Resolution in Modal, Description and Hybrid Logic

Carlos Areces[1] Hans de Nivelle[2]
Maarten de Rijke[1]

[1]ILLC, University of Amsterdam
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
Email: {carlos,mdr}@science.uva.nl
[2]Max Planck Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
Email: nivelