Ordered resolution with selection for $\mathcal{H}(@)$

Carlos Areces[1] and Daniel Gorín[2]

[1] LORIA - INRIA Lorraine
areces@loria.fr
[2] Universidad de Buenos Aires
dgorin@dc.uba.ar

Abstract. The hybrid logic $\mathcal{H}(@)$ is obtained by adding nominals and the satisfaction operator @ to the basic modal logic. The resulting logic gains expressive power without increasing the complexity of the satisfiability problem, which remains within PSpace. A resolution calculus for $\mathcal{H}(@)$ was introduced in [5], but it did not provide strategies for ordered resolution and selection functions. Additionally, the problem of termination was left open.

In this paper we address both issues. We first define proper notions of admissible orderings and selection functions and prove the refutational completeness of the obtained ordered resolution calculus using a standard "candidate model" construction [10]. Next, we refine some of the nominal-handling rules and show that the resulting calculus is sound, complete and can only generate a finite number of clauses, establishing termination. Finally, the theoretical results were tested empirically by implementing the new strategies into HyLoRes [6, 18], an experimental prototype for the original calculus described in [5]. Both versions of the prover were compared and we discuss some preliminary results.

1 Introduction

Modal logics are languages which offer relatively high expressive power, but which, unlike full classical first-order logic, have a decidable satisfiability problem [12] (deciding satisfiability for the basic modal logic is PSpace-complete). Traditional modal logics, though, suffer from some important expressive limitations: 1) they can't make explicit reference to concrete elements of the domain, and 2) they can't express equality between elements. Hybrid logics [11] are a family of extensions of classical modal logics that aim to solve these limitations by the introduction of nominals and special modal operators.

Intuitively, a nominal is a *name* for an element of a model even though, from a syntactic point of view, it behaves like a proposition symbol and can be used wherever the latter is acceptable. For instance, if i and j are nominals, and p is a proposition symbol, we can write formulas such as $i \wedge p \wedge \langle r \rangle (p \wedge [r]j)$. In this paper we will consider only the basic hybrid logic $\mathcal{H}(@)$, i.e., the extension of the basic modal logic with nominals and the satisfaction operator @, that allows the evaluation of a formula at a specific element of the model.

Formally, the set of formulas of $\mathcal{H}(@)$ is defined with respect to a signature $\mathcal{S} = \langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL} \rangle$, where $\mathsf{PROP} = \{p, q, r, \ldots\}$ (the proposition symbols), $\mathsf{NOM} = \{i, j, k, \ldots\}$ (the nominals), and $\mathsf{REL} = \{r_1, r_2, r_3, \ldots\}$ (the relation symbols) are infinite, enumerable, pairwise disjoint sets. $\mathsf{ATOM} = \mathsf{PROP} \cup \mathsf{NOM}$ is the set of atomic symbols. Given a signature $\mathcal{S}$ the set of $\mathcal{H}(@)$-formulas over $\mathcal{S}$ is defined as:

$$\mathcal{H}(@) ::= a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \langle r \rangle \varphi \mid @_n \varphi$$

where $a \in \mathsf{ATOM}$, $n \in \mathsf{NOM}$, $r \in \mathsf{REL}$ and $\varphi, \varphi' \in \mathcal{H}(@)$. The remaining standard operators ($\vee$, $\rightarrow$, $[r]$, etc.) are defined in the usual way.

Definition 1 (validity). *A* hybrid model *is a structure* $M = \langle W, \{r^M \mid r \in \mathsf{REL}\}, V \rangle$ *where* W *is a non-empty set (the* domain *of the model, whose elements are called* states*),* $r^M \subseteq W \times W$ *is a binary relation for each* $r \in \mathsf{REL}$*,* $V(p) \subseteq W$ *for each* $p \in \mathsf{PROP}$*, and* $V(n) = \{w\}$ *for some* $w \in W$ *when* $n \in \mathsf{NOM}$*.*

Given a hybrid model $M = \langle W, \{r^M \mid r \in \mathsf{REL}\}, V \rangle$ *and an element* $w \in W$*, the satisfiability relation* $M, w \models \varphi$ *(read "model* M *satisfies the formula* φ *at state* w*") is defined as follows:*

$$\begin{array}{rll} M, w \models a & \textit{iff} & w \in V(a),\ a \in \mathsf{ATOM} \\ M, w \models \neg\varphi & \textit{iff} & M, w \not\models \varphi \\ M, w \models \varphi_1 \wedge \varphi_2 & \textit{iff} & M, w \models \varphi_1 \textit{ and } M, w \models \varphi_2 \\ M, w \models \langle r \rangle \varphi & \textit{iff} & \textit{exists } w' \in W \textit{ such that } r^M(w, w') \textit{ and } M, w' \models \varphi \\ M, w \models @_n \varphi & \textit{iff} & M, w' \models \varphi,\ \textit{with } w' \in V(n). \end{array}$$

The logic $\mathcal{H}(@)$ introduces, through nominals and @, a weak notion of equality reasoning. For example, the formulas

$$\begin{array}{ll} @_i i & \text{(reflexivity),} \\ @_i j \leftrightarrow @_j i & \text{(symmetry),} \\ (@_i j \wedge @_j k) \rightarrow @_i k & \text{(transitivity), and} \\ @_i j \rightarrow (\varphi \leftrightarrow \varphi(i/j)) & \text{(substitution by identicals)} \end{array}$$

are tautologies of $\mathcal{H}(@)$. This notion is not present in the basic modal logic and it can be shown that $\mathcal{H}(@)$ is strictly more expressive [2]. Nevertheless, its satisfiability problem remains within PSpace [3].

The most successful automated theorem proving implementations for modal logics are based on the tableau method and much of their outstanding performance is due to the heavy use of several heuristics and refinements [8]. However, a number of these heuristics don't work or become rather involved when the underlying logic allows some form of equality. When nominals are added, the performance of the tableaux-based theorem provers is severely affected. In this scenario, it makes sense to investigate other kinds of algorithms. In particular, we will discuss resolution, the most successful automated theorem proving method for first-order logic with equality [10, 9].

In [5] a resolution based calculus for $\mathcal{H}(@)$ is proposed. The formulation of the calculus that we will present takes formulas in *negation normal form*, i.e., the negation operator can only be applied to atoms[1]. As a consequence, both $\vee$ and $[\,]$ become primitive symbols. Let $\mathcal{S} = \langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL} \rangle$ be a signature, we define the set $\mathcal{H}^{\text{NNF}}(@)$ as follows:

$$\mathcal{H}^{\text{NNF}}(@) ::= a \mid \neg a \mid \varphi \vee \varphi' \mid \varphi \wedge \varphi' \mid \langle r \rangle \varphi \mid [r]\varphi \mid @_i\varphi$$

where $a \in \mathsf{ATOM}$, $r \in \mathsf{REL}$, $i \in \mathsf{NOM}$ and $\varphi, \varphi' \in \mathcal{H}^{\text{NNF}}(@)$. We will call formulas of the form $@_i\varphi$, *@-formulas*. We will consider, from now on, only formulas of $\mathcal{H}^{\text{NNF}}(@)$, unless the contrary is stated.

Like the resolution calculus for first-order logic, the hybrid resolution calculus works on sets of *clauses*. A clause, in this context, is a set of arbitrary $\mathcal{H}^{\text{NNF}}(@)$ @-formulas. A clause represents the disjunction of its formulas, but there's no additional restriction regarding the form of the formulas (i.e., they do not need to be literals). It is worth noting that to allow only @-formulas in a clause is not an expressivity limitation in terms of satisfiability: a formula φ is satisfiable if and only if for an arbitrary nominal i not occurring in φ, $@_i\varphi$ is satisfiable.

