Hybrid Languages*

Patrick Blackburn[†] and Jerry Seligman[‡]

Abstract

Hybrid languages have both modal and first-order characteristics: a Kripke semantics, and explicit variable binding apparatus. This paper motivates the development of hybrid languages, sketches their history, and examines the expressive power of three hybrid binders. We show that all three binders give rise to languages strictly weaker than the corresponding first-order language, that full first-order expressivity can be gained by adding the universal modality, and that all three binders can force the existence of infinite models and have undecidable satisfiability problems.

1 Introduction

Although both first-order languages and modal languages are tools for describing relational structures, they work very differently. First-order languages take an 'external' view of relational structures, and make use of explicit variables and binding. Modal languages, on the other hand, take an 'internal' view and eschew explicit variable binding in favour of operators. The purpose of this paper is to introduce and explore a number of *hybrid* languages. These are like first-order languages in their explicit use of variables and binding, but adopt the internal perspective characteristic of modal logic.

Our investigation is largely model theoretic. We begin with an informal introduction to hybrid languages, indicating how they arise from recent work in extended modal logic, and sketch what we know about their history. We then define a number of hybrid languages and turn to the topic that will occupy us for remainder of the paper: their expressivity. Firstly, we compare the hybrid languages with each other and with their first-order correspondence language. As we shall see, three of these languages are strictly weaker than the correspondence language—though full first-order expressive power can be

*To appear in *Journal of Logic, Language and Information.*

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

[‡]Institute of Philosophy, National Chung Cheng University, Chia-Yi, Taiwan, R.O.C. Email: jerry@phil.ccu.edu.tw

gained by adding the universal modality. Moreover, they give rise to an expressive hierarchy; we show that the hierarchy is genuine by establishing a number of preservation results. Secondly, we show that none of these languages has the finite model property, and all are undecidable. To conclude the paper we note some potential applications and directions for further logical investigation.

2 Towards hybrid languages

Modal logic has changed dramatically over the last twenty years. A host of new applications in theoretical computer science, knowledge representation and computational linguistics has led to the development of richer languages (for example, PDL, Harel 1984) and the investigation of such topics as computational complexity and automated theorem proving. Moreover, our understanding of what modal logic actually *is* has deepened considerably. For example, thanks to correspondence theory (see van Benthem 1983, 1984) we know that modal languages can be usefully viewed as fragments of first- (and indeed, second-) order logic. Modal languages are no longer seen as exotic 'non-classical' systems; like their classical cousins, they are simply a way of talking about relational structures. As this correspondence theoretic view—and in particular, the *first-order* perspective it offers—underlies much of the present paper, let us briefly note how it arises.

Suppose we are working with a modal language with a single unary modality $\langle R\rangle$. That is, we have a (denumerable) collection *Prop* of propositional symbols (written p, q, r etc.), a truth-functionally adequate collection of boolean connectives (e.g. $\neg$ and $\wedge$) and $\langle R\rangle$. We generate the formulae in the expected way (all elements of *Prop* are formulae, boolean combinations of formulae are formulae, $\langle R\rangle\varphi$ is a formula if φ is, and nothing else is a formula) and define $[R]\varphi$ to be $\neg\langle R\rangle\neg\varphi$.

The standard semantics makes use of *Kripke models*. For the language in $\langle R\rangle$, a Kripke model M consists of a non-empty set $|M|$, a binary relation R_M on $|M|$, and a *Prop*-indexed collection of unary relations on $|M|$. The satisfaction definition inductively defines a three-place relation between a model M, a formula φ, and elements a of $|M|$:

$M, a \models p$	iff	$a \in P$, where P is the unary relation indexed by p
$M, a \models \neg\varphi$	iff	$M, a \not\models \varphi$
$M, a \models \varphi \wedge \psi$	iff	$M, a \models \varphi$ and $M, a \models \psi$
$M, a \models \langle R\rangle\varphi$	iff	there is an $a' \in |M|$ such that $R_M(a, a')$ and $M, a' \models \varphi$

While this definition is of fundamental importance, thinking of modal logics purely in terms of Kripke models predisposes one to view modal logic as an isolated formal system. Correspondence theory offers a broader perspective which emphasizes the connections between modal and classical languages. The basic ideas are very simple. First, we note that Kripke models are simply *relational* *structures* in the usual sense of model theory (see Hodges 1993). Next, we note that there is an obvious way of viewing the Kripke satisfaction definition as a translation of modal formulae into first-order formulae. For the modal language in $\langle R \rangle$, the appropriate 'correspondence language' is a first-order language with a unary relation symbol R and a *Prop*-indexed collection of unary relation symbols. The required translation of modal formulae (known as the *standard translation*) is as follows:

$$\begin{array}{lcl} ST_x(p) & = & Px, \text{ where } P \text{ is the relation symbol indexed by } p \\ ST_x(\neg\varphi) & = & \neg ST_x(\varphi) \\ ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\psi) \\ ST_x(\langle R \rangle \varphi) & = & \exists y (R(x,y) \wedge ST_y(\varphi)) \end{array}$$

For the sake of definiteness, we may take the y in the final clause to be the first variable different from x in some fixed enumeration of the variables. Note that for any modal formula φ, $ST_x(\varphi)$ contains exactly one free variable, namely x. It is clear that for each model M, modal formula φ, and element a of $|M|$, $M, a \models \varphi$ iff $M \models ST_x(\varphi)[x \mapsto a]$ (here $[x \mapsto a]$ means assign a to the free variable x).

Simple as it is, the standard translation tells us a lot about what makes modal languages special. As a number of logicians (most notably Prior 1967) have emphasized, modal languages take an 'internal' or 'local' view of information. To some extent these intuitions are clarified by the Kripke satisfaction definition (we evaluate formulae at points *inside* models, and only ever examine R-accessible worlds) but the standard translation makes them explicit. Modal languages really do work 'internally': to capture their effect classically we must work not with first-order sentences, but with formulae bearing a special free variable. Moreover, modal formulae work 'locally': all occurrences of quantifiers in the formulae produced by the standard translation are bounded.

But correspondence theory has not only proved mathematically fruitful; it has changed the way modal logicians ply their trade. The idea that modal languages are an interesting way of talking about relational structures—coupled with the increasing emphasis on applicable formalisms—has encouraged modal logicians to experiment with a wide variety of enriched formalisms. Our work on hybrid languages has it roots in this emerging field of *extended modal logic*. In particular, it draws on recent work on *logical modalities* and *modal logics with names*. Let us consider these in turn.

Logical modalities are rather like first-order 'logical predicates'. For example, equality is not definable in arbitrary structures using only first-order apparatus. However, because equality is such a fundamental relation, it is usual to add a special binary relation symbol to first-order languages (namely, =) and to *stipulate* that it denotes the equality relation. Logical modalities are based on the same idea: if we need to talk about fundamental but modally undefinable relations, introduce special modalities and *stipulate* that they deal with the relation in question. Two such operators have proved particularly useful: the D-operator (which explores the $\neq$ relation) and the *universal modality* (which deals with the universal relation $|M| \times |M|$). Both enrichments result in a considerable increase in expressive power (Goranko and Passy 1993, de Rijke 1992) and the universal modality plays an important supporting role in what follows.

However, the immediate ancestors of hybrid languages are modal logics with names (see Passy and Tinchev 1985, 1991, Gargov and Goranko 1993, Blackburn 1993). The enrichment involved is simple: a second sort of atomic symbol is introduced (these are called *names* or *nominals*) and it is stipulated that such formulae are satisfied at a unique element of any Kripke model. Intuitively, such a formula 'names' the unique point at which it satisfied. This enrichment also leads to an increase in expressive power. Moreover, it is the first step on the path to *hybrid languages*.

