What are Hybrid Languages?

Patrick Blackburn[*] Jerry Seligman[†]

Abstract

Hybrid languages exhibit two kinds of hybridisation. First, they combine the distinguishing features of modal logic and classical logic: although they have a Kripke semantics, they also make use of explicit variables and quantifiers that bind them. Second, they don't draw a syntactic distinction between terms and formulas: terms are part of the formula algebra, thus enabling the free combination of two different types of information. The goals of this paper are to introduce a number of hybrid languages, to discuss some of their fundamental logical properties (expressivity, decidability, and undecidability), and then, briefly, to indicate why such systems deserve further attention.

Although hybrid languages have a long and varied history, it is quite likely that most readers will know little, if anything, about them. This dictates the structure of this paper: before we can explain why we're interested in hybrid languages, we're going to have to explain what they are and give some insight into their capabilities, for only then will a discussion of broader motivational issues make much sense. We first introduce a number of important hybrid systems, discuss their expressive power, and prove two basic results: a decidability result, and an undecidability result. Only with this background safely behind us will we turn to the broader question: why bother looking at these systems anyway?

Hybrid tense logic

"Hybrid language" is a loose term covering a number of logical systems living somewhere between modal and classical logic. This is too vague to be helpful, so let's try to be a little more specific: any modal language can be 'hybridised' by adding variables over states and mechanisms for binding them. In this paper we are going to hybridise *tense logic* and see what emerges. So, to fix notation and terminology, let us briefly review the syntax and semantics of ordinary Priorean tense logic.

Given a denumerable infinite set PROP $= \{p, q, r, \ldots\}$ of *propositional symbols*, the well formed formulas of tense logic are defined as follows:

$$\text{WFF } \phi \ := \ p|\neg\phi|\phi \wedge \psi|F\phi|P\phi.$$

[*]Computerlinguistik, Universität des Saarlandes, D-66041 Saarbrücken, Germany. Email: patrick@coli.uni-sb.de

[†]Philosophy Institute, National Chung Cheng University, Min-Hsiung, Chia-Yi, Taiwan. Email: jerry@phil.ccu.edu.tw

It is usual to introduce the following notation for the duals of the F and P operators: $G\phi := \neg F \neg \phi$, $H\phi := \neg P \neg \phi$. The other boolean operators (such as $\rightarrow$, $\vee$, $\top$, and $\bot$) are treated as abbreviations. In what follows, we refer to the language of tense logic as TL. We shall also be paying a lot of attention to ML, the forward looking (or modal) fragment of this language. That is:

$$\mathrm{ML} \; = \; \{\phi \in \mathrm{TL} \mid \phi \; \text{ contains no occurrences of the } P \text{ operator}\}.$$

At the heart of the semantics for TL is a three-place relation $\Vdash$ between a model, a point in that model, and a wff. A *model* is a triple $(T, <, V)$ such that T is a non-empty set and $<$ is a binary relation on T. The pair $(T, <)$ is usually called a *frame*, and it plays the role of a 'flow of time': the elements of T are interpreted as *points* of time, and $<$ is interpreted as the binary relation of temporal precedence ('earlier than'). Initially, we do not constrain $<$ to give a reasonable model of the flow of time; it can be any binary relation. Later in the paper we shall insist that $<$ is a strict partial order. The third and final component of a model is a mapping V from PROP to $Pow(T)$. Intuitively, V specifies which (temporally relative) propositions hold at which times.

We can now define $\Vdash$, the fundamental semantical relation, inductively. Let $M = (T, <, V)$ and $t \in T$. Then:

$$\begin{array}{lll}
M, t \Vdash p & \text{iff} & t \in V(p), \text{ where } p \in \mathrm{PROP} \\
M, t \Vdash \neg\phi & \text{iff} & M, t \nVdash \phi \\
M, t \Vdash \phi \wedge \psi & \text{iff} & M, t \Vdash \phi \text{ and } M, t \Vdash \psi \\
M, t \Vdash F\phi & \text{iff} & \exists t' \in T (t < t' \;\&\; M, t' \Vdash \phi) \\
M, t \Vdash P\phi & \text{iff} & \exists t' \in T (t' < t \;\&\; M, t' \Vdash \phi)
\end{array}$$

If $M, t \Vdash \phi$ we say that ϕ is satisfied in M at t.

So far, everything should be familiar. We have a simple propositional language which we evaluate at points inside models. (That is, TL takes an internal perspective on relational structure.) And, in spite of TL's syntactic simplicity (it contains no variable binding apparatus) we can quantify over points: the operators F and P do this implicitly by 'scanning' accessible points for information. (Note that the two 'scanning clauses' in the definition of $\Vdash$ are defined in the metalanguage using a bounded first-order quantification over points.) Arguably, it is this combination of a propositional syntax, coupled with internal, locally scanning, method of talking about relational structures (in short, Kripke semantics), that is paradigmatic of modal logic. It is certainly what gives the study of modal systems — ranging from simple one operator systems to propositional dynamic logic — its characteristic flavour.

So what are hybrid languages? These emerge when we make *syntactic* concessions to classical logic, but retain the internal, local, *semantic* ideas characteristic of modal systems. In particular, when we hybridise TL we shall introduce a second sort of variable (these will range over points) and various mechanisms for binding them. Nonetheless, although the syntax will become more complex and (at least from one perspective) more 'classical', our semantics will remain firmly Kripkean.

To details. Syntactically, a hybrid tense language is formed when we specify a second denumerably infinite set $\mathrm{VAR} = \{x, y, z, u, v, w \ldots\}$, whose elements we call *variables*, some non-empty collection of *binders* (we employ B as a metavariable over binders), possibly augmented by a collection of additional sentential *operators*, which may be of any arity (we employ O^n as a metavariable over n-ary sentential operators). Given this, the general syntax of a hybrid tense logic has the following form:

$$\text{WFF } \phi \ := \ x|p|\neg\phi|\phi \wedge \psi|F\phi|P\phi|Bx\phi|O^1(\text{arg}_1)|\ldots|O^n(\text{arg}_1,\ldots,\text{arg}_n)|\ldots$$

Let's spell this out a little. First, and most fundamentally, any variable x in VAR is a wff in just the same way that a propositional symbol p in PROP is. That is, such expressions as $x \wedge p$, $\neg x$, Py, and $F(z \wedge p) \wedge F(z \wedge q) \rightarrow F(z \wedge p \wedge q)$ are well formed.

Second, if B is a binding operator, and x is a variable, then $Bx\phi$ is also a wff. Now, just as in classical logic, this clause forces us to draw a distinction between free and bound variables: prefixing Bx before a formula ϕ turns all the free occurrences of x in $x\phi$ (including the first one), into variables bound by the prefixed occurrence of B. We won't define these concepts here — experience with classical logic is a reliable guide — though we must mention that by a *sentence* we mean a wff containing no free variables. This paper is primarily concerned with sentences.

Third, if O^n is an n-ary operator, then prefixing O^n before n 'appropriate arguments' yields a wff. What's an appropriate argument? That depends on O^n. Arguments will always be wffs, but not all wffs may be appropriate. We will be examining one such additional operator later, namely @. This operator takes two arguments. Its second argument can be any wff, but the first must be a variable.

So much for syntax; what about semantics? This is a straightforward modification of Kripke semantics. First, as in classical logic, we interpret variables with the help of an assignment function. Then, we turn $\Vdash$ into a *four-place* relation in the obvious way: interpretation at a point in a model is conducted relative to some assignment function.