Given a formula $\varphi \in \mathcal{H}^{\text{NNF}}(@)$, we define $\mathit{ClSet}(\varphi) = \{\{@_i\varphi\}\}$, for i an arbitrary nominal not occurring in φ. We can now define $\mathit{ClSet}^*(\varphi)$ — the saturated set of clauses for φ — as the smallest set that includes $\mathit{ClSet}(\varphi)$ and is saturated under the rules of the resolution calculus $\mathbf{R}[\mathcal{H}^{\text{NNF}}(@)]$ given in Figure 1, where $i, j \in \mathsf{NOM}$ and $p \in \mathsf{PROP}$.

$$(\wedge)\ \frac{Cl \cup \{@_i(\varphi_1 \wedge \varphi_2)\}}{\begin{array}{c}Cl \cup \{@_i\varphi_1\}\\ Cl \cup \{@_i\varphi_2\}\end{array}} \qquad (\vee)\ \frac{Cl \cup \{@_i(\varphi_1 \vee \varphi_2)\}}{Cl \cup \{@_i\varphi_1, @_i\varphi_2\}}$$

$$(\text{RES})\ \frac{Cl_1 \cup \{@_ip\} \quad Cl_2 \cup \{@_i\neg p\}}{Cl_1 \cup Cl_2}$$

$$([r])\ \frac{Cl_1 \cup \{@_i[r]\varphi\} \quad Cl_2 \cup \{@_i\langle r \rangle j\}}{Cl_1 \cup Cl_2 \cup \{@_j\varphi\}} \qquad (\langle r \rangle)\ \frac{Cl \cup \{@_i\langle r \rangle\varphi\}}{\begin{array}{c}Cl \cup \{@_i\langle r \rangle j\}\\ Cl \cup \{@_j\varphi\}\end{array}}\ \text{for a new } j \in \mathsf{NOM}$$

$$(@)\ \frac{Cl \cup \{@_i@_j\varphi\}}{Cl \cup \{@_j\varphi\}}$$

$$(\text{SYM})\ \frac{Cl \cup \{@_ij\}}{Cl \cup \{@_ji\}} \qquad (\text{REF})\ \frac{Cl \cup \{@_i\neg i\}}{Cl} \qquad (\text{PAR})\ \frac{Cl_1 \cup \{@_ij\} \quad Cl_2 \cup \{\varphi(j)\}}{Cl_1 \cup Cl_2 \cup \{\varphi(j/i)\}}$$

Fig. 1. The Resolution Calculus $\mathbf{R}[\mathcal{H}^{\text{NNF}}(@)]$.

We can group these rules according to their role. The $(\wedge)$, $(\vee)$ and $(@)$ rules handle formula simplification. The $(\langle r \rangle)$ rule does a mild skolemization, assigning a new name (through a new nominal) to an element of the model which was

[1] The restriction to formulas in negation normal form simplifies the definition of admissible orderings and selections functions, but it also have effects on the calculus as we can see in Figure 1 where the (RES) rule applies only to literals.

existentially quantified (through a diamond). The (RES) rule works like the resolution rule for first-order logic, while the ($[r]$) rule encodes a non-trivial unification plus a resolution step. Finally, the (SYM), (REF) and (PAR) rules are the standard set of rules for equality handling in (function free) first-order logic resolution [9].

The construction of $ClSet^*(\varphi)$ is a correct and complete algorithm to decide satisfiability for $\mathcal{H}^{\text{NNF}}(@)$ (and hence for $\mathcal{H}(@)$): φ is unsatisfiable if and only if the empty clause $\{\}$ is an element of $ClSet^*(\varphi)$ [5]. However, $ClSet^*(\varphi)$ might be an infinite set because each application of the $\langle r \rangle$-rule introduces a new nominal. Thus, there are formulas whose satisfiability this algorithm can't decide in a finite number of steps. In Section 4 we show how to turn this calculus into a decision method for $\mathcal{H}(@)$.

A standard technique to regulate the generation of clauses in resolution for first-order logic is called *ordered resolution with selection functions* [10]. The general idea is to establish certain conditions under which it is safe to *chose* a literal from each clause such that rules are to be applied to a clause only to eliminate its chosen literal. The ordered resolution calculus with selection functions is refutationally complete for first-order logic when an ordering $\succ$ with certain properties is used (see [10] for further details). In the following sections we develop similar strategies for $\mathbf{R}[\mathcal{H}^{\text{NNF}}(@)]$.

2 Ordered hybrid resolution with selection functions

In the context of resolution systems, an ordering between formulas is called *admissible* when it can be used in a calculus of ordered resolution, preserving refutational completeness. In this section we propose an ordered resolution calculus for $\mathcal{H}^{\text{NNF}}(@)$.

The following definitions are standard (see, e.g. [13]). A binary relation $\succ$ is called an *ordering* if it is transitive and irreflexive; if, additionally, for any two distinct elements x and y one of $x \succ y$ or $y \succ x$ holds, $\succ$ is said to be *total*. An ordering $\succ$ is called *well-founded* when there is no infinite chain $x_1 \succ x_2 \succ x_3 \dots$. Let $\succ$ be an ordering between formulas, and let's indicate with $\varphi[\psi]_p$ a formula φ where ψ appears at position p. We say that $\succ$ has the *subformula property* if $\varphi[\psi]_p \succ \psi$ whenever $\varphi[\psi]_p \neq \psi$, and that it is a *rewrite ordering* when $\varphi[\psi_1]_p \succ \varphi[\psi_2]_p$ iff $\psi_1 \succ \psi_2$. A well-founded rewrite ordering is called a *reduction ordering*, and if it also has the subformula property, it is called a *simplification ordering*. We will use the same symbol to denote both an ordering on formulas and its standard extension to clauses.

We can now define the notion of admissible ordering for resolution on $\mathcal{H}^{\text{NNF}}(@)$.

Definition 2 (admissible ordering). *An ordering $\succ$ over $\mathcal{H}^{NNF}(@)$ is* admissible *if it is a total simplification ordering satisfying the following conditions for all $\varphi, \psi \in \mathcal{H}^{NNF}(@)$ and all $i, j \in \mathsf{NOM}$:*

A1) $\varphi \succ i$ for all $\varphi \notin \mathsf{NOM}$
A2) if $\varphi \succ \psi$, then $@_i\varphi \succ @_j\psi$

A3) if $\langle r\rangle i$ is a proper subformula of φ, then $\varphi \succ \langle r\rangle j$
A4) $[r]i \succ \langle r\rangle j$.

Condition A1 states that nominals must be smaller than formulas other than nominals, condition A2 requires the operator @ not to affect the ordering among formulas, condition A3 introduces a very weak notion of structural complexity, while condition A4 prioritizes $[\ ]$ over $\langle\ \rangle$.

It is easy to show that the conditions in Definition 2 are not too restrictive, and that there actually exists orderings satisfying them. A standard method for building simplification orderings is by using the *lexicographic path orderings* ($\succ_{lpo}$), (see [13] for a definition), which is defined given a set of operators O and an ordering $>$ on O (the precedence), over the set of well formed terms $T(O)$. When the ordering $>$ is well-founded (and total), then $\succ_{lpo}$ is a (total) simplification ordering.

In this context, since we will define an ordering based on lpo, it will be convenient to treat @, $\langle\ \rangle$ and $[\ ]$ as binary operators: $@(\cdot,\cdot) : \mathcal{H}^{\text{NNF}}(@) \times \mathsf{NOM} \to \mathcal{H}^{\text{NNF}}(@)$[2], $\langle\ \rangle(\cdot,\cdot) : \mathsf{REL} \times \mathcal{H}^{\text{NNF}}(@) \to \mathcal{H}^{\text{NNF}}(@)$ and $[\ ](\cdot,\cdot) : \mathsf{REL} \times \mathcal{H}^{\text{NNF}}(@) \to \mathcal{H}^{\text{NNF}}(@)$ but we will keep the notation $@_n\varphi$, $\langle r\rangle\varphi$ and $[r]\varphi$.