Consider a formula built from nominals, say $i \to \neg\langle R\rangle i$. Here i names a point which is *fixed* in any interpretation, and the formula says that the point named by i is not related to itself by the relation denoted by R. Under correspondence, the nominal i is treated as a first-order *constant*. This is where hybrid languages come in: they treat nominals as *variables* open to binding. Taking this step leads to a certain loss of syntactic innocence (instead of a purely propositional system we shall have to deal with variables, assignments, and binding) but as long as we retain the internal Kripke semantics, the resulting systems will retain a distinctively modal flavour.

So, let us augment the basic modal language with a set X of variables, to be taken as new syntactic atoms. The interpretation of modal formulae is relativised to an assignment of values to variables g, with all the usual clauses and a new base clause for variables x:

$M, g, a \models x$ iff $g(x) = a$.

How should we quantify these variables, and what can we do with the resulting systems? The reader may be surprised by the first question: isn't it *obvious* what our quantifier should be? Such a reader probably has the following in mind: build formulae of the form $\exists x.\varphi$, and interpret them in models (with the aid of an assignment of values to variables g) as follows:

$M, g, a \models \exists x.\varphi$ iff there is an assignment $g' \stackrel{x}{=} g$ such that $M, g', a \models \varphi$.

(By '$g' \stackrel{x}{=} g$' we mean that for each variable y in X, either $g'(y) = g(y)$ or $x = y$.) This is certainly the most direct hybrid analog of the first-order existential quantifier, but not the only one. For a start, why should a hybrid existential quantifier reset only the assignment? Why not use the following binder Σ, which resets the point of evaluation as well?

$M, g, a \models \Sigma x.\varphi$ iff there is an a' in M and an assignment $g' \stackrel{x}{=} g$ such that $g'(x) = a'$ and $M, g', a' \models \varphi$.

Intuitively, Σ binds φ's variables and tries to find a satisfying point somewhere in the model. It too seems a reasonable candidate for the role of 'hybrid existential quantifier'.

But we are under no obligation to define *any* analog of the existential quantifier. If we reflect on the locality inherent in modal logic, we are led to ideas that owe little to first-order logic. In particular, the following binder is a natural choice:

$$M, g, a \models \downarrow x.\varphi \text{ iff } M, g', a \models \varphi, \text{ where } g' \stackrel{x}{=} g \text{ and } g'(x) = a.$$

That is, $\downarrow$ binds x to the point of evaluation: it names the here-and-now. This addition results in a considerable increase in expressive power. For example, suppose we extend the propositional tense logic with variables and $\downarrow$. Then the *Until* (and *Since*) operators become definable:

$$\mathit{Until}\,(\varphi, \psi) := \downarrow x.F(\varphi \wedge H(Px \rightarrow \psi)).$$

This is a striking example. The *Until* operator is not definable in tense logic enriched with nominals, the universal modality, or even the D-operator.

Finally, we have a binder which occupies a natural position in logical space, but which we had not considered before beginning this investigation:

$M, g, a \models \Downarrow x.\varphi$ iff there is an a' in M and an assignment $g' \stackrel{x}{=} g$ such that $g'(x) = a$ and $M, g', a' \models \varphi$.

The $\Downarrow$ operator combines the powers of the $\downarrow$ and $\Diamond$ ('somewhere') operators; indeed, it is easy to see that $\Downarrow x.\phi$ is logically equivalent to $\downarrow x.\Diamond\phi$.

These, then, are the hybrid binders we shall explore. To conclude this overview, we sketch what is already known about them. Although the literature is small and scattered, all three binders have been considered previously—albeit sometimes in disguised form.

Arthur Prior (1967, 1968) seems to have been the first to suggest $\exists$, and it was investigated further by Bull (1970). Bull worked in tense logic enriched with both $\exists$ and the universal modality, and showed that the Henkin construction adapted naturally to the hybrid system. The idea then seems to have lain dormant until (independently) reinvented in the mid '80s by the Sofia school. Beginning with PDL with names (Passy and Tinchev 1985a) they swiftly moved to PDL enriched with $\exists$ (Passy and Tinchev 1985b, 1991). Technical themes explored in the Bulgarian tradition include (high) undecidability results, the natural way the first-order Henkin construction adapts to the hybrid systems, and Gabbay (1981) style irreflexivity rules.

Although we are not aware of any direct use of Σ, there are at least two systems in which it is a natural defined operator: Allen's interval-based system and Prior's UT calculus. Allen's calculus is based around a 'retrieval' operator **Holds**(i, φ), where i names an interval and φ is a formula. The intended semantics is that φ holds at the interval named by i. Similarly, Prior's UT calculus is based around the retrieval operator $\mathrm{T}(t, \varphi)$. Here t names a point, and φ must be true at its denotation.[1]

Both systems allow quantification over the 'naming' slot, and the quantifier used is essentially $\exists$. Thus the expressions $\exists i \mathbf{Holds}(i, \varphi)$ and $\exists i \mathrm{T}(i, \varphi)$ check whether φ is satisfied anywhere in a model. This is what we would write as $\Sigma x.\varphi$.[2]

Finally, $\downarrow$ has been independently invented on at least three occasion. Richards *et al* (1989) introduce it as part of their investigation of natural language temporal semantics and temporal databases, Goranko (1994, 1995) introduces it as a formalisation of 'now' and 'then', and Sellink (1994) introduces it as part of a system for reasoning about I/O-automata. All three systems have made different syntactic choices. For example, neither Richards *et al* nor Sellink treat the bindable variables as formulae (though it is straightforward to show that the Richards *et al* system is essentially tense logic enriched with $\downarrow$), but Goranko's formulation is basically the same as ours. He extends standard unimodal and temporal languages with the universal modality and $\downarrow$, provides a number of complete axiomatisations for these systems, and proves undecidability.

3 Syntax and semantics

In this section we define the syntax and semantics of our hybrid languages, extend the standard translation to cover them, and fix some notation and terminology.

We shall view hybrid languages as alternative ways of talking about relational structures of *arbitrary* signature. For convenience, let us fix at the outset the class of structures we shall work with. Suppose we are given a set $\mathcal{R}$ of relation symbols and a function ν assigning a natural number to each symbol in R (its arity). Then we shall work with $\mathcal{M}$, the class of relational structures M consisting of a set $|M|$ together with a subset R_M of $|M|^{\nu(R)}$, for each R in $\mathcal{R}$.

We now define various languages for describing such structures. We shall be interested in each language's capacity to define properties of individuals, thus our primary semantic relationship is that of an individual a of a model M being correctly described by a formula φ of the language. We write this as

$$M, a \models \varphi.$$

[1] Such retrieval operators are interesting in their own right, and a natural addition to hybrid languages, but space limitations mean we cannot discuss them further here. Hybrid languages with retrieval operators are explored in Seligman (1991, 1994). Similar operators are to be found in *topological logic* (see Rescher and Urquhart 1971, and references therein).

[2] In passing, both Allen's interval system and Prior's UT calculus can be viewed as hybrid systems. When one formalises the intended syntax and semantics of these systems, one is lead not to a standard first-order system, but to topological logic (Rescher and Urquhart 1971). Hybrid languages offer an alternative to topological formalisations.

Our reference language will be the standard first-order language generated from a set X of variables and the relation symbols in $\mathcal{R}$. We christen this language L_0, and often refer to it as the *correspondence language*. For simplicity, we assume that the only logical symbols in L_0 are the boolean connectives $\wedge$ and $\neg$, and the quantifier $\exists$. The truth of a formula of L_0 in a structure M of $\mathcal{M}$, relative to an assignment function $g\colon X \to |M|$ is given in the standard way. However the semantic relation of interest—that of an individual of some structure being described by a formula—requires us to isolate a special variable to 'localise' L_0 descriptions to a particular individual. In effect, the linguistic unit of interest is a formula φ of L_0 together with a localising variable x; we write the combined symbol as $\varphi[x]$. Thus the semantic relation we wish to explore is defined for L_0 as follows:

> $M, g, a \models \varphi[x]$ iff φ is true in M relative to the unique assignment function $g'\colon X \to |M|$, such that $g'(x) = a$ and $g' \stackrel{x}{=} g$.

