let's look at it more closely.[5]

First, note that $\downarrow$ 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

[4] 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 both $\forall$ and A.

[5] Incidentally, while $\downarrow$ 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.

too. To put it another way, we are free to regard $\downarrow$ 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 $\mathrm{ML}+\forall$; we can define it either as $\forall x(x \rightarrow \varphi)$ or $\exists x(x \wedge \varphi)$, thus $\mathrm{ML}+\downarrow$ is a fragment of $\mathrm{ML}+\forall$. It's quite an interesting fragment. For a start, sentences of $\mathrm{ML}+\downarrow$ *are* preserved under the formation of ML-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; $\mathrm{ML}+\downarrow$ 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, $\mathrm{ML}+\downarrow$ has two drawbacks. First, for many purposes it simply isn't expressive enough. Second, the only known way of obtaining completeness results for this language relies on the use of a rather complex rule of proof. Let us examine both issues more closely.

Although adding $\downarrow$ increases the expressive power, *Until* still isn't definable. To see why, we will have to make use *quasi-injective bisimulations*, as 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 ML-generated by s' and s' is not in the submodel ML-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:*

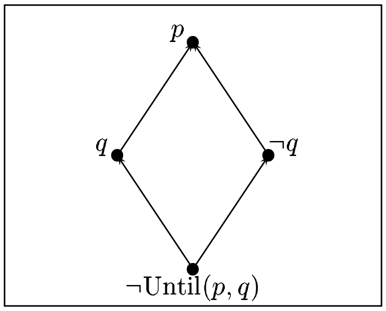
1. *For all states s_1, $s_1{}'$ in $\mathcal{M}_1$, and s_2 in $\mathcal{M}_2$, if $s_1 Z s_2$ and $s_1{}' Z s_2$, and $s_1 \neq s_1{}'$ then s_1 and $s_1{}'$ are mutually inaccessible, and*

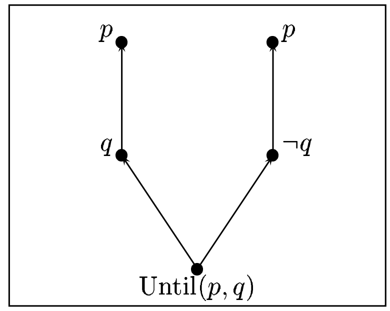
2. *For all states s_2, $s_2{}'$ in $\mathcal{M}_2$, and s_1 in $\mathcal{M}_1$, if $s_1 Z s_2$ and $s_1 Z s_2{}'$, and $s_2 \neq s_2{}'$ then s_2 and $s_2{}'$ are mutually inaccessible.*

Now, $\mathrm{ML}+\downarrow$ 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* $\mathrm{ML}+\downarrow$, $\mathcal{M}_1, s_1 \models \varphi$ *iff* $\mathcal{M}_2, s_2 \models \varphi$.

We can use this result to show that no sentence of $\mathrm{ML}+\downarrow$ 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 $\mathrm{ML}+\downarrow$ 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 $\mathrm{ML} + \downarrow$ 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. Hence *Until*(p, q) is not expressible.

Incidentally, the previous counterexample does *not* apply to TL+$\downarrow$. Although the relation between the two models is a TL-bisimulation, the two topmost point in the right-hand model clearly *do* belong to the same TL-generated submodel: we can move backwards from either to the root, and then forward to the other. This is no accident: as we shall see in Section 5, TL+$\downarrow$ *is* capable of defining *Until* (and *Since*).

There is a further difficulty with $\mathrm{ML} + \downarrow$: even over strictly partially ordered models, there is no obvious way to provide a complete axiomatization without resorting to a fairly complex rule of proof. For a start, $\downarrow x\Box\varphi \rightarrow \Box{\downarrow}x\varphi$, the Barcan analog in $\mathrm{ML} + \downarrow$ is unsound over strict partial orders. (For example, consider $\downarrow x\Box\neg x \rightarrow \Box{\downarrow}x\neg x$; in any strictly partially ordered model, this is false at any state that has a successor.) Moreover, at present no alternative axioms are known which build a suitable bridge between the modalities and the $\downarrow$ binder. Now, as is shown in Blackburn and Tzakova (1997), there is a way round the problem: by making use of the *COV* rule (see Gargov, Passy and Tinchev (1987), Passy and Tinchev (1991), Gargov and Goranko (1993)) it is possible to prove a completeness result.[6] Unfortunately, the *COV* rule is rather complex. Let's take a brief look at it.

Suppose we are working with ML+$\downarrow$. Let # be some symbol not belonging to this language. Then we define the set of $\Box$-forms as follows: (1) # is a $\Box$-form, (2) if L is a $\Box$-form and φ is an ML+$\downarrow$-formula then $\varphi \rightarrow L$ and $\Box L$ are $\Box$-forms, and (3) nothing else is a $\Box$-form. Note that every $\Box$-form L has exactly one occurrence of the symbol #. We use $L(\psi)$ to denote the formula obtained from L by replacing the unique occurrence of # by a formula ψ. We can now define the *COV* rule. For every $\Box$-form L, and every state symbol $\mathbf{s}$

[6] The earliest work on axiomatic systems for $\downarrow$ seems to be that of 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 $\forall x\varphi := {\downarrow}yA{\downarrow}xA(y \rightarrow \varphi)$, thus Goranko's language has full first-order expressive power.

not occurring in L, we have:

$$\text{If } \vdash L(\neg \mathbf{s}) \text{ then } \vdash L(\bot)$$

Roughly speaking, this rule is useful because it permits us keep track of which nominals we have substituted where (for full details, see Blackburn and Tzakova (1998)). Unfortunately it does so in a rather brute-force way: paths through the model are encoded using explicit nestings of modalities; we would prefer a simpler approach.[7]

Summing up, no previously studied hybrid system meets our three wish-list criteria. The $\forall$ binder is interesting and elegant — but to adopt it is to abandon locality. Essentially the $\forall$ binder is the key to reproducing first-order logic in a modal setting; this is an interesting project, both philosophically and methodologically (see Prior and Fine (1977) and Passy and Tinchev (1991) for arguments in its support), but it is not the project that interests us here.

The $\downarrow$ binder is far more promising — binding to the current state is such an intrinsically modal idea that it is natural to place this binder center stage. But can we overcome its expressive weakness? And are there natural ways to avoid dependence on *COV* or other complex rules of proof? The answer to both these questions is "Yes". In fact, we shall explore three ways of realizing these goals: adding $\Downarrow^1$, a universal quantifier over accessible states; changing the underlying language from ML to TL; and finally, the smoothest solution of all: adding a retrieval operator @ to match the action of $\downarrow$.

4 The $\Downarrow^1$ binder

Question: given that the current state is the most modally significant state, what states are next in importance? Answer: the states *accessible* from the current state, of course! This observation prompts the definition of the $\Downarrow^1$ binder:

$$\mathcal{M}, g, s \models \Downarrow^1_x \varphi \text{ iff } \mathcal{M}, g', s \models \varphi, \text{ for all } g' \stackrel{x}{\sim} g \text{ such that } sRg'(x).$$

That is, the $\Downarrow^1$ binder is simply a universal quantifier over *accessible* states.[8] Note that if we define $\downarrow^1_x \varphi$ to be the dual binder $\neg \Downarrow^1_x \neg \varphi$, then:

$$\mathcal{M}, g, s \models \downarrow^1_x \varphi \quad \text{iff} \quad \mathcal{M}, g', s \models \varphi, \text{ for some } g' \stackrel{x}{\sim} g \text{ such that } sRg'(x).$$

Thus $\downarrow^1_x \varphi$ is an existential quantifier across accessible states. The reader should think of $\Downarrow^1$ and $\downarrow^1$ as a pair of binders that "match" the actions of $\Box$ and $\Diamond$ respectively. Observe that both $\Downarrow^1$ and $\downarrow^1$ are definable in ML + $\forall$, for $\Downarrow^1_x \varphi$ is simply $\forall x(\Diamond x \to \varphi)$ and $\downarrow^1_x \varphi$ is $\exists x(\Diamond x \wedge \varphi)$. Further, observe that $\Downarrow^1$ is an intrinsically *local* binder; sentences built with this operator are preserved under

[7] The *COV* rule would be more palatable if it could be shown to be admissible over some simple axiomatic base; but we have no such result. The *COV* rule could be replaced by a modal version of the *Paste* rule used in our discussion of TL+$\downarrow$ in Section 5. But although we will show that the *Paste* rule is admissible in TL+$\downarrow$, we do not know whether (the modal version of) *Paste* is admissible in in ML+$\downarrow$.

[8] We won't dwell on the syntactic preliminaries. Such concepts such as 'free', 'bound', 'sentence', and so on are defined in the expected way, and a Substitution Lemma holds just as it does for the $\forall$ and $\downarrow$ binders (recall Lemma 2).

the formation of generated submodels. Moreover it is strong enough to define *Until*; the required definition is simply the one given earlier for ML+$\forall$ with $\downarrow^1$ replacing $\exists$:

$$Until(\varphi, \psi) \ := \ \downarrow^1_y(\Diamond(y \wedge \varphi) \wedge \Box(\Diamond y \rightarrow \psi)).$$

So what happens when we add the $\Downarrow^1$ binder to ML+$\downarrow$? As we shall see — *if we restrict our attention to transitive models* — the $\Downarrow^1$ and $\downarrow$ binders work smoothly together and it is possible to give a straightforward axiomatization of the minimal temporal logic. The remainder of this section is devoted to establishing this.

The axiomatization

Our axiomatization is called $\mathcal{H}[\downarrow, \Downarrow^1]$*(I4)*; the $\mathcal{H}[\downarrow, \Downarrow^1]$ indicates the hybrid language we are working in, the *(I4)* that we are dealing with the logic of irreflexive and transitive models. Our goal is to show that $\mathcal{H}[\downarrow, \Downarrow^1]$*(I4)* consists of precisely the ML $+ \downarrow + \Downarrow^1$ formulas valid on strict partial orders.

$\mathcal{H}[\downarrow, \Downarrow^1]$*(I4)* is an extension of *K4*, the modal logic of strict partial orders. Recall that *K4* is the smallest set of formulas containing all propositional tautologies, all instances of $\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$, and all instances of $\Box\varphi \rightarrow \Box\Box\varphi$, 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 $\Box\varphi$).

Suppose we have fixed a (countable) language $\mathcal{L}$ of ML $+ \downarrow + \Downarrow^1$. By $\mathcal{H}[\downarrow, \Downarrow^1]$*(I4)* we mean the smallest set of $\mathcal{L}$ formulas that contains all instances of the *K4* axiom schemas, and all instances of the axiom schemas listed below, that is closed under the following rules of proof: modus ponens, necessitation, and *state variable localization for both* $\downarrow$ *and* $\Downarrow^1$ (that is, if φ is provable then so are $\downarrow x\varphi$ and $\Downarrow^1_x\varphi$, for all state variables x). We assume the usual notion of formal proof. For the remainder of this section $\vdash \varphi$ means that φ is provable in $\mathcal{H}[\downarrow, \Downarrow^1](I4)$.

Our axiom schemas fall naturally into four groups. The first group reflects the basic quantificational powers of $\downarrow$ and $\Downarrow^1$.

Q1	$\downarrow\mathbf{v}(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \downarrow\mathbf{v}\psi)$	$\Downarrow^1_{\mathbf{v}}(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \Downarrow^1_{\mathbf{v}}\psi)$
Q2	$\downarrow\mathbf{v}\varphi \rightarrow (\mathbf{s} \rightarrow \varphi[\mathbf{s}/\mathbf{v}])$	$\Downarrow^1_{\mathbf{v}}\varphi \rightarrow (\Diamond\mathbf{s} \rightarrow \varphi[\mathbf{s}/\mathbf{v}])$
Q3	$\downarrow\mathbf{v}(\mathbf{v} \rightarrow \varphi) \rightarrow \downarrow\mathbf{v}\varphi$	$\Downarrow^1_{\mathbf{v}}(\Diamond\mathbf{v} \rightarrow \varphi) \rightarrow \Downarrow^1_{\mathbf{v}}\varphi$
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 (and indeed, of the schemas given earlier for ML+$\forall$). The major difference is that the present version of *Q2* only lets us substitute state symbols for binders when the appropriate conditions are fulfilled: either $\mathbf{s}$ must be true in the current state (for the $\downarrow$ binder) or it must be true at a successor state (for the $\Downarrow^1$ binder). These restrictions motivate the introduction of *Q3*. *Q3* allows us to eliminate conditional occurrences of state variables (usually occurrences introduced by

Q2 that have subsequently become bound; we shall see many examples of this). Finally, $\downarrow$ has one property that $\Downarrow^1$ lacks: it is self-dual; the *Self-Dual* axiom, which is included only for $\downarrow$, reflects this. Summing up: both $\downarrow$ and $\Downarrow^1$ legitimate a *restricted* version of classical quantificational reasoning, and when proving completeness we shall have to ensure that we can work our way around these restrictions.

