Hybrid Logics: The Old and The New

Carlos Areces
Team TALARIS
INRIA Lorraine,
areces@loria.fr

October 25, 2007

Abstract

Hybrid Logics are usually presented as an extension of modal logics where we can explicitly refer to the elements in the model. In this paper we will start by introducing hybrid logics from that perspective, and we will complete this presentation with a historical overview. But we will later have a closer look to the $\downarrow$ binder, and propose an alternative way to think about it. We will explain then how, in this new light, a whole new spectrum of logical languages is offered to the enterprising logician.

Keywords: Hybrid Logics, Modal Logics, Semantics, Model Theory, Expressive Power

1 Explicit Reference

Hybrid languages are usually presented as modal languages that have special symbols to name individual states in models. These new symbols, often called *nominals*, are simply a new sort of atomic symbols $\{i, j, k, \ldots\}$ disjoint from the set of standard propositional variables. For example, if i is a nominal and p and q are propositional variables, then

$$\Box i \rightarrow (\Diamond q \rightarrow \Box q) \text{ and } \Box p \rightarrow (\Diamond q \rightarrow \Box q)$$

are both well formed formulas; but they have quite a different meaning. Actually, as we will now explain, the first is a tautology while the second is contingent. The difference comes from the interpretation that should be attributed to nominals. Because they are standing for particular elements in the model they should be true at a unique state. Formally, their interpretation should be a singleton set. Comming back then to the examples above then, the antecedent $\Box i$ implies that the number of accessible states is at most one, and this condition is sufficient to make the consequent true. This is, of course, not the case for the antecedent $\Box p$ of the second formula, as nothing forbids the existence of various accessible states where p holds.

But nominals are only the tip of the iceberg. Once we have names for states at our disposal we can, for example, introduce, for each nominal i, an operator $@_i$ that allows us to jump to the point named by i. The formula $@_i\varphi$ (read 'at i, φ') moves the point of evaluation to the state named by i and evaluates φ there. Intuitively, the $@_i$ operators internalize the satisfaction relation '$\models$' into the logical language:

$$\mathcal{M}, w \models \varphi \text{ iff } \mathcal{M} \models @_i\varphi, \text{ where } i \text{ is a nominal naming } w.$$

For this reason, these operators are usually called *satisfaction operators*.

If nominals are names for individual states, why not to have also variables ranging over states, with corresponding quantifiers. We would then be able to write formulas like $\forall y.\Diamond y$. The intuitive reading of this formula is simply, $\forall y.R(x, y)$, forcing the current state to be related to all states in the domain. The $\forall$ quantifier is very expressive. As discussed in [Blackburn and Seligman, 1998], even the basic modal language extended with state variables and this universal quantifier is undecidable. Moreover, $\forall$ and @ together give us already full first-order expressive power. Nevertheless, the $\forall$ quantifier is historically important. The earliest treatments are probably those of [Prior, 1967; Prior, 1968; Bull, 1970].

From a modal perspective, other binders besides $\forall$ are possible. The $\downarrow$ binder binds variables to the *current* point of evaluation. In essence, it enables us to create a name for the here-and-now, and refer to it later in the formula. For example, the formula $\downarrow i.\Diamond i$ is true at a state m iff m is related to itself. The intuitive reading is quite straightforward: the formula says "call the current state y and check that y is reachable." Like $\forall$, the $\downarrow$ binder has been invented independently on several occasions. For example, in [Richards *et al.*, 1989], $\downarrow$ is introduced as part of an investigation into temporal semantics and temporal databases, [Sellink, 1994] uses it to aid reasoning about automata, it is related to the freeze operator in [Henzinger, 1990], and [Cresswell, 1990] employs it as part of his treatment of indexicality. However, none of the systems just mentioned allows the free syntactic interplay of variables with the underlying propositional logic; that is, they make use of $\downarrow$, but in languages that are not fully hybrid. The earliest paper to introduce it into a fully hybrid language seems to be [Goranko, 1994]. Summarizing the above discussion, we can say that the term *hybrid logic* refers to a family of extensions of the basic hybrid language with devices that, in one way or another, allow for explicit reference to individual states of the Kripke model.