None of the other languages will need a special symbol to denote the individual described. Instead, the relation $\models$ will be defined directly in the way familiar from modal logic. We now define these systems.

Syntactically, each of these languages is an extension of the basic modal language L_m corresponding to L_0. L_m is the smallest set of formulae containing the following: (1) each individual variable $x \in X$, (2) $\varphi \wedge \psi$, for each φ and ψ in L_m, (3) $\neg\varphi$, for each φ in L_m, (4) the propositional constant $\top$, and (5) $R(\varphi_1, \ldots, \varphi_{\nu(R)-1})$, for each $\nu(R)-1$ long sequence $\varphi_1, \ldots, \varphi_{\nu(R)-1}$ of formulae in L_m. Sometimes we drop this 'official' syntax in favour of something more obviously modal. In particular, when working with binary relation R we shall often use the notation $\langle R\rangle\varphi$ instead of $R(\varphi)$.

Now for the semantics. Given a structure M in $\mathcal{M}$, an element $a \in |M|$, an assignment function $g\colon X \to |M|$, and a wff φ of L_m we define:

> $M, g, a \models x$ iff $g(x) = a$
>
> $M, g, a \models \varphi \wedge \psi$ iff $M, g, a \models \varphi$ and $M, g, a \models \psi$
>
> $M, g, a \models \neg\varphi$ iff $M, g, a \not\models \varphi$
>
> $M, g, a \models \top$ always
>
> $M, g, a \models R(\varphi_1, \ldots, \varphi_{\nu(R)-1})$ iff there is a $\nu(R)-1$ long sequence $a_1, \ldots, a_{\nu(R)-1}$ of elements of M such that $R_M(a, a_1, \ldots, a_{\nu(R)-1})$ and $M, g, a_i \models \varphi_i$ for $1 \leq i \leq \nu(R) - 1$.

Two comments should be made. First, note that when are working with a binary relation R, the clause for the modalities is the familiar one:

> $M, g, a \models \langle R\rangle\varphi$ iff there is an a' in M such that $R_M(a, a')$ and $M, g, a' \models \varphi$.

In short, the satisfaction clause for the modalities is the natural generalisation to n-place relations.

Second, note that the variables denote individual elements of M, not subsets. That is, we have none of the familiar propositional variables in our language. It would be a trivial to add them—but for present purposes, superfluous. In this paper we wish to focus on the purely first-order apparatus of hybrid languages. Because all the variables of L_m (and of all of the extensions we shall consider) denote individuals, these systems are essentially first-order.

When variables are introduced we must be careful to ensure that the standard translation is stated in such a way as to prevent unintentional coreference and binding. Bearing this in mind, the translation is straightforward:

$$\begin{array}{lcl} ST_x(y) & = & (x = y) \\ ST_x(R(\varphi_1, \cdots, \varphi_n)) & = & \exists y_1, \cdots, y_n(R(x, y_1, \cdots, y_n) \\ & & \wedge ST_{y_1}(\varphi_1) \wedge \cdots \wedge ST_{y_n}(\varphi_n)). \end{array}$$

Here the variables $y_1, \ldots, y_n$ are the first n variables in some standard enumeration which do not occur in $R(x, \varphi_1, \cdots, \varphi_n)$.

L_m is only our base language; we shall build hybrid languages on top of it by adding various binding operators, and possibly an extra modal operator as well. A *binding operator* is a binary operator B which takes a variable x and a formula φ as arguments, resulting in a formula written $Bx.\varphi$, and with the consequence that all occurrences of x in φ are bound. Let us fix some notation and terminology for extensions of L_m. Let $\mathcal{O}$ and $\mathcal{B}$ be two sets of symbols, called *operator symbols* and *binding symbols* respectively. The language $L(\mathcal{O}, \mathcal{B})$ is defined to be the smallest set of formulae containing: (1) each individual variable $x \in X$, (2) $R(\varphi_1, \ldots, \varphi_{\nu(R)-1})$, for each $\nu(R) - 1$ long sequence $\varphi_1, \ldots, \varphi_n$ of formulae in $L(\mathcal{O}, \mathcal{B})$, (3) $\varphi \wedge \psi$, for each φ and ψ in $L(\mathcal{O}, \mathcal{B})$, (4) $\neg\varphi$, for each φ in $L(\mathcal{O}, \mathcal{B})$, (5) the propositional constant $\top$, (6) $O\varphi$, for each O in $\mathcal{O}$ and each φ in $L(\mathcal{O}, \mathcal{B})$, and (7) $Bx.\varphi$, for each B in $\mathcal{B}$, each x in X, and each φ in $L(\mathcal{O}, \mathcal{B})$.

The relationship of 'being free in' and 'being bound in' between a variable and a formula is defined in the expected way, and a *sentence* of $L(\mathcal{O}, \mathcal{B})$ is defined to be a formula (of $L(\mathcal{O}, \mathcal{B})$) in which no variable occurs free. The propositional constant $\top$ is included in the base language to ensure that every language has sentences.

As a notational convenience, we shall use the operators themselves to name the language. Thus if O is a modal operator, we abbreviate '$L(\{O\}, \emptyset)$' to 'O', and if B is a binding operator, we abbreviate '$L(\emptyset, \{B\})$' to 'B'. When multiple operators are involved, with use '+' to indicate language combination. Thus if O is a modal operator, and B_1 and B_2 are both binding operators then '$O + B_1 + B_2$' abbreviates '$L(\{O\}, \{B_1, B_2\})$'. Note that addition of operator-languages is commutative, associative and idempotent.

We are ready to introduce the operators and binders and their associated semantic conditions. This investigation will centre around one logical modal operator $\Diamond$ (the 'somewhere' operator) and four binding operators $\exists$, $\downarrow$, Σ and $\Downarrow$, associated with the following semantic conditions:

$M, g, a \models \Diamond\varphi$ iff there is an element a' of $|M|$ such that $M, g, a' \models \varphi$

$M, g, a \models \exists x.\varphi$ iff there is an assignment function $g' \colon X \to |M|$ such that $g' \stackrel{x}{=} g$ and $M, g', a \models \varphi$

$M, g, a \models \downarrow x.\varphi$ iff there is an assignment function $g' \colon X \to |M|$ such that $g' \stackrel{x}{=} g$ and $g'(x) = a$ and $M, g', a \models \varphi$

$M, g, a \models \Sigma x.\varphi$ iff there is an element a' of $|M|$ and an assignment function $g' \colon X \to |M|$ such that $g' \stackrel{x}{=} g$, $g'(x) = a'$ and $M, g', a' \models \varphi$

$M, g, a \models \Downarrow x.\varphi$ iff there is an element a' of $|M|$ and an assignment function $g' \colon X \to |M|$ such that $g' \stackrel{x}{=} g$, $g'(x) = a$ and $M, g', a' \models \varphi$

Note that $\Diamond\varphi$ means that φ holds somewhere in the structure. We define $\Box\varphi$ to be $\neg\Diamond\neg\varphi$; this asserts that φ holds everywhere in the structure, so $\Box$ is the universal modality.