The second group consists of *Barcan* analogs:

$$\textit{Barcan}_{10} \qquad \Downarrow^1_{\mathbf{v}}\Box\varphi \to \Box{\downarrow}\mathbf{v}\varphi$$

$$\textit{Barcan}_{11} \qquad \Downarrow^1_{\mathbf{v}}\Box\varphi \to \Box\Downarrow^1_{\mathbf{v}}\varphi$$

These two permutation principles play a crucial role in the completeness proof. Note well: the soundness of $\textit{Barcan}_{11}$ depends on our transitivity assumption!

The third group consists of a single schema, and reflects the fact that state variables are essentially labels which uniquely identify nodes. It's our *model theory of labeling*; in a sense, it is a modal counterpart of the first-order theory of equality.

$$\textit{Nom} \qquad \Downarrow^1_{\mathbf{v}}[\Diamond(\mathbf{v}\wedge\varphi) \to \Box(\mathbf{v}\to\varphi)]$$

The fourth group consists of a single schema reflecting the fact that we are working with irreflexive models.

$$I \qquad {\downarrow}\mathbf{v}\Box\neg\mathbf{v}$$

In many ways, $\mathcal{H}[\downarrow,\Downarrow^1](I4)$ behaves in ways familiar from classical logic. For example, the *normality schemas* for $\downarrow$ and $\Downarrow^1$, *replacement of equivalents*, and α*-conversion* for the state variables, can all be straightforwardly established.

Lemma 4 (Normality) *For all formulas* φ *and* ψ *we have:*

$$\vdash {\downarrow}x(\varphi\to\psi)\to({\downarrow}x\varphi\to{\downarrow}x\psi) \qquad \vdash \Downarrow^1_x(\varphi\to\psi)\to(\Downarrow^1_x\varphi\to\Downarrow^1_x\psi).$$

Proof. Note that $\Downarrow^1_x(\varphi\to\psi)\to(\Diamond x\to(\varphi\to\psi))$ is an instance of *Q2*, as is $\Downarrow^1_x\varphi\to(\Diamond x\to\varphi)$. Hence $\vdash(\Downarrow^1_x(\varphi\to\psi)\wedge\Downarrow^1_x\varphi))\to(\Diamond x\to\psi)$. Use localization to prefix this formula with $\Downarrow^1_x$, and then *Q1* to distribute $\Downarrow^1_x$ over the main implication to get $\vdash(\Downarrow^1_x(\varphi\to\psi)\wedge\Downarrow^1_x\varphi)\to\Downarrow^1_x(\Diamond x\to\psi)$. Note that $\Downarrow^1_x(\Diamond x\to\psi)\to\Downarrow^1_x\psi$ is an instance of *Q3*, so we can simplify the consequent and so obtain the result. (Using *Q3* to simplify the conditionals produced by applications of *Q2* is fairly common in $\mathcal{H}[\downarrow,\Downarrow^1](I4)$ proofs.) The proof for $\downarrow$ is analogous. $\dashv$

Lemma 5 (Replacement of equivalents) *If* $\vdash\xi\leftrightarrow\chi$*, and the formula* $\varphi(\xi)$ *differs from the formula* $\varphi(\chi)$ *only in having* ξ *at zero or more places where* $\varphi(\chi)$ *has* χ*, then* $\vdash\varphi(\xi)\leftrightarrow\varphi(\chi)$.

Proof. Because we have all instances of the normality schemas for $\Box$, $\downarrow$, and $\Downarrow^1$, the results follows in the standard way. $\dashv$

Lemma 6 (α-conversion) *If* y *is substitutable for* x *in* φ *and* φ *has no free occurrences of* y*, then* $\vdash{\downarrow}x\varphi\leftrightarrow{\downarrow}y\varphi[y/x]$ *and* $\vdash\Downarrow^1_x\varphi\leftrightarrow\Downarrow^1_y\varphi[y/x]$.

Proof. Straightforward. Use *Q1*, *Q2*, *Q3* and the localization rules. $\dashv$

Moreover, just as we can "generalize on parameters" in first-order logic, we can "localize on nominals" in our hybrid language. For any formula φ, any nominal i, and any variable z that does *not* occur in φ, let $\varphi[z/i]$ denote the result of replacing all occurrences of i in φ by z. Then we have:

Lemma 7 (Localization on nominals) *If $\vdash \varphi$, then there is a state variable y that does not occur in φ such that $\vdash \downarrow y\varphi[y/i]$ and $\vdash \Downarrow^1_y\varphi[y/i]$. Moreover, for any variable x, we can choose y to be distinct from x.*

Proof. If i does not occur in φ, $\varphi[y/i]$ is identical to φ, hence as φ is provable, so are $\downarrow y\varphi[y/i]$ and $\Downarrow^1_y\varphi[y/i]$ for any choice of y. So suppose i does occur in φ. By assumption we have a proof of φ. Choose any variable y that does not occur in this proof, and replace every occurrence of i in the proof by y. It follows by induction on the length of proofs that this new sequence is a proof of $\varphi[y/i]$ (all that needs to be observed is that there are no axioms or rules that let us do something to a nominal that we cannot do to state variable). Using localization to prefix $\downarrow y$ and $\Downarrow^1_y$ to the last item in this proof yields proofs of $\downarrow y\varphi[y/x]$ and $\Downarrow^1_y\varphi[y/x]$ respectively. Moreover, given any variable x, we can always choose y to be distinct from x, for there are infinitely many variables not occurring in a given proof. $\dashv$

So far, so good — but we really do need to be careful! The $\Downarrow^1$ operator has a distinctly modal flavor, and assuming 'obvious' classical principles can be dangerous. A simple example concerns vacuous occurrences of binders. In classical logic, we can simply add or discard vacuous quantifiers. What about our hybrid language? For the most part, the expected vacuity principles hold:

Lemma 8 (Vacuity principles) *Let φ be any formula containing no free occurrences of x. Then: $\vdash \varphi \to \downarrow x\varphi$, $\vdash \varphi \to \Downarrow^1_x\varphi$ and $\vdash \downarrow x\varphi \to \varphi$.*

Proof. The fact that $\vdash \varphi \to \varphi$ immediately yields (via localization and *Q1*) both $\vdash \varphi \to \downarrow x\varphi$ and $\vdash \varphi \to \Downarrow^1_x\varphi$. Using *Self-Dual*, we then obtain $\vdash \downarrow x\varphi \to \varphi$. $\dashv$

But what about $\Downarrow^1_x\varphi \to \varphi$? In fact, this is *not* provable, and we don't want it be, for it is not valid! To see this, consider $\Downarrow^1_x \bot \to \bot$. In any model, the antecedent will be true in any state that has no R-successors — but the consequent, of course, is false. It may be helpful to consider the contraposed and dualised form of this formula: $\top \to \downarrow^1_x\top$. The consequent demands the existence of an R-successor (in effect, the $\downarrow^1$ operator contains a 'hidden $\Diamond$') but the trivially true antecedent obviously doesn't guarantee that a successor exists.

A more significant example, which will affect our later work, is the following: when φ contains no free occurrences of x, then we have that $\vdash (\varphi \to \downarrow x\psi) \to \downarrow x(\varphi \to \psi)$, but we certainly do *not* have $\vdash (\varphi \to \downarrow^1_x\psi) \to \downarrow^1_x(\varphi \to \psi)$ (consider what happens when φ and ψ are $\bot$ and $\top$ respectively).

With the help of the Substitution Lemma it is routine to show that $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ is sound with respect to the class of all strictly partially ordered models, so let's turn to the question of its completeness.

Completeness

First some preliminaries. A set of formulas Σ is *consistent* iff for all formulas σ, if σ is a conjunction of (finitely many) formulas from Σ, then $\nvdash \sigma \rightarrow \bot$; otherwise Σ is *inconsistent*. A set of $\mathcal{L}$-formulas Σ is a *maximal consistent set* in $\mathcal{L}$ (an $\mathcal{L}$-MCS) iff it is consistent, and any set of $\mathcal{L}$-formulas that properly extends it is inconsistent. As $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ is an extension of classical propositional logic, *Lindenbaum's Lemma* holds: any consistent set of $\mathcal{L}$-formulas can be extended to an $\mathcal{L}$-MCS.

We want to prove the following completeness theorem: *any consistent set of formulas can be satisfied in a strictly partially ordered standard model with respect to a standard assignment function*. So, given such a set, how do we build a model? The natural approach, of course, is to blend the modal technique of *canonical models*, with the first-order technique of Henkin proofs via *witnessed sets*, and this is what we shall do.

Definition 9 (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.*

Definition 10 (Witnessed sets) *An MCS Γ is called* $\downarrow$-witnessed *iff for all formulas $\downarrow x\varphi \in \Gamma$, there is some nominal i such that $(\varphi[i/x] \wedge i) \in \Gamma$. It is called* $\downarrow^1$-witnessed *iff for all formulas $\downarrow^1_x\varphi \in \Gamma$ there is a nominal i such that $(\varphi[i/x] \wedge \Diamond i) \in \Gamma$. It is called* witnessed *iff it is both $\downarrow$- and $\downarrow^1$-witnessed.*

Lemma 11 (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 witnessed $\mathcal{L}^+$-MCS.*

Proof. Enumerate the nominals that are in $\mathcal{L}^+$ but not in $\mathcal{L}$. Let Φ be a consistent set of $\mathcal{L}$ formulas. We now inductively extend Φ to a witnessed $\mathcal{L}^+$-MCS. Let $\Theta^0 = \Phi$. Note that this set contains no nominals from the enumeration. Define Θ^{n+1} as follows. If $\Theta^n \cup \{\varphi_n\}$ is *inconsistent*, then $\Theta^{n+1} = \Theta^n$. Otherwise:

1. $\Theta^{n+1} = \Theta^n \cup \{\varphi_n\}$, if φ_n is not of the form $\downarrow x\psi$ or $\downarrow^1_x\psi$.
2. $\Theta^{n+1} = \Theta^n \cup \{\varphi_n\} \cup \{\psi[i/x] \wedge i\}$, if φ_n is of the form $\downarrow x\psi$.
3. $\Theta^{n+1} = \Theta^n \cup \{\varphi_n\} \cup \{\psi[i/x] \wedge \Diamond i\}$, if φ_n is of the form $\downarrow^1_x\psi$.

(In clauses 2 and 3, i is the first nominal in the enumeration that has not been used in the definitions of Θ^m, for all $m \leq n$, and that is not in φ_n.)

Let $\Theta = \bigcup_{n\geq 0} \Theta^n$. Clearly this set is maximal and witnessed. To show that Θ is consistent, we need to prove that for all n, Θ^n is consistent. Case 1 is trivial; what about cases 2 and 3?

For case 3 we argue by contrapositive. Suppose that for $\varphi_n = \downarrow^1_x\psi$, $\Theta^n \cup \{\varphi_n\}$ is consistent while $\Theta^{n+1} = \Theta^n \cup \{\varphi_n\} \cup \{\psi[i/x] \wedge \Diamond i\}$ is not. Then there is a formula χ, which is a conjunction of a finite number of formulae in $\Theta^n \cup \{\varphi_n\}$, such that $\vdash \chi \to \neg(\psi[i/x] \wedge \Diamond i)$. We're going to localize on i. By Lemma 7 there is a state variable y distinct from x that does not occur in $\chi \to \neg(\psi[i/x] \wedge \Diamond i)$ such that:

$$\vdash \Downarrow^1_y(\chi \to \neg(\psi[i/x] \wedge \Diamond i))[y/i].$$

That is, we have that $\vdash \Downarrow^1_y(\chi[y/i] \to \neg(\psi[i/x][y/i] \wedge \Diamond i[y/i]))$. But i is a new nominal: it does not occur in χ, hence $\chi[y/i]$ is just χ; and it does not occur in ψ, hence $\psi[i/x][y/i]$ is $\psi[y/x]$. Thus the previous expression simplifies to:

$$\vdash \Downarrow^1_y(\chi \to \neg(\psi[y/x] \wedge \Diamond y)).$$

Hence by $Q1$ we get $\vdash \chi \to \Downarrow^1_y \neg(\psi[y/x] \wedge \Diamond y)$. By replacement of equivalents this is just $\vdash \chi \to \Downarrow^1_y(\Diamond y \to \neg\psi[y/x])$, so we can use $Q3$ to simplify the consequent, thus obtaining $\vdash \chi \to \Downarrow^1_y \neg\psi[y/x]$. Now use α-conversion on $\Downarrow^1_y \neg\psi[y/x]$ to replace y by x (this is possible by Lemma 6), thus we have $\vdash \chi \to \Downarrow^1_x \neg\psi$, and hence $\Theta^n \cup \{\varphi_n\} \vdash \Downarrow^1_x \neg\psi$. As $\varphi_n = \neg\Downarrow^1_x \neg\psi$ this contradicts the consistency of $\Theta^n \cup \{\varphi_n\}$.