We give the following constructive definition of an admissible ordering based on lpo over the set $O = \mathsf{PROP} \cup \mathsf{NOM} \cup \mathsf{REL} \cup \{\neg, \wedge, \vee, @, \langle\ \rangle, [\ ]\}$ with the obvious arities (note that $\mathcal{H}^{\text{NNF}}(@) \subset T(O)$).

Definition 3. *Given a hybrid signature $\mathcal{S} = \langle \{p_i \mid i \in \mathbb{N}\}, \{n_i \mid i \in \mathbb{N}\}, \{r_i \mid i \in \mathbb{N}\}\rangle$, let O be the set $\mathcal{S} \cup \{\neg, \wedge, \vee, @, [\ ], \langle\ \rangle\}$, and define the precedence relation $> \subseteq O \times O$ as the transitive closure of the set*

$$\begin{gathered}\{(@,\neg),(\neg,\wedge),(\wedge,\vee),(\vee,[\ ]),([\ ],\langle\ \rangle)\} \cup \\ \{(\langle\ \rangle,r_i),(r_i,p_j),(p_j,n_k) \mid i,j,k \in \mathbb{N}\} \cup \\ \{(r_i,r_j),(p_i,p_j),(n_i,n_j) \mid i > j\}.\end{gathered}$$

By definition, $>$ is total, irreflexive and well-founded. Let $\succ_{lpo}$ be the lpo over $\mathcal{H}^{\text{NNF}}(@)$ that uses $>$ as precedence. It follows that $\succ_{lpo}$ must be a total simplification ordering. Finally, define $\succ_h$ as

$$\varphi \succ_h \psi \text{ iff } \begin{cases} size(\varphi) > size(\psi), \text{ or} \\ size(\varphi) = size(\psi) \text{ and } \varphi \succ_{lpo} \psi \end{cases}$$

where $size(\varphi)$ is the number of operators in φ.

Proposition 1. *$\succ_h$ is an admissible ordering.*

Observe that no admissible ordering can be defined using lpo alone. It suffices to note that there's no way to guarantee $\langle r'\rangle\langle r\rangle i \succ_{lpo} \langle r\rangle j$ when $r \succ_{lpo} r'$ and $j \succ_{lpo} i$, which violates A3. Unless stated otherwise, from now on we will use $\succ$ to refer to some arbitrary but fixed admissible ordering.

[2] The order of the parameters of this operator has been chosen to simplify Definition 3 and the proof of Proposition 1.

$$(\wedge)\ \frac{Cl \cup \{@_i(\varphi_1 \wedge \varphi_2)\}}{\begin{array}{c}Cl \cup \{@_i\varphi_1\} \\ Cl \cup \{@_i\varphi_2\}\end{array}} \qquad (\vee)\ \frac{Cl \cup \{@_i(\varphi_1 \vee \varphi_2)\}}{Cl \cup \{@_i\varphi_1, @_i\varphi_2\}}$$

$$\text{(RES)}\ \frac{Cl_1 \cup \{@_i p\} \quad Cl_2 \cup \{@_i \neg p\}}{Cl_1 \cup Cl_2}$$

$$([r])\ \frac{Cl_1 \cup \{@_i[r]\varphi\} \quad Cl_2 \cup \{@_i\langle r\rangle j\}}{Cl_1 \cup Cl_2 \cup \{@_j\varphi\}} \qquad (\langle r\rangle)\ \frac{Cl \cup \{@_i\langle r\rangle\varphi\}}{\begin{array}{c}Cl \cup \{@_i\langle r\rangle j\} \\ Cl \cup \{@_j\varphi\}\end{array}}\ \begin{array}{l}\text{for a new } j \in \mathsf{NOM} \\ \text{and } \varphi \notin \mathsf{NOM}\end{array}$$

$$(@)\ \frac{Cl \cup \{@_i@_j\varphi\}}{Cl \cup \{@_j\varphi\}} \qquad \text{(REF)}\ \frac{Cl \cup \{@_i\neg i\}}{Cl}$$

$$\text{(SYM)}\ \frac{Cl \cup \{@_j i\}}{Cl \cup \{@_i j\}}\ \text{if } i \succ j \qquad \text{(PAR)}\ \frac{Cl_1 \cup \{@_j i\} \quad Cl_2 \cup \{\varphi(j)\}}{Cl_1 \cup Cl_2 \cup \{\varphi(j/i)\}}\ \begin{array}{l}\text{if } j \succ i \text{ and} \\ \varphi(j) \succ @_j i\end{array}$$

Restrictions: Assume an admissible ordering $\succ$ and a selection function S. In the following, φ and ψ are the formulas explicitly displayed in the rules. The main premise of each rule is the rightmost, the other premise (in rules with two premises) is the side premise.

- If $C = C' \cup \{\varphi\}$ is the main premise, then either $S(C) = \{\varphi\}$ or, $S(C) = \emptyset$ and $\{\varphi\} \succ C'$.
- If $D = D' \cup \{\psi\}$ is the side premise, then $\{\psi\} \succ D'$ and $S(D) = \emptyset$.

Fig. 2. The Resolution Calculus $\mathbf{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$.

Finally, in resolution for first-order logic, a selection function may chose only negative literals from a clause. As we work with clauses which can contain arbitrary @-formulas from $\mathcal{H}^{\text{NNF}}(@)$ we define "negative literals" as the complement of the set PLIT of positive literals, where $\mathsf{PLIT} ::= @_i j \mid @_i p \mid @_i\langle r\rangle j$, for $i, j \in \mathsf{NOM}$, $p \in \mathsf{PROP}$ and $r \in \mathsf{REL}$.

Definition 4 (selection function). *A function S from clauses to clauses is a* selection function *if and only if, for every clause C we have $S(C) \subseteq C$, $|S(C)| \leq 1$ and $S(C) \cap \mathsf{PLIT} = \emptyset$.*

We are now ready to formulate the strategy of ordered resolution with selection functions for $\mathcal{H}^{\text{NNF}}(@)$. Figure 2 contains the rules of the calculus.

The rules of $\mathbf{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ differ from the ones in Figure 1 only in the addition of some restrictions, both local (in the ($\langle r\rangle$), (SYM) and (PAR) rules) and global. Notice that, as an effect of the global restrictions, there is only one formula in each clause that may be involved in an inference. We will call this formula the *distinguished formula* of the clause.

3 Refutational completeness of $\mathbf{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$

The standard proof of refutational completeness for first-order logic resolution is via the generation of potential Herbrand models [10]. In this section we start by showing that an appropriate notion of Herbrand model can be defined for hybrid languages containing nominals and @.

The following result was established in [20] for CPDL a version of PDL (Propositional Dynamic Logic [17]) extended with hybrid operators, but it holds for any hybrid logics containing nominals and @. For N a hybrid model, let $\mathsf{diag}(N)$, the *diagram of* N, be the set $\mathsf{diag}(N) = \{\varphi \mid \varphi \in \mathsf{PLIT} \text{ and } N \models \varphi\} \cup \{\neg\varphi \mid \varphi \in \mathsf{PLIT} \text{ and } N \not\models \varphi\}$, and call a model *named* if each state of its domain satisfies at least one nominal.