For each structure M in $\mathcal{M}$, each a in $|M|$, and each formula φ of $L(\mathcal{O}, \mathcal{B})$, we define:

$$M, a \models \varphi \text{ iff for each assignment function } g \colon X \to |M|,\ M, g, a \models \varphi.$$

Finally, we extend the standard translation to the new operator and binders as follows (remembering that $ST_x(\varphi)$ is only defined for x not in φ):

$$
\begin{array}{lcll}
ST_x(\Diamond\varphi) & = & \exists z ST_z(\varphi) & (z = \text{the first variable not in } \varphi)\\
ST_x(\exists y.\varphi) & = & \exists y.ST_x(\varphi) & \\
ST_x(\Sigma y.\varphi) & = & \exists y.ST_y(\varphi) & \\
ST_x(\downarrow y.\varphi) & = & \exists y.(x = y \wedge ST_x(\varphi)) & \\
ST_x(\Downarrow y.\varphi) & = & \exists y.\exists z.(x = y \wedge ST_z(\varphi)) &
\end{array}
$$

It is clear that this translation is satisfaction preserving, that is, for each modal formula φ and variable x not occurring in φ,

$$M, g, a \models \varphi \text{ iff } M, g, a \models ST_x(\varphi)[x]$$

Moreover, the translation $ST_x(\varphi)$ is sure to be a first-order formula whose free variables are those of φ together with at most one additional free variable, the 'localising' variable x. So, as promised, all our hybrid languages can be regarded as fragments of the first-order correspondence language. But how expressive are these various fragments? This is the question to which we now turn.

4 Comparing expressivity

For a given signature, we have now defined thirty-one hybrid languages: one for each combination of operators selected from the above five, excluding the basic modal language. In this section we shall compare the expressivity of these languages both with each other, and with the correspondence language.

First we need a suitable basis for comparison. We propose to compare the properties definable by each language.

Definable subsets A *property* $P = \{P_M\}_{M \in \mathcal{M}}$ is a family of sets, with $P_M \subseteq |M|$ for each structure $M \in \mathcal{M}$. A property P is *definable* in language L, if there is a sentence φ of L such that, for each $M \in \mathcal{M}$ and each $a \in P_M$, $M, a \models \varphi$ iff $a \in P_M$.

This is an appropriate measure of hybrid expressivity. It is intrinsically 'local', and thus a natural choice for modal languages. Note that it is important that we restrict our attention to sentences, that is to formulae with no free variables. As the standard translation of the previous section shows, any sentence of any our hybrid languages gives rise to a first-order formula with one free variable, and such first-order formulae indeed define properties. A formula of a hybrid language having free variables would correspond to a first-order formula with more than one free variable, and so does not express a property.[3]

We shall compare our languages in the obvious way. We say that language L' is *at least as expressive as* language L, written $L \leq L'$, if every property definable in L is also definable in L'. The languages L and L' are *expressively equivalent*, written $L \sim L'$, if $L \leq L'$ and $L' \leq L$. We say that L is *less expressive than* L', written $L < L'$, if $L \leq L'$ and not $L' \leq L$.

Note that addition of operator-languages is monotonic in $\leq$. That is, if $L_1 \leq L_1'$ and $L_2 \leq L_2'$ then $(L_1 + L_2) \leq (L_1' + L_2')$. From this and idempotence, it follows that the least upper-bound of two operator-languages L and L' is their sum $L + L'$ (up to expressive equivalence).

Note also that the existence of a translation of L in L' ensures that $L \leq L'$. From the standard translation given in the previous section, it follows that the correspondence language is as least as expressive as any of the hybrid languages.

The $\Diamond\downarrow$-hierarchy

Consider Figure 1. We shall establish that it is an exact picture of the $\leq$-ordering of our languages. The position of those combinations of languages not depicted in the diagram can be computed by taking least upper-bounds.

Showing that this diagram is an exact picture of the $\leq$ ordering falls into two parts. The easy part is to show that if there is an arrow from L to L'

[3]The question of how best to compare the expressivity of formulae with free variables is an interesting one, but is not addressed here.

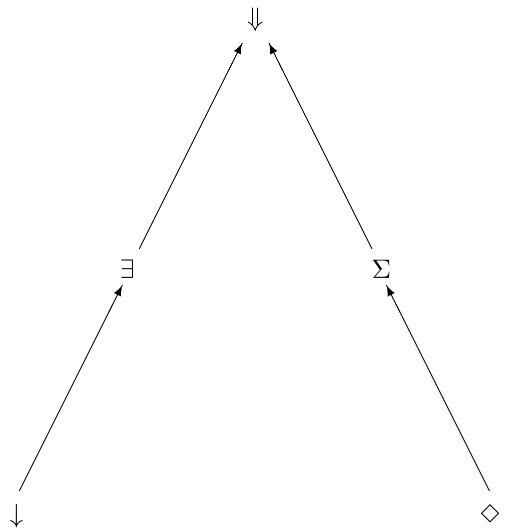


Figure 1: The $\Diamond{\downarrow}$-hierarchy

in the diagram, then $L \leq L'$; we do this by defining the operator or binder characteristic of L using the operator or binder in L'. Showing that the the diagram is *exact*—that is, demonstrating that if there is no arrow from L to L', then then $L \not\leq L'$—is harder; we do so using model theoretic arguments.

Proposition 4.1 *Each of the thirty-one hybrid languages is expressively equivalent to one of the five basic hybrid languages:* $\downarrow$, $\Diamond$, $\exists$, Σ, *or* $\Downarrow$. *Moreover, the languages are ordered as follows:* $\downarrow \leq \exists \leq \Downarrow$ *and* $\Diamond \leq \Sigma \leq \Downarrow$

Proof. The order of the five basic languages is established by means of the following definitions:

$\downarrow x.\varphi \quad := \quad \exists x.(x \wedge \varphi)$

$\exists x.\varphi \quad := \quad \Downarrow z.\Downarrow x.(z \wedge \varphi)$, where z is a variable not occurring in φ

$\Diamond\varphi \quad := \quad \Sigma z.\varphi$, where z is a variable not occurring in φ

$\Sigma x.\varphi \quad := \quad \Downarrow z.\Downarrow x.(x \wedge \varphi)$

Also, the definition

$\Downarrow x.\varphi \quad := \quad \downarrow x.\Diamond\varphi$

shows that $\Downarrow \leq \downarrow + \Diamond$ and so, by monotonicity, that $\downarrow + \Diamond$ and any other language containing an operator or binder from each of the two branches of the diagram is expressively equivalent to $\Downarrow$. Again by monotonicity, any language combining operators from just one branch is expressively equivalent to the basic language given by the most expressive of those operators. This ensures that all thirty-one hybrid languages are expressively equivalent to one of the basic five. $\dashv$

Theorem 4.1 *The diagram in Figure 1 is an exact picture of the expressivity ordering of the hybrid languages definable using the operators* $\downarrow$, $\Diamond$, $\exists$, Σ *and* $\Downarrow$.

Proof. That the diagram is a correct picture of the ordering follows from Proposition 4.1. It remains to be shown is that the diagram is an exact picture of the ordering. For this it suffices to establish that $\exists \not\leq \downarrow$, $\Diamond \not\leq \exists$, $\Sigma \not\leq \Diamond$ and $\downarrow \not\leq \Sigma$. These results are established as Propositions 4.3, 4.6, 4.8 and 4.10 of the next section. $\dashv$

Preservation Results

A negative expressivity result, say $L \not\leq L'$, may be proved by showing that the truth of sentences of L' are preserved under certain relations between structures which do not preserve the truth of sentences of L. Two of the relations we shall use (*generated substructure* and *bisimulation*) are standard tools of modal model theory; the other two (*proper generated substructure isomorphism* and *full internal bisimulation*) are new.

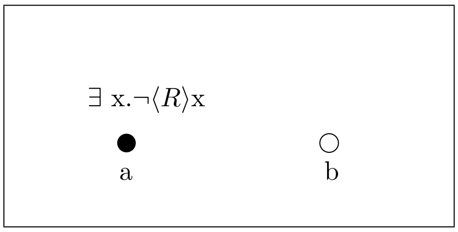
Generated substructures Given a structure M in $\mathcal{M}$, let M_a be the smallest substructure of M, such that $a \in |M_a|$ and for each $R \in \mathcal{R}$, and each $\nu(R)$-long sequence $a_1, \ldots, a_{\nu(R)}$ of elements of $|M|$, if $a_1 \in |M_a|$ and $\langle a_1, \ldots, a_{\nu(R)} \rangle \in R_M$ then $a_2, \ldots, a_{\nu(R)} \in |M_a|$. M_a is the substructure of M *generated by* a.

