Expressive Power and Decidability for Memory Logics

Carlos Areces\textsuperscript{1}, Diego Figueira\textsuperscript{2}, Santiago Figueira\textsuperscript{3,4}, and Sergio Mera\textsuperscript{3,*}

\textsuperscript{1} INRIA Nancy Grand Est, France
\textsuperscript{2} LSV, ENS Cachan, CNRS, INRIA, France
\textsuperscript{3} Departamento de Computación, FCEyN, UBA, Argentina
\textsuperscript{4} CONICET, Argentina

Abstract. Taking as inspiration the hybrid logic $\mathcal{H}L(\downarrow)$, we introduce a new family of logics that we call memory logics. In this article we present in detail two interesting members of this family defining their formal syntax and semantics. We then introduce a proper notion of bisimulation and investigate their expressive power (in comparison with modal and hybrid logics). We will prove that in terms of expressive power, the memory logics we discuss in this paper are more expressive than orthodox modal logic, but less expressive than $\mathcal{H}L(\downarrow)$. We also establish the undecidability of their satisfiability problems.

1 Memory Logics: Hybrid Logics with a Twist

Hybrid languages have been extensively investigated in the past years. $\mathcal{H}L$, the simplest hybrid language, is usually presented as the basic modal language $\mathcal{K}$ extended with special symbols (called nominals) to name individual states in a model. These new symbols are simply a new sort of atomic symbols $\{i, j, k, \ldots\}$ disjoint from the set of standard propositional variables. While they behave syntactically exactly as propositional variables do, their semantic interpretation differ: nominals denote elements in the model, instead of sets of elements. This simple addition already results in increased expressive power. For example the formula $i \land \langle r \rangle i$ is true in a state $w$, only if $w$ is a reflexive point named by the nominal $i$. As the basic modal language is invariant under unraveling, there is no equivalent modal formula \[1\].

But as we said above, $\mathcal{H}L$ is just the simplest hybrid language. Once nominals have been added to the language, other natural extensions arise. Having names for states at our disposal we can introduce, for each nominal $i$, an operator $@_i$ that allows us to jump to the point named by $i$ obtaining the language $\mathcal{H}L(@)$. The formula $@_i \varphi$ (read ‘at $i$, $\varphi$’) moves the point of evaluation to the state named by $i$ and evaluates $\varphi$ there. Intuitively, the $@_i$ operators internalize the satisfaction relation ‘$|$’ into the logical language: $\mathcal{M}, w \models \varphi$ iff $\mathcal{M} \models @_i \varphi$, where $i$ is a nominal naming $w$. For this reason, these operators are usually called satisfaction operators.

\* S. Mera is partially supported by a grant of Fundación YPF.
If nominals are names for individual states, why not introduce also binders. We would then be able to write formulas like $\forall i. \langle r \rangle i$, which will be true at a state $w$ if it is related to all states in the domain. The $\forall$ quantifier is very expressive: the satisfiability problem of $\mathcal{HL}(\forall)$ ($\mathcal{HL}$ extended with the universal binder $\forall$) is undecidable \cite{2}. Moreover, $\mathcal{HL}(\@, \forall)$ is expressively equivalent to full first-order logic (over the appropriate signature).

From a modal perspective, other binders besides $\forall$ are possible. The $\downarrow$ binder binds nominals 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 $\downarrow i. \langle r \rangle i$ is true at a state $w$ if and only if it is related to itself. The intuitive reading is quite straightforward: the formula says “call the current state $i$ and check that $i$ is reachable”. The logic $\mathcal{HL}(\downarrow)$ is also very expressive but weaker than $\mathcal{HL}(\forall)$. Sadly, its satisfiability problem is also undecidable.

Different binders for hybrid logics have been investigated in detail (see \cite{2}), but in this article we want to take a look at $\downarrow$ from a slightly different perspective: we will consider nominals and $\downarrow$ as ways for storing and retrieving information in the model.

Models as Information Storage. We should note that nominals and $\downarrow$ work nicely together. Whereas $\downarrow i$ stores the current point of evaluation in the nominal $i$, nominals act as checkpoints enabling us to retrieve stored information by verifying if the current point is named by a given nominal $i$. To make this point clear, let’s define formally the semantics of $\mathcal{HL}(\downarrow)$.

**Definition 1.** A hybrid signature $S$ is a tuple $\langle \text{PROP}, \text{REL}, \text{NOM} \rangle$ where PROP, REL, NOM are mutually disjoint infinite enumerable sets (the sets of propositional symbols, relational symbols and nominals, respectively).

Formulas of $\mathcal{HL}(\downarrow)$ are defined over a given $S$ by the following rules:

$$\text{FORMS} ::= p \mid i \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \langle r \rangle \varphi \mid \downarrow i. \varphi,$$

where $p \in \text{PROP}$, $i \in \text{NOM}$, $r \in \text{REL}$ and $\varphi, \varphi_1, \varphi_2 \in \text{FORMS}$. Formulas in which any nominal $i$ appears in the scope of a binder $\downarrow i$ are called sentences.

A model for $\mathcal{HL}(\downarrow)$ over a signature $S$ is a tuple $\langle W, (R_r)_{r \in \text{REL}}, V, g \rangle$ where $\langle W, (R_r)_{r \in \text{REL}}, V \rangle$ is a standard Kripke model (i.e., $W$ is a non empty set, each $R_r$ is a binary relation over $W$, and $V$ is a valuation), and $g$ is an assignment function from NOM to $W$.

Given a model $\mathcal{M} = \langle W, (R_r)_{r \in \text{REL}}, V, g \rangle$ the semantic conditions for the propositional and modal operators are defined as usual (see \cite{1}), and in addition:

- $\langle W, (R_r)_{r \in \text{REL}}, V, g \rangle, w \models i \iff g(i) = w$
- $\langle W, (R_r)_{r \in \text{REL}}, V, g \rangle, w \models \downarrow i. \varphi \iff (W, (R_r)_{r \in \text{REL}}, V, g'_w), w \models \varphi$

where $g'_w$ is the assignment identical to $g$ except perhaps in that $g'_w(i) = w$.

We can think that $\downarrow i$ is modifying the model (by storing the current point of evaluation into $i$), and that $i$ is being evaluated in the modified model. We can