Note that this argument is essentially *classical*: it exploits the quantificational powers of $\Downarrow^1$ in a manner reminiscent of Henkin proofs. Case 2 is proved analogously. We conclude that Θ is consistent. $\dashv$

Definition 12 (Witnessed models) *Let Σ be a witnessed MCS in some countable language $\mathcal{L}$, let $\mathcal{M}^c = (S^c, R^c, V^c)$ be the canonical model in $\mathcal{L}$, let S^Σ be those states in S^c that belong to the submodel ML-generated by Σ, and let $Wit(S^c)$ be the set of all witnessed MCSs in S^c. Then the* witnessed model $\mathcal{M}^w$ *yielded by Σ is the triple (S^w, R^w, V^w), where $S^w = S^\Sigma \cap Wit(S^c)$, and R^w and V^w are the restrictions of R^c and V^c respectively to S^w.*

Lemma 13 *Let $\mathcal{M}^w$ be the witnessed model yielded by a witnessed MCS Σ. Then (1) R^w is transitive, (2) every MCS in S^w contains at least one nominal, (3) R^w is irreflexive, and (4) no state symbol occurs in more than one MCS in S^w.*

Proof. For (1), note it is a standard result that R^c is transitive (because all MCSs contain all instances of $\Box\varphi \to \Box\Box\varphi$, the transitivity axiom), hence as R^w is a subrelation of R^c it is transitive too.

For (2), note that $\vdash \downarrow xx$ (this follows easily from the fact that $\vdash x \to x$) so every MCS in S^w contains $\downarrow xx$. But as all MCSs in S^w are witnessed MCSs, they are $\downarrow$-witnessed, hence every MCS contains a witness for this formula. Such a witness is simply a nominal.

For (3), we have just observed that any MCS Γ contains some nominal, say i. But Γ also contains the axiom $\downarrow x\Box\neg x$. Hence, by *Q2*, Γ contains $\Box\neg i$ as well. Thus Γ is not R^c-related to itself (that is, R^c is irreflexive), and as R^w is a subrelation of R^c, R^w is irreflexive too.

For (4), suppose first that there are two distinct successors of Σ in $\mathcal{M}$, say Γ and Δ, that both contain the same state symbol s. Since Γ and Δ are distinct, there is a formula δ that distinguishes them; that is, a formula δ such that $(s \wedge \delta) \in \Gamma$ and $(s \wedge \neg\delta) \in \Delta$. It follows (using the fact that R^w is a restriction of R^c) that $\Diamond(s \wedge \delta) \in \Sigma$ and $\Diamond(s \wedge \neg\delta) \in \Sigma$. Let z be any variable that does not occur in δ and is distinct from s; then $\Downarrow^1_z(\Diamond(z \wedge \delta) \to \Box(z \to \delta))$ is an instance of the *Nom* axiom. As $\Diamond s \in \Sigma$, it follows by *Q2* that $\vdash \Diamond(s \wedge \delta) \to \Box(s \to \delta)$.

Thus, as $\Diamond(s \wedge \delta) \in \Sigma$, $\Box(s \to \delta) \in \Sigma$. But then, as $\Diamond(s \wedge \neg\delta) \in \Sigma$, $\Diamond(s \wedge \neg\delta \wedge \delta) \in \Sigma$, which is impossible. We conclude that no two distinct R^w-successors of Σ contain the same state symbol.

So next assume that some state symbol s is in both Σ and a successor Γ of Σ. As $s \in \Sigma$, using the irreflexivity axiom and *Q2* we have that $\Box\neg s \in \Sigma$. On the other hand, as s belongs to a successor of Σ, we also have that $\Diamond s \in \Sigma$; as Σ is consistent, this is impossible. Thus no state symbol occurs in both Σ and a successor of Σ, and we have shown that no state symbol occurs in more than one MCS in S^w. $\dashv$

Note, however, that there is no guarantee that every state symbol occurs in at least one witnessed MCS in a witnessed model. For example, the set $\{\neg i, \Box\neg i\}$ is consistent and thus can be extended to a witnessed MCS. But any witnessed MCS extending $\{\neg i, \Box\neg i\}$ yields a witnessed model in which i does not occur in any MCS, and in such models the natural valuation V^w is not a standard valuation as $V^w(i) = \{\}$. As we need to build a standard model (and a standard assignment) we need to ensure that all state symbols denote some state, so we shall glue a new root node $*$ onto our witnessed models. This new node will serve as a denotation for any symbol not contained in MCSs. More precisely:

Definition 14 (Completed models and assignments) *Given a witnessed MCS Σ, let $\mathcal{M}^w = (S^w, R^w, V^w)$ be the witnessed model yielded by Σ. The* completion *of $\mathcal{M}^w$ is a triple $\mathcal{M}^+ = (S^+, R^+, V^+)$, where $S^+ = S^w \cup \{*\}$ (where $*$ is an entity that is not an MCS); $R^+ = R^w \cup \{(*, \Gamma) \mid \Gamma \in S^w\}$; for all propositional variables p, $V^+(p) = V^w(p)$; and for all nominals i, $V^+(i) = \{\Gamma \in \mathcal{M}^w \mid i \in \Gamma\}$ if this set is non-empty, and $V^+(i) = \{*\}$ otherwise.*

If $\mathcal{M}^+ = (S^+, R^+, V^+)$ is a completion of a witnessed model $\mathcal{M}^w$, then the completed assignment g^+ on $\mathcal{M}^+$ is defined as follows: for all variables x, $g^+(x) = \{\Gamma \in \mathcal{M}^w \mid x \in \Gamma\}$ if this set is non-empty, and $g(x) = \{*\}$ otherwise.

Clearly (with the help of Lemma 13) completions of witnessed models are standard models, completed assignments are standard assignments, and R^+ is a strict partial ordering. Thus all theorems of the logic $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ are true in completed models with respect to the relevant complete assignment. Completions of witnessed models are well-behaved structures, and we shall use them to prove our Truth Lemma.

But we are not yet ready to do this. First we have to establish a crucial fact: that (completions of) witnessed models contain all the information required to cope with the *modalities*. That is, we need an Existence Lemma which tells us that if $\Diamond\varphi$ belongs to a *witnessed MCS* Δ, then Δ has a R^w-successor Γ containing φ. This is not obvious. The ordinary Existence Lemma for modal logic tells us that Δ has a R^c-successor Γ that contains φ — but it does not guaranteed that Γ is a *witnessed* MCS, and hence we have no idea whether or not Γ is an R^w-successor of Δ. To put it another way, we formed the witnessed model by throwing away the non-witnessed MCSs present in the canonical model. How do we know that we didn't throw away all the φ-containing R^c-successors of Δ?

The obvious approach to this problem is to tackle it head on by inductively constructing the required witnessed successors; this is the approach taken in Blackburn and Tzakova (1998) for $\mathrm{ML} + \forall$. However there is no obvious way to adapt this argument to $\mathrm{ML}+\downarrow+\Downarrow^1$ because of the following technical problem: as we have already observed, $(\varphi \rightarrow \downarrow^1_x\psi) \rightarrow \downarrow^1_x(\varphi \rightarrow \psi)$ is *not* a theorem of $\mathcal{H}[\downarrow, \Downarrow^1](I4)$i for it is not sound. However $(\varphi \rightarrow \exists x\psi) \rightarrow \exists x(\varphi \rightarrow \psi)$, the corresponding principle in $\mathrm{ML}+\forall$, *is* sound and in fact plays an (easily overlooked, but important) role in the completeness proof for $\mathrm{ML}+\forall$. We might sum up the situation as follows: although $\mathrm{ML}+\downarrow+\Downarrow^1$ is rich enough to support *Barcan* analogs, the restrictions on its classical component seem to make the obvious route to completeness difficult.[9]

But there is an elegant solution — and a modally flavored one at that. Let us call an MCS *named* iff it contains at least one nominal, and call a nominal in an MCS a *name* for that MCS. We are going to ask one simple, but important, question: *could we have thrown away any named* MCS? As we shall show, the answer is "no": that is, all named successors of witnessed MCSs are witnessed. Why is this useful? Because it means that we are free to think in terms of named MCSs, rather than just in terms of witnessing — and, as will become clear, because we have the *Nom* schema at our disposal, named MCSs are very easy to work with. In particular, this approach leads to a relatively straightforward proof of the required Existence Lemma.

We shall need the following syntactical preliminary:

Lemma 15 *For any formulas φ and ψ such that φ contains no free occurrences of x we have that $\vdash (\varphi \wedge \downarrow^1_x\psi) \rightarrow \downarrow^1_x(\varphi \wedge \psi)$.*

Proof. Because of our restriction on φ, $\Downarrow^1_x(\varphi \rightarrow \neg\psi) \rightarrow (\varphi \rightarrow \Downarrow^1_x\neg\psi)$ is an instance of *Q1*, hence $\vdash \Downarrow^1_x(\varphi \rightarrow \neg\psi) \rightarrow (\neg\varphi \vee \Downarrow^1_x\neg\psi)$. Now, by replacement of equivalents we have $\vdash \Downarrow^1_x(\neg\varphi \vee \neg\psi) \leftrightarrow \Downarrow^1_x(\varphi \rightarrow \neg\psi)$, hence it follows that $\vdash \Downarrow^1_x(\neg\varphi \vee \neg\psi) \rightarrow (\neg\varphi \vee \Downarrow^1_x\neg\psi)$. Taking the contrapositive yields $\vdash \neg(\neg\varphi \vee \Downarrow^1_x\neg\psi) \rightarrow \neg\Downarrow^1_x(\neg\varphi \vee \neg\psi)$ and the required result follows. $\dashv$

Now for the key observation — and the place where we cash in our transitivity assumption.

Lemma 16 *Let Σ be a witnessed MCS and $\mathcal{M}^\Sigma$ the submodel of the canonical model it generates. Suppose that Γ is a named MCS in $\mathcal{M}^\Sigma$ that is distinct from Σ. Then Γ is witnessed, and hence $\Sigma R^w \Gamma$.*

Proof. Suppose $\Gamma \neq \Sigma$ and that Γ is named by i. Note that $\Sigma R^c \Gamma$, as $\mathcal{M}^\Sigma$ is transitive. We need to show that Γ is both $\downarrow$- and $\downarrow^1$-witnessed. Now, it is easy to see that Γ is $\downarrow$-witnessed. For suppose that $\downarrow x\varphi \in \Gamma$. By *Q2* we have that $\downarrow x\varphi \rightarrow (i \rightarrow \varphi[i/x]) \in \Gamma$, hence $\varphi[i/x] \in \Gamma$ and Γ is $\downarrow$-witnessed.

So suppose $\downarrow^1_x\varphi \in \Gamma$. We have to show that there is a nominal k such that $\varphi[k/x] \wedge \Diamond k \in \Gamma$. Now, from the definition of the canonical relation, we have $\Diamond(i \wedge \downarrow^1_x\varphi) \in \Sigma$. So, suppose that the following formula was provable:

$$\textit{Named-Witness} \qquad \Diamond(i \wedge \downarrow^1_x\varphi) \rightarrow \downarrow^1_u\Diamond(i \wedge \varphi[u/x] \wedge \Diamond u),$$

for some variable u not in $\downarrow^1_x\varphi$. Then the required result would follow easily, for we would have that $\downarrow^1_u\Diamond(i \wedge \varphi[u/x] \wedge \Diamond u) \in \Sigma$, and hence, as Σ is $\downarrow^1$-witnessed,

[9] Recent work suggests that it may be worth re-examining the head-on approach. Using a technique described in Gabbay (1976) (see Gabbay's Lemmas 7.3 and 7.4, pages 40–41) for first-order modal logic, Blackburn and Tzakova (1998a) give a completeness proof for multi-modal logic enriched with $\forall$ that does not appeal to the provability of $(\varphi \rightarrow \exists x\psi) \rightarrow \exists x(\varphi \rightarrow \psi)$. It would be interesting to try adapting this alternative proof to $\mathrm{ML}+\downarrow+\Downarrow^1$.

that $\Diamond(i \wedge \varphi[k/u] \wedge \Diamond k) \in \Sigma$, for some nominal k. Using *Nom* we could conclude that $\Box(i \rightarrow (\varphi[k/u] \wedge \Diamond k)) \in \Sigma$, and it would follow that $\varphi[k/u] \wedge \Diamond k \in \Gamma$ since i names Γ. Thus Γ would be $\downarrow^1$-witnessed as required.