Proposition 4.2 *The truth-value of each formula* φ *of* $\downarrow$ *is preserved by taking generated substructures; that is for each model* M *in* $\mathcal{M}$ *and each* $a \in |M|$*, if* $M, a \models \varphi$ *then* $M_a, a \models \varphi$.

Proof. By induction on the complexity of formulae, with the following strengthened inductive hypothesis: for each $g: X \to |M_a|$ and $b \in |M_a|$, $M, g, b \models \varphi$ iff $M_a, g, a \models \varphi$. All the required inductive steps, save that for $\downarrow$, are standard. The step for $\downarrow$ is as follows: if $g: X \to |M_a|$ and $M, g, b \models \downarrow x.\varphi$ then there is a $g': X \to |M|$ with $g' \overset{x}{=} g$ such that $g'(x) = b$ and $M, g', b \models \varphi$. Clearly, $rng\ (g') \subseteq |M_a|$, so $M_a, g', b \models \varphi$, by the inductive hypothesis, and so $M_a, g, b \models \downarrow x.\varphi$. The converse is proved similarly, and this completes the induction. Now suppose that $M, a \models \varphi$. Each assignment g on M_a is also an assignment on M and so $M, g, a \models \varphi$. By the above argument, $M_a, g, a \models \varphi$, and so $M_a, a \models \varphi$, as required. $\dashv$

Proposition 4.3 $\exists \not\leq \downarrow$

Proof. Suppose for the sake of a contradiction that $\exists \leq \downarrow$. Then for each $\exists$-sentence φ the property defined by φ is also definable in L', and so there is a $\downarrow$-sentence φ' equivalent to φ. By Proposition 4.2 the truth-value of φ', and hence of φ, is preserved by taking generated substructures. But this is not the case. Consider the following counterexample. Let M be the two-element structure, with $|M| = \{a, b\}$, and a binary relation $R_M = \{\langle a, a\rangle\}$.



The $\exists$-sentence $\exists x.\neg\langle R\rangle x$ is true at a in M, for we can assign the point b to x and $\langle a, b\rangle \notin R_M$. However it is *not* true at a in M_a. As M_a contains only the point a, all assignments assign a to x; and as a is reflexive, $\neg\langle R\rangle x$ must be false. $\dashv$

Proper Generated Substructure Isomorphisms Given structures M and M' in $\mathcal{M}$, and elements $a \in |M|$ and $a' \in |M'|$, we say that an isomorphism $f: M_a \rightarrow M'_{a'}$ is a *proper generated substructure isomorphism* iff $f(a) = a'$, $M \neq M_a$, and $M' \neq M'_{a'}$.

Proposition 4.4 *The truth-values of $\exists$-formulae is preserved under proper generated substructure isomorphisms. In other words, if M and M' are structures in $\mathcal{M}$ such that there is a proper generated substructure isomorphism f from M_a to M'_a, then for each $\exists$-formula φ, if $M, a \models \varphi$ then $M', a' \models \varphi$.*

Proof. By induction on the complexity of φ. Once more we shall need a slightly stronger statement to serve as an induction hypothesis. Assignments $g: X \rightarrow M$ and $h: X \rightarrow M'$ are said to be *f-compatible* iff for each $x \in X$, if either $g(x) \in |M_a|$ or $h(x) \in |M'_{a'}|$ then $h(x) = fg(x)$. Our induction hypothesis is that, for each $\exists$-formula φ, if M, M', a, a', and f are as above, $b \in |M_a|$, and $g: X \rightarrow M$ and $h: X \rightarrow M'$ are f-compatible assignments, then $M, g, b \models \varphi$ iff $M', h, f(b) \models \varphi$.

The only interesting step in the induction is the clause for $\exists x.\varphi$. The required argument is as follows. If $M, g, b \models \exists x.\varphi$ then there is a $g': X \rightarrow M$ such that $g \stackrel{x}{=} g'$ and $M, g', b \models \varphi$. Define $h': X \rightarrow M'$ by

$$h'(y) = \begin{cases} h(y) & \text{if } y \neq x \\ fg'(x) & \text{if } y = x \text{ and } \ g'(x) \in |M_a| \\ a^* & \text{if } \ y = x \text{ and } \ g'(x) \notin |M_a| \end{cases}$$

where a^* is an arbitrarily chosen element of $|M'| - |M'_{a'}|$. (There must be such an element, because $M' \neq M'_{a'}$.) By construction, h' and g' are f-compatible assignments, and so $M', h', f(b) \models \varphi$, by the induction hypothesis. As $h \stackrel{x}{=} h'$,

$M', h, f(b) \models \exists x.\varphi$. The converse holds by a symmetric argument, using f^{-1} instead of f.

This establishes the inductive conclusion. The result follows from the further observation that for each assignment g' on M', the assignment fg' on M is f-compatible. Supposing that $M, a \models \varphi$ we have that $M, fg', a \models \varphi$, and so $M, g', a' \models \varphi$ by the inductive conclusion, for each assignment g' on M'. Hence $M', a' \models \varphi$, as required. $\dashv$

It is worth noting an easy consequences of this result. For any structure M in $\mathcal{M}$, let M^+ be the structure obtained from M by adding a single point which is not in the extension of any R in $\mathcal{R}$.

Proposition 4.5 *For any M in $\mathcal{M}$ and $a \in |M|$, either $M = M_a$ or each $\exists$-formula true at a in M, is also true at a in M_a^+.*

Proof. The identity function on $|M_a|$ is a proper generated substructure isomorphism from M to M_a^+, and so the result follows from Proposition 4.4. $\dashv$

Proposition 4.5 may be paraphrased as saying that although $\exists$ can 'see' whether or not there are points that lie outside the substructure generated by the point of evaluation it is blind to the information they contain. All the non-local generated substructures could be be collapsed to a single point and $\exists$ could not detect the difference. By contrast, $\Diamond$ is sensitive to the information in other generated substructures, and so we are able to establish the following result.

Proposition 4.6 $\Diamond \not\leq \exists$

Proof. If $\Diamond \leq \exists$ then by Proposition 4.5, for any M in $\mathcal{M}$ and $a \in |M|$, either $M = M_a$ or each $\Diamond$-sentence true at a in M, is also true at a in M_a^+. But this is not the case. For a counterexample, consider the two-element structure M shown in the proof of Proposition 4.3. If $\top$ is a tautology then the sentence $\Diamond\langle R\rangle\top$ is true at b in M, but not true at b in M_b^+. $\dashv$

Bisimulations Given structures M and M' in $\mathcal{M}$, a non-empty binary relation $Z \subseteq |M| \times |M'|$ is called a *bisimulation* between M and M' iff the following conditions are satisfied:

1. if $Z(a_1, a'_1)$ and $R(a_1, a_2, \cdots, a_{\nu(R)})$ then there are $a'_2, \cdots, a'_{\nu(R)} \in |M'|$ such that $R(a'_1, a'_2, \cdots, a'_{\nu(R)})$ and $Z(a_i, a'_i)$ for all i $(2 \leq i \leq \nu(R))$, and
2. if $Z(a_1, a'_1)$ and $R(a'_1, a'_2, \cdots, a'_{\nu(R)})$ then there are $a_2, \cdots, a_{\nu(R)} \in |M|$ such that $R(a_1, a_2, \cdots, a_{\nu(R)})$ and $Z(a_i, a'_i)$ for all i $(2 \leq i \leq \nu(R))$.

In short, a relation between two models is a bisimulation if related points satisfy the modally natural back-and-forth conditions.

A bisimulation Z between M and M' is *full* if for each $a \in |M|$ there is an $a' \in |M'|$ such that $Z(a, a')$ and for each $a' \in |M'|$ there is an $a \in |M|$ such that $Z(a, a')$.