More precisely, if $g\colon \text{VAR} \longrightarrow T$ then g is called an *assignment of values to variables*, or an *assignment function*. (Thus, as promised, variables range over points of time.) We turn $\Vdash$ into a four-place relation as follows:

$$\begin{array}{lll}
M,g,t \Vdash x & \text{iff} & t = g(x), \text{ where } x \in \text{VAR} \\
M,g,t \Vdash p & \text{iff} & t \in V(p), \text{ where } p \in \text{PROP} \\
M,g,t \Vdash \neg\phi & \text{iff} & M,g,t \nVdash \phi \\
M,g,t \Vdash \phi \wedge \psi & \text{iff} & M,g,t \Vdash \phi \text{ and } M,g,t \Vdash \psi \\
M,g,t \Vdash F\phi & \text{iff} & \exists t' \in T(t < t' \ \& \ M,g,t' \Vdash \phi) \\
M,g,t \Vdash P\phi & \text{iff} & \exists t' \in T(t' < t \ \& \ M,g,t' \Vdash \phi)
\end{array}$$

Now, you're probably impatient to see how we're going to bind our variables and what sort of expressive power such binding leads to. Nonetheless, it's well worth pausing here, for in spite of the modesty of the fragment introduced so far, something quite significant — and decidedly non-classical — has already taken place.

Our variables range over times. Moreover, because of the first clause in the new semantic definition, variables are true at *precisely* the point they are assigned. In short, free variables are very like names for points.[1]

[1] Actually, the fragment whose semantics we have so far defined is nothing but *nominal tense logic* or *tense logic with names*. See Blackburn (1993) or Gargov and Goranko (1993) for further discussion of such systems.

Why is this significant? Well, up till now we have presented hybrid languages as if hybridisation simply amounted to grafting a classical variable binding syntax onto a modal semantics. Hybrid languages are certainly hybrid in this sense — but they are also hybrid in a second, perhaps deeper, respect. As we have defined them, hybrid languages permit two distinct sorts of information (namely propositional information, and names) to be mixed in one algebra: as formulas such as $F(p \wedge x)$ testify, both sorts of information can be freely combined. This is distinctly non-classical. In first-order logic, for example, terms and formulas are carefully segregated. Hybrid languages are more tolerant, and indeed, it is their 'sortal tolerance' that gives hybrid languages much of their technical and philosophical interest.

But let us resume our development of hybrid semantics. So far, we've only talked about free variables: it is high time we introduced some binders. We shall discuss two: $\exists$ and $\downarrow$.

The $\exists$ binder is the one most people think of when the idea of hybrid languages is explained. It's simply a 'classical' existential quantifier across times:

$$M, g, t \Vdash \exists x \phi \text{ iff } \exists g' (g \stackrel{x}{\sim} g' \ \& \ M, g', t \Vdash \phi).$$

As usual, $g \stackrel{x}{\sim} g'$ means that g' is an assignment of values to variables that agrees with g on all arguments save possibly x. Just as in classical logic, the semantic values of sentences are *not* dependent on the assignment of values to variables: if ϕ is a sentence, then there is an assignment g such that $M, g, t \Vdash \phi$ iff for *all* assignments g, $M, g, t \Vdash \phi$. In such a case we shall write $M, t \Vdash \phi$ and say that ϕ is satisfied in M at t.

This binder (and its dual $\forall x \phi := \neg \exists x \neg \phi$) has been introduced on several occasions, for quite distinct purposes. The earliest discussions are probably those of Prior (1967) Chapter V.6, Prior (1968), and Bull (1970) — indeed, these splendid discussions of hybrid tense logic are the earliest work on hybrid languages we know of. About fifteen years later (and independently of Prior and Bull) Passy and Tinchev hybridised propositional dynamic logic with the help of $\exists$. They remark that the idea of $\exists$ was suggested to them by Skordev, who in turn was inspired by certain investigations in recursion theory; the paths that have lead to hybrid languages are indeed many and varied. Passy and Tinchev (1991) gives an excellent overview of this line of work and its connections with extended modal logic.

But for all it's naturalness, $\exists$ is by no means the only binder of interest. The intrinsic locality of the $\downarrow$ binder make it a prime candidate for further investigation:

$$M, g, t \Vdash \downarrow x \phi \quad \text{iff} \quad M, g', t \Vdash \phi, \quad \text{where} \quad g \stackrel{x}{\sim} g' \ \& \ g'(x) = t.$$

That is, $\downarrow$ 'names the here-and-now'; or (more accurately) it stores the point of evaluation at the location labeled x. If ϕ is a sentence, we write $M, t \Vdash \phi$ iff there is an assignment g (equivalently: for all assignments g) $M, g, t \Vdash \phi$.

Like $\exists$, the $\downarrow$ binder has been independently invented on a number of occasions. For example, Richards et. al. (1989) introduce $\downarrow$ as part of an investigation into temporal semantics and temporal databases, Sellink uses it to aid reasoning about automata, and Cresswell (1990) uses it as part of his treatment of indexicality. Nonetheless, none of the systems just mentioned allows the free syntactic interplay of variables with the underlying propositional logic; the earliest paper to introduce $\downarrow$ into a fully hybrid language seems to be Goranko (1994). Other papers investigating $\downarrow$ in hybrid settings include Blackburn and Seligman (1995), Goranko (1996a) and Goranko (1996b).

The two binders at our disposal are more than enough for this paper, so let's turn to the final syntactic possibility we have allowed ourselves: are there additional operators which work naturally with our binders?

Indeed there are. In computational terms, binders are essentially *storage* mechanisms: they 'save' points of the frame in 'memory' (the assignment function) at the 'location' named by the bound variable. Where there is storage it is natural to have *retrieval*, thus we shall introduce a two place operator @ whose first argument is a variable and whose second argument is an arbitrary wff. That is, @ forms wffs of the forms $@(x, \phi)$ (pronounced: 'At x ϕ'), though in what follows we shall minimise brackets by writing the first argument as a subscript, thus the previous formula would be written $@_x\phi$. Two further syntactic remarks are in order. First, the @ operator is *not* a binder. Occurrences of variables in the scope of @ may well be bound but only by the binding operators $\exists$ and $\downarrow$, not by @. For example, in $@_w\exists x(@_x{\downarrow}y(Fw \rightarrow F({\downarrow}z(@_yFz)))$ the only free variable is w. Second, note that as variables are wffs, the second argument of @ may itself be a variable, thus expressions such as $@_xy$ and $@_xx$ are wffs.

Given the previous discussion, and the reading 'At x ϕ', it will come as no surprise to learn that we interpret our retrieval operator as follows:

$$M, g, t \Vdash @_x\phi \quad \text{iff} \quad M, g, g(x) \Vdash \phi.$$

That is, @ retrieves the point stored at location x and evaluates ϕ at this point.

This retrieval operator may well be familiar to you. For a start, $@_x\phi$ is just Prior's $T(x, \phi)$ construct, which he used to define his 'third grade tense logics'. It is also the $\mathbf{Holds}(x, \phi)$ operator introduced by Allen (1984) for temporal representation in AI.[2] Finally, the same operator is used in the 'Topological Logic' of Rescher and Urquhart (1971). However, although all use @ in various guises, none of the systems just mentioned is a hybrid language in our sense. Indeed, as far as we know, the only published work on @ in the setting of hybrid languages are the proof theoretical investigations of Seligman (1991,1994).