And, indeed, *Named-Witness* is provable. Using *Q3* and α-conversion we have $\vdash \downarrow^1_x\varphi \rightarrow \downarrow^1_u(\varphi[u/x] \wedge \Diamond u)$, where u is a variable not occurring in $\downarrow^1_x\varphi$. Hence by propositional calculus and easy modal reasoning $\vdash \Diamond(i \wedge \downarrow^1_x\varphi) \rightarrow \Diamond(i \wedge \downarrow^1_u(\varphi[u/x] \wedge \Diamond u))$. Hence, by the previous lemma we have $\vdash \Diamond(i \wedge \downarrow^1_x\varphi) \rightarrow \Diamond\downarrow^1_u(i \wedge (\varphi[u/x] \wedge \Diamond u))$, and using $Barcan_{11}$ to give $\downarrow^1_u$ scope over the consequent, we get *Named-Witness*. $\dashv$

Lemma 17 (Existence Lemma for Witnessed Models) *Let Σ be a witnessed MCS, and let $\mathcal{M}^w = (S^w, R^w, V^w)$ be the witnessed model yielded by Σ. Then for any MCS Δ in S^w, if $\Diamond\varphi \in \Delta$, then there is $\Theta \in S^w$ such that $\Delta R^w \Theta$ and $\varphi \in \Theta$.*

Proof. Easy. We are simply going to use the following formula to reduce the problem to the basic modal Existence Lemma:

$$Paste \qquad \Diamond\varphi \rightarrow \downarrow^1_x\Diamond(x \wedge \varphi),$$

where x is a variable not occurring in φ. To see that *Paste* is provable, note that $\vdash \varphi \rightarrow (x \rightarrow (x \wedge \varphi)$. By localization and normality we get $\vdash \downarrow x\varphi \rightarrow \downarrow x(x \rightarrow (x \wedge \varphi)$, hence using *Q3* to simplify the consequent we get $\vdash \downarrow x\varphi \rightarrow \downarrow x(x \wedge \varphi)$. Lemma 8 tells us that $\vdash \varphi \rightarrow \downarrow x\varphi$, as x does not occur in φ, hence $\vdash \varphi \rightarrow \downarrow x(x \wedge \varphi)$. Basic modal reasoning yields $\vdash \Diamond\varphi \rightarrow \Diamond\downarrow x(x \wedge \varphi)$, and $Barcan_{10}$, then yields *Paste*.

But now the lemma is virtually immediate. If $\Diamond\varphi \in \Delta$, then $\downarrow^1_u\Diamond(u \wedge \varphi) \in \Delta$; hence as Δ is witnessed, for some nominal i, $\Diamond(i \wedge \varphi) \in \Delta$. Now, by the basic modal Existence Lemma, Δ has an R^c-successor containing i and φ; call this i-containing successor Θ. Moreover, as Θ is a named MCS in the submodel of the canonical model generated by Σ, by the previous lemma it is witnessed, and hence $\Delta R^w \Theta$. $\dashv$

Lemma 18 (Truth Lemma) *Let $\mathcal{M}$ be the completion of a witnessed model in some countable language $\mathcal{L}$, let g be the completed $\mathcal{M}$-assignment, and let Δ be an $\mathcal{L}$-MCS in $\mathcal{M}$. For every formula φ we have $\varphi \in \Delta$ iff $\mathcal{M}, g, \Delta \models \varphi$.*

Proof. By induction on the complexity of φ. If φ is a state symbol or a propositional variable the equivalence follows from the definition of the model $\mathcal{M}$ and the assignment g. The Boolean cases follow from obvious properties of MCSs. For the modal case, the Existence Lemma gives us the information required to prove the left to right direction. The right to left direction is more or less immediate, though there is a small point the reader should observe: if $\mathcal{M}, g, \Delta \models \Diamond\psi$, then there is a successor of Δ that satisfies ψ. Since no MCS precedes $*$, we conclude that $*$ cannot be this successor of Δ. Thus the successor to Δ that satisfies ψ is itself an MCS, and so we can apply the inductive hypothesis.

So suppose $\downarrow^1_x\psi \in \Delta$. Since Δ is witnessed, there is a nominal i such that $\psi[i/x] \wedge \Diamond i \in \Delta$. By the inductive hypothesis $\mathcal{M}, g, \Delta \models \psi[i/x]$. Moreover, by the Existence Lemma, Δ is related to some witnessed MCS containing i, hence $\mathcal{M}, g, \Delta \models \Diamond i$. Thus, by the contrapositive of the *Q2* axiom, $\mathcal{M}, g, \Delta \models \downarrow^1_x\psi$.

For the other direction assume $\mathcal{M}, g, \Delta \models \downarrow^1_x\psi$. That is, $\mathcal{M}, g', \Delta \models \psi$, where $g' \overset{x}{\sim} g$ such that $g'(x) = \{\Gamma\}$, where Γ is a witnessed MCS such that

$\Delta R^+\Gamma$. By clause 2 of Lemma 13, Γ contains a nominal, say i. Now, by clause 2 of the Substitution Lemma, $\mathcal{M}, g, \Delta \models \psi[i/x]$, hence by the inductive hypothesis $\psi[i/x] \in \Delta$. Moreover, as $\Delta R^+\Gamma$, $\Diamond i \in \Delta$. So, by the contrapositive of the *Q2* axiom, $\downarrow^1_x \psi$ is in Δ as required.

The argument $\downarrow x\psi \in \Delta$ is similar (in fact, slightly simpler as no appeals to the Existence Lemma are required) and is left to the reader. $\dashv$

Theorem 19 (Completeness) *Every $\mathcal{H}[\downarrow, \Downarrow^1](I4)$-consistent set of formulas in a countable language $\mathcal{L}$ is satisfiable in a countable, rooted, strictly partially ordered, standard model with respect to a standard assignment function.*

Proof. Suppose Σ is a consistent set of $\mathcal{L}$-formulas. By the Extended Lindenbaum's Lemma we expand it to a witnessed MCS Σ^+ in the countable language $\mathcal{L}^+$. Let $\mathcal{M}^+$ be the completed model yielded by Σ^+, and let g^+ be the completed assignment on this model. $\mathcal{M}^+$ is a rooted strictly partially ordered standard model, and g^+ is a standard assignment. It is countable since every MCS contains a nominal, and $\mathcal{L}^+$ is countable. By the Truth Lemma, $\mathcal{M}^+, g^+, \Sigma^+ \models \Sigma^+$, and so $\mathcal{M}^+, g^+, \Sigma^+ \models \Sigma$. $\dashv$

The completeness proof for $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ unearths a number of ideas that will play a key role in subsequent work. For example, from now on we will always think in terms of named sets rather than witnessed sets. Moreover, the idea underlying the *Paste* formula is crucial, and will return in various guises. Nonetheless, there is an obvious shortcoming: the proof hinged on the transitivity assumption. What can we do about this?

Reflecting on the proof, we see that the combination of transitivity and $\Downarrow^1$ was really a way of ensuring *communication*: it enabled us to establish a link between bound variables and the nominals used to instantiate them.[10] Consider, in particular, how Lemma 16 was proved. This relied on the fact that — because of transitivity — the generating point could see each named MCS in the canonical model in a single step. This guaranteed (with the help of *Named-Witness*) that all such sets were witnessed, and subsequent use of *Paste* led to a swift proof of the Existence Lemma. The moral is clear: if we want local hybrid languages capable of coping with the logics of arbitrary models, it seems we must look for communication mechanisms that function effectively in the absence of transitivity.

5 Tense logic with $\downarrow$

We begin our quest for improved communication by (temporarily) changing the underlying language from ML to TL; we are going to abandon ML+$\downarrow$+$\Downarrow^1$

[10] One reviewer suggested that the transitivity assumption was essentially a way of smuggling globality into the language. We don't agree. While it's true that $\Downarrow^1$ 'matches' the action of $\Box$ on transitive frames in much the same way that $\forall$ matches the action of of the universal modality A, the model construction for $\Downarrow^1$ is very different from that for ML+$\forall$ or ML+$\forall$+A. Perhaps the suggestion made in Footnote 9 will enable this difference to be bridged, but even so the fact remains that only a fragment of classical reasoning is sound for $\Downarrow^1$. Moreover, many applications demand that we work with models containing *multiple* transitive flows of time, each completely isolated from the others. (For example, such models underly many Montague-style analyses of temporal expressions.) Typically we would not want the *temporal* operators and binders to be able to access states in alternative time flows; our sense of locality mirrors this requirement perfectly.

and work instead with TL+$\downarrow$. As we shall see, the interplay of TL's forward and backward looking operators is a communication mechanism of the type we require: even without the help of a transitivity assumption, TL interacts cleanly with $\downarrow$. Let's consider some examples.

First, *Until* is definable in TL+$\downarrow$:

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

Note that this works rather differently from previous hybrid definitions of *Until*. This definition is *active*. First we use $\downarrow$ to mark the current state with x. Then we say that there is a future state, labeled y, such that (1) φ holds at y, and (2) if we look back at x from y we see that the following holds: all successors of x that precede y satisfy ψ. In short, we use $\downarrow$ to name two key states (the current state and the φ verifying state) and by using the tense operators to shuttle backwards and forward between them we enforce the required conditions on ψ. Contrast this with the way we proceeded in ML+$\downarrow$+$\Downarrow^1$. There we stayed put at the current state, used $\downarrow^1$ to label the φ point, and then enforced the condition on the ψs with the help of the modalities.

Next observe that $\downarrow^1$ is definable in TL+$\downarrow$ as follows: $\downarrow^1_y\varphi := \downarrow x F \downarrow y P(x \wedge \varphi)$. Note that this uses the same strategy as the definition of *Until*: we mark the key points and use the tense operators to shuttle between them.[11] Given the work of the previous section, this suggests that TL+$\downarrow$ is a promising language. Moreover, the way these binders are defined bears some resemblance to the L-forms that underly theCOV rule. So surely we can neatly axiomatize the minimal logic of TL+$\downarrow$?[12]

Indeed we can. We'll do so as follows. First we'll define an axiomatization called $\mathcal{H}[\downarrow](K_t)$; as this notation is meant to suggest, this will be an extension

[11] This, of course, means it is possible to mimic the passive ML+$\downarrow$+$\Downarrow^1$ style definition of *Until* in TL+$\downarrow$. We do this as follows: $\mathit{Until}(\varphi,\psi) \ := \ \downarrow x F \downarrow y P(x \wedge \Diamond(y \wedge \varphi) \wedge G(Fy \rightarrow \psi)))$. That is, all the shuttling to and fro is now packaged into a macro in prenex position. Note that not only $\downarrow^1$ is definable; we can also define binders which quantify over 2-step successors, 3-step successors, and so on. In fact, we can even define binders that allow us to quantify over successors reachable via zig-zagging paths. Here, for example, is how we define an existential quantifier over those states reachable by making 3 forward steps followed by 1 backward one: $\downarrow^{3,1}_y\varphi := \downarrow x FFFP \downarrow y FPPP(x \wedge \varphi)$.

[12] Although we are mostly concerned with the technical ramifications of adding $\downarrow$ to TL, we attach deeper significance to TL+$\downarrow$; in our view this language is rather special. Priorean tense logic has sometimes been criticized for trivializing the *present* tense — the past and future tenses are both there, but there is a 'gap' where the present tense should be. In our view $\downarrow$ goes a great deal of the way towards formalizing a linguistically and philosophically plausible notion of the present tense: to bind a variable to the current state is essentially to make *direct indexical reference* to it — and arguably this is the cornerstone of the semantics of the present tense. We can't follow this up here, but for explicit arguments along these lines that make use of $\downarrow$ see Richards *et al* (1989) (these authors use the notation G for $\downarrow$). Similar sentiments can be gleaned from Prior's analysis of the word "Now" (see Prior (1968)). Kamp's technical development of Prior's views (see Kamp (1971)) abandoned Prior's nominal-based direct reference account in favor of simulating reference in the metalanguage via a two dimensional semantics, an approach which was subsequently adopted by Vlach (1973) and others; Prior's arguments (and nominals with it) passed into obscurity. There is an intriguing story to be told here, hints of which may be found in Blackburn (1990) and (1993).

Note that if $\downarrow$ is accepted as a reasonable formalization of the present tense, this gives added significance to the definability of Kamp's *Until* and *Since* operators in TL+$\downarrow$: these operators, as Kamp himself proved, are *not* definable in terms of past and future — but they *are* definable in terms of past, *present* and future. That is, Prior's celebrated triple encompasses temporal reasoning in a very strong form.

of the minimal Priorean tense logic K_t. We'll then show that this logic admits a rule of proof called *Paste*; this rule is analogous to the *Paste* formula used in the previous section to prove the Existence Lemma. Using this rule and the model construction methods introduced in the previous section it will be straightforward to prove completeness.

Let's get to work. $\mathcal{H}[\downarrow](K_t)$ is an extension of K_t, the minimal Priorean tense logic. Recall that K_t is the smallest set of formulas containing all propositional tautologies, all instances of $G(\varphi \to \psi) \to (G\varphi \to G\psi)$, and $H(\varphi \to \psi) \to (H\varphi \to H\psi)$, and all instances of $\varphi \to GP\varphi$ and $\varphi \to HF\varphi$ (the *Converse* axioms), that is closed under *modus ponens* and *necessitation for both G and H* (that is, if φ is provable then so are $G\varphi$ and $H\varphi$). The *Converse* axioms play a key role in the work that follows: as we shall see, it is these simple looking formulas that give us "the logic of to-and-fro".