Theorem 1 (Scott's Isomorphism Theorem). *Let M and N be two countable, named hybrid models. Then M and N are isomorphic iff $M \models \mathsf{diag}(N)$.*

Based on Theorem 1 we can define hybrid Herbrand models as follows:

Definition 5 (Herbrand model). *Let $\mathcal{S} = \langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL} \rangle$ be a hybrid signature. A* hybrid Herbrand model *for $\mathcal{H}(@)$ over $\mathcal{S}$ is any set $I \subseteq \mathsf{PLIT}$.*

We identify a Herbrand model with a set of positive literals. This set will uniquely define certain hybrid model.

Definition 6. *Given a hybrid Herbrand model I, let $\sim_I$ be the minimum equivalence relation over NOM that extends the set $\{(i, j) \mid @_i j \in I\}$. We now define the* hybrid model uniquely determined by I *as $\langle W^I, \{r^I \mid r \in \mathsf{REL}\}, V^I \rangle$ where*

$$\begin{array}{ll} W^I & = \mathsf{NOM}/_{\sim_I} \\ r^I & = \{([j], [k]) \mid @_j \langle r \rangle k \in I\} \\ V^I(p) & = \{[j] \mid @_j p \in I\}, p \in \mathsf{PROP} \\ V^I(i) & = \{[i]\}, i \in \mathsf{NOM}. \end{array}$$

where $\mathsf{NOM}/_{\sim_I}$ is the set consisting of equivalence classes of $\sim_I$, and $[i]$ is the equivalence class assigned to i by $\sim_I$.

From now on, we will not distinguish between a hybrid Herbrand model I and its associated model. We will say, for instance, that a formula $@_i\varphi$ is true in I whenever it is satisfied by its associated model (as we are always referring to @-formulas no explicit point of evaluation is needed).

The following theorem (easily proved using Theorem 1) shows that we can work with Herbrand models instead of arbitrary models.

Theorem 2. *Given Γ, a set of @-formulas of $\mathcal{H}(@)$ over a signature $\mathcal{S} = \langle \mathsf{PROP}, \mathsf{NOM}, \mathsf{REL} \rangle$, Γ has a hybrid model if and only if it has a hybrid Herbrand model over the signature $\mathcal{S}' = \langle \mathsf{PROP}, \mathsf{NOM} \cup \mathsf{NOM}', \mathsf{REL} \rangle$, where NOM' is a numerable set disjoint from NOM.*

We are now ready to prove the refutational completeness of $\mathbf{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$. The idea is to build a candidate Herbrand model from an arbitrary (and potentially infinite) set of clauses, such that if the least clause of the set is not true under this model, then the calculus must allow the derivation of a new clause which will also fail to be true in the model. By definition, the empty clause is the smallest clause in any admissible ordering, and it can be shown that admissible orderings ensure that any consequent of a rule is smaller than the main premise. We will, thus, prove that the process leads to either the construction of a Herbrand model for the initial formula (i.e., the initial formula was satisfiable), or to the inclusion of the empty clause in the saturated set (i.e., the initial formula was unsatisfiable).

The definition of a candidate model given below is more complex than the one in [10]. This is because the latter was used for first-order logic without equality, while in $\mathcal{H}(@)$ we have to deal with equalities of the form $@_i j$.

Definition 7 (σ_I). *Given a hybrid Herbrand interpretation I, we define the following substitution of nominals by nominals:*

$$\sigma_I = \{i \mapsto j \mid i \sim_I j \wedge (\forall k)(k \sim_I j \rightarrow k \succeq j)\}.$$

σ_I substitutes each nominal with the least nominal of its class, which is taken as the class representative.

Definition 8. *We define the set of* simple formulas *of $\mathcal{H}^{NNF}(@)$ over $\mathcal{S}$ as:*

$$\mathsf{SIMP} ::= @_i j \ \textit{(with } i \succ j\textit{)} \mid @_i p \mid @_i \neg a \mid @_i \langle r \rangle j \mid @_i [r] \varphi$$

where $i, j \in \mathsf{NOM}$, $p \in \mathsf{PROP}$, $a \in \mathsf{ATOM}$, $r \in \mathsf{REL}$ and $\varphi \in \mathcal{H}^{NNF}(@)$.

Let N be a fixed set of clauses. The following three definitions must be taken as a unit. They are presented separately for clarity but are mutually recursive.

Definition 9 (I_C). *Let C be a clause (not necessarily in N), we name I_C the hybrid Herbrand interpretation given by $\bigcup_{C \succ D} \varepsilon_D$.*

Definition 10 (reduced form). *Let C be a clause and φ its maximal formula. If $\varphi \in \mathsf{SIMP}$ and either a) $\varphi \in \mathsf{PLIT}$ and $\varphi = \varphi\sigma_{I_C}$, or b) $\varphi = @_i[r]\psi$ and $i = i\sigma_{I_C}$; then we say that both φ and C are in* reduced form.

Definition 11 (ε_C). *Let C be a clause (not necessarily in N). If it simultaneously holds that: a) $C \in N$, b) C is in reduced form, c) The maximal formula in C is in PLIT, d) C is false under I_C, and e) $S(C) = \emptyset$; then $\varepsilon_C = \{\varphi\}$, where φ is the maximal formula in C; otherwise, ε_C is the empty set.*

We say that C *produces* φ if $\varepsilon_C = \{\varphi\}$ and call it a *productive clause*. I_C is the *partial interpretation of N below C*. Only those clauses whose maximal formula φ is a positive literal and have no selected formulas may be productive.

Definition 12 (candidate model). *I_N, a* candidate model *for N, is defined as $\bigcup_{C \in N} \varepsilon_C$.*

If a clause C is false under I, we say that C is a *counterexample* of I. Analyzing all the rules of the calculus and considering separately those distinguished formulas that are not in reduced form, the following result can be proved.

Proposition 2. *Let N be a set of clauses and $C \in N$ be the minimum counterexample of I_N, with respect to an admissible ordering $\succ$. If $C \neq \{\}$, then there exists an inference using one of the rules of the calculus such that:*

1. *C is the main premise*
2. *the side premise (when present) is productive*
3. *all the consequents are smaller, with respect to $\succ$, than C and at least one of them is a counterexample of I_N.*

Proof. Using Definition 2 we can easily check that every consequent in the calculus is smaller than the main premise of its inference. The hard part of the proof is to verify that a proper side premise (when required) exists.

Let φ be the distinguished formula of C. If $\varphi \notin \mathsf{SIMP}$, C is trivially the premise of some unary rule and the proposition holds. Now, suppose $\varphi \in \mathsf{SIMP}$ is not in reduced form; this means that some clause D produces $@_i j$ for an i occurring in φ. It is easy to check that, in this case, (PAR) can be applied on D and C. Finally, if φ is in reduced form, it must be of the form $@_i \neg a$ (for $a \in \mathsf{ATOM}$) or $@_i [r] \psi$. The first case is handled either by the (REF) or the (RES) rules, and the proof is analogous to the standard one for first-order logic.

The latter case deserves more attention. The non-trivial part of the proof is to see that a clause in N must produce some $@_i \langle r \rangle j$ such that $@_j \psi$ is false in I_N; but this follows from the fact that C is a counterexample in reduced form and that, for any $k, l \in \mathsf{NOM}$, if $@_k l \in I_N$, then $l = k\sigma_{I_N}$.

Refutational completeness can be easily established from Proposition 2.

Theorem 3. $\boldsymbol{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ *is refutationally complete.*

4 Termination of the calculus

In this section we show how the calculus $\mathbf{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ can be turned into a decision procedure for satisfiability. We will introduce the necessary changes to ensure that for any formula $\varphi \in \mathcal{H}(@)$, $\mathit{ClSet}^*(\varphi)$ is a finite set. If this condition holds, implementing an algorithm that computes $\mathit{ClSet}^*(\varphi)$ in finite time (e.g., the "given clause algorithm" [23]) is straightforward.