We now have an ample supply of hybrid languages and sublanguages at our disposal, and much of the present paper is devoted to examining the properties of various combinations of $\exists$, $\downarrow$, and @, over both TL and ML. We shall use a self explanatory notation to indicate the various sublanguages. For example, TL $+ \exists + \downarrow +$ @ is the richest hybrid language so far defined; it is the language of tense logic enriched by both binders plus @. On the other hand, ML $+ \downarrow$ is a relatively modest sublanguage of this system. It consists of the forward looking fragment of TL enriched with the binder $\downarrow$.

Remarks on expressivity

As a warming up exercise, let's briefly examine the expressive strengths and weaknesses of our hybrid systems over *arbitrary* structures. (Most of the observations gathered here are discussed more fully in Blackburn and Seligman (1995).)

[2] Allen introduced his **Holds** operator in the setting of interval-based, not point-based, temporal logic. By hybridising a suitable modal logic of intervals, it is straightforward — and in our view, instructive — to reconstruct Allen's system in hybrid terms.

To begin with, a rather obvious but fundamental point: hybrid languages really do reside somewhere between modal and classical logic.

Recall that TL can be viewed as a fragment of a certain first-order language, namely its *correspondence language* (see van Benthem (1984)). This correspondence language contains a binary relation symbol $<$, a denumerably infinite collection of one-place symbols P, Q, R, and so on (these correspond in the obvious way to the elements p, q, r and so on of PROP — indeed, we shall simply take the elements of PROP to be these predicate symbols), boolean operators, a denumerably infinite collection of first-order variables (to keep life simple, we'll simply use the hybrid variables VAR here) and the familiar first-order quantifier symbols $\exists$ and $\forall$ (so we're overloading these symbols, using them both in the first-order and the hybrid setting; this won't prove confusing in practice). Any tense logical model $M = (T, <, V)$ can be regarded as first-order model for the correspondence language: the relation $<$ interpets the symbol $<$, and for all $p \in$ PROP, the subset $V(p)$ interprets the unary predicate symbol P. The well known *standard translation* then shows that TL can be regarded as a fragment of the correspondence language. Typical clauses of the translation are:

$$\begin{array}{lcl} ST_x(p) & = & Px \\ ST_x(\neg\phi) & = & \neg ST_x(\phi) \\ ST_x(F\phi) & = & \exists y(x < y \wedge ST_y(\phi)), \ \ y \text{ a fresh variable.} \end{array}$$

Clearly, for any formula $\phi \in$ TL, $ST_x(\phi)$ is a first-order formula containing exactly one free variable (namely x) in which all quantifiers are bounded. Moreover, it is more or less immediate that $M, t \Vdash \phi$ iff $M \models ST_x(\phi)[t]$. (Here $M \models ST_x(\phi)[t]$ means that the model M, viewed as a first-order model, satisfies the the first-order formula $ST_x(\phi)$ when t is assigned as the denotation of its single free variable x.)

The standard translation extends to all the hybrid apparatus we have introduced:

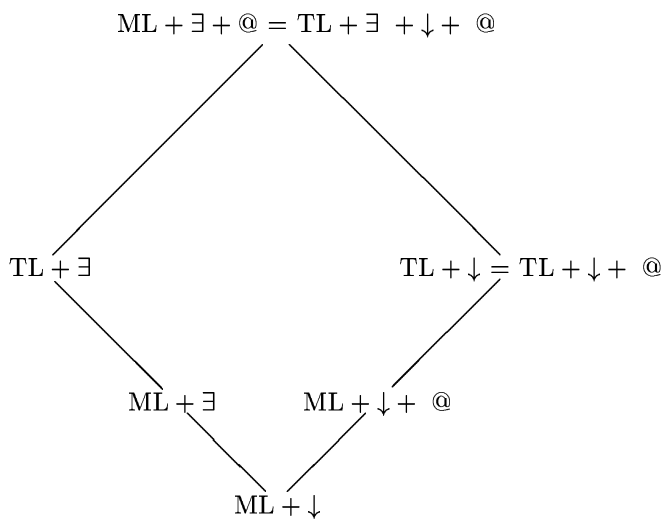
$$\begin{array}{lcl} ST_x(y) & = & x = y, \ \text{for all variables } y \\ ST_x(\exists y\phi) & = & \exists y ST_x(\phi)) \\ ST_x(\downarrow y\phi) & = & \exists y(x = y \wedge ST_x(\phi)) \\ ST_x(@_y\phi) & = & ST_y(\phi). \end{array}$$

Note that, just like formulas of ordinary tense logic, *sentences* of hybrid languages translate into formulas of the correspondence language containing one free variable. The translation clearly preserves satisfaction, so our hybrid languages are not stronger than the one free variable fragment of the correspondence language.

Now a second, perhaps less obvious, point: over arbitrary structures, most of the hybrid systems and subsystems we have introduced are strictly *weaker* than the correspondence language. Here's the simplest example: sentences of ML $+\downarrow$ are preserved under the formation of *forward-generated submodels*. Given a model $(T, <, V)$ and a point t of T, the submodel *forward-generated* (henceforth: *f-generated*) *by* t is the smallest submodel $(S, <^s, V^s)$ of $(T, <, V)$ such that $t \in S$ and for all $s \in S, s' \in T$, if $s < s'$ then, then $s' \in S$. (More concretely, the submodel f-generated by t contains just those points accessible from t by a finite number of transitions in the forward direction of the $<$ relation; t is often called the *root* of this model.) It is a standard result that the satisfaction of ML formulas is preserved under the formation of f-generated submodels, and it only remains to observe that any points 'stored' by $\downarrow$ in the course of evaluation must be points in the f-generated submodel. It is straightforward to turn this observation into an inductive proof. Now, there are one-free-variable formulas of the correspondence language that are *not* preserved under this construction (for example, $Qx \wedge \exists y Py$), and so $\mathrm{ML} + \downarrow$ is the strictly weaker language.

As a second example, consider $\mathrm{ML} + \exists$. Now, sentences of this language need not be preserved under f-generated submodels, but they *are* preserved under the following transformation. Let $M = (T, <, V)$ be a model, and suppose that $M^s = (S, <^s, V^s)$ is a f-generated submodel of M such that $card(T \backslash S) \geq 1$. Let M^{s+*} be the model obtained by adding a new point $*$ to M^s in such a way that $*$ is disconnected from any point in S. (We can make $*$ itself either reflexive or irreflexive; it's completely irrelevant.) Now, for any sentence of $\mathrm{ML} + \exists$, and any $s \in S$, it is easy to show that $M, s \Vdash \phi$ iff $M^{s+*}, s \Vdash \phi$. The point is this. Unlike $\downarrow$, the $\exists$ binder is *not* local; it may well bind variables to points outside the submodel f-generated by the point of evaluation. Nonetheless, though it can 'detect' the existence of such distant points, it *can't* scan the information there. Collapsing all such points to single point $*$ does no damage at all.

In fact, as the following diagram shows, we have a genuine expressivity hierarchy.[3] Let's briefly explore it.



First, even languages low in the hierarchy are very expressive. For example, the $\mathrm{ML} + \downarrow$ sentence $\downarrow x \neg F x$ is satisfiable only at irreflexive points — and as is well known, no formula of ML or TL has this property. When we move one step higher to $\mathrm{ML} + \downarrow + @$, the *Until* operator becomes definable:

$$Until(\phi, \psi) \ := \ \downarrow x F \downarrow y(\phi \wedge @_x G(Fy \rightarrow \psi)).$$

Next observe that — as the diagram shows — the $\exists$ binder is strictly stronger than $\downarrow$. Given $\exists$ we can define $\downarrow x \phi$ to be $\exists x(x \wedge \phi)$, so $\exists$ is at least as strong as

[3] The '=' symbol is used to denote expressive equivalence: roughly, that the languages can be used to define the same sets of points in all models. A precise definition of this relation is given in Blackburn and Seligman (1995).