To the axioms and rules of proof of K_t we add the following. First $\mathcal{H}[\downarrow](K_t)$ is closed under *state variable localization for* $\downarrow$, just as $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ is. In addition, $\mathcal{H}[\downarrow](K_t)$ contains the *Q1*, *Q2*, *Q3* and *Self-Dual* axioms for $\downarrow$, and the following version of *Nom*:

$$\textit{Nom} \quad \mathbf{T}(\mathbf{s} \wedge \varphi) \to \mathbf{U}(\mathbf{s} \to \varphi).$$

Here $\mathbf{s}$ is a metavariable over state symbols, $\mathbf{T}$ is a metavariable over sequences of F and P operators (including the null sequence) and $\mathbf{U}$ is a metavariable over sequences of G and H operators (including the null sequence). Consider what this version of *Nom* says: if by following the sequence of transitions indicated by $\mathbf{T}$ we reach a state labeled $\mathbf{s}$ and containing the information φ, then *any* sequence of transitions $\mathbf{U}$ that takes us to $\mathbf{s}$ will takes us to a φ-containing state — for of course, no matter how we get there, there is only one state labeled $\mathbf{s}$. We need to work with such sequences because we have placed no restrictions on the accessibility relation; our modal theory of labeling must cope with arbitrary zig-zag paths.

And that's $\mathcal{H}[\downarrow](K_t)$. Note that all its rules of proof and axioms (with the exception of the new version of *Nom*) are familiar, and that there are no *Barcan* analogs. And now for the key observation: we don't need *Barcan* analogs, because we can prove that $\mathcal{H}[\downarrow](K_t)$ is closed under a rule of proof called *Paste*.

We introduce some notation. Let T be a metavariable over F and P (note: we mean single occurrences of these two operators, not sequences of them), and let us write $T_i\varphi$ as shorthand for $T(i \wedge \varphi)$. Using this notation, we can state the *Paste* Rule as follows:

$$\frac{\vdash T_i \cdots T_j T_k \varphi \to \theta}{\vdash T_i \cdots T_j T \varphi \to \theta}$$

(Here k is a nominal distinct from $i, \ldots, j$ that does not occur in φ or θ.)

Lemma 20 (Admissibility of Paste) *The Paste rule is admissible in* $\mathcal{H}[\downarrow](K_t)$.

That is, we *automatically* have closure under this rule in $\mathcal{H}[\downarrow](K_t)$; we defer the proof of this until the end of the section. For now, let's just accept the lemma, and see why closure under this rule is so useful. The reason is embodied in the the following definition:

Definition 21 (Pasted sets) *An MCS* Γ *is called* pasted *iff for every formula of the form* $T_i \cdots T_j T\varphi \in \Gamma$ *there is a nominal* k *such that such that* $T_i \cdots T_j T_k\varphi \in \Gamma$.

That is, a pasted set is an MCS in which whenever we can reach some information φ by following a chain of transitions, then we can always reach that same information *in a named state* by following the same chain. Intuitively, if we start with a pasted set this should make it easier to prove an Existence Lemma: faced with a demand $\Diamond\varphi$, we look for a *named* state that fulfills that demand. And now the significance of the *Paste* rule is clear: read contrapositively (that is, read from bottom to top) it tells is that pasting a brand new nominal 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.

Lemma 22 (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 named and pasted* $\mathcal{L}^+$*-MCS.*

Proof. Enumerate the formulas of $\mathcal{L}^+$; call this the formula enumeration. Enumerate the nominals that are in $\mathcal{L}^+$ but not in $\mathcal{L}$; call this the new-nominal enumeration. Let Φ be a consistent set of $\mathcal{L}$ formulas. We shall first add a name to Φ, and then inductively extend it to a pasted $\mathcal{L}^+$-MCS Θ.

Let j be the first nominal in the new-nominal enumeration. Define Φ_j to be $\Phi \cup \{j\}$. Φ_j is consistent. For suppose not. Then for some conjunction of formulas δ from Φ we have that $\vdash j \to \neg\delta$. As j is from the new-nominal enumeration, it does not occur in δ, hence we can apply localization on nominals (see Lemma 7) to obtain $\vdash \downarrow x(x \to \neg\delta)$, where x is a variable that does not occur in δ. By *Q3*, $\vdash \downarrow x\neg\delta$, hence by the *Vacuity* (see Lemma 8), $\vdash \neg\delta$, which contradicts the consistency of Φ. Thus Φ_j is consistent.

We now inductively extend Δ_j to a pasted $\mathcal{L}^+$-MCS. Let $\Theta^0 = \Delta_j$. Define Θ^{n+1} as follows. Let φ_n be the n-th formula in the formula enumeration. If $\Theta^n \cup \{\varphi_n\}$ is inconsistent, then $\Theta^{n+1} = \Theta^n$. Otherwise:

1. $\Theta^{n+1} = \Theta^n \cup \{\varphi_n\}$, if φ_n is not of the form $T_i \cdots T_j T\psi$.
2. $\Theta^{n+1} = \Theta^n \cup \{\varphi_n\} \cup \{T_i \cdots T_j T_k\psi\}$, otherwise. Here k is the first nominal in the new-nominal enumeration that does not occur in Θ^n or φ_n.

Let $\Theta = \bigcup_{n \geq 0} \Theta^n$. It is easy to see that this set is named, maximal, and pasted. Moreover, Θ^n is consistent for all n (this is precisely what *Paste* guarantees) hence Θ itself is consistent too. $\dashv$

What should we do next? One of the most valuable things we learned in the previous completeness proof was to try thinking in terms of named sets. That's what we're going to do here, right from the start, Given a named and pasted MCS, we're going to inductively collect all it's *named* successors, *named* predecessors, *named* successors of *named* successors, *named* successors of *named* predecessors, and so on, and use these MCSs to make the model. As we shall show, our logic is strong enough to guarantee that all these named sets are both pasted and $\downarrow$-witnessed.

Definition 23 (Named models) *Let Σ be a named and pasted MCS in some countable language $\mathcal{L}$ and let $\mathcal{M}^c = (S^c, R^c, V^c)$ be the canonical model in $\mathcal{L}$. Define S^0 to be $\{\Sigma\}$. For all natural numbers $n \geq 0$, define S^{n+1} to be*

$$S^n \cup \{\Gamma \in S^c \mid \Gamma \textit{ is named, and for some } \Delta \in S^n, \Delta R^c \Gamma \textit{ or } \Gamma R^c \Delta\}.$$

Define S to be $\bigcup_{n \in \omega} S_n$. Define $\mathcal{M}$, the named model yielded by Σ, *to be (S, R, V) where R and V are the restrictions of S^c and R^c respectively to S.*

Lemma 24 *Let $\mathcal{M}$ be the named model yielded by a named and pasted MCS Σ. Then (1) every MCS in S is named, (2) no state symbol occurs in more than one MCS in S, (3) every MCS in S is $\downarrow$-witnessed, and (4) every MCS in S is pasted.*

Proof. (1) is trivial, (2) can be proved using the (temporal version of) the *Nom* schema, and (3) is an immediate consequence of *Q2* and the fact that every MCS in S is named (in fact we made use of this argument in the first part of the proof of Lemma 16).

For (4), we shall show that every named R^c-successor and R^c-predecessor of a named and pasted set is itself pasted. Given our inductive definition of S, it follows immediately that every MCS in S must be pasted. So suppose Γ is named by a nominal l, and that Γ is a R^c-predecessor of a named and pasted set Δ. Further suppose for the sake of a contradiction that Γ is *not* pasted. This means there is some $T_i \cdots T_j T\varphi \in \Gamma$ such that for all nominal k, $\neg T_i \cdots T_j T_k \varphi \in \Gamma$. Now $\Gamma R^c \Delta$, hence $P(l \wedge T_i \cdots T_j T\varphi) \in \Delta$. But Δ is pasted, hence for some nominal k, $P(l \wedge T_i \cdots T_j T_k \varphi) \in \Delta$. Hence by *Nom*, $H(l \to T_i \cdots T_j T_k \varphi) \in \Delta$, thus as $\Gamma R^c \Delta$ and $l \in \Delta$, we have that $T_i \cdots T_j T_k \varphi \in \Gamma$. From this contradiction we deduce that Γ must be pasted after all. In a similar fashion (indeed, simply by replacing P by F and H by G in the previous argument) we can show that named R^c-successors of named and pasted sets are themselves pasted, which finishes the proof. $\dashv$

Lemma 25 (Existence Lemma for Named Models) *Let Σ be a named and pasted MCS, and let $\mathcal{M} = (S, R, V)$ be the named model yielded by Σ. Then for any MCS Δ in S:*

1. *If $F\varphi \in \Delta$, then there is a $\Theta \in S$ such that $\Delta R \Theta$ and $\varphi \in \Theta$.*
2. *If $P\varphi \in \Delta$, then there is a $\Theta \in S$ such that $\Theta R \Delta$ and $\varphi \in \Theta$.*

Proof. More or less immediate; we shall prove item 2. Suppose $P\varphi \in \Delta$. By the last part of the previous lemma Δ is pasted, hence for some nominal i, $P(i \wedge \varphi) \in \Delta$. By basic modal reasoning we have that $\{i \wedge \varphi\} \cup \{\theta \mid H\theta \in \Delta\}$ is consistent, hence it can be extended to an MCS Θ. It is standard result of tense logic that $\Theta R^c \Delta$. As Θ is named by i, it belongs to S, hence $\Theta R \Delta$ as required. Item 1 can be proved similarly. $\dashv$

We can now finish off pretty much as we did in the previous section. First, to guarantee that every nominal and variable denotes something we complete the model by adding on a dummy state.

Definition 26 (Completed models and assignments) *Let* $\mathcal{M} = (S, R, V)$ *be the named model yielded by some named and pasted set* Σ. *A* completion *of* $\mathcal{M}$ *is a triple* $\mathcal{M}^+ = (S^+, R^+, V^+)$, *where* $S^+ = S \cup \{*\}$ *(where* $*$ *is an entity that is not an MCS);* $R^+ = R$*; for all propositional variables* p, $V^+(p) = V(p)$*; and for all nominals* i, $V^+(i) = \{\Gamma \in \mathcal{M} \mid i \in \Gamma\}$ *if this set is non-empty, and* $V^+(i) = \{*\}$ *otherwise.*

If $\mathcal{M}^+ = (S^+, R^+, V^+)$ *is a completion of a named model* $\mathcal{M}^n$, *then the completed assignment* g^+ *on* $\mathcal{M}^+$ *is defined as follows: for all variables* x, $g^+(x) = \{\Gamma \in \mathcal{M} \mid x \in \Gamma\}$ *if this set is non-empty, and* $g(x) = \{*\}$ *otherwise.*

Note that by item 2 of Lemma 24, completed models and assignments are standard. Also note that completed models are *not* connected; $*$ is not related to any other state other states.

Lemma 27 (Truth Lemma) *Let* $\mathcal{M}$ *be the completion of a named model in some countable language* $\mathcal{L}$, g *the completed* $\mathcal{M}$*-assignment, and* Δ *an* $\mathcal{L}$*-MCS in* $\mathcal{M}$. *For every formula* φ *we have* $\varphi \in \Delta$ *iff* $\mathcal{M}, g, \Delta \models \varphi$.

Proof. Much the same as the proof of Lemma 18, though simpler. The Existence Lemma just proved handles the inductive step modalities. The fact that all MCSs in the model are $\downarrow$-witnessed (item 4 of Lemma 24) handles the step for $\downarrow$. $\dashv$

Theorem 28 (Completeness) *Every* $\mathcal{H}[\downarrow](K_t)$*-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,* $\mathcal{H}[\downarrow](K_t)$*-consistent set of* sentences *in* $\mathcal{L}$ *is satisfiable in a countable* connected *standard model*

Proof. The first part of the theorem is proved in the expected way: given a $\mathcal{H}[\downarrow](K_t)$-consistent set of formulas Σ, we use the Extended Lindenbaum Lemma to expand it to a named and pasted set Σ^+ in a countable language $\mathcal{L}^+$, and satisfy it on the completed model $\mathcal{M}^+$ it gives rise to using the completed assignment g^+.

$\mathcal{M}^+$ isn't connected, but let $\mathcal{M}$ be the submodel generated by Σ^+; note that this is just the named model. Then all *sentences* in Σ^+ are true in $\mathcal{M}$. $\dashv$

That's it — though there are two loose ends that need tidying: we need to axiomatize the minimal *temporal* logic, and we need to show that the *Paste* rule is admissible.

We axiomatize the minimal temporal logic by adding all instances of the two following schemas; as usual, $\mathbf{s}$ is a metavariable over state symbols.

$$FF\mathbf{s} \rightarrow F\mathbf{s}$$

$$\mathbf{s} \rightarrow \neg F\mathbf{s}$$

As the reader can easily check, if we build a model for a consistent set of formulas in this enriched logic, the named model will be a strict partial order, and completeness is proved. Fine — but why didn't we just add the familiar *4* axiom schema that we used in the previous proof; surely that would have done just as well for transitivity? Yes, that would have worked — but this way of axiomatizing things points towards a general result we are going to prove in the following section. Note that only state symbols were used in these schemas; neither contains propositional variables. As we shall see in the next section, such schemas *guarantee* completeness.[13]

Finally, let's show that the *Paste* rule is admissible. Much of the proof hinges on the following "to-and-fro" (or "perspective shifting") rules that hold for the minimal tense K_t and its extensions: for any formulas φ and θ, $F\varphi \to \theta$ is provable iff $\varphi \to H\theta$ is provable; and $P\varphi \to \theta$ is provable iff $\varphi \to G\theta$ is provable. We leave the proofs to the reader; they are simple exercise in manipulating the *Converse* axioms. Let T^t denote H if T is F, and G if T is P. Then we can summarize these two rules as: $T\varphi \to \theta$ is provable iff $\varphi \to T^t\theta$. We're now ready to go to work. What follows is essentially an adaptation of a technique used by Gabbay and Hodkinson (1990) to prove a completeness result for the *Until-Since* logic of the real numbers. We apply their to-and-fro idea until we isolate a bare nominal which we can "quantify away" with the the aid of $\downarrow$ binder. Then we to-and-fro everything back together again.