Proposition 4.7 *Let M and M' be structures in $\mathcal{M}$. If Z is a full bisimulation between (M, g) and (M', h) and $Z(a, a')$, then for each $\Diamond$-sentence φ, $M, g, a \models \varphi$ iff $M', h, a' \models \varphi$.*

Proof. By induction on the structure of φ. The proof is standard (see van Benthem 1983,1984; bisimulations are called zig-zag relations in these references). Briefly, the base case holds because φ contains no variables and Z trivially preserves $\top$; the back and forth conditions drive through the step for the modalities. The inductive step for formulae of the form $\Diamond\varphi$ follows from the assumption that Z is full. $\dashv$

Proposition 4.8 $\Sigma \not\leq \Diamond$

Proof. Σ-sentences are *not* necessarily preserved under bisimulations. Let M be a model with a single binary relation R, such that $|M| = \{a\}$ and $R(a, a)$. Let M' be the model consisting of the natural numbers with R interpreted as their usual strict order ($<$). Let Z relate a to every natural number. We can see that $M, a \models \Sigma x.\langle R\rangle x$, by assigning a to x. But this sentence is not true at any point in M', because there is no natural number n for which $n < n$.

The conclusion follows from Proposition 4.7 by the now familiar argument. $\dashv$

Internal bisimulations A bisimulation Z between M and M' is an *internal bisimulation on M* if $M = M'$.

Proposition 4.9 *Suppose that Z is a full internal bisimulation on M, and $Z(a, a')$. For each Σ-sentence φ, $M, a \models \varphi$ iff $M', a' \models \varphi$.*

Proof. By induction on the structure of Σ-*sentences.* If the principal connective of φ is boolean or a modal operator then the proof follows familiar steps of Proposition 4.7. The only case of interest is where $\varphi = \Sigma x.\psi$. In this case we cannot use the induction hypothesis because x may occur free in ψ and so ψ may not be a Σ-sentence. However, the conclusion is immediate from the semantics of Σ, because whether or not $M, a \models \Sigma x.\psi$ does not depend on a. $\dashv$

Proposition 4.10 $\downarrow \not\leq \Sigma$

Proof. Given Proposition 4.9, it suffices to show that $\downarrow$-sentences need not be preserved under full internal bisimulations. Let M be a structure with $|M| = \{a, b\}$ bearing a single binary relation $R = \{\langle a, a\rangle, \langle b, a\rangle\}$. The relation $Z = \{\langle a, a\rangle, \langle a, b\rangle, \langle b, a\rangle\}$ is a full internal bisimulation on M. Although $M, b \models \downarrow x.\neg\langle R\rangle x$ and $Z(b, a)$ is bisimilar to a, we have that $M, a \not\models \downarrow x.\neg\langle R\rangle x$. $\dashv$

With this result established we have completed the proof of Theorem 4.1: Figure 1 is indeed an exact picture of our expressive hierarchy.

Capturing the correspondence language

We know that all our hybrid languages can be regarded as fragments of the correspondence language. Here we establish a converse result: the language $\Downarrow$ at the top of our hierarchy is strong enough to define all properties that are definable in the correspondence language L_0 using a formula whose only free variable is the special 'localising' variable. We show this by defining the *hybrid translation* of a suitable fragment of L_0.

We first present the translation for signatures containing only unary relation symbols P, and binary relation symbols R. Suppose we are working with such a signature. Choose x to be the 'localising' variable and let L_0^x be the set of L_0 formulae in which x occurs only free. For the atomic formulae of we define:

$$\begin{array}{lcl} HT(v_1 = v_2) & = & \Downarrow x.(v_1 \wedge v_2) \\ HT(P(v_1)) & = & \Downarrow x.(v_1 \wedge P) \\ HT(R(v_1, v_2)) & = & \Downarrow x.(v_1 \wedge \langle R \rangle v_2) \end{array}$$

Note that these definitions ensure that in the special cases in which the first variable v_1 is the localising variable x, the translation produces formulae which are logically equivalent to much simpler formulae. For example, $HT(x = x)$ is a logical truth, $HT(x = y)$ is equivalent to y, and $HT(P(x))$ is equivalent to P.

The definition of HT is extended to complex formulae of L_0 in a straightforward way. The translation of a boolean combination of formulae is the boolean combination of the translation of the formulae, and (making use of the fact that $\Downarrow$ can define the hybrid binder $\exists$) the existential quantifier commutes similarly:

$$HT(\exists y.\varphi) \quad = \quad \exists y.HT(\varphi).$$

For the general case, if $\nu(R) > 2$ then

$$HT(R(v_1, \ldots, v_n)) \quad = \quad \Downarrow x.(v_1 \wedge \langle R \rangle (v_2, \ldots, v_{\nu(R)}))$$

Clearly $M, g, a \models \varphi$ iff $M, g, a \models HT(\varphi)$; and if φ contains only x free, then $HT(\varphi)$ is a sentence of $\Downarrow$. Thus we have proved

Proposition 4.11 $\quad \Downarrow \sim L_0$.

5 Infinite models and undecidability

In this section we show that any language at least as expressive as $\downarrow$ both lacks the finite model property and has an undecidable satisfiability problem. We then turn to the other branch of our hierarchy and show that under a more course-grained measure of expressivity Σ is as strong as the correspondence language; the undecidability of Σ-satisfiability is an immediate corollary.

The finite model property does not hold even for *sentences* of $\downarrow$. Define:

S	$x \wedge \neg\langle R\rangle x \wedge \langle R\rangle\neg x \wedge [R]\langle R\rangle x$
C	$[R][R]{\downarrow}y.(\neg x \rightarrow \langle R\rangle(x \wedge \langle R\rangle y))$
I	$[R]{\downarrow}y.\neg\langle R\rangle y$
D	$[R]\langle R\rangle\neg x$
4	$[R]{\downarrow}y.\langle R\rangle(x \wedge [R](\langle R\rangle(\neg x \wedge \langle R\rangle y \rightarrow \langle R\rangle y)))$

Let *SCID4* be $S \wedge C \wedge I \wedge D \wedge 4$. Note that ${\downarrow}x.\mathit{SCID4}$ is a sentence. This sentence is satisfied in the following model. Let $(\mathbf{N}, <)$ be the natural numbers in their usual order, and suppose $s \notin \mathbf{N}$. Let M be the model bearing a single binary relation R that is defined as follows: $|M|$ is $\mathbf{N} \cup \{s\}$, and R is $< \cup\{(n, s), (s, n) : n \in \mathbf{N}\}$. Let g be any assignment in $|M|$ such that $g(x) = s$. It is clear that $M, g, s \models \mathit{SCID4}$, hence $M, s \models {\downarrow}x.\mathit{SCID4}$. Thus ${\downarrow}x.\mathit{SCID4}$ has at least one (infinite) model.

Proposition 5.1 *If* $M, s \models {\downarrow}x.\mathit{SCID4}$ *then* $|M|$ *is infinite.*

Proof. Suppose $M, s \models {\downarrow}x.\mathit{SCID4}$. Let $B = \{b \in |M| : sRb\}$. Because S is satisfied, $s \notin B$, $B \neq \emptyset$, and for all $b \in B$, bRs. Because C is satisfied, if $a \neq s$ and a is an R-successor of an element of B then a is also an element of B. As I is satisfied at s, every point in B is irreflexive; as D is satisfied at s, every point in B has an R-successor distinct from s; and as 4 is satisfied, R transitively orders B. Hence B is an unbounded strict partial order, thus B is infinite and so is $|M|$. $\dashv$

Note the way that s played the role of a 'spy' point from which a large chunk of the model could be surveyed. The ability of $\downarrow$ to force the existence of models containing spy points is the key to the following undecidability proof. We prove undecidability by reducing the *unbounded tiling problem* to the $\downarrow$-satisfiability problem. A tile t is a 1×1 square, of fixed orientation, with coloured edges *right*(t), *left*(t), *up*(t), and *down*(t) taken from some denumerable set. The unbounded tiling problem is: given a finite set $\mathcal{T}$ of tile types, does there exist a function *tile* from $\mathbf{N} \times \mathbf{N}$ 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 Π^0_1 complete; for further information see Harel (1983).