$\downarrow$. The fact that $\downarrow$ sentences are preserved under f-generated submodels while $\exists$ sentences are not shows that it is *strictly* stronger.

The diagram also tells us that $\mathrm{TL}+\downarrow = \mathrm{TL}+\downarrow+@$. Why is this? Essentially it is due to two factors. First, $\downarrow$ binds locally. Second, because in TL we can look both forwards and backwards along $<$, we don't need @ to retrieve stored values. An example will make this clear. Consider the definition of *Until* given above:

$$Until(\phi, \psi) \ := \ \downarrow x F \downarrow y(\phi \wedge @_x G(Fy \rightarrow \psi)).$$

There is an obvious analog in $\mathrm{TL} + \downarrow$, namely:

$$Until(\phi, \psi) \ := \ \downarrow x F \downarrow y(\phi \wedge P(x \wedge G(Fy \rightarrow \psi))).$$

Here it is the P operator that retrieves the stored value of x; note that the occurrence of P 'inversely matches' the first occurrence of F. In general, occurrences of @ can be (rather cumbersomely) replaced by appropriate sequences of tense operators, and hence $\mathrm{TL} + \downarrow = \mathrm{TL} + \downarrow + @$.

Finally, let's take a look at the top of the diagram. Why is $\mathrm{ML} + \exists + @ = \mathrm{TL} + \exists + \downarrow + @$. First, as we've already seen, $\downarrow$ is definable using $\exists$, thus $\mathrm{TL} + \exists + \downarrow + @ = \mathrm{TL} + \exists + @$. Next, note that this high up in the hierarchy we don't need to have a primitive P operator at our disposal. To see this, first define the "somewhere" modality:

$$E\phi \ := \exists y @ y \phi.$$

We can then define P as follows:

$$P\phi \ := \ \downarrow x E(\phi \wedge Fx).$$

Thus $\mathrm{TL} + \exists + @ = \mathrm{ML} + \exists + @$.

In fact, at the top of the diagram we have everything we need to capture the correspondence language. The key point to observe is the smooth way $\exists$ and @ work together. As we've already noted, $\exists$ can detect distant points, but can't scan the information they contain; adding @ removes this weakness. Once this has been realised, it becomes clear how to define a satisfaction-preserving *hybrid translation* from first-order logic to $\mathrm{ML}+\exists+@$. Here are its key clauses:

$$\begin{array}{lcl} HT(y = z) & = & \downarrow x E(y \wedge z) \\ HT(Py) & = & \downarrow x E(y \wedge p) \\ HT(y < z) & = & \downarrow x E(y \wedge Fz) \end{array}$$

Note that if the input first-order formula ϕ contains only the variable x free, $HT(\phi)$ is a sentence.

A decidability result

What we have so learnt can be summed up as follows: adding variables over states and variable binding apparatus to modal languages need *not* automatically catapult them to full correspondence strength — though if full first order strength is required it can be obtained by adding such combinations as $\exists + @$ or $\downarrow + E$. This is interesting — but a further observation follows hot on its heels. Hybrid languages are very strong, and over arbitrary structures this has the following consequence: even $\mathrm{ML} + \downarrow$ is *undecidable*.[4]

What happens over other classes of structures? And, given that we are working in the language of tense logic, can we do better over the class of *strict partial order* (SPOs)? Recall that a strict partial order is a binary relation $<$ that is both transitive and irreflexive. SPOs clearly capture some of the more fundamental intuitions about 'flows of time', but they don't constrain the concept too tightly (the class contains both linear, non-linear, discrete and continuous frames) so it's an important class for temporal logic. Now, as the reader can easily verify, restricting our attention to the class of SPOs does *not* alter our expressivity hierarchy in any way: the arguments involving f-generated submodels and collapsings, for example, go through unchanged.[5] However, as we shall now show, the restriction to SPOs has the following effect: the satisfiability problem for $\mathrm{ML} + \downarrow$ sentences becomes decidable.

How are we to prove this? We can't do so using finite model property arguments, for while $F\top \wedge GF\top$ is SPO-satisfiable, it's not *finite*-SPO-satisfiable. Instead, we shall use the other classic modal decidability technique, namely Gabbay-style reductions to decidable second-order theories of countably branching trees (see Gabbay (1976)). Now, such proofs typically involve an 'unravelling' step, and in our case we are going to need to unravel SPOs.[6] But how do we know that this is a satisfiability preserving operation for $\mathrm{ML} + \downarrow$ sentences? We'll provide a general answer to this question by introducing the notion of *quasi-injective bisimulations*.

Definition 1 (Bisimulations) *A* bisimulation *is a non-empty binary relation Z between two models M_1 and M_2 such that:*

1. *For all points u in M_1 and v in M_2, if uZv then u and v satisfy the same propositional symbols.*

2. *For all points u, u' in M_1 and v in M_2, if $u < u'$ and uZv then there is a point v' in M_2 such that $v < v'$ and $u'Zv'$.*

3. *For all points v, v' in M_2, and u in M_1, if $v < v'$ and uZv then there is a point u' in M_1 such that $u < u'$ and $u'Zv'$.*

Definition 2 (Mutual inaccessibility) *Points t and t' in a model $M = (T, <, V)$ are* mutually inaccessible *iff t is not in the submodel f-generated by t' and t' is not in the submodel f-generated by t.*

For SPOs, mutual inaccessibility is just incompatibility, in the usual order-theoretic sense.

Definition 3 (Quasi-injective bisimulations) *Let Z be a bisimulation between models M_1 and M_2. Then Z is a* quasi-injective bisimulation *iff:*

[4] There is no published proof of precisely this result, but the 'spy point' argument of Blackburn and Seligman (1995) can be adapted to yield it. Valentin Goranko (personal communication) has also proved the result via a spy-point argument.

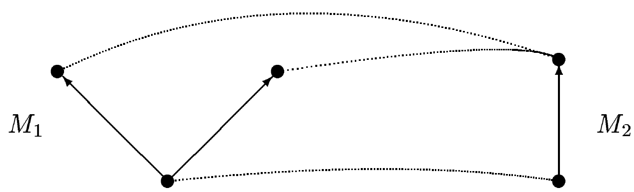
[5] By way of contrast, note that over strict *total* orders (that is, strict linear orders), the hierarchy shrinks: $\mathrm{TL} + \downarrow$ swallows the correspondence language. To see this, note that on linear structures, $E\phi$ is definable as $P\phi \vee \phi \vee F\phi$, $@_x\phi$ is definable as $E(x \wedge \phi)$, and $\exists x\phi$ is definable as $\downarrow yE\downarrow xE(y \wedge \phi)$.

[6] For a discussion of unravelling and its applications in modal logic, see Bull and Segerberg (1984).