Let's go. We need to show that if $\vdash T_i \cdots T_j T_k \varphi \to \theta$, then $\vdash T_i \cdots T_j T \varphi \to \theta$, where k is a nominal distinct from $i, \ldots, j$ that does not occur in φ or θ. Here's how we start:

$$\begin{array}{l} \vdash T_i T_l \cdots T_j T_k \varphi \to \theta \\ \vdash T(i \wedge T_l \cdots T_j T_k \varphi) \to \theta \\ \vdash (i \wedge T_l \cdots T_j T_k \varphi) \to T^t\theta \\ \vdash T_l \cdots T_j T_k \varphi \to (i \to T^t\theta) \end{array}$$

That is, we've simply expanded our shorthand, applied a perspectival shifting rule, and then used propositional logic. We repeat the argument until we obtain:

$$\vdash k \wedge \varphi \to T^t(j \to T^t(\cdots(l \to T^t(i \to T^t\theta))\cdots))$$

Propositional logic yields:

$$\vdash k \to (\varphi \to T^t(j \to T^t(\cdots(l \to T^t(i \to T^t\theta))\cdots)))$$

Having isolated the (unique) occurrence of k in the antecedent, we start applying what we know about $\downarrow$ to get rid of it. First we apply localization on nominals. Letting x be the new variable which replaces k we obtain:

$$\vdash {\downarrow}x(x \to (\varphi \to T^t(j \to T^t(\cdots(l \to T^t(i \to T^t\theta))\cdots))))$$

Well, k is gone — but it has left x in its place. But we can get rid of x as follows. First, using normality and the fact that ${\downarrow}xx$ is provable yields:

$$\vdash {\downarrow}x(\varphi \to T^t(j \to T^t(\cdots(l \to T^t(i \to T^t\theta))\cdots)))$$

The x in the antecedent position is gone, hence as there are no other occurrences of this variable under the scope of ${\downarrow}x$, we can apply *Vacuity* to obtain:

$$\vdash \varphi \to T^t(j \to T^t(\cdots(l \to T^t(i \to T^t\theta))\cdots))$$

[13]Incidentally, this axiomatization is also, albeit in slightly disguised form, a complete axiomatization of the minimal temporal logic of IQ (see Richards *et al* (1989)).

Having successfully removed all traces of k, we patiently use the perspective shifting rules and propositional calculus to convert the formula back to its original format; in effect we run the first few steps of the argument in reverse. This process eventually yields

$$\vdash T_i T_l \cdots T_j \varphi \to \theta$$

as required. We have shown that the *Paste* rule is admissible.

Time to sum up: what have we achieved? Well, there's certainly a lot on the positive side: we have shown that TL cashes out the communication metaphor, we have learned how to handle a minimal hybrid logic in $\downarrow$, and we have hinted that it is going to be easy to prove the completeness of certain kinds of axiomatic extensions.

Nonetheless, we shouldn't be satisfied with this result. For a start, for many applications we're simply not interested in backward looking operators — and clearly such operators are crucial to the to the work of this section. Furthermore, while the previous result is impressive testimony to the naturalness of tense logics converse operator pairs, it has to be said that there's something — let's face it, *clumsy* — about the way tense logic implements communication: we are forced to pass information up and down long chains of tense operators, as the admissibility proof for the *Paste* rule makes painfully clear. Communication certainly seems to be the key to hybrid completeness: but how are we to obtain it without assuming transitivity, without looking backwards, and, above all, without abandoning the locality intuition that underlies our investigation?

6 The @ operator

Let's start with a little story. 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 point 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.

Two points need to be made right away. First, *sentences* of ML+$\downarrow$+@ are preserved under ML-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 $\downarrow$; as $\downarrow$ binds locally, the result follows. Second, @ increases the expressive power of ML+$\downarrow$, and does so by upgrading the lines of communication; to see this, let's return to our running example, the definability of *Until*.[14]

As we have seen, *Until* is not definable in ML+$\downarrow$ — but it certainly *is* in ML+$\downarrow$+@. In fact, ML+$\downarrow$+@ can mimic either the TL+$\downarrow$ or the ML+$\Downarrow^1$ style definition. Recall that in TL+$\downarrow$ we defined *Until* as follows:

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

But it is easy to capture this definition in ML+$\downarrow$+@. After all, the $P(x \wedge \cdots)$ subformula is essentially a way of saying "move to the point named x", and in our new language we can express this *directly* (to ensure that the correspondence is transparent, we use tense logical notation for $\Box$ and $\Diamond$):

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

Next, recall that in ML+$\Downarrow^1$ we defined *Until* as follows:

$$\mathit{Until}(\varphi,\psi) \;:=\; \downarrow^1_y(\Diamond(y \wedge \varphi) \wedge \Box(\Diamond y \rightarrow \psi)).$$

How do we mimic this? As follows:

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

[14] A lot more could be said about @, and we can't possibly say it all here. But two things are worth mentioning. First, the reader has probably seen something like @ before in other languages: 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. Interestingly, adding @ to ML+$\forall$ also suffices to yield full first-order expressive power; for more on this see Blackburn and Seligman (998).

In Sofia school work on hybrid languages and modal logic with names, use of the universal modality is almost invariably taken for granted. However we have found one passage where Passy and Tinchev express a less satisfied opinion:

> *So the operator* $\langle\nu\rangle$ [that is: E] *is really a strange one: it realizes something like half-order quantification, and more precisely, a quantification of order* $1/2 + i$, *where* i *is the imaginary unit.* (Passy and Tinchev, 1985, Section 3).

In essence, by stripping E down to @ we simplify the situation. As we shall see, @ is *precisely* what is needed to drive through elegant completeness results. It leaves no difficult residue, imaginary or otherwise.

The prenex block $\downarrow x\Diamond\downarrow y@_x$ is simply a way of defining $\downarrow^1_y$.[15] Clearly $\downarrow$ and @ make a great team; they communicate smoothly and their cooperation will give rise to an elegant proof theory. As will soon become apparent, @ is the dominant partner proof-theoretically. For a start, @ allows us to simplify the modal theory of labeling; we won't need nested operators in the new *Nom* axiom. Second, it will allow us to state a pair of rules called *Paste-0* and *Paste-1*. Unlike the *Paste* rule used in the previous section (and indeed, unlike *COV*), these rules don't make use of arbitrarily deep operator nestings; we will be able to paste in all the nominals we at a depth of at most 1. Moreover, because @ records what happens at each and every named point, we will find that every MCS contains a blueprint of an entire collection of MCSs, all neatly indexed by subscripted @ operators. Once this has been grasped, and once the role of the *Paste-0* and *Paste-1* has been understood, the completeness proof practically writes itself.

Let's get to work. We shall present the axiomatization in two stages. First we'll present the system $\mathcal{H}[\downarrow, @](K)$, and show that it gives rise to well-behaved collections of MCSs. We'll then introduce the *Paste* rules and show how they lead to the Existence Lemma. The completeness proof for the minimal logic will be an immediate consequence; almost as immediate will be the completeness of many of its extensions.

$\mathcal{H}[\downarrow, @](K)$ 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 $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$, that is closed under *modus ponens* (if φ and $\varphi \to \psi$ 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 the following. First, $\mathcal{H}[\downarrow, @](K)$ is closed under *state variable localization for* $\downarrow$, and contains the *Q1*, *Q2*, *Q3* and *Self-Dual* axioms for $\downarrow$. In addition, for every state symbol s, it is closed under *$@_s$-necessitation* (if φ is provable then so is $@_s\varphi$) and all instances of the following axiom schemas.

There are three groups of schemas. The first identifies the basic logic of @.

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

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*: the left-to-right direction is the modal determinism axiom, while the right-to-left direction 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

[15] So to speak, although we can't really stay put at the current state and bind variables to accessible states as we can in ML+$\Downarrow^1$, we *can* do something just as good: 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. With this done, we can carry on as if we never left the current state in the first place. In a similar way we can define an existential quantifier over states accessible in 2 steps ($\downarrow^2_y\varphi := \downarrow x\Diamond\Diamond\downarrow y@_x\varphi$); and indeed, for any natural number n, an existential quantifier over states accessible in n steps. Note that the duals of all these operators also have neat definitions: for example, $\Downarrow^2_y\varphi := \downarrow x\Box\Box\downarrow y@_x\varphi$.

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 $(\mathbf{s} \wedge @_{\mathbf{s}}\varphi) \to \varphi$; we call this is *Elimination* schema.

The next group is our modal theory of labeling. The @ operator allows us to formulate this simply and directly:

Name $@_{\mathbf{s}}\mathbf{s}$

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

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

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

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

Back $\Diamond @_{\mathbf{s}}\varphi \to @_{\mathbf{s}}\varphi$

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

And that's $\mathcal{H}[\downarrow, @](K)$. Much of what we subsequently need can be established in this system, so before introducing the *Paste* rules, let's prove the following lemma; 'consistency' and 'MCS' here mean mean $\mathcal{H}[\downarrow, @](K)$-consistency, $\mathcal{H}[\downarrow, @](K)$-MCS, and so on.

Lemma 29 *Let Γ be an MCS that contains a state symbol, 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 an 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 *Name* 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 paragraph.

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, given a state-symbol-containing MCS, the subscripted @ operators index a well-behaved collection of MCSs; the Δ_s certainly seem plausible model-building material. Nonetheless, they don't yet have all the properties we want. First, note that we usually build our models out of *named* MCSs, that is, MCSs that contain *nominals*; named sets are automatically $\downarrow$-witnessed (this follows immediately from *Q2*; in fact, we made use of this in the first part of Lemma 16), thus we can prove the clause of the Truth Lemma for $\downarrow$ without having to worry about accidental binding. But note that even if Γ itself is named (say by i), we have no guarantee that the Δ_s are named 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, more serious, problem. Suppose we take the collection of Δ_s yielded by a named MCS as the building blocks of our model: how do we know that this model supports an Existence Lemma? Bluntly, we don't.

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

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

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. Let $\mathcal{H}[\downarrow, @](K) + \texttt{Paste}$ be the system obtained by adding both rules to $\mathcal{H}[\downarrow, @](K)$; until further notice, proof-theoretic concepts such as 'consistent' and 'MCS' are used with reference to this enriched system. We defer till the end of the section discussion of the admissibility of these rules; for now we'll concentrate using them to prove completeness.

The key rule is *Paste-1*. It trades on the same idea as the *Paste* rule for TL+$\downarrow$. In fact, if we use the tense logical notation F for $\Diamond$, and write $F_s\varphi$ for $F(s \wedge \varphi)$ as we did in the previous section, *Paste-1* becomes:

$$\frac{\vdash @_{\mathbf{s}}F_{\mathbf{t}}\varphi \to \theta}{\vdash @_{\mathbf{s}}F\varphi \to \theta}$$

Clearly this is essentially the same as the *Paste* rule used in previous section. Its core is unchanged: as before (read the rule from bottom to top) it tells us that introducing a brand new label under the scope of the F (that is, $\Diamond$) is a consistency preserving operation. The important difference is the way we *access* this core. In the previous section we used sequences of indexed tense operators to do this; here we simply use the @ operator to insist that this core reasoning is acceptable at any labeled state. 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.

Definition 30 (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 31 (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 named and pasted* $\mathcal{L}^+$*-MCS.*

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. By exactly the same $\downarrow$-based argument used in the proof of the Extended Lindenbaum Lemma for $\mathcal{H}[\downarrow](K_t)$ (see Lemma 22) it follows that Φ_j is consistent.