The calculus $\mathbf{R}[\mathcal{H}^{\text{NNF}}(@)]$ of Figure 1 can trivially generate an infinite saturated set of clauses as the ($\langle r \rangle$) rule can be applied on formulas of the form $@_i \langle r \rangle j$ for $j \in \mathsf{NOM}$.[3] $\mathbf{R}^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ avoids this behavior, but an infinite number of nominals can still be introduced by interaction between the ($[r]$), ($\langle r \rangle$) and (PAR) rules. As no other symbols but nominals are introduced during resolution, and given that formulas in consequent clauses are never larger (in number of operators) than those in the antecedent, if we can control the generation of nominals we will obtain termination.

[3] Actually, just repetitive application of the ($\langle r \rangle$) rule to the same clause can lead to the generation of an infinite set, but this can be easily avoided by ensuring that the rule is applied only once to each $\langle r \rangle$-formula in a clause.

There are essentially two ways in which an infinite number of nominals can be introduced by the rules of $\mathbf{R}^{\mathrm{OS}}[\mathcal{H}^{\mathrm{NNF}}(@)]$:

Type 1. A formula of the form $@_i\langle r\rangle\varphi$ introduces a new nominal which, in turn, contributes to the derivation of a new clause containing $@_i\langle r\rangle\varphi$. All new nominals are immediate successors of i and they are actually representing the same state in the model, but the calculus cannot detect it.

Type 2. There is a formula φ and an infinite sequence of distinct nominals $n_0, n_1, n_2, \ldots$ such that, for all $i \in \mathbb{N}$, some $@_{n_i}\langle r\rangle\varphi_i$ in the saturated set introduces, by way of the ($\langle r\rangle$) rule, the nominal n_{i+1}. The calculus is exploring a cycle in the model, and cannot detect when to stop the search.

For concrete examples, try the rules of $\mathbf{R}^{\mathrm{OS}}[\mathcal{H}^{\mathrm{NNF}}(@)]$ over the formulas $@_i([r](i \wedge (q \vee \langle r\rangle p)) \wedge \langle r\rangle p)$ and $@_i([r](i \wedge \langle r\rangle p) \wedge \langle r\rangle p)$ using any admissible ordering where i is the least nominal.

As we see, to obtain termination we need to impose some control on the way new nominals are generated by the ($\langle r\rangle$) rule, and on how chains of nominal successors are treated. Solving problems of Type 1 is relatively easy. The next proposition provides the key to the solution.

Proposition 3. *Let $\varphi, \psi \in \mathcal{H}^{\mathrm{NNF}}(@)$ be such that $\varphi[@_i\langle r\rangle\psi]_p$, and let $j \in \mathsf{NOM}$ not occur in φ. Then, φ is satisfiable iff $\varphi[@_i\langle r\rangle\psi/(@_i\langle r\rangle j \wedge @_j\psi)]$ is satisfiable.*

Notice that the proposition involves *simultaneous replacement* of all subformulas $@_i\langle r\rangle\psi$ in φ by $(@_i\langle r\rangle j \wedge @_j\psi)$. The ($\langle r\rangle$) rule instead, uses *a new nominal* in each application, even when applied to the same formula $@_i\langle r\rangle\psi$.

The solution to problems of Type 1 then is to define a function *nom* assigning a unique nominal to each formula $@_i\langle r\rangle\psi$ and redefine the ($\langle r\rangle$) rule with the help of *nom*. Solving problems of Type 2 is more involved, and it is here where the (PAR) rule plays an important role. We will see that we can use, also here, the function *nom* to our advantage.

Let's start by properly defining this function. We first differentiate between nominals which appear in the input formula and nominals generated by application of the ($\langle r\rangle$) rule. Let NOM_i (the set of *initial nominals*) and NOM_c (the set of *computation nominals*) be infinite sets such that $\mathsf{NOM}_i \cap \mathsf{NOM}_c = \{\}$ and $\mathsf{NOM}_i \cup \mathsf{NOM}_c = \mathsf{NOM}$.

We additionally assume, without loss of generality, that $\mathsf{NOM}_c = \bigcup_{k \in \mathbb{N}} N_c^k$, where the sets N_c^k are infinite, pairwise disjoint and well-ordered by $\succ$. And we impose the additional, mild condition on $\succ$ requiring $s \succ t$ whenever $s \in \mathsf{NOM}_c^j$, $t \in \mathsf{NOM}_c^k$ and $j > k$. These conditions will simplify the definition of *nom* and the proof of termination. From now on we assume that NOM_c and $\succ$ comply with these requirements.

Now, let $\mathcal{H}_i^{\mathrm{NNF}}(@)$ be the subset of $\mathcal{H}^{\mathrm{NNF}}(@)$ where only nominals in NOM_i occur and define

$$\mathcal{H}_{@\diamond}^{\mathrm{NNF}}(@) = \{@_i\langle r\rangle\varphi \mid i \in \mathsf{NOM}, r \in \mathsf{REL}, \varphi \in \mathcal{H}_i^{\mathrm{NNF}}(@), \varphi \notin \mathsf{NOM}\},$$

the set of those @-formulas of $\mathcal{H}^{\mathrm{NNF}}(@)$ that can be the distinguished formula of a premise of the ($\langle r\rangle$) rule.

$$(\wedge)\ \frac{Cl \cup \{@_i(\varphi_1 \wedge \varphi_2)\}}{\begin{array}{c} Cl \cup \{@_i\varphi_1\} \\ Cl \cup \{@_i\varphi_2\}\end{array}} \qquad (\vee)\ \frac{Cl \cup \{@_i(\varphi_1 \vee \varphi_2)\}}{Cl \cup \{@_i\varphi_1, @_i\varphi_2\}}$$

$$(\text{RES})\ \frac{Cl_1 \cup \{@_i p\} \quad Cl_2 \cup \{@_i \neg p\}}{Cl_1 \cup Cl_2}$$

$$([r])\ \frac{Cl_1 \cup \{@_i[r]\varphi\} \quad Cl_2 \cup \{@_i\langle r\rangle s\}}{Cl_1 \cup Cl_2 \cup \{@_s\varphi\}} \qquad (\langle r\rangle')\ \frac{Cl \cup \{@_i\langle r\rangle\varphi\}}{\begin{array}{c} Cl \cup \{@_i\langle r\rangle j\} \\ Cl \cup \{@_j\varphi\}\end{array}}\ \begin{array}{l} \varphi \notin \mathsf{NOM} \text{ and} \\ j = nom^{\succ}(@_i\langle r\rangle\varphi)\end{array}$$

$$(@)\ \frac{Cl \cup \{@_i@_j\varphi\}}{Cl \cup \{@_j\varphi\}} \qquad (\text{REF})\ \frac{Cl \cup \{@_i\neg i\}}{Cl} \qquad (\text{SYM})\ \frac{Cl \cup \{@_j i\}}{Cl \cup \{@_i j\}}\ \text{ if } i \succ j$$

$$(\text{PAR'})\quad \frac{Cl_1 \cup \{@_j i\} \quad Cl_2 \cup \{\varphi(j)\}}{Cl_1 \cup Cl_2 \cup \{\varphi(j/i)\}}\ \begin{array}{l} \text{if } j \succ i,\ \varphi(j) \succ @_j i, \text{ and whenever} \\ \varphi(j) = @_k\langle r\rangle l, \text{ then } l \in \mathsf{NOM}_i, \text{ or} \\ i \in \mathsf{NOM}_i \text{ and } l = j\end{array}$$

$$(\text{PAR-}@\Diamond)\ \frac{Cl_1 \cup \{@_j i\} \quad Cl_2 \cup \{@_j\langle r\rangle k\}}{\begin{array}{c} Cl_1 \cup Cl_2 \cup \{@_i\langle r\rangle l\} \\ Cl_1 \cup Cl_2 \cup \{@_k l\}\end{array}}\ \begin{array}{l} \text{if } j \succ i \text{ and } k \in \mathsf{NOM}_c, \text{ and for} \\ \text{some } \varphi,\ k = nom^{\succ}(@_j\varphi), \text{ and} \\ l = nom^{\succ}(@_i\varphi)\end{array}$$

Restrictions: Assume an admissible ordering $\succ$, a proper $nom^{\succ}$ function and a selection function S. In the following, φ and ψ are the formulas explicitly displayed in the rules. The main premise of each rule is the rightmost, the other premise (in rules with two premises) is the side premise.