1. *For all points u, u' in M_1, and v in M_2, if uZv and $u'Zv$, and $u \neq u'$ then u and u' are mutually inaccessible, and*

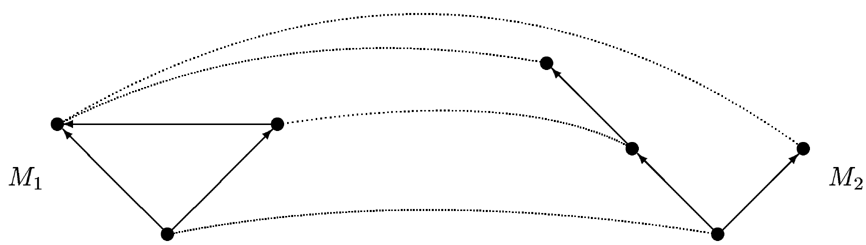
2. *For all points v, v' in M_2, and u in M_1, if uZv and uZv', and $v \neq v'$ then v and v' are mutually inaccessible.*

Here's a simple example (the dotted lines indicate the bisimulation):



This bisimulation identifies the two dead-ends in M_1 with the single dead-end in M_2. This is fine because the dead-ends in M_1 are mutually inaccessible.

As a second example, note that the technique of 'unravelling' can sometimes be analysed using quasi-injective bisimulations. In particular, the following 'triangle' can be unravelled to the four point 'tilted capital L':



In fact the previous example points to the key observation underlying the proof of decidability. As is well known, when we unravel a rooted model around its root, the model is a bounded morphic image of (and hence bisimilar to) its unravelling. But this doesn't help us: we've already met an example that shows that the satisfiability of $\mathrm{ML}+\downarrow$ sentences is *not* preserved under ordinary bisimulations: $\downarrow x \neg Fx$ is satisfiable only at irreflexive points, but irreflexivity is not preserved under bisimulations (a standard counterexample is the bisimilarity of $(\omega, <)$ to a single reflexive point). However, as you can easily check, if we start with a rooted SPO, then not only is it bisimilar to its unravelling, but the usual bisimulation is quasi-injective — and, as we shall now see, quasi-injective bisimulations *do* preserve the satisfiability of $\mathrm{ML}+\downarrow$ sentences.

Lemma 1 *Let Z be a quasi-injective bisimulation between models M_1 and M_2, let u and v be points in M_1 and M_2 respectively such that uZv, and let g_1 and g_2 be assignments on M_1 and M_2 respectively such that for all variables x:*

1. *$g_1(x)Zv'$ iff $v' = g_2(x)$, for all v' in the submodel f-generated by v, and*

2. *$u'Zg_2(x)$ iff $u' = g_1(x)$, for all u' in the submodel f-generated by u.*

In other words, the restriction of Z to the ranges of g_1 and g_2 is required to be a one-one correspondence, with $g_1(x)Zg_2(x)$ for each variable x.

Then, for all wffs ϕ in $\mathrm{ML}+\downarrow$, $M_1, g_1, u \Vdash \phi$ iff $M_2, g_2, v \Vdash \phi$.

Proof. By induction on the length of ϕ. The case for propositional symbols follows because Z is a bisimulation, and bisimilar points agree on propositional information. The case for variables follows because of restrictions placed on g_1 and g_2. The case for booleans is clear. Moreover, the case for formulas of the form $F\psi$ is (almost) standard. The fact that Z is a bisimulation drives the argument through in the familiar fashion; the only additional observation we need to make is that the inductive hypothesis applies to any u' such that $u < u'$ because the submodel f-generated by any such u' is a submodel of that f-generated by u.

The crucial case is for wffs of the form $\downarrow x\psi$. So, for the left to right direction, suppose $M_1, g_1, u \Vdash \downarrow x\psi$. Then $M_1, h_1, u \Vdash \psi$, where $h_1(x) = u$, and $h_1 \overset{x}{\sim} g_1$. Define an assignment h_2 on M_2 by $h_2(x) = v$, and $h_2 \overset{x}{\sim} g_2$. We have to show that h_1 and h_2 satisfy the inductive hypothesis; that is, we have to show that:

1. $h_1(z)Zv'$ iff $v' = h_2(z)$, for all v' in the submodel f-generated by v, and
2. $u'Zh_2(z)$ iff $u' = h_1(z)$, for all u' in the submodel f-generated by u.

If $z \neq x$ the claim is trivial, for g_1 and g_2 satisfy the inductive hypothesis, and new h assignments differ from the old g assignments only in what they assign to x. On the other hand, if $z = x$ then $h_1(z) = u$ and $h_2(z) = v$, and so we must show that (1) uZv' iff $v' = v$, for all v' in the submodel f-generated by v, and (2) $u'Zv$ iff $u' = u$, for all u' in the submodel f-generated by u. Now the left to right direction of (1) is clear, since by assumption uZv. For the right to left direction note that if v' is in the submodel f-generated by v, then v and v' are *not* mutually inaccessible. So since uZv, by the quasi-injectivity of Z, if uZv' then $v' = v$. The argument for (2) is similar. In short, the induction hypothesis holds for h_1 and h_2, hence $M_2, h_2, v \Vdash \psi$, and hence $M_2, g_2, v \Vdash \downarrow\psi$ as required. The right to left direction is similar. $\dashv$

Theorem 1 *Let Z is a quasi-injective bisimulation between models M_1 and M_2, and let u and v be points in M_1 and M_2 respectively such that uZv. Then for all* sentences *of* ML $+\downarrow$, *$M_1, u \Vdash \phi$ iff $M_2, v \Vdash \phi$.*

Proof. For the left to right direction, suppose that ϕ is an ML $+\downarrow$ sentence such that $M_1, u \Vdash \phi$. As we have already remarked, the satisfiability of ML $+\downarrow$ sentences is preserved under the formation of f-generated submodels, thus $S_1, u \Vdash \phi$ where S_1 is the submodel of M_1 f-generated by u. Let g_1 be the assignment on M_1 such that $g_1(z) = u$ for all variables z; trivially $M_1, g_1, u \Vdash \phi$.

Now, let S_2 be the submodel of M_2 f-generated by v. As uZv, the restriction of Z to S_1 and S_2 is a quasi-injective bisimulation between these submodels. Let g_2 be the assignment on S_2 such that $g_2(z) = v$ for all variables z. Then by the quasi-injectivity of Z, for all variables z, $g_1(z)Zv'$ iff $v' = v = g_2(z)$, for all v' in the submodel f-generated by v; and $u'Zg_2(z)$ iff $u' = u = g_1(z)$, for all u' in the submodel f-generated by u. Thus, by the previous lemma, $S_2, g_2, v \Vdash \phi$. But ϕ is a sentence, and so $S_2, v \Vdash \phi$, and (again using the fact that ML$+\downarrow$ sentences are preserved under the formation of f-generated submodels) $M_2, v \Vdash \phi$ as required. The converse is similar. $\dashv$

Theorem 2 *Determining whether any given sentence of* ML $+\downarrow$ *is satisfiable on some strict partial order is a decidable problem.*

Proof. We prove decidability by reducing the ML $+\downarrow$ satisfiability problem over SPOs to the membership problem for $S\omega S$. Recall that $S\omega S$ is the monadic second order theory of N_ω, the countable infinite tree in which every node has ω successors; Rabin (1969) shows that membership in $S\omega S$ is decidable. Our reduction has four main steps.

First, if a sentence ϕ of ML $+\downarrow$ is SPO-satisfiable then it is satisfiable on a rooted, countable, SPO. To see this simply observe that if ϕ is satisfiable in some strictly partially ordered model M_0, then so is $ST_x(\phi)$. By the Löwenheim-Skolem theorem, there is a countable SPO M_1 that satisfies $ST(\phi)$ and hence ϕ. The submodel M_2 of M_1 that is f-generated by the point satisfying ϕ is the rooted, countable, strictly partially ordered model for ϕ we want.