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 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 named, 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 named and pasted $\mathcal{L}^+$-MCS. $\dashv$

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

Definition 32 (Named models and natural assignments) *Let Γ be a named and pasted MCS. 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 named 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 named models have all the structure we want. For a start, by Clause 3 of Lemma 29, $\Gamma \in S$, and by Clause 5, V is a *standard* valuation and g is a *standard* assignment. Note that we *don't* need to 'complete' this model by gluing on a dummy state $*$; every state symbol finds a home right from the start. Further, all states in the model contain nominals, because Γ is pasted (and hence, 0-pasted). It follows from this, using *Q2*, that every state in the model is $\downarrow$-witnessed, so we have the structure needed to push through the clause of the Truth Lemma for $\downarrow$. Moreover, we know from Lemma 29 that $\mathcal{M}$ is well behaved as far as @ is concerned. So it only remains to ensure that such models support an Existence Lemma. This, of course, is where 1-pasting comes in:

Lemma 33 (Existence Lemma) *Let $\mathcal{M} = (S, R, V)$ be the named model yielded by a named and pasted set Γ. 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 29) 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 34 (Truth Lemma) *Let Θ be an MCS in $\mathcal{M}$. For all formulae φ, $\varphi \in \Theta$ iff $\mathcal{M}, \Theta \models \varphi$.*

Proof. Straightforward: the Existence Lemma just proved handles the modal case, and the fact that named sets are $\downarrow$-witnessed handles the clause for $\downarrow$. The argument for @ runs as follows: $\mathcal{M}, \Theta \models @_s\psi$ iff $\mathcal{M}, \Delta_{\mathbf{s}} \models \psi$ (for by Clause 3 of Lemma 29, Δ_s is the only MCS containing s, and hence, by the the atomic case, 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 35 (Completeness) *Every $\mathcal{H}[\downarrow, @](K)$+`Paste`-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 $\mathcal{H}[\downarrow, @](K)$+`Paste`-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 $\mathcal{H}[\downarrow, @](K) + \texttt{Paste}$-consistent set of formulas Σ, use the Extended Lindenbaum Lemma to expand it to a named and pasted set Σ^+ in a countable language $\mathcal{L}^+$. By the Truth Lemma just proved, the named model and natural assignment that Σ^+ gives rise to satisfy Σ at Σ^+. This named 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 — as we hinted in the previous section, one of the nicest things about hybrid languages is the relative ease with which general completeness results for richer logics can be proved.[16] Moreover, such results typically link completeness and *frame-definability* in a very straightforward way.

A formula is said to *define* some property of frames (say 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 state and prove 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*
$\downarrow x \Box(\Diamond x \to x)$	*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*

Note that the last three expressions work in a very natural way: they simply take

[16] Historically, this has been a major motivation for exploring hybrid languages. Most authors mention what happens when ML or TL is extended with both $\forall$ and A: as Bull (1970) points out (see page 285), all first-order extensions of the basic logic are easily proved complete, and there is brief argument to the same effect at the end of Gargov, Passy and Tinchev (1987). However Passy and Tinchev (1991) push matters much 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 concerns of the present paper are rather different: what happen in weaker local languages?

advantage of the fact that $\downarrow$ and @ can simulate the $\Downarrow^1$, $\downarrow^1$, and $\Downarrow^2$ operators.[17]

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

Theorem 36 (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 $\mathcal{H}[\downarrow, @](K) + \texttt{Paste}$ 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`.

Proof. An easy corollary of Theorem 35: 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 named model $\mathcal{M} = (S, R, V)$ and the natural assignment g. Now, the named model is built of MCSs, and each axiom in `Pure` belongs to every $\mathcal{P}$-MCS, thus by the Truth Lemma, $\mathcal{M}, g \models \texttt{Pure}$. But as `Pure` contains only *sentences*, the choice of assignment is irrelevant, hence $\mathcal{M} \models \texttt{Pure}$. Moreover, as `Pure` contains only *pure* sentences, the choice of valuation is also irrelevant, and $(S, R) \models \texttt{Pure}$. 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 of this result, note that we obtain the minimal temporal logic for ML+$\downarrow$+@ by adding as axioms $\downarrow x \Box \neg x$ and $\downarrow x \Box\Box \downarrow y @_x \Diamond y$; the previous theorem guarantees that the named model will validate these axioms, hence as they define irreflexivity and transitivity respectively, the named model will have these properties.

This is pleasant, but let's push things further. Theorem 36 requires us to use sentences as axioms. However it can be more natural to use *pure schemas*. Consider, for example, the schema $\Diamond\Diamond\mathbf{s} \to \Diamond\mathbf{s}$. Any instance of this schema defines transitivity, and it is easy to verify that including all instances as axioms guarantees a transitive named model. Similarly, any instance of the schema

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

defines the no-branching-to-the-right property, and including all instances as axioms guarantees a named model with this property. Both transitivity and no-branching-to-the-right are definable using pure sentences,[18] 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 named models.[19]

[17]For example, 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").

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

[19]We don't know many temporally relevant examples in ML+$\downarrow$+@ that require the use of schemas, but examples are easy to find in TL+$\downarrow$. For example, the schema $P\mathbf{s} \vee \mathbf{s} \vee F\mathbf{s}$ guarantees us *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's going on here? A moments thought shows that the use of schemas is really equivalent to allowing occurrences of the global hybrid binder $\forall$ in prenex position. For example, including all instances of $\Diamond\Diamond\mathbf{s} \to \Diamond\mathbf{s}$ is really equivalent to using $\forall(\Diamond\Diamond x \to \Diamond x)$ as an axiom; the prenex $\forall$-sentence encapsulates all the information contained in the schema. Let's make this precise.

Let φ be a pure formula of ML+$\downarrow$+@; for example

$$i \to \Diamond{\downarrow}y(\neg i \wedge @_i\Diamond(\neg i \wedge \neg y)).$$

is such a formula. We obtain a *pure schema* when we uniformly replace all occurrences of a free-variables and nominals by metavariables over state symbols, and uniformly replace occurrences of bound variables by metavariables over state symbols. For example, the following schema is obtainable from our example:

$$\mathbf{s} \to \Diamond{\downarrow}\mathbf{v}(\neg\mathbf{s} \wedge @_{\mathbf{s}}\Diamond(\neg\mathbf{s} \wedge \neg\mathbf{v})).$$

Note that any schema we can build is obtainable from a formula that contains no nominals (simply use free variables instead) and that any such formula is an instance of the schema it gives rise to. We are ready for the key concept:

Definition 37 ($\forall$-encapsulations) *Let* S *be any pure schema, let* σ *be any formula containing no nominals that gives rise to* S*, and let* $x_1, \ldots, x_n$ *be all the variables that occur free in* ψ*. Then the sentence* $\forall x \cdots \forall x_n\sigma$ *is a* $\forall$-encapsulation *of* S.

Lemma 38 *Let* $\mathcal{M} = (S, R, V)$ *be a model such that for all* $s \in S$*, there is a nominal* i *such that* $V(i) = \{s\}$*. Then for any schema* S *and any variable assignment* g*, if* $\mathcal{M}, g \models \mathsf{S}$*, then* $\mathcal{M} \models \forall x \cdots \forall x_n\sigma$*, where* $\forall x \cdots \forall x_n\sigma$ *is the* $\forall$*-encapsulation of* S*.*

Proof. We show the contrapositive. If $\mathcal{M} \not\models \forall x \cdots \forall x_n\sigma$ then there is some state $s \in S$ such that $\mathcal{M}, s, g \not\models \forall x \cdots \forall x_n\sigma$. This means there is some g' that differs from g only on what it assigns to $x_1, \ldots, x_n$, such that $\mathcal{M}, s, g' \not\models \sigma$. So we've falsified an instance of the schema; unfortunately we've done so using g', not g. But because every state in M is named by some nominal we can repair this: let $i_1, \ldots, i_n$ be nominals such that $V(i_1) = g'(x_1), \ldots, V(i_n) = g'(x_n)$. Then, by the Substitution Lemma,

$$\mathcal{M}, s, g \not\models \sigma[i_1/x_1, \ldots, i_n/x_n],$$

hence we have falsified an instance of S in $\mathcal{M}$ under g as required. $\dashv$

Lemma 39 *Let* $\hat{\sigma}$ *be a* $\forall$*-encapsulation of of a schema* S*, and let* (S, R) *be a frame such that* $(S, R) \models \hat{\sigma}$*. Then every instance of* S *is valid on* (S, R)*.*

Proof. Straightforward. $\dashv$

A *pure schematic extension* of $\mathcal{H}[\downarrow, @](K) + \texttt{Paste}$ is any system obtained by adding all ML+$\downarrow$+@ instances of a set of pure schemas of ML+$\downarrow$+@ as axioms to $\mathcal{H}[\downarrow, @](K) + \texttt{Paste}$.[20]

[20] It's perhaps worth stressing that we're *not* adding any formulas that contain occurrences of the hybrid binder $\forall$ as axioms; the axioms are all ML+$\downarrow$+@ formulas. The detour via $\forall$-encapsulations formulas is simply an easy way of proving that all these ML+$\downarrow$+@ formulas have the effect we want.

Theorem 40 (Extended Completeness II) *Let* `Schemas` *be a set of pure schemas of ML+↓+@, and let $\mathcal{S}$ be the pure schematic extension of $\mathcal{H}[\downarrow, @](K)+$* `Paste` *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. Again we build a satisfying model $\mathcal{M} = (S, R, V)$ and assignment g as described in Theorem 35. By the Truth Lemma, every instance of every schema in `Schemas` is satisfied at every state in this model; that is, for all $\mathsf{S} \in$ `Schemas` we have $\mathcal{M}, g \models \mathsf{S}$. Now for the key step: every state in the named model is named by some nominal, hence Lemma 38 is applicable, and for all $\mathsf{S} \in$ `Schemas` we have $\mathcal{M}, g \models \forall x \cdots \forall x_n \sigma$, where $\forall x \cdots \forall x_n \sigma$ is the $\forall$-encapsulation of S. Now, all these encapsulations are *pure sentences* (of course, sentences of ML+∀+↓+@, not sentences of ML+↓+@) hence both the assignment and the valuation are irrelevant, and we conclude that $(S, R) \models \forall x \cdots \forall x_n \sigma$. Hence, by Lemma 39, every instance of every schema is valid on (S, R), which is what we wanted to show. Finally, if Σ contains only sentences we obtain a connected model by restricting our attention to the submodel generated by Σ^+; as an easy argument shows, as all instances of schemas in `Schema` are valid on (S, R), they remain valid on this generated subframe. $\dashv$

What sort of coverage do Theorems 36 and 40 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+$\downarrow$+@ formula φ, any standard model $\mathcal{M} = (S, R, V)$, any standard assignment g, and any $s \in S$:

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

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

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

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

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

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 and forming the universal closure. Thus Theorem 36 and 40 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?[21] But we leave such questions for another time and turn to the admissibility of the *Paste* rules.

We do not know whether *Paste-1* is admissible in $\mathcal{H}[\downarrow, @](K)$, and believe that there is an interesting open question here:

> *Is Paste-1 admissible in* $\mathcal{H}[\downarrow, @](K)$ *or some finite axiomatic extension?*

The key word here is *finite*. Because we can define the operators $\Downarrow^n$ for all natural numbers n, it seems clear that we can define an *infinite* extension of $\mathcal{H}[\downarrow, @](K)$ in which *Paste-1* is admissible.[22] The interesting issue is whether the effect of *Paste-1* can be captured using only finitely many new axioms.

[21] Incidentally, there are first-order properties which are modally definable but not Sahlqvist definable, which can be defined by pure sentences. For example, the property transitivity + atomicity ($\forall x \exists y(xRy \wedge \forall z(yRz \rightarrow z = y))$) is definable by the conjunction of the transitivity axiom 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 this; only transitivity + atomicity is modally definable. But the following pure sentence defines atomicity: $\downarrow x \Diamond \downarrow y \Box \downarrow z y$. Note that @ is not needed. And we have already seen that transitivity is definable by a pure sentence.

[22] The obvious way of doing this would be to generalize the ideas underlying $\mathcal{H}[\downarrow, \Downarrow^1](I4)$; in particular we would make use of a whole cascade of *Barcan* analogs that generalized the idea of $Barcan_{10}$.

Matters are somewhat clearer for *Paste-0*; the following semantic argument seems to show that *Paste-0* is admissible in $\mathcal{H}[\downarrow, @](K)$. When proving Theorem 35, we only used *Paste-0* to guarantee that each Δ_s contained a nominal. But as far as we can see, our proof goes through substantially unchanged if we let our model contain states Δ_s that are only 'named' by a free variable. Proving completeness with such models is trickier; for example, we need to witness $\downarrow$ formulas with variables not nominals, and this requires careful use of α-conversion to prove the Truth Lemma. But the basic argument is the same and seems only to require resources available in $\mathcal{H}[\downarrow, @](K)$. Admittedly, such an argument sketch runs the risk of overlooking a principle that falls outside $\mathcal{H}[\downarrow, @](K)$, but it seems safe to conclude that *Paste-0* is either admissible in $\mathcal{H}[\downarrow, @](K)$ itself, or in some simple axiomatic extension. As yet we can't yet back up this semantic argument up with an explicit syntactic demonstration of admissibility, and we would like to be able to do so.

7 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. Of course, that's part of the story, and an important part at that, but we believe that a far more general idea is at work, and that it deserves to be made explicit.

In essence, our preceding work rested on a simple idea: combining two distinct 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; they were both handled "propositionally". Now the natural question is: if this works for state-name 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; so why not introduce special atomic symbols that range such entities and allow ourselves to bind them? In short, why not attempt hybridization in a far more ambitious way?[23]

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 inclusion relation on intervals.[24] Or perhaps we'd

[23]In fact, 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 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.