- If $C = C' \cup \{\varphi\}$ is the main premise, then either $S(C) = \{\varphi\}$ or, $S(C) = \emptyset$ and $\{\varphi\} \succ C'$.
- If $D = D' \cup \{\psi\}$ is the side premise, then $\{\psi\} \succ D'$ and $S(D) = \emptyset$.

Fig. 3. The Resolution Calculus $\mathbf{R}_T^{\mathrm{OS}}[\mathcal{H}^{\mathrm{NNF}}(@)]$.

Definition 13 **($nom^{\succ}$).** *Given $\succ$ an admissible ordering, let $nom^{\succ} : \mathcal{H}^{NNF}_{@\Diamond}(@) \to \mathsf{NOM}_c$ be any function such that*

1. *$nom^{\succ}$ is injective,*
2. *$i \succ j$ iff $nom^{\succ}(@_i\langle r\rangle\varphi) \succ nom^{\succ}(@_j\langle r\rangle\varphi)$, for any $i, j \in \mathsf{NOM}_c$, and*
3. *for all $j \in \mathsf{NOM}_c^i$ there exists a formula $@_k\langle r\rangle\varphi$ such that $j = nom^{\succ}(@_k\langle r\rangle\varphi)$ and, either $i = 0$ and $k \in \mathsf{NOM}_i$, or else $k \in \mathsf{NOM}_c^{i-1}$*

Condition 1) is required for soundness: we can use the same nominal for each $@_i\langle r\rangle\varphi$ formula, but no two different formulas in $\mathcal{H}^{\mathrm{NNF}}_{@\Diamond}(@)$ should use the same nominal. Condition 2) is needed to guarantee refutational completeness. Finally, condition 3) avoids cycles (like in $i = nom^{\succ}(@_i\langle r\rangle\psi)$) and, more important, it is required in order to obtain a terminating calculus.

It can be easily shown that Definition 13 is not too restrictive; i.e., that it can be satisfied by a concrete function. For example, let n_φ be a new nominal for each formula $\varphi \in \mathcal{H}^{\mathrm{NNF}}_{@\Diamond}(@)$ and make $n_\varphi = nom^{\succ}(\varphi)$.

Figure 3 shows the calculus $\mathbf{R}_T^{\mathrm{OS}}[\mathcal{H}^{\mathrm{NNF}}(@)]$ for which we will establish refutational completeness and termination. Notice that the $(\langle r \rangle)$ rule has been replaced by $(\langle r \rangle')$ which uses the $nom^{\succ}$ function to always assign the same nominal to a formula in $\mathcal{H}_{@\diamond}^{\mathrm{NNF}}$. The (PAR) rule has been replaced by two rules: (PAR') and (PAR-@$\diamond$). (PAR') is just a restriction of (PAR) which does not handle certain formulas of the form $@_i\langle r \rangle j$. Such formulas are treated in a special way by the (PAR-@$\diamond$) rule. The (PAR-@$\diamond$) rule deserves some explanation. The intuition behind this rule is the following: if j and i denote the same state in the model (as indicated by the distinguished formula $@_j i$ in the side premise) then, by Proposition 3, k and l can be taken to be equal too. However, by Definition 13, $k \succ l$ and, thus, l should be preferred over k.

We now proceed to discuss soundness, refutational completeness and termination of $\mathbf{R}_T^{\mathrm{OS}}[\mathcal{H}^{\mathrm{NNF}}(@)]$, starting with soundness:

Theorem 4. *If $\varphi \in \mathcal{H}_i^{NNF}(@)$ is satisfiable, then $ClSet^*(\varphi)$ (closed by the rules of $\boldsymbol{R}_T^{\mathrm{OS}}[\mathcal{H}^{NNF}(@)]$) is satisfiable.*

Proof. The proof is based on the fact that, for any $\succ$ and $nom^{\succ}$, given a model for φ, we can build another model for φ but such that certain criteria of compatibility with $nom^{\succ}$ also holds. In essence, for M to be "compatible" with $nom^{\succ}$ the following conditions should hold:

- if $M \models @_i\langle r \rangle \psi$ and $j = nom^{\succ}(@_i\langle r \rangle \psi)$, then $M \models @_i\langle r \rangle j$, and $M \models @_j \psi$,
- if $k = nom^{\succ}(@_i\langle r \rangle \psi)$, $l = nom^{\succ}(@_j\langle r \rangle \psi)$ and $M \models @_i j$, then $M \models @_k l$.

In order to prove completeness, one should note that, by Definition 13, the consequents of the (PAR-@$\diamond$) rule are always smaller than the main premise. Using this fact, the proof of Theorem 2 is easily adapted (handling the $(\langle r \rangle')$ rule is straightforward).

Theorem 5. *$\boldsymbol{R}_T^{\mathrm{OS}}[\mathcal{H}^{NNF}(@)]$ is refutationally complete.*

We finally turn to the problem of proving that $\mathbf{R}_T^{\mathrm{OS}}[\mathcal{H}^{\mathrm{NNF}}(@)]$ doesn't generate infinite saturated sets. We are ready to exactly formulate conditions under which the problems of Type 1 and 2 we discussed above cannot occur. Let $\mathsf{NOM}(\Gamma)$ be the set of all nominals occurring in Γ, we want to establish that for all $\varphi \in \mathcal{H}_{@\diamond}^{\mathrm{NNF}}(@)$:

1. every set $\mathsf{NOM}_c^i \cap \mathsf{NOM}(ClSet^*(\varphi))$ is finite, for all $i \in \mathbb{N}$, and
2. the set $\{i \mid i \in \mathbb{N}$ and $\mathsf{NOM}_c^i \cap \mathsf{NOM}(ClSet^*(\varphi)) \neq \{\}\}$ is finite.

In the next proposition we show that these two conditions can be guaranteed.

Proposition 4. *Let $\varphi \in \mathcal{H}_i^{NNF}(@)$, then the set of nominals in $ClSet^*(\varphi)$ computed using the rules in $\boldsymbol{R}_T^{\mathrm{OS}}[\mathcal{H}^{NNF}(@)]$ is finite.*

Proof. For any $i \in \mathsf{NOM}$, the function $level(i)$ is defined as 0 if $i \in \mathsf{NOM}_i$ or the only $n \geq 1$ such that $i \in \mathsf{NOM}_c^{n-1}$ otherwise. Now, given $@_i\varphi \in \mathcal{H}_i^{\mathrm{NNF}}(@)$ define $d'(@_i\varphi) = level(i) + d(\varphi)$ where $d(\varphi)$ is the modal depth of φ. The proof of the proposition directly follows from these properties:

1. For every clause $\{@_i\langle r\rangle j\} \cup C \in \mathit{ClSet}^*(\varphi)$, either $\mathit{level}(j) = 0$ or $\mathit{level}(j) = \mathit{level}(i) + 1$.
2. For all formula ψ occurring in $\mathit{ClSet}^*(\varphi)$, if some $i \in \mathsf{NOM}_c$ occurs in ψ, then ψ is of the form: $@_i\psi'$, $@_j\langle r\rangle i$, or $@_j i$ (i does not occur in ψ' and $i \neq j$).
3. If ψ occurs in some clause in $\mathit{ClSet}^*(\varphi)$, then $d'(\psi) \leq d(\varphi)$ and, moreover, if $\psi = @_i j$, then $\mathit{level}(j) \leq d(\varphi)$. Therefore, for all formula ψ, if i is a nominal occurring in a formula in $\mathit{ClSet}^*(\psi)$, then $\mathit{level}(i) \leq d(\psi)$.
4. For all nominal i, the number of distinct formulas of the form $@_i\langle r\rangle\psi$ ($\psi \notin \mathsf{NOM}$) occurring in $\mathit{ClSet}^*(\varphi)$ is finite.
5. For all $k \in \mathbb{N}$, the set $\mathsf{NOM}_c^k \cap \mathsf{NOM}(\mathit{ClSet}^*(\varphi))$ is finite.

Termination of $\mathbf{R}_T^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ is a direct corollary of the above proposition.

Theorem 6. *$\mathbf{R}_T^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ is a decision procedure for the problem of satisfiability of $\mathcal{H}(@)$.*

5 Implementation and testing

HyLoRes 1.0 is an automated theorem prover for the logic $\mathcal{H}(@, \downarrow)$[4] (but we will only use it for formulas of $\mathcal{H}(@)$) written in Haskell, of approximately 5000 lines of code, based on the resolution calculus proposed in [5]. It must be noted that this is not a tool aiming to compete with state-of-the-art theorem provers. Automated provers such as SPASS [1], Vampire [22], RACER [16] or *SAT [15] include an important number of heuristics and optimizations with which they achieve an outstanding performance. HyLoRes implements a relatively small set of optimizations and it is still mainly a proof of concept implementation.

We have developed a new version (2.0) of HyLoRes that uses the rules of the $\mathbf{R}_T^{\text{OS}}[\mathcal{H}^{\text{NNF}}(@)]$ calculus presented in Figure 3. Several tests were run to compare the performance of versions 1.0 and 2.0. In this section we comment on some of the results obtained.

Nowadays, the standard test suite for basic modal logic satisfiability is the "random 3CNF$\Box_m$" [21], an adaptation of the random 3CNF for propositional logic [19]. This type of test generates batches of random formulas subject to certain restriction parameters (e.g., number of propositional variables, modal depth, maximum number of clauses, etc.).

The standard definition of random 3CNF$\Box_m$ generates formulas that are strictly modal (i.e., without neither nominals nor the @ operator). An extension, called random h3CNF$\Box_m$ and implemented as the generator hGen, is described in [7] that suits the needs of theorem provers for hybrid logics. hGen generates formulas for sublanguages of $\mathcal{H}(@, A, \downarrow)$ ($\mathcal{H}(@)$ extended with the $\downarrow$ binder and the universal modality A).

The parameters involved in the generation of test batches were: number of propositional variables (V), number of nominals (N), maximum modal depth (D) and number of clauses (L). After fixing the values for V, N and D, a batch

[4] $\mathcal{H}(@, \downarrow)$ is $\mathcal{H}(@)$ extended with the $\downarrow$ binder, see [4] for details.

of 100 formulas of $\mathcal{H}(@)$ was generated for each value of L in a given range. They were then used as input for both theorem provers, with a timeout value of 40 seconds per formula. To plot the results, the median of the execution time and of the number of clauses generated were taken.

First Test. We compared the performance of both versions of the prover using simple formulas ($V = 2$, $N = 3$, $D = 1$). Figure 4 shows four graphs: the satisfiability/unsatisfiability curves together with percentage of timeouts in the first line, and the comparison of space and time resources used in the second. In all cases, the x-axis represent number of clauses produced by the random generator (notice that, the bigger the number of clauses generated the bigger the probability of the clause set to be unsatisfiable). In the first line, the y-axis shows the percentage of cases of satisfiability, unsatisfiability, and timeouts. In the second line, the y-axis is a logscale and shows median of the number of clauses generated in the left graph, and median of the execution time (in seconds) in the right graph.

The performance of HyLoRes 2.0 was clearly better than that of its predecessor. Figure 4 shows that HyLoRes 1.0 couldn't solve an important fraction of the simpler problems, while HyLoRes 2.0 solved them all. It is interesting to observe that HyLoRes 1.0 had the larger number of timeouts in the region where most of the formulas are satisfiable, while HyLoRes 2.0 is benefiting here from the restrictions on ordering and selection functions which accelerate saturation.

It is noticeable that in this test the initialization time of HyLoRes 2.0 is higher than the time needed to solve the problem itself.

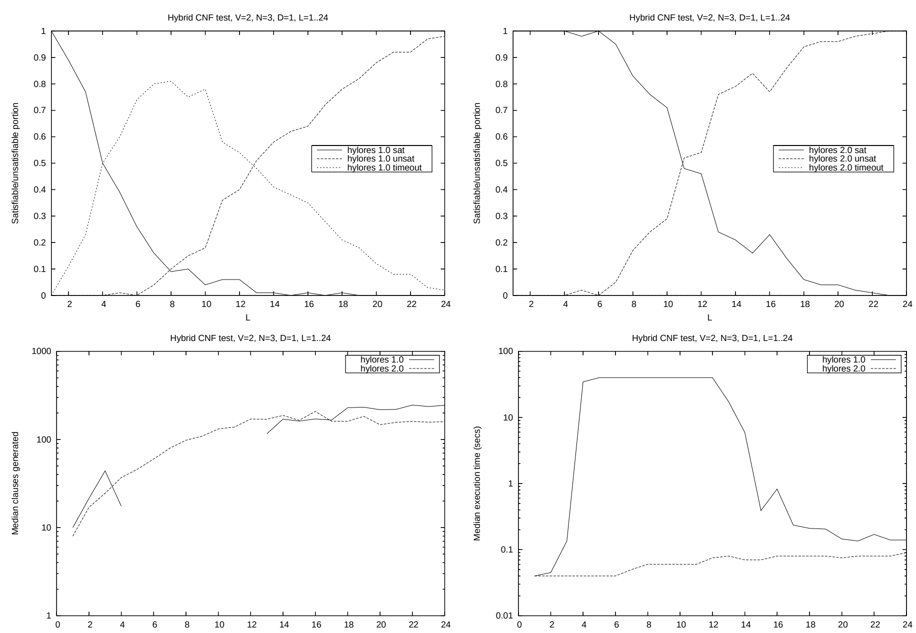

Fig. 4. h3CNF$\Box_m$ and HyLoRes 1.0 and 2.0 – Simple formulas.

Second Test. As one can clearly see in Figure 4, a large number of timeouts negatively affects the representativeness of the plot (see how the satisfiability/unsatisfiability percentage curves differ between the two graphs in the first line of the figure). When the modal depth of the formulas is augmented, the number of cases that HyLoRes 1.0 can solve in a reasonable time becomes too small to be relevant. Hence the more difficult tests were run only over different configurations of HyLoRes 2.0.

Figure 5 shows the results for formulas where only the strictly modal complexity was increased: $V = 8$, $N = 3$ and $D = 7$. We only show now the distribution of satisfiability, unsatisfiability and timeouts in the left graph and the cpu usage on the right graph. In this case, the number of timeouts in the harder zone is below 15%, however, the mean answer time is still below one second.