The undecidability proof is model theoretic. We shall represent tiled $\mathbf{N} \times \mathbf{N}$ 'grids' as models bearing four binary relations: S, U, R and T. The relations U and R (up and right) will represent the grid $\mathbf{N} \times \mathbf{N}$, the T relation the associated tiles. The role of the S relation is to permit all this information to be surveyed: there will be a special spy point s from where the entire decorated grid can be seen via S. This will make it possible to define, for any finite set of tile types $\mathcal{T}$, a formula $\varphi^{\mathcal{T}}$ with the following property: $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$ iff $\varphi^{\mathcal{T}}$ has a model.

The undecidability of $\downarrow$-satisfiability then follows from the undecidability of the unbounded tiling problem.

So, let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a finite set of tile types. Define:

S_1	$x \wedge \neg\langle S\rangle x \wedge \langle S\rangle\neg x \wedge [S]\langle S\rangle x \wedge [S][S]x$
S_2	$[S][U]\neg x \wedge [S][R]\neg x$
D	$[S]\langle U\rangle\top \wedge [S]\langle R\rangle\top$
$C^\dagger$	$[S][\dagger]{\downarrow}y.\langle S\rangle\langle S\rangle y$, for $\dagger \in \{U, R\}$
$P^\dagger$	$[S]{\downarrow}y.\langle\dagger\rangle{\downarrow}z.\langle S\rangle\langle S\rangle(y \wedge [\dagger]z)$, for $\dagger \in \{U, R\}$
G	$[S]{\downarrow}y.\langle U\rangle\langle R\rangle{\downarrow}z.\langle S\rangle\langle S\rangle(y \wedge \langle R\rangle\langle U\rangle z)$
T_i	$\langle T\rangle \cdots$ *i-times* $\cdots \langle T\rangle[T]\perp$
O	$[S]((\bigvee_{i=1}^{k} T_i) \wedge \bigwedge_{1\le i<j\le k}(T_i \to \neg T_j))$
V	$[S](\bigvee_{u(t_i)=d(t_j)}(T_i \wedge \langle U\rangle T_j))$
H	$[S](\bigvee_{r(t_i)=l(t_j)}(T_i \wedge \langle R\rangle T_j))$

Let $\varphi^{\mathcal{T}}$ be ${\downarrow}x.(S_1 \wedge S_2 \wedge D \wedge C^\dagger \wedge P^\dagger \wedge G \wedge O \wedge V \wedge H)$.

Proposition 5.2 *$\mathcal{T}$ tiles $\mathbf{N}\times\mathbf{N}$ iff $\varphi^{\mathcal{T}}$ is satisfiable.*

Proof. Suppose that $M, s \models \varphi^{\mathcal{T}}$. Let G, the set of grid points, be $\{g \in |M| : sSg\}$. Because S_1 is satisfied, $s \notin G$, $G \neq \emptyset$, for all $g \in G$, gRs, and for all $g \in G$, if gRa then $a = s$. Note that from these properties it follows that $[S]\phi$ holds at s iff ϕ is true at all points $g \in G$. Moreover, we also have a (dual of the) universal modality on G itself: for any formula ϕ, the formula $\langle S\rangle\langle S\rangle\phi$ is true at a point $g \in G$ iff there is some point $g' \in G$ satisfying ϕ — in short, every point in G can see every other point in G by making a two step S-excursion.

Bearing these remarks in mind, it is easy to establish the following. Because S_2 is satisfied at s, s is not accessible from G via the relations U or R. Because D is satisfied, every point in G has at least one U and R successor, while the $C^\dagger$ ensure that all these successors are in G. The $P^\dagger$ guarantee that both U and R are partial functions on G (and hence, because of D, total functions on G), while G gives us the desired grid pattern. Note that all these formulas are prefixed by $[S]$, while five of them (the $C^\dagger$, the $P^\dagger$, and G) also make crucial use of the $\langle S\rangle\langle S\rangle$ combination.

Let g_0 be an arbitrary element of G. Let $f\colon \mathbf{N}\times\mathbf{N} \longrightarrow |M|$ be such that $f(0,0) = g_0$, $f(n,m)Vf(n,m+1)$ and $f(n,m)Hf(n+1,m)$. Clearly this function is well-defined, thus we can now define the tiling. Let $\mathit{tile} : \mathbf{N}\times\mathbf{N} \to \mathcal{T}$ be defined by $t(\langle n,m\rangle) = t_i$ iff $\mathbf{M}, g, f(n,m) \models T_i$. Using the fact that O, V and H are satisfied it easily follows that this is a tiling of $\mathbf{N} \times \mathbf{N}$.

For the converse, suppose that $\mathit{tile} : \mathbf{N}\times\mathbf{N} \rightarrow \mathcal{T}$ is a tiling of $\mathbf{N}\times\mathbf{N}$. We build the model M for $\varphi^{\mathcal{T}}$ out of the ordered pairs $\langle n,m\rangle$ together with the following ingredients. First, choose some $s \notin \mathbf{N} \times \mathbf{N}$; this will be our spy point. In addition, for each $\langle n,m\rangle$ choose a set $\mathrm{TL}_{n,m}$ of cardinality i, where *tile* $(\langle n,m\rangle)$ is the i-th tile type. (We choose the $\mathrm{TL}_{n,m}$ so that they are mutually disjoint, and do not contain s or any ordered pair $\langle n,m\rangle$ of natural numbers.) Arbitrarily enumerate the elements of each $\mathrm{TL}_{n,m}$ as $\{l_1 \ldots, l_i\}$. Let $R_{n,m}$ be the binary relation on $\mathrm{TL}_{n,m}$ given by $l_j R l_k$ iff $k = j + 1$. This i-length T-sequence will be used to represent the fact that the tiling associates a tile of the i-th tile type with (n,m). We define the desired model M as follows:

$$|M| = \{s\} \cup \{\langle n,m\rangle : n,m \in \mathbf{N}\} \cup \textstyle\bigcup_{n,m\in\mathbf{N}} \mathrm{TL}_{n,m}$$

$$S = \{\langle s, \langle n,m\rangle\rangle, \langle\langle n,m\rangle, s\rangle : n,m \in \mathbf{N}\}$$

$$U = \{\langle\langle n,m\rangle, \langle n,m+1\rangle\rangle : n,m \in \mathbf{N}\}$$

$$R = \{\langle\langle n,m\rangle, \langle n+1,m\rangle\rangle : n,m \in \mathbf{N}\}$$

$$T = \textstyle\bigcup_{n,m\in\mathbf{N}} R_{n,m} \cup \{\langle\langle n,m\rangle, l^1\rangle : n,m \in \mathbf{N}, l^1 \in \mathrm{TL}_{n,m}\}$$

By construction, $M, s \models \varphi^{\mathcal{T}}$. ⊣

Theorem 5.1 *$\downarrow$ has an undecidable satisfiability problem.*

Proof. Immediate from the previous proposition, and the undecidability of the unbounded tiling problem. ⊣

It is interesting to compare this proof with that of Goranko (1994, 1995). Goranko also reduces the unbounded tiling problem to the satisfiability problem for a hybrid language containing $\downarrow$. However, his proof makes use of a primitive universal modality to create the grid. The spy point method shows that $\downarrow$ can be dangerous even in its absence.

Let us turn to the other branch of the hierarchy. The relevant results for $\Diamond$ are well known (it is an S5 modality, and so has the finite model property and is decidable in NP time) so let's consider Σ.

It is more or less immediate that Σ lacks the finite model property. Consider the binder Π dual to Σ (that is, $\Pi x.\varphi := \neg\Sigma x.\neg\varphi$). Clearly $M, g, a \models \Pi x.\varphi$ iff for all points a' and all assignments $g' \stackrel{x}{=} g$, if $g'(x) = a'$ then $M, g', a' \models \varphi$, thus we can enforce global conditions. Now consider the following formula:

$$\Pi x.(\langle R\rangle\langle R\rangle x \rightarrow \langle R\rangle x) \wedge \Pi x.(x \rightarrow \neg\langle R\rangle x) \wedge \Pi x.(x \rightarrow \langle R\rangle\neg x).$$

All the satisfying models for this sentence are infinite strict partial orders.

The power of Σ to enforce global conditions is the key to the undecidability result. Although the preservation results of the previous section show that Σ is not *locally* as strong as the correspondence language, it is as strong *globally*. Consider the following translation of L_0^x into Σ. (We omit the obvious clauses for the boolean connectives.)

$$
\begin{array}{lcl}
GT(R(y_1, \ldots, y_n) & = & \Diamond(y \wedge R(y_2, \ldots, y_n)) \\
GT(\exists y.\varphi) & = & \Sigma y.GT(\varphi)
\end{array}
$$

Proposition 5.3 *For all models M, and all sentences φ in L_0^x, $M \models \varphi$ iff for all points a in M, $M, a \models GT(\varphi)$.*

Proof. A straightforward induction on the structure of φ. $\dashv$

Corollary 5.1 Σ *has an undecidable satisfiability problem.*

6 Concluding remarks

To conclude this paper we suggest some applications for hybrid languages and note a number of directions for further logical work.

Questions in theoretical computer science have twice lead to the invention of hybrid languages, namely PDL+$\exists$ (Passy and Tinchev 1985b) and Sellink's (1994) $\downarrow$-based system for reasoning about automata; it will be interesting to see how this line of work develops. The other traditional source of hybrid languages has been the study of knowledge representation; here there are several potentially interesting applications. For example, hybrid languages are a natural way of thinking about the temporal representation systems of Allen (1984) and McDermott (1982). This application has connections with a line of work which uses hybrid languages to model natural language temporal semantics; see Richards *et al* (1989), Blackburn (1994) and Goranko (1994, 1995). Moreover, recent work (see Buvač, Buvač and Mason 1994,1995) uses what are essentially three valued hybrid languages for contextual reasoning in AI.

Novel application may arise in computational linguistics, where it is becoming increasingly important to have precise models of syntactic and phonological structure, together with suitable constraint language. As the models used in syntax and phonology tend to be labeled, decorated graphs of some sort, it is usually straightforward to view them as relational structures, and hybrid languages may be appropriate constraint languages: Reape (1993) has formalised parts of HPSG (Pollard and Sag 1987) using a feature logic enriched with $\exists$, and it would be interesting to experiment with hybrid versions of the Bird and Blackburn (1991) account of autosegmental phonology.

On the logical front there remains much to do. While the Sofia school has produced numerous completeness results for $\exists$ and $\downarrow$ in the presence of $\Diamond$ (see Passy and Tinchev 1985b, 1991, and Goranko 1994, 1995) there seem to be few $\Diamond$-free results. It is likely that modal completeness technology will extend relatively straightforwardly to such systems, but interesting issues remain (for example, how are Gabbay style irreflexivity rules best handled in weaker systems?). The new binders also raise proof theoretic issues. Seligman (1994) proves a cut-elimination result for a hybrid language, but which combinations of binders (and retrieval operators) admit such well behaved sequent calculi? Moreover, the present paper has ignored both second order modal definability and algebraic semantics, two glaring omissions. Finally, it would be pleasant to make contact with other 'non-standard' logics. The topological logic of Rescher and Urquhart (1971) has already been noted. In addition, there are interesting points of contact with the work of van Benthem and Alechina (1994), Alechina and Lambalgen (1994) and Alechina (1994) and these deserve further attention.

Acknowledgements

We would like to thank Valentin Goranko for a number of stimulating discussions, and his comments on an earlier draft of the paper. We would also like to thank the referee for a number of suggestions which greatly improved the presentation.

References

[1] Alechina, N., 1994, "On one decidable generalised quantifier logic corresponding to a decidable fragment of first-order logic", *Journal of Logic, Language and Information*, this issue.

[2] Alechina, N. and Lambalgen, M., 1994, "Correspondence and completeness for generalised quantifiers", ILLC Technical Note X-94-03, University of Amsterdam.

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

[4] Bird, S., and Blackburn, P., 1991, "A logical approach to arabic phonology", pp. 89–94 in *Proceedings of the Fifth Conference of the European Chapter of the Association of Computational Linguistics, Berlin.*

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

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

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

[8] Buvač, S., Buvač, V. and Mason, I., 1994, "The semantics of propositional contexts", pp. 468–477 in *Proceedings of the Eighth International Symposium on Methodologies for Intelligent System*, Lecture Notes for Artificial Intelligence, Springer Verlag: Berlin.

[9] Buvač, S., Buvač, V. and Mason, I., 1995, "Metamathematics of contexts", *Fundamenta Informaticae*, **23**, 263–301.

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

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

[12] 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., Berlin: Springer-Verlag.

[13] Goranko, V., 1995, "Hierarchies of modal and temporal logics with reference pointers", to appear in *Journal of Logic, Language, and Information.*

[14] Goranko, V., and Passy, S., 1989, "Using the universal modality: gains and questions", *Journal of Logic and Computation* **2**, 5–30.

[15] Harel, D., 1983, "Recurring dominoes: making the highly undecidable highly understandable", in *Springer Lecture Notes in Computer Science* **158**, 177–194.

[16] Harel, D., 1984, "Dynamic logic", pp. 497–604 in *Handbook of Philosophical Logic*, **2**, D. Gabbay and F. Guenthner, ed., Dordrecht: Reidel.

[17] Hodges, W., 1993, *Model Theory*, Cambridge University Press.

[18] Mcdermott, D., 1982, "A temporal logic for reasoning about processes and plans", *Cognitive Science* **6**, 101–55.

[19] Passy, S. and Tinchev, T., 1985a, "PDL with data constants", *Information Processing Letters* **20**, 35–41.

[20] Passy, S.,and Tinchev, T., 1985b, "Quantifiers in combinatory PDL: completeness, definability, incompleteness", in in *Springer Lecture Notes in Computer Science* **199**, 512–519.

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

[22] Pollard, C., and Sag, I., 1987, *Information Based Syntax and Semantics, Volume 1: Fundamentals*, Stanford: CSLI.

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

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

[25] Reape, M., 1994, "A feature value logic with intensionality, nonwellfoundedness and functional and relational dependencies", pp. 77–110 in *Constraints, Language, and Computation*, C.J. Rupp, M. Rosner, and R. Johnson, ed., London: Academic Press.

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

[27] de Rijke, M. (1992a), The modal logic of inequality, *Journ. Symbolic Logic* **57**, 566–584.

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

[29] Seligman, J., 1991, "A Cut-free Sequent Calculus for Elementary Situated Reasoning", Technical Report HCRC-RP 22, HCRC, Edinburgh.

[30] Seligman, J., 1994, "The Logic of Correct Description", to appear in *Advances in Intensional Logic*, M. de Rijke, ed., Dordrecht: Kluwer.

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

[32] van Benthem, J., 1983, *Modal Logic and Classical Logic*, Napoli: Bibliopolis.

[33] van Benthem, J., 1984., "Correspondence Theory", pp. 167–247 in *Handbook of Philosophical Logic* **2**, D. Gabbay and F. Guenthner, eds., Dordrecht: Reidel.

[34] van Benthem, J., and Alechina, N., 1994, "Modal quantification over structured domains", to appear in *Advances in Intensional Logic*, M. de Rijke, ed., Dordrecht: Kluwer.