Second, if ϕ is satisfiable on a rooted, countable SPO then it is satisfiable on a countable tree M_3: to form M_3 simply unravel M_2 around its root. M_3 is quasi-injectively bisimilar to M_2 and hence, by the previous theorem, a model for ϕ.

Third, we can isomorphically embed M_3 as initial section of N_ω, the infinite tree in which every node has ω successors. The required embedding f can be inductively defined as follows. First, f maps the root of M_3 to the root node of N_ω. Second, suppose that $f(t) = m$, but that f is not defined on the daughters of t. As M_3 is a countable model, the daughters of t can be enumerated. Then we extend f by insisting that for each daughter t' of t, if t' is k-th in the enumeration then $f(t')$ is the k-th successor of m.

Fourth, not only are all the models we need essentially submodels of N_ω, they are even submodels definable in $S\omega S$. Indeed, we shall now define a function *Sat*-ML $+\downarrow$ such that for any sentence ϕ of ML $+\downarrow$, *Sat*-ML $+\downarrow(\phi)$ is a sentence in the language of $S\omega S$ that is satisfiable in N_ω iff ϕ is satisfiable.

This is rather easy. Let $\leq$ denote the relation of 'dominance' in N_ω, and let $<$ denote the relation of 'proper dominance' in N_ω; $\leq$ is a primitive symbol in the language of $S\omega S$ and $<$ is definable. Note that $ROOT(x) \;:=\; \neg\exists y(y < x)$ is predicate true of precisely the root of N_ω. Using $ROOT$ and $\leq$ we can define a predicate *INIT*, containing one free set variable S, which picks out the initial sections of N_ω:

$$INIT(S) \;:=\; \exists y(ROOTy \wedge Sy) \wedge \forall z \forall u((Sz \wedge u \leq z) \rightarrow Su)).$$

Next, we define a predicate $<_S$, containing a free set variable S, that defines the restriction of $<$ to a subset S of N_ω, namely $<_S \;:=\; Sx \wedge Sy \wedge x < y,$. Using these definitions, it is straightforward to define a translation T^ω from ML $+\downarrow$ to the language of $S\omega S$:

$$\begin{array}{lcl}
T^\omega_{x,S}(p) & = & Px \\
T^\omega_{x,S}(\neg\phi) & = & \neg T^\omega_{x,S}(\phi) \\
T^\omega_{x,S}(\phi \wedge \psi) & = & T^\omega_{x,S}(\phi) \wedge T^\omega_{x,S}(\phi) \\
T^\omega_{x,S}(F\phi) & = & \exists y(x <_S y \wedge T^\omega_{y,S}(\phi)) \\
T^\omega_{x,S}(\downarrow y\phi) & = & \exists y(x = y \wedge T^\omega_{x,S}(\phi))
\end{array}$$

This translation is an obvious variant on the standard translation.

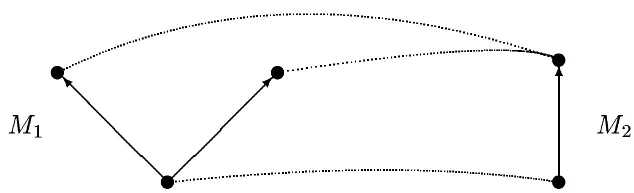
We now have everything we need. So, let ϕ be a sentence of ML $+\downarrow$ constructed from the propositional symbols $p_1, \ldots, p_n$. Define *Sat*-ML $+\downarrow(\phi)$ to be the following sentence:

$$\exists S \exists P_1 \cdots \exists P_n \exists x(INIT(S) \wedge \textstyle\bigwedge_{1 \leq i \leq n} \forall z(P_i z \rightarrow Sz) \wedge Sx \wedge T^\omega_{x,S}(\phi)).$$

Recall that $T^{\omega}_{x,S}(\phi)$ contains free occurrences of S (in $<_S$) and x; these become bound in the above sentence. Bearing this in mind, it should be clear that this sentence asserts the existence of an initial section S of N_ω, a collection of n subsets P_i of this subtree, and a point x in the subtree, that satisfy the translation of ϕ. In short, it says there is a countable tree-like model for the (translation of) ϕ, and we have reduced the ML $+\downarrow$-satisfiability problem to the SωS membership problem, which by Rabin's celebrated result is decidable. $\dashv$

An undecidability result

What happens if we move one step up in the hierarchy? Adding @ yields a far more discriminating language. For a start, not all sentences ML $+ \downarrow +$ @ are preserved under quasi-injective bisimulations, as the following example shows:



The sentence $\downarrow x F \downarrow y @_x F \neg y$ true at the root of M_1 (it correctly detects the presence of two distinct successors) but false at the root of M_2. However, as we shall now show, we pay a price for this expressivity: even when we confine our attention to SPOs, ML $+ \downarrow +$ @ has an undecidable satisfiability problem. We shall prove this using a reduction from the unbounded tiling problem.

A tile t is a 1×1 square, of fixed orientation, with coloured edges. The colours are drawn from some denumerable set, and for any tile t we'll use the notation $\mathit{right}(t)$, $\mathit{left}(t)$, $\mathit{up}(t)$, and $\mathit{down}(t)$ to indicate the colours of its four edges. The unbounded tiling problem is this: given a finite set $\mathcal{T}$ of tile types, can one cover $\omega \times \omega$ with tiles of these types in such a way that adjacent tiles have the same colour on their adjacent edges? More precisely, does there exist a function *tile* from $\omega \times \omega$ to $\mathcal{T}$ such that $\mathit{right}(\mathit{tile}(n,m)) = \mathit{left}(\mathit{tile}(n+1,m))$, and $\mathit{up}(\mathit{tile}(n,m)) = \mathit{down}(\mathit{tile}(n,m+1))$? This problem is known to be undecidable; see Harel (1983) for further discussion.

We want to reduce the unbounded tiling problem to the problem of deciding whether a sentence of ML$+\downarrow+$@ is satisfiable on a SPO. Essentially, for any finite set of tiles $\mathcal{T} = \{t_1, \ldots, t_i, \ldots t_k\}$, we are going to construct a formula $\phi^{\mathcal{T}}$ which performs two main tasks. First, it forces any strictly partially ordered model satisfying it to be appropriately 'grid-like'. Second, it forces a special group of propositional variables to act like tiles whose colours match on adjacent edges. We shall be able to show that $\phi^{\mathcal{T}}$ is satisfiable on an SPO iff $\mathcal{T}$ tiles $\omega \times \omega$; hence, as the tiling problem is undecidable, the satisfiability problem must be too.

How should we carry out the first task? And what does it mean for an SPOed model to be gridlike? As a guide, think of $\omega \times \omega$ under the ordering $<_g$ defined by $(n,m) <_g (n',m')$ iff either $n <_\omega n'$ and $m \leq_\omega m'$, or $m <_\omega m'$ and $n \leq_\omega n'$. (Here $<_\omega$ is the ordinary 'less-than' ordering on ω and $\leq_\omega$ is the ordinary 'less-than-or-equal' ordering.) Pictorially, $(n,m) <_g (n',m')$ iff

(n', m') is a pair in the upper-left quadrant f-generated by (n, m). Note that $<_g$ is a strict partial order.

Now, $<_g$ has two very interesting properties: every pair (n, m) has two $<_g$-minimal-successors, namely $(n+1, m)$ and $(n, m+1)$; and these two minimal successors share a common minimal successor, namely $(n+1, m+1)$. We shall call a strict partial order $<$ *gridlike* if it has these two properties. The heart of the undecidability proof is the observation that ML $+ \downarrow +$ @ can force the existence of gridlike SPOs.