This test suggests again that the strategies of order and selection function are effective (see how times in the satisfiability section are better than the ones in the unsatisfiable region).

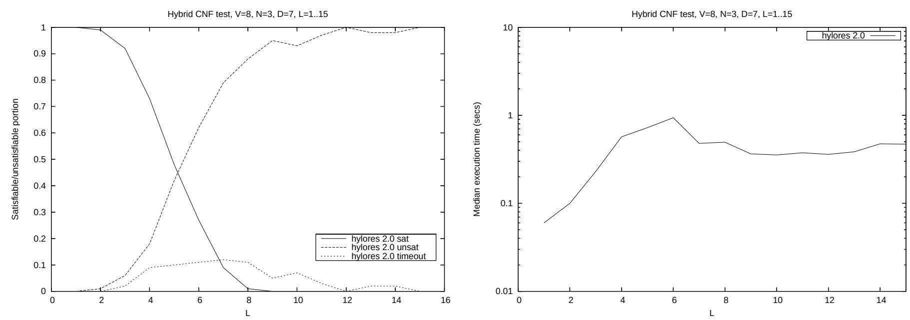

Fig. 5. h3CNF$\Box_m$ and HyLoRes 2.0 – Complex formulas, small number of nominals.

Third Test. Finally, Figure 6 shows the results obtained with formulas with an increased number of nominals and a low modal depth: $V = 6$, $N = 7$, $D = 2$. A larger number of nominals means a more frequent application of paramodulation rules. This is why HyLoRes 2.0 has a larger number of timeouts here, while the median execution time in the harder zone is over 10 seconds.

This test indicate that heuristics to control paramodulation (for example those described in [9]) should be implemented in HyLoRes, as the naive paramodulation used at the moment is too expensive. It is important to observe, though, that the formulas used in the test shown in Figure 6 have twice the modal depth, three times the number of propositional variables and more than twice the number of nominals than those of Figure 4, which HyLoRes 1.0 could barely handle.

6 Conclusions and Future Work

We presented in this paper a sound, complete and terminating strategy of resolution with order and selection functions for the hybrid language $\mathcal{H}(@)$. The paper

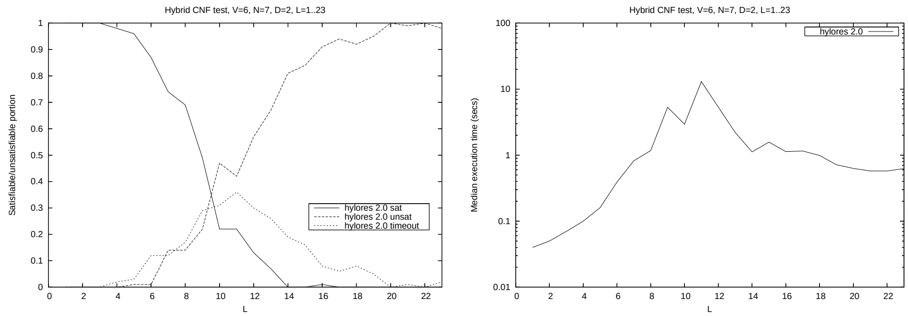

Fig. 6. h3CNF$\Box_m$ and HyLoRes 2.0 – Medium formulas, larger number of nominals.

shows in addition that standard resolution techniques and notions (e.g., the candidate model construction, the notion of admissible orderings, the definition of Herbrand models, etc.) which are crucial part of the actual work on resolution for classical logics can be adapted to the framework of modal logics when the hybrid operators (nominals and @) are present. Moreover, the strategy has been implemented in the HyLoRes prover and the preliminary tests show significant improvements.

We have not yet investigated the complexity of our resolution strategy. We conjecture that it is ExpTime-hard (and hence not optimal). Further refinements of the ordering and selection functions used, possibly together with the implementation of stronger resolution strategies (e.g, hyper resolution) might reduce the complexity to the optimal bound of PSpace (see, [14]), but these are topics for future research. More generally, further work on how to choose suitable parameters (which orderings and selection functions are most effective for a certain input) and the implementation of optimizations, heuristics and simplifications by rewriting remain to be done to enhance the usability of HyLoRes.

Suitable generalizations of the standard notions of redundancy [10] (e.g., backwards and forwards subsumption) should also be developed in detail. HyLoRes already implements of some basic ideas, but both theory and practice as it applies to resolution for modal-like languages, should be further developed.

We are also interested in investigated fragments of $\mathcal{H}(@)$ for which resolution might have a specially good behavior (e.g., find a suitable notion of Horn formulas) on the one hand, and on the other in developing extensions the actual framework to languages more expressive than $\mathcal{H}(@)$ (e.g., considering the addition of the $\downarrow$ binder and the universal modality A).

References

1. B. Afshordel, U. Brahm, C. Cohrs, T. Engel, E. Keen, C. Theobald, D. Topić, and C. Weidenbach. System description: SPASS Version 1.0.0. In *Automated deduction—CADE-16 (Trento, 1999)*, pages 187–201, Berlin, 1999. Springer.
2. C. Areces. *Logic Engineering. The Case of Description and Hybrid Logics.* PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands, October 2000.

3. C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In *Proc. of the 8th Annual Conference of the EACSL*, pages 307–321, 1999.
4. C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.
5. C. Areces, H. de Nivelle, and M. de Rijke. Resolution in modal, description and hybrid logic. *Journal of Logic and Computation*, 11(5):717–736, 2001.
6. C. Areces and J. Heguiabehere. HyLoRes: A hybrid logic prover based on direct resolution. In *Advances in Modal Logic*, Toulouse, France, 2002.
7. C. Areces and J. Heguiabehere. hGen: A random CNF formula generator for Hybrid Languages. In *Proc. of Methods for Modalities 3*, Nancy, France, 2003.
8. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
9. L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In *Automated deduction—a basis for applications, Vol. I*, pages 353–397. Kluwer, Dordrecht, 1998.
10. L. Bachmair and H. Ganzinger. Resolution theorem proving. In J. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 2, pages 19–99. Elsevier, 2001.
11. P. Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL*, 8(3):339–365, 2000.
12. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2002.
13. N. Dershowitz and J. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science. Volume B: Formal Models and Sematics (B)*, pages 243–320. Elsevier and MIT Press, 1990.
14. L. Georgieva, U. Hustadt, and R. Schmidt. Computational space efficiency and minimal model generation for guarded formulae. In R. Nieuwenhuis and A. Voronkov, editors, *Proc. of the 8th LPAR 2001*, number 2250 in LNAI, pages 85–99, 2001.
15. E. Giunchiglia, A. Tacchella, and F. Giunchiglia. SAT-based decision procedures for classical modal logics. *Journal of Automated Reasoning*, 28(2):143–171, 2002.
16. V. Haarslev and R. Möller. RACER system description. In *IJCAR 2001*, number 2083 in LNAI, Siena, 2001.
17. 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.
18. HyLoRes' Home Page. `http://www.loria.fr/~areces/HyLoRes/`, 2004.
19. D. Mitchell, B. Sleman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proc. of the 10th National Conference on Artificial Intelligence*, pages 459–465, 1992.
20. S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.
21. P. Patel-Schneider and R. Sebastiani. A new general method to generate random modal formulae for testing decision procedures. *Journal of Artificial Intelligence Research*, 18:351–389, May 2003.
22. A. Riazanov and A. Voronkov. The design and implementation of VAMPIRE. *AI Communications*, 15(2):91–110, 2002. Special issue on CASC.
23. A. Voronkov. Algorithms, datastructures, and other issues in efficient automated deduction. In *IJCAR 2001*, number 2083 in LNAI, pages 13–28, Siena, 2001.