[24]Various constraints would be imposed to make this interpretation plausible. Typically we

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 @.[25]

But there is another way of developing multi-sorted hybrid languages. This hinges on the following simple observation: *some types of information can be thought of as structured sets of states.* For example, an interval is the set of all states between two end points.[26] Why not add atomic symbols that range over such sets? After all, we already have propositional symbols that range over arbitrary subsets, and state-symbols which range over singleton subsets — so why not symbols that range over convex sets too? This is arguably a useful idea (see Blackburn (1999, 1993)) and it is certainly simple to handle logically.[27] But to illustrate the structured-set based 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 highly non-trivial problems.

A hybrid path language

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.[28]

---

would demand that $(S, <)$ be a strict partial order, that $(S, \sqsubseteq)$ be partial order, and that $<$ 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.

[25] The 'straightforward' is justified: many of the frame properties required are expressible by pure sentences or schemas, hence completeness for 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 21). Incidentally, it would be interesting to compare an $\downarrow$- and @-based treatment with Yde Venema's two-dimensional analysis (see Venema (1990)).

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

[27] Incidentally, 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 mentioned above. Incidentally, this 'duality' between the two approaches to sorting may well be useful in the more difficult case of hybrid path languages discussed below, but we won't follow this suggestion up here.

[28] It's worth stressing that there are wide range of reasons for being interested in path based temporal logics, and that the motivations just given barely begin to scratch the surface of this rich and varied domain. Moreover, in both the philosophical and theoretical computer science literatures there is a vast range of non-deterministic models of time and associated logics. We are not going to attempt to address the wide variety of issues these raise; our aim is simply to define a simple hybrid language for talking about the paths that exist in tree-like models, and discuss its more obvious properties, both pleasant and problematic.

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 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.[29]

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 PROP $\cup$ SVAR $\cup$ NOM $\cup$ PVAR $\cup$ PNOM.

The second step is to add a binder. We shall add a binder called $\Downarrow^{\pi}$; we'll add it to ML $+\downarrow+\Downarrow^1$, thus forming the language ML $+\downarrow+\Downarrow^1+\Downarrow^{\pi}$. As the notation should 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 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 41 (Standard models and assignments) *Let ML*$+\downarrow+\Downarrow^1+\Downarrow^{\pi}$ *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* : PROP $\cup$ NOM $\cup$ PNOM $\longrightarrow$ *$Pow(S)$. A model is called* standard *iff for all nominals $i \in$* NOM, *$V(i)$ is a singleton subset of S, and for all path nominals $\tau \in$* PNOM, *$V(\tau) \in \Pi(S, R)$.*

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

[29]We are not the first to do this. Motivated by Prior's arguments, Robert Bull added a universal quantifier over paths to 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. We return to Bull's work later in the section. A recent paper by Goranko on hybrid languages strong enough to embed CTL* (see Goranko (1996b)) is also worth noting. In one way Goranko's language is weaker than the language discussed here (it doesn't doesn't contain path binders, only path nominals) and in another way it is stronger (it contains the the universal modality).

singleton subset of S, and for all path variables $\rho \in$ 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 $\Downarrow^{\pi}$:

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

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.

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:

$$\mathit{Fair}(\varphi) \;:=\; \Downarrow^{\pi}_{\rho}(\Diamond(\rho \wedge \varphi) \wedge \Downarrow^{1}_{x}\Box((x \wedge \rho \wedge \varphi) \to \Diamond(\rho \wedge \varphi)).$$

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, it is also easy to see that many of the (by now familiar) 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 groups of axiom schemas are valid:

Q1	$\Downarrow^{\pi}_{\boldsymbol{\rho}}(\varphi \to \psi) \to (\varphi \to \Downarrow^{\pi}_{\boldsymbol{\rho}}\psi)$
Q2	$\Downarrow^{\pi}_{\boldsymbol{\rho}}\varphi \to (\mathbf{p} \to \varphi[\mathbf{p}/\boldsymbol{\rho}])$
Q3	$\Downarrow^{\pi}_{\boldsymbol{\rho}}(\boldsymbol{\rho} \to \psi) \to \Downarrow^{\pi}_{\boldsymbol{\rho}}\psi$
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 $\Downarrow^{1}$. Moreover *Local-Path* is a clearly analogous to the *Name* axiom for $\downarrow$.

Here's another similarity with our earlier work; we have a *Barcan* analog:

Barcan$_{\pi}$	$\Downarrow^{\pi}_{\boldsymbol{\rho}}\Box\varphi \to \Box\Downarrow^{\pi}_{\boldsymbol{\rho}}\varphi$

The contraposed and dualised form $\Diamond\downarrow^{\pi}_{\boldsymbol{\rho}}\varphi \to \downarrow^{\pi}_{\boldsymbol{\rho}}\Diamond\varphi$ 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"; in essence, it is a path existence principle.

Our language also supports simple axioms that reflect the geometry of paths (we use $\boldsymbol{\tau}$ as a metavariable over path nominals and $\mathbf{i}$ and $\mathbf{j}$ as metavariable over state nominals):

$P1 \quad \Diamond\boldsymbol{\tau} \to \boldsymbol{\tau}$

$P2 \quad \boldsymbol{\tau} \wedge \Diamond\top \to \Diamond\boldsymbol{\tau}$

$P3 \quad \Diamond(\mathbf{i} \wedge \boldsymbol{\tau}) \wedge \Diamond(\mathbf{j} \wedge \boldsymbol{\tau}) \to \Diamond(\mathbf{i} \wedge \Diamond\mathbf{j}) \vee \Diamond(\mathbf{i} \wedge \mathbf{j}) \vee \Diamond(\mathbf{j} \wedge \Diamond\mathbf{i})$

Clearly *P1* reflects convexity, *P2* reflects R-maximality under successorship, and *P3* reflects linearity; note the way the state and path nominals cooperate here. Summing up, in many ways $\mathrm{ML} + \downarrow + \Downarrow^1 + \Downarrow^\pi$ is a pleasant language. It offers a natural way of talking about paths, and validates many familiar principles.

That was the good part — let's turn to the bad. It seems that proving completeness results for this language will require new ideas; the apparatus of named models used in previous sections does not seem to extend naturally to the new language. Worse, the method takes us disconcertingly far, and then lets us down. We shall illustrate the problem by showing what goes wrong if one tries to build a model for formulas consistent with respect to $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ augmented with the rules and axioms for $\Downarrow^\pi$ just listed.[30]

To extend the model building technique for $\mathcal{H}[\downarrow, \Downarrow^1](I4)$ to our hybrid path language, we first extend our concept of witnessing as follows: if $\downarrow^\pi_\rho \varphi \in \Gamma$ then there is a path nominal τ such that $\downarrow^\pi_\rho \varphi \to (\varphi[\tau/\rho] \wedge \tau) \in \Gamma$. Extending Lemma 11 to an Extended Lindenbaum's Lemma for this richer notion of witnessing is routine.

Second, the turning point of our completeness proof for $\mathrm{ML} + \downarrow + \Downarrow^1$ was Lemma 16, in which we proved that named MCSs in witnessed models were themselves witnessed. Interestingly, this lemma extends straightforwardly to $\mathrm{ML} + \downarrow + \Downarrow^1 + \Downarrow^\pi$. Because we have $Barcan_\pi$, it is easy to prove the following analog of *Named-Witness*:

$$\textit{Named-Path-Witness} \qquad \Diamond(\tau \wedge \downarrow^\pi_\rho \varphi) \to \downarrow^\pi_{\rho'} \Diamond(\tau \wedge \varphi[\rho'/\rho] \wedge \rho')$$

(Here ρ' is not free in $\Diamond(\tau \wedge \downarrow^\pi_\rho \varphi)$.) With this at our disposal, the required extension of Lemma 16 is an obvious analog of our earlier work.

This has three more-or-less immediate consequences. First, the Existence Lemma follows exactly as in our earlier work. Second, some path nominal is true at every state. Third, for any path symbol, we can prove that the subset of a witnessed model consisting of all MCSs that contain that symbol is convex, maximal under R-successorship, and linear. There are one or two refinements of the earlier model construction that need to be made (for example, to complete the model we not only need to glue on an extra root node, we need to glue on a 'dummy successor' to the dummy root to ensure the R-maximality of the path symbol interpretations). However, the required ideas are pretty straightforward; our earlier work provides a scaffolding on which it possible to build a natural looking model.

So what's the problem? It's simple, but deadly: although all *states* are named, we don't have any guarantee that all *paths* are labeled by some path symbol. Nothing in our model construction guarantees this, and without it, the proof of the Truth Lemma does not go through. In other respects, the model

[30] We would like to stress that we are *not* claiming that this system is complete. Quite the reverse — we are using this system because it provides us with what is needed to work most of the way through a named model construction and then *fail*. In short provides us with just enough to illustrate why we have a genuine problem on our hand.

is frustratingly well-behaved: it provides a satisfying model for formulas that *don't* contain path symbols, and we can automatically add lots of first order properties to it as long as we do so using state-symbol axioms. In spite of this, it's not the model we want.

This is not easy to fix. For a start, it does not reflect our insistence on working with local languages. As we mentioned in the introduction, Robert Bull in his pioneering article on hybrid languages introduced the idea of path symbols and path binders. Now, Bull worked with a non-local language: he used the universal modality A, the binder $\forall$ over states (thus he had access to the full power of first-order quantification over states) and, instead of the local $\Downarrow^{\pi}$ binder he used the non-local binder $\forall^{\pi}$ that quantifies over arbitrary paths. In spite of this power, Bull runs into the same problem. He comments (see 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. A standard approach 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 our named 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 if there are realistic prospects of success here. Incidentally, we also think it would also be interesting to further explore the (finitary) rules used in Goranko (1996b).

And that's 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 that the potential interest of hybridization to a richer temporal ontologies is now clear.

8 Concluding remarks

In this paper we 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 *locality* assumption, namely that temporal operators and binders should only be able to examine, or bind to, temporally accessible states. We examined three extensions of ML+$\downarrow$, a local language in which *Until* was not definable, and showed that:

1. Adding $\Downarrow^1$, a universal quantifier over successor states, thus forming ML+$\downarrow$+$\Downarrow^1$, yielded a local language which could define *Until* and whose proof theory over transitive models could be axiomatized without complex rules of proof.

2. Adding the backward looking tense operators, thus shifting to TL+$\downarrow$, yielded a local language which could define both *Since* and *Until* (thus showing, as we remarked in Footnote 12, that these operators are definable in terms of past, *present*, and future) and whose minimal logic could be simply axiomatized. The completeness proof appealed to a complex rule of proof called *Paste*, but this was shown to be admissible.

3. Adding the retrieval operator @, thus forming ML+$\downarrow$+@, yielded a local hybrid language which could define *Until* and whose minimal logic had a simple axiomatization. The axiomatization hinged on a rule called *Paste-1*, a simple @-based version of the tense logical *Paste* rule. We posed the admissibility of *Paste-1* as an open problem.

In addition, we proved two general Sahlqvist-style results for extended logics; these results covered a number of temporally interesting properties including density and discreteness. Summing up, our technical investigations fulfill the first and second wishes listed at the end of Section 2, and (in our view) the third as well: we feel they exhibit a genuine synergy of modal and classical ideas.

It's only fair to warn the reader that we pay a price for this: the minimal logics discussed in this paper lack the finite model and are undecidable; this can be proved using the 'spy point' method introduced in Blackburn and Seligman (1995). 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.

This may be a price worth paying. Because these languages internalize the notion of 'label' in the object language, we are *not* restricted to axiomatic systems but can devise a wide range of proof systems; Seligman (1997) discusses natural deduction and sequent-based methods for global languages, and recent work shows that such methods work for local languages as well. Thus there are natural strategies for developing hybrid languages computationally, and in our view the opportunities offered by 'internalized labels' amply compensate for undecidability, but a full discussion is beyond the scope of the present paper.

Our conceptual argument in favor of hybridization is essentially a secular version of Prior's program of viewing 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 thinking about temporal reasoning over rich ontologies. A particularly intriguing kind of sorting involves sorts whose elements are structured sets of states; paths are perhaps the most important example. We showed (elaborating on Bull (1970)), that a path sort could be incorporated into local hybrid languages, that they offered a natural way of extending expressivity, and that they validated a number of the hybrid axiom schemas encountered in earlier sections. We showed that completeness poses difficult questions, and suggested avenues for further work.

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 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. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 14:56–83, 1993.

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

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

[7] 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.

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

[9] P. Blackburn and M. Tzakova. Hybridizing concept languages. Manuscript, 1998a.

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

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

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

[13] D. Gabbay. *Investigations in Modal and Tense Logics with applications to Problems in Philosophy and Linguistics*. Synthese Library, Volume 92. D. Reidel, 1976.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] J.A.W. Kamp. Formal properties of "now". *Theoria*, 37:227–273, 1971.

[21] 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.

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

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

[24] A. Prior. 'Now'. *Nous*, 2:101–119, 1968.

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

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

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

[28] 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.

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

[30] 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.

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

[32] F. Vlach. *Now and Then: A Formal Study in the Logic of Tense Anaphora.* PhD Thesis, UCLA, 1973.

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