For a more detailed introduction, including further intuitive examples using the different hybrid languages, the reader is referred to [Areces and ten Cate, 2007]. The Hybrid Logic Web Pages [Areces, 2007] provides additional information and a broad on-line bibliography.

1.1 Basic Definitions

The simplest hybrid language is $\mathcal{H}$, which extends the basic modal language with nominals only. Further extensions will be named by listing the additional operators. The most expressive system we will discuss in detail is $\mathcal{H}(\mathsf{E}, @, \downarrow)$, with the existential modality E, @-operators, and the $\downarrow$ binder (when considering languages containing the $\downarrow$ binder, it is implicitly understood that the language also contains state variables). At various points, we will briefly mention other hybrid languages as well (e.g., hybrid extensions of temporal and dynamic logics).

The following two definitions give the syntax and semantics of $\mathcal{H}(\mathsf{E}, @, \downarrow)$. The corresponding definitions for sublanguages of $\mathcal{H}(\mathsf{E}, @, \downarrow)$ can be obtained by leaving out irrelevant clauses.

Definition 1.1 *Let* $\mathsf{PROP} = \{p_1, p_2, \ldots\}$ *(the* propositional variables*) and* $\mathsf{NOM} = \{i_1, i_2, \ldots\}$ *be pairwise disjoint, countably infinite sets of symbols. The well-formed formulas of the hybrid language are given by the following recursive definition:*

$$\mathsf{FORMS} \ ::= \ p \mid i \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid \mathsf{E}\varphi \mid @_i\varphi \mid \downarrow i.\varphi,$$

where $p \in \mathsf{PROP}$*,* $i \in \mathsf{NOM}$ *and* $\varphi, \varphi_1, \varphi_2 \in \mathsf{FORMS}$*.*

Note that the above syntax is simply that of ordinary propositional modal logic extended with clauses for nominals and for $\mathsf{E}\varphi$, $@_s\varphi$ and $\downarrow x_j.\varphi$. Also, note that, like propositional variables, nominals can be used as atomic formulas.

Definition 1.2 *A (hybrid)* model $\mathcal{M}$ *is a triple* $\mathcal{M} = \langle M, R, V\rangle$ *such that* M *is a non-empty set,* R *is a binary relation on* M*, and* $V : \mathsf{PROP} \to \wp(M)$*.*

We usually write M for the domain of a model $\mathcal{M}$, and call the elements of M states*,* worlds *or* points*. R is an* accessibility relation*, and V is the* valuation*. An* assignment *g for $\mathcal{M}$ is a mapping $g : \mathsf{NOM} \to M$. Given an assignment $g : \mathsf{NOM} \to M$, a nominal $i \in \mathsf{NOM}$, and a state $m \in M$, we define g^i_m (an i-*variant of *g) by letting $g^i_m(i) = m$ and $g^i_m(j) = g(j)$ for all $j \neq i$.*

Let $\mathcal{M} = \langle M, R, V\rangle$ be a model, $m \in M$, and g an assignment for $\mathcal{M}$. Then the satisfaction relation *is defined as follows:*

$$
\begin{array}{lll}
\mathcal{M}, g, m \models p & \textit{iff} & m \in V(p) \quad \textit{for } p \in \mathsf{PROP} \\
\mathcal{M}, g, m \models i & \textit{iff} & m = g(i) \quad \textit{for } i \in \mathsf{NOM} \\
\mathcal{M}, g, m \models \neg\varphi & \textit{iff} & \mathcal{M}, g, m \not\models \varphi \\
\mathcal{M}, g, m \models \varphi_1 \wedge \varphi_2 & \textit{iff} & \mathcal{M}, g, m \models \varphi_1 \textit{ and } \mathcal{M}, g, m \models \varphi_2 \\
\mathcal{M}, g, m \models \Diamond\varphi & \textit{iff} & \textit{there is } m' \textit{ s.t. } R(m, m') \textit{ and } \mathcal{M}, g, m' \models \varphi \\
\mathcal{M}, g, m \models \mathsf{E}\varphi & \textit{iff} & \textit{there is } m' \in M \textit{ s.t. } \mathcal{M}, g, m' \models \varphi \\
\mathcal{M}, g, m \models @_i\varphi & \textit{iff} & \mathcal{M}, g, g(i) \models \varphi \quad \textit{for } i \in \mathsf{NOM} \\
\mathcal{M}, g, m \models {\downarrow} i.\varphi & \textit{iff} & \mathcal{M}, g^i_m, m \models \varphi.
\end{array}
$$

The first five clauses in the definition of the satisfaction relation are similar to the ones for the basic modal language, except that they are relativized to an additional assignment function. Recall that nominals can be used as atomic formulas, in which case they act as propositional variables that are true at a unique state. The $\downarrow$ binder binds nominals to the state where evaluation is being performed (the *current world*), and $@_i$ shifts evaluation to the state named by i.

Definitions 1.1 and 1.2 specify the syntax and semantics of the very expressive hybrid language $\mathcal{H}(\mathsf{E}, @, \downarrow)$ (which is actually as expressive as first-order logic). Two important fragments of this language are $\mathcal{H}(@, \downarrow)$, which is obtained by dropping the clauses for the existential modality E, and $\mathcal{H}(@)$, which is obtained by dropping in addition the $\downarrow$-binder. In other words, $\mathcal{H}(@)$ is simply the extension of the basic modal language with nominals and satisfaction operators.

2 History

In this section we will provide an overview of the historical development of hybrid languages, starting with the pioneering work of Prior, through the "revival" in the late eighties and early nineties in Sofia, and ending with the work of Blackburn and Seligman in the late nineties.

The Foundational Work of Prior: The work of Prior in modal logic and in particular in the modal analysis of time is well known, to the point that he is usually regarded as the inventor of temporal logic. For a detailed discussion of Prior's contributions to this field, together with some biographical information, see [Øhrstrøm and Hasle, 1993; Copeland, 1999; Blackburn, 2001].

Nowadays, the view that modal logic can be seen as a fragment of first-order or second-order logic is commonplace. This is fairly straightforward once we observe the possible worlds semantics of modal operators. When reading the earlier work of Prior, however, we should keep in mind that, at that time, most modal intuitions came solely from axiomatics.

Already in this setup, Prior discusses the relation between the modal and the first order language in relation to the way we experience time and tense. His ideas are made explicit in his work discussing McTaggart 'series of time' [McTaggart, 1908]. Working already in the framework of temporal logic, he will introduce in [Prior, 1967, Chapter V.6] the I-calculus. In the I-calculus, propositions of the tense calculus are treated as predicates expressing properties of dates (which are represented by variables). The formula px should be read as "p at x," and I is a binary relation taking dates as arguments where Ixy is read as "y is later than x." Using an arbitrary date x to represent the time of utterance, Fp (intuitively, "the proposition p happens in the future") is equated with $\exists y.(Ixy \wedge py)$ (i.e., "p at some time later than x") and similarly for Pp, "the proposition p happens in the past." Prior mentions already that, by imposing various conditions on the relation I, analogues of the axioms of the tense calculus can be derived in the I-calculus. From today's perspective this is nothing else that interpreting the temporal language in terms of a first-order correspondence language.

But Prior expressed strong reserves concerning "reducing modality to quantity." And his intuitions on the foundational nature of modality later grew into a mature philosophy in Prior's view that quantification over possible worlds and instants was to be interpreted in terms of modality and tense — which constituted primitive notions — and not vice versa (although he did recognized that the study of both quantity and modality could benefit of each other).

This idea of the primacy of the tense calculus over the I-calculus (of McTaggart's A-series over the B-series [McTaggart, 1908] was to become a central and distinctive tenet of his philosophy. These issues form the theme of his final, unfinished, book [Prior, 1977], but they already appear in the earlier articles mentioned above.

But of course, the reconstruction of the I-calculus within the tense calculus is impossible, as the I-calculus is strictly more expressive than the tense calculus. Prior recognized this fact and investigated ways to extend the expressive power of the tense calculus to permit the reconstruction. This directly led to what we call today *very expressive hybrid languages* (i.e., hybrid languages including the $\forall$ binder). In [Prior, 1967, Chapter V.6], he actually proposes a way to develop the I-calculus inside the tense calculus, and for this he allows instant variables to be used together with propositional variables. He will call this step "the third grade of tense-logical involvement" in [Prior, 1968, Chapter XI], where instant variables are treated as representing (special) propositions.

We see, then, that Prior's development of hybrid languages was rooted in his philosophical convictions, and was instrumental in the implementation of some of his very early intuitions on time and tense. Prior's death in 1969 put an end to these investigations. Notice though, that Prior was never fully satisfied with his solution. It was technically correct (and actually quite bold and ingenious) but he was concerned that, in managing to "upgrade" the tense calculus to full first-order expressivity, the language had lost its claim to a metaphysical fundamentality. Robert Bull, a student of Prior, pushed the ideas of hybridization further in [Bull, 1970], where he provides an axiomatization and completeness result for a logic containing variables for *paths* on a model, which he calls "history-propositional" variables.

The Sofia School: As we saw, the roots of hybrid logic go back to Prior and Bull. About fifteen years later in Sofia, Bulgaria, nominals were rediscovered by Gargov, Passy and Tinchev in their investigations on Boolean modal logic and propositional dynamic logic. One of the issues that led them into these investigations was the following asymmetry in the expressive power of the modal language. The union of two accessibility relations is definable in the basic modal language, in the sense that the formula

$$\langle T\rangle p \leftrightarrow \langle R\rangle p \vee \langle S\rangle p$$

is valid on a frame precisely if the accessibility relation interpreting $\langle T\rangle$ is the union of the accessibility relations interpreting $\langle R\rangle$ and $\langle S\rangle$. Moreover, when added to the basic modal language, this formula completely axiomatizes the modal logic of the relevant class of frames.

Surprisingly, *intersection* of accessibility relations is not definable in the same way: there is no formula in the basic modal language that is valid on a frame precisely if the accessibility relation of $\langle T\rangle$ is the intersection of the accessibility relation of $\langle R\rangle$ and $\langle S\rangle$.

Now, Gargov, Passy and Tinchev showed in [Gargov *et al.*, 1987] that intersection *can* be defined using *nominals*. Indeed, for i a nominal, the axiom scheme

$$\langle T\rangle i \leftrightarrow \langle R\rangle i \wedge \langle S\rangle i$$

defines intersection in the above sense, and exactly axiomatizes the logic of the relevant class of frames (when added to an appropriate base axiomatization). The same story goes for complementation: there is no formula of the basic modal language that is valid on a frame precisely if the accessibility relation of $\langle R\rangle$ is the complement of the accessibility relation of S, but such a formula exists when nominals are added to the language: $\langle R\rangle i \leftrightarrow \neg\langle S\rangle i$.

This form of capturing the Boolean operations was investigated by Gargov, Passy and Tinchev in [Gargov *et al.*, 1987]. In that paper, the first complete axiomatization of the minimal hybrid language is given.

Besides the minimal hybrid language $\mathcal{H}$, Gargov, Passy and Tinchev also studied a richer hybrid language, obtained by extending propositional dynamic logic (PDL, [Harel, 1984]) with nominals. Intersection of accessibility relations is particularly interesting in this setting, as it can be interpreted as parallelism, or concurrency of programs. Passy and Tinchev [Passy and Tinchev, 1985] propose an extension of PDL with nominals and the universal modality, which they call Combinatory PDL. The paper contains an axiomatization of CPDL$(\cap, ^{-}, \subset, ^{-1})$, combinatory PDL extended with program intersection, complementation, subprograms and inverse.

The Sofia tradition in hybrid logics continues with the work of Goranko. In [Gargov and Goranko, 1993], Gargov and Goranko investigate the basic modal language extended first with nominals and the universal and existential modalities ($\mathcal{H}(\mathsf{E})$), and then with the difference operator D. They prove that both languages are equivalent with respect to frame definability, and then provide characterizations of frame definability for these languages.

The work of Gargov and Goranko is historically relevant because, within the Sofia school, it marks the start of research on hybrid logics as such, and not as part of their research on extensions of PDL. Around the same time, but independently, Blackburn was studying simple hybrid languages over a Prior-style tense logic [Blackburn, 1990; Blackburn, 1992]. These two lines of research can be considered the origins of the current perspective on hybrid logics.

Goranko is also the first to investigate the $\downarrow$ binder in the context of hybrid logic. In [Goranko, 1994], he extends the basic modal language with the universal modality and the $\downarrow$ binder with only a single state variable (though using a slightly different notation). Goranko provides an axiomatization for this logic, and illustrations of its high expressivity (sufficient, for example, to define Kamp's $U(p,q)$ and $S(p,q)$ and Stavi's $U'(p,q)$ and $S'(p,q)$ temporal operators and to simulate Prior's instant variables), and shows that the satisfiability problem for this language is undecidable. He mentions in the same paper that introducing multiple state variables would be possible, and investigates the resulting language in more detail in [Goranko, 1996].

It is interesting to note that most of the languages studied by the Sofia school included the universal modality. In the following years and mainly through the work of Blackburn and Seligman, research in hybrid languages deals with, on the one hand, weak languages containing only nominals (e.g., [Blackburn, 1993; Blackburn and Spaan, 1993]) and, on the other hand, very expressive languages containing binders (e.g., [Blackburn and Seligman, 1995; Blackburn and Tzakova, 1998; Blackburn and Tzakova, 1999]).

Very Expressive Hybrid Languages: In the mid-nineties, Blackburn and Seligman [Blackburn and Seligman, 1995] studied a number of very expressive hybrid languages, obtained by means of various state variable binders. We will review a few of these binders here, but only very briefly.

Up to now, we have introduced two hybrid binders, the "classical" $\exists$ and the "more modal" $\downarrow$. Let us review their semantic definitions. Given a model $\mathcal{M} = \langle M, R, V\rangle$, an assignment g in $\mathcal{M}$ and $m \in M$:

$$\begin{array}{lll} \mathcal{M}, g, m \models \exists i.\varphi & \text{iff} & \mathcal{M}, g^i_{m'}, m \models \varphi \text{ for some } m' \in M. \\ \mathcal{M}, g, m \models {\downarrow} i.\varphi & \text{iff} & \mathcal{M}, g^i_m, m \models \varphi. \end{array}$$

Both binders let us change the value assigned to i, without changing the point of evaluation. In [Blackburn and Seligman, 1995] Blackburn and Seligman investigate two other binders which, besides changing the value of the bound variable, also change the point of evaluation:

$$\begin{array}{lll} \mathcal{M}, g, m \models \Sigma i.\varphi & \text{iff} & \mathcal{M}, g^i_{m'}, m' \models \varphi \text{ for some } m' \in M. \\ \mathcal{M}, g, m \models {\Downarrow} i.\varphi & \text{iff} & \mathcal{M}, g^i_m, m' \models \varphi \text{ for some } m' \in M. \end{array}$$

It is not hard to see that $\Sigma i.\varphi$ is equivalent to $\mathsf{E}{\downarrow} i.\varphi$, whereas ${\Downarrow} i.\varphi$ is equivalent to ${\downarrow} i.\mathsf{E}\varphi$. The main result in [Blackburn and Seligman, 1995] is that these binders form an expressive hierarchy. If we let $<$ stand for the relation "is strictly less expressive than" then we have that $\mathcal{H}(\downarrow) < \mathcal{H}(\exists) < \mathcal{H}(\Downarrow)$ and $\mathcal{H}(\mathsf{E}) < \mathcal{H}(\Sigma) < \mathcal{H}(\Downarrow)$. The expressivity inclusions are proved using the following equivalences (where j is a nominal not in φ):

$$\begin{array}{lcl} \downarrow i.\varphi & \equiv & \exists i.(i \wedge \varphi) \\ \exists i.\varphi & \equiv & \Downarrow j.\Downarrow i.(j \wedge \varphi) \\ \mathsf{E}\varphi & \equiv & \Sigma j.\varphi \\ \Sigma i.\varphi & \equiv & \Downarrow j.\Downarrow i.(i \wedge \varphi). \end{array}$$

Moreover, the equivalence $\Downarrow i.\varphi \equiv \downarrow i.\mathsf{E}\varphi$ shows that $\mathcal{H}(\Downarrow) \leq \mathcal{H}(\downarrow, \mathsf{E})$ and hence any language containing an operator from each of the two "branches" in the hierarchy is expressively equivalent to $\mathcal{H}(\Downarrow)$. The strictness of the hierarchy is proved in [Blackburn and Seligman, 1995] using different variants of bisimulations, preserving truth of formulas of the various languages.

In [Tzakova, 1999a; Tzakova, 1999b], Tzakova explores some examples of very expressive hybrid languages with binding operators in more detail, both axiomatically and by means of tableaux systems.

3 A Detailed Look at ↓

As should be clear from the previous section, different binders for hybrid logics have been investigated in detail. Let us now take a different look at ↓ and its relation with @ from a slightly different perspective.

First, note that satisfaction operators work in perfect coordination with ↓. Whereas ↓ "stores" the current point of evaluation (by binding a variable to it), the satisfaction operators enable us to "retrieve" stored information by shifting the point of evaluation in the model. By using the "storing and retrieving" intuition it is easy to define complex properties. For example, Kamp's temporal *until* operator U (with semantics: $U(\varphi, \psi)$ is true at a state m if there is a future state m' where φ holds, such that ψ holds in all states between m and m') can be defined as follows:

$$U(\varphi, \psi) := \downarrow i.\Diamond\downarrow j.(\varphi \wedge @_i\Box(\Diamond j \rightarrow \psi)).$$

Let us see how this work. First, we name the current state i using ↓, and use the $\Diamond$ operator to find a suitable successor state, which we call j, where φ holds. Without the @ operator we would be stuck in that successor state, but we can use @ to go back to i and demand that in all successors of i having j as a successor, ψ holds.

One way to understanding the interplay described above is from a more dynamic perspective. We can think that $\downarrow i.$ is *modifying* the current model (by storing the current point of evaluation into i), and that $@_i$ is being evaluated in the modified model. We can make this idea more concrete by working with *pointed models* [Blackburn *et al.*, 2001] and 'moving' the valuation and the point of evaluation 'within' our definition of a model. I.e., instead of writing $\langle W, R, V\rangle, g, m$ we will write $\langle W, R, V, g, m\rangle$, now clearly $\downarrow$ arrow is modifying the model, by switching evaluation to a model with a different assignment g'. And the same thing is doing the very traditional modal operator $\Diamond$, moving the component 'point of evaluation' of our model to a suitable successor.

It should be clear that we are not doing anything too special or revolutionary with our change of perspective with respect to what a model actually is. Truly enough, it is perhaps not so clear that $\downarrow$ acts as a 'quantifier,' but we are completely happy with that. We never called $\downarrow$ a 'quantifier', only a 'binder' and it does bind nominals to specific values. Similarly, we can see the assignment g as a particular type of 'information storage' in our model, and consider $\downarrow$ and @ as included in the logical language, as our way to access this information storage for reading and writing.

But let us take a step back and consider the new picture. When we introduced the $\downarrow$ binder, our main aim was to define a binder which was weaker than the first-order quantifier. We thought of the semantics of $\downarrow$ first, and we suitable adjusted the way we updated the assignment later. In the last section we presented alternative possibilities, when we discussed other binders investigated by Blackburn and Seligman. But why do we need to restrict ourselves to binders and assignments?

Models as Information Storage: Let us start with a standard (pointed) Kripke models $\langle W, R, V, w\rangle$, and let us consider a very simple addition: just a set $S \subseteq W$. We can, for example, think of S as a set of states that are, for some reason, 'known' to us. Already in this very simple set up we can define the following operators

$$\begin{array}{rcl} \langle W, R, V, w, S\rangle \models \texttt{(remember)}\varphi & \text{iff} & \langle W, R, V, w, S \cup \{w\}\rangle \models \varphi \\ \langle W, R, V, w, S\rangle \models \texttt{(forget)}\varphi & \text{iff} & \langle W, R, V, w, S \backslash \{w\}\rangle \models \varphi \\ \langle W, R, V, w, S\rangle \models \texttt{known} & \text{iff} & w \in S \end{array}$$

Already with this simple language we would have that for any model $\mathcal{M}$, $\mathcal{M} \models \Diamond\top \wedge \texttt{(forget)(remember)}\Box\texttt{known}$ will be true only in points which are related only with themselves.

Notice that we could write the same formula using $\downarrow$, as $\Diamond\top \wedge {\downarrow} i.\Box i$, but the new language is much less expressive than $\mathcal{H}(\downarrow)$, as we cannot discern between states stored in S, while an assignment g keeps a complete mapping between states and nominals.

But of course, we can include structures which are richer than a simple set, in our models. Let us consider one further example. Let S be now a stack of elements that we will represent as a list that 'grows to the right,' and let us define the operators:

$$\begin{array}{rcl}
\langle W, R, V, w, S\rangle \models \texttt{(push)}\varphi & \text{iff} & \langle W, R, V, w, S\cdot w\rangle \models \varphi \\
\langle W, R, V, w, S\cdot w'\rangle \models \texttt{(pop)}\varphi & \text{iff} & \langle W, R, V, w, S\rangle \models \varphi \\
\langle W, R, V, w, []\rangle \models \texttt{(pop)}\varphi & \text{never} & \\
\langle W, R, V, w, S\cdot w'\rangle \models \texttt{top} & \text{iff} & w = w'.
\end{array}$$

Concretely, the proposal is to take seriously the usual saying that 'modal languages are languages to talk about labelled graphs' but give us the freedom to chose what we want to 'remember' about a given graph and how we are going to store it. The standard Kripke structure $\langle W, R, V\rangle$ represents the graph that we are trying to describe, and we will describe it using a local perspective from the point of evaluation w, but we can also add to these new structures that we use to record some other particular characteristics that we are interested in.

This way of looking to models seems to provide a high degree of freedom and it opens the way to many new and interesting proposal for modal logics. Are all these logics 'hybrid'? we can ask ourselves. Conveniently, the term 'hybrid' is vague enough to easily accommodate them. Are they 'modal'? I believe that the answer is affirmative. They are all specialized languages to describe labelled graphs, and they all follow very closely the spirit of Prior.

4 Conclusions

In this paper, we have introduced hybrid logics following closely the material presented in [Areces and ten Cate, 2007] (actually, in many ways this paper should be considered as a shorter version of the first sections of that chapter). But we have then moved on to present a new way of looking at the $\mathcal{H}(@,\downarrow)$ language and in particular to the status that the assignment used to interpret nominals, @ and $\downarrow$ plays in the model. We have then propose a generalization of that idea, which leads to a new spectrum of logical languages.

There is much work to be done to explore this new landscape. The ideas we discussed here are actually very much related to some ideas comming from the area of epistemic logics. Many authors in that field, have introduced already operators that in one way or another can be seen as modifying the model (see for example [Baltag *et al.*, 1998; Kooi, 2007]).

References

[Areces and ten Cate, 2007] C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logic*. Elsevier, 2007.

[Areces, 2007] C. Areces. HyLo: The hybrid logics' web site, 2007. `http://hylo.loria.fr`. Last visited: Oct/07.

[Baltag *et al.*, 1998] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of TARK'98*, pages 43–56, 1998.

[Blackburn and Seligman, 1995] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.

[Blackburn and Seligman, 1998] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic 1*, pages 41–62. CSLI Publications, Stanford University, 1998.

[Blackburn and Spaan, 1993] P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information*, 2:129–169, 1993.

[Blackburn and Tzakova, 1998] P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 6:625–650, 1998.

[Blackburn and Tzakova, 1999] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999.

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

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

[Blackburn, 1992] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 34(1):56–83, 1992.

[Blackburn, 1993] P. Blackburn. Modal logic and attribute value structures. In M. de Rijke, editor, *Diamonds and Defaults*, Synthese Language Library, pages 19–65. Kluwer Academic Publishers, Dordrecht, 1993.

[Blackburn, 2001] P. Blackburn. Modal logic as dialogical logic. *Synthese*, 127:57–93, 2001.

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

[Copeland, 1999] B. Copeland. Arthur Prior. In E. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 1999. `http://plato.stanford.edu/archives/win1999/entries/prior`.

[Cresswell, 1990] M. Cresswell. *Entities and Indices*. Reidel, Dordrecht, 1990.

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

[Gargov *et al.*, 1987] G. Gargov, S. Passy, and T. Tinchev. Modal environment for boolean speculations. In D. Scordev, editor, *Mathematical Logic and its Applications*, pages 253–263. Plenum Press, New York, 1987.

[Goranko, 1994] V. Goranko. Temporal logic with reference pointers. In *Temporal Logic*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 133–148. Springer-Verlag, 1994.

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

[Harel, 1984] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic. Vol. II*, volume 165 of *Synthese Library*, pages 497–604. D. Reidel Publishing Co., Dordrecht, 1984. Extensions of classical logic.

[Henzinger, 1990] T. Henzinger. Half-order modal logic: how to prove real-time properties. In *PODC '90: Proceedings of the 9th annual ACM Symposium on Principles of Distributed Computing*, pages 281–296. ACM Press, 1990.

[Kooi, 2007] B. Kooi. Expressivity and completeness for public update logic via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2), 2007.

[McTaggart, 1908] J. McTaggart. The unreality of time. *Mind*, pages 457–474, 1908.

[Øhrstrøm and Hasle, 1993] P. Øhrstrøm and P. Hasle. A. N. Prior's rediscovery of tense logic. *Erkenntnis*, 39:23–50, 1993.

[Passy and Tinchev, 1985] S. Passy and T. Tinchev. PDL with data constants. *Information Processing Letters*, 20(1):35–41, 1985.

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

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

[Prior, 1977] A. Prior. *Worlds, Times and Selves.* Duckworth, London, 1977. Edited by K. Fine.

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

[Sellink, 1994] M. Sellink. Verifying modal formulae over I/O-automata by means of type theory. Logic group preprint series, Utrecht University, 1994.

[Tzakova, 1999a] M. Tzakova. *Hybrid Languages.* PhD thesis, Technischen Fakultät der Universität des Saarlandes, 1999.

[Tzakova, 1999b] M. Tzakova. Tableaux calculi for hybrid logics. In N. Murray, editor, *Proceedings of the Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 278–292. Springer Verlag, 1999.