And so to work. Given a finite set of tiles $\mathcal{T} = \{t_1, \ldots, t_i, \ldots t_k\}$ we shall take as our set of propositional symbols $\mathcal{T} \cup \{0, 1, 2\}$. We will make heavy use of the following abbreviation: $G^+\psi \ := \ \psi \wedge G\psi$. Note that if $G^+\psi$ is satisfied at some point t in a strictly partially ordered model, then ψ holds at t and at every point in the submodel f-generated by t.

The formula $\phi^{\mathcal{T}}$ we wish to construct will have the following form:

$$\phi^{min} \wedge \phi^{share} \wedge \phi^{mod\text{-}3} \wedge \phi^{match}.$$

The first two conjuncts forces the existence of the desired gridlike SPO, the third draws the distinction between vertical and horizontal, while the last covers it with colour-matching tiles. We introduce the required conjuncts in four steps.

Step 1. Let ϕ^1 be the following sentence:

$$\begin{aligned}
&{\downarrow} x F {\downarrow} y(\neg x \wedge @_x F {\downarrow} z(\neg x \wedge \neg y \wedge @_x(\neg FFy \\
&\qquad \wedge \neg FFz \\
&\qquad \wedge G{\downarrow} w(@_x \neg FFw \to y \vee z) \\
&\qquad \wedge G{\downarrow} w(\neg y \wedge \neg z \to (@_y Fw \vee @_z F_w)))))
\end{aligned}$$

If ϕ^1 is satisfied at some point in an SPO (the point labeled x) then it has exactly two minimal successors (namely the points labeled by y and z). Accordingly, we define:

$$\phi^{min} \ := \ G^+\phi^1$$

.

Step 2. It is handy to have a modality to access minimal successors, so we define:

$$\langle min \rangle \psi \ := \ {\downarrow} x F {\downarrow} y(\psi \wedge @_x \neg FFy).$$

With the aid of this modality, it is easy to insist that minimal successors share a common minimal successor. As a first step we define:

$$\phi^2 \ := \ {\downarrow} x \langle min \rangle {\downarrow} y \langle min \rangle {\downarrow} u @_x \langle min \rangle {\downarrow} z(\neg y \wedge \langle min \rangle u).$$

If this formula is satisfied at some point (the one labeled x), then it has two distinct minimal successors (labeled y and z respectively) and these two minimal successors share a common minimal successor (the point labeled u). Accordingly we define:

$$\phi^{share} \ := \ G^+\phi^2$$

.

Step 3. We're not quite ready to place the tiles in position, for although we now have gridlike structure, it is too symmetric: when $\omega \times \omega$ is ordered by $<_g$ there is an obvious distinction between vertical and horizontal, but this is not mirrored in our gridlike model in any obvious way. However, by using a technique from Spaan (1993), we can impose the required distinction rather elegantly.

First, let ϕ^3 be a formula that is satisfied at a point iff that point satisfies exactly one of the propositional variables 0, 1 or 2. By insisting that $G^+\phi^3$ be satisfied we obtain a submodel in which every point is labeled by exactly one of these numbers.

Next, suppose that a point t in our model is labeled by the number a, where $0 \leq a \leq 2$. Let $+_3$ denote addition mod 3. We're going to think of a one-step vertical move from t as a move to a minimal successor of t that is labeled $a +_3 1$, and we'll think of a one-step horizontal move from t as a move to a minimal successor of t that is labeled $a +_3 2$. Let's define modalities embodying these idea:

$$\begin{aligned}\langle up\rangle\psi \quad &:= \quad \textstyle\bigwedge_{0\leq a\leq 2}(a \rightarrow \langle min\rangle((a +_3 1) \wedge \psi))\\ \langle right\rangle\psi \quad &:= \quad \textstyle\bigwedge_{0\leq a\leq 2}(a \rightarrow \langle min\rangle((a +_3 2) \wedge \psi))\end{aligned}$$

We now define:

$$\phi^{mod\text{-}3} \;:=\; G^+\phi^3 \wedge G^+(\langle up\rangle\top \wedge \langle right\rangle\top).$$

This guarantees that each point is labeled by a number less than three, and that each point has both a vertical and horizontal immediate successor. Note that if a point t is labeled by the number a, then the shared common successor of t's two immediate successors is also labeled a.

Step 4. It is now easy to force the desired tiling. First, define ϕ^4 as follows:

$$\begin{aligned}\phi^4 \quad := \quad & \textstyle\bigvee_{1\leq i\leq k} t_i \wedge \bigwedge_{1\leq i\leq k}(t_i \rightarrow \neg(\bigvee_{j\neq i} t_j))\\ & \wedge \textstyle\bigwedge_{1\leq i\leq k}(t_i \rightarrow \langle up\rangle(\bigvee_{up(t_i)=down(t_j)} t_j))\\ & \wedge \textstyle\bigwedge_{1\leq i\leq k}(t_i \rightarrow \langle right\rangle(\bigvee_{right(t_i)=left(t_j)} t_j))\end{aligned}$$

This formula straightforwardly forces the propositional variables t_i to act like matched tiles: every point satisfies exactly one tile, and the adjacent tiles have matching colours both vertically and horizontally. So we define:

$$\phi^{match} \;:=\; G^+\phi^4.$$

Our work is done. Let $\phi^{\mathcal{T}}$ be $\phi^{min} \wedge \phi^{share} \wedge \phi^{mod\text{-}3} \wedge \phi^{match}$. It is easy to see that if $\mathcal{T}$ tiles $\omega \times \omega$ then $\phi^{\mathcal{T}}$ is satisfiable, for the tiling induces a suitable model in the obvious way. Moreover, as should be clear from our discussion, if $\phi^{\mathcal{T}}$ is satisfiable on a strictly partially ordered model, then the model embodies all the information needed to construct a tiling. Hence, as there is no algorithm for solving the unbounded tiling problem, there can be no algorithm for solving $\mathrm{ML} + {\downarrow} + @$ satisfiability over SPOs, and we have shown:

Theorem 3 *Determining whether any given sentences of* $\mathrm{ML} + {\downarrow} + @$ *is satisfiable on some strict partial order is an undecidable problem.*

It follows that the next language up in the hierarchy has an undecidable satisfaction problem as well. That is, we have also shown:

Theorem 4 *Determining whether any given sentences of* TL $+ \downarrow$ *is satisfiable on some strict partial order is an undecidable problem.*

And of course, ML $+ \exists +$ @, the strongest language in our hierarchy, must have an undecidable satisfaction problem too — but as it has the expressive strength of the correspondence language, this is hardly surprising.

Why hybrid languages?

Why should modal logicians be interested in hybrid languages? We think that the most important answer is simply, "because they're fascinating", and hope our discussion has whetted your logical appetite. Nonetheless, there are less self-indulgent reasons for studying them, and we'll close the paper by noting some of the more prominent.

Hybrid languages are a natural tool for studying indexicality in natural language. Now, the classical tool for this purpose is *multi-dimensional modal logic*. In multi-dimensional systems, formulas are evaluated at sequences of points. The sequence length is typically somewhere between 2 and ω, and the basic intuition is that one point of the sequence is thought of as the point of evaluation, while the others are used as memory locations to store such important points as 'now' and 'then'; key references in this tradition include Kamp (1971), Vlach (1973), and Gabbay (1976). However, with the exception of recent work by Cresswell (1990,1996), the study of such indexical logics has languished somewhat in recent years.[7]

Hybrid languages offer a novel perspective from which to re-assess, and unify, the work of the philosophical tradition. Roughly speaking, hybrid languages move multi-dimensional logic's sequence of evaluation points from the metalanguage to the object language — after all, hybrid variables are pretty much names for indices. Once safely in the object language, these 'names for indices' freely combine with the other logical apparatus, enabling subtle distinctions to be drawn. Moreover, when equipped with the @ operator, hybrid languages offer the 'de-scoping' behaviour typical of such multi-dimensional operators as **Now** and **Then**. Thus it should come as no surprise that it is possible to identify many multi-dimensional logics of indexicality as fragments of hybrid languages. For example, a key part of the system of Cresswell (1990) can be viewed as the hybrid fragment in which every occurrence of a variable x is of the form $\downarrow x$ or $@_x$. (That is: storage is permitted, retrieval is admitted, but variables cannot be freely manipulated as part of the underlying logic.) It would be useful to have a more accurate map of the various indexical logics that have been proposed, and embedding them in hybrid languages seems a natural way of going about this task.

A second reason for being interested in hybrid languages stems from their sortal tolerance: this tolerance seems to be precisely what is required to analyse a number of philosophical issues.

One such issue has a long history: how should temporal information be represented? Arthur Prior's argued convincingly that time, the earlier-than/later-

[7]Ironically, over the same period interest in multi-dimensional systems for mathematical or computational ends has flourished, as recent work in Arrow Logic and Cylindric Modal Logic testifies; see Marx and Venema (1996). Unfortunately, there seems to have been little contact between the mathematical and philosophical traditions.

than relation, and tensed statements should be handled together in the same language (Prior (1967) is the key reference), and this basic insight prompted the technical investigations of Bull (1970). As Bull puts it:

> I am coming to believe that the "right" way to do tense logic is that of Prior.... There are several reasons for this conviction; the first is simply that time and tense ought to be discussed together in the same language. Bull (1970), page 282.

(Bull also gives a second reason for his interest in hybrid languages: "I consider a richer language, extending to the contexts of tense logic, to be necessary to further development".)

A more recent source of applications for sortally tolerant systems comes from *situation theory*; see Seligman and Moss (1996) for an overview. Situation theory glories in a rich ontology and favours an Austinian theory of truth: every statement is about some situation. What notion of logical consequence do such ideas lead to? The problem is examined in Seligman (1994), where, because they permit terms to be used as formulas, hybrid languages are chosen as the medium of analysis. Seligman presents a proof theoretic account of "the logic of correct description" that suffices for the case of omniscient situations (that is, situations containing all other situations). It seems plausible that the use of partial hybrid languages would permit his analysis to be extended to arbitrary situations.

Incidentally, for a variety of reasons the proof theoretical investigations underlying this philosophical analysis strike us as an interesting point of departure for further purely technical work. For a start, because of the presence of hybrid variables, the resulting hybrid sequent calculus is reminiscent of 'labeled' analyses of modal consequence, such as those of Fitting(1983) or Gabbay(1992). The interesting difference is that in hybrid languages the 'labels' (that is, the hybrid variables) are not extraneous metalinguistic entities, but members in good standing of the object language. Seligman's analysis shows that such 'internal labels' can be incorporated naturally — indeed, elegantly — into the deductive mechanism. It's also worth mentioning that there seem to be interesting parallels between hybrid proof theory and the use of Gabbay style 'rules for the undefinable' (see Gabbay (1981)) that deserve further attention.

But, to conclude, a more general remark. In this paper we have presented hybrid languages as variable binding extensions of propositional tense logic that combine aspects of Tarskian and Kripke semantics, have discussed two key binders and a useful additional operator, and have emphasized their 'sortally tolerant' character. But in one important respect our discussion has been too narrow: namely, *why confine ourselves to just two sorts?* For example, in the setting of temporal logic it is natural to work with a three sorted system: as well the propositional information residing in PROP, and the nominal information residing in VAR, encode 'course of history' information in a third sort PATH. In fact, Bull (1970) — probably the very first technical investigation of hybrid languages — contains an examination of such a system, motivated by Prior's interest in 'branching' models of time. More generally, hybrid languages with multiple sorts are an obvious arena for further investigations.

These, then, are some of our motives for studying hybrid languages. We believe these systems are natural and deserve further attention from the modal logic community; we hope that this paper has made it clear why.

References

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

[2] van Benthem, J., 1984., "Correspondence Theory", in *Handbook of Philosophical Logic*, **2**, D. Gabbay and F. Guenthner, eds., Dordrecht: Reidel.

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

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

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

[6] Bull, R. and Segerberg, K. 1984., "Basic modal logic", in *Handbook of Philosophical Logic*, **2**, D. Gabbay and F. Guenthner, eds., D. Reidel.

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

[8] Cresswell, M. 1996, *Semantic Indexicality*, Kluwer.

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

[10] Gabbay, D., 1976, *Investigations in Modal and Tense Logics with Applications to Problems in Philosophy and Linguistics*, D. Reidel.

[11] Gabbay, D., 1981, "An irreflexivity lemma", pp. 67–89 in *Aspects of Philosophical Logic*, U. Mönnich, ed., D. Reidel.

[12] Gabbay, D., 1992, *Labelled Deductive Systems*, Oxford University Press.

[13] Gargov, G. and Goranko, V., 1993, "Modal logic with names", *Journal of Philosophical Logic*, **22**, 607–636.

[14] Goranko, V., 1994, "Temporal logic with reference pointers", pp. 133–164 in *Temporal Logic. First International Conference, ICTL '94 Bonn, Germany*, D. Gabbay and H. Ohlbach, ed., Springer-Verlag.

[15] Goranko, V., 1996a, "Hierarchies of modal and temporal logics with reference pointers", *Journal of Logic, Language and Information*, **5**, 1–24.

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

[17] Harel, D., 1983, "Recurring dominoes: making the highly undecidable highly understandable", in *Foundations of Computing Theory. Proceedings of the 1983 International FCT-Conference, Borgholm, Sweden.*, Marek Karpinski, ed., Springer Lecture Notes in Computer Science, **158**, 177–194.

[18] Kamp, J., 1971, "Formal properties of "now" ", *Theoria*, **37**, 227–273. 177–194.

[19] Marx, M. and Venema, Y. 1997, *Multi-Dimensional Modal Logic*, Kluwer.

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

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

[22] Prior, A., 1968, " 'Now' ", *Nous*, **2**, 101–119.

[23] Rabin, M., 1969, "Decidability of second order theories and automata on infinite trees", *Trans. Am. Math. Soc.*, **141**, 1–35.

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

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

[26] Seligman, J., 1991, "A cut-free sequent calculus for elementary situated reasoning", Technical Report HCRC-RP 22, HCRC, Edinburgh.

[27] Seligman, J., 1994, "The logic of correct description", to appear in *Advances in Intensional Logic*, M. de Rijke, ed., Kluwer.

[28] Seligman, J. and Moss, L., 1996, "Situation theory", *Handbook of Logic in Linguistics*, J. van Benthem and A. ter Meulen, ed., Kluwer.

[29] Sellink, M. P. A., 1994, "Verifying modal formulae over I/O-automata by means of type theory", Logic Group Preprint Series, Utrecht University.

[30] Spaan, E., 1993, *Complexity of Modal Logic*, PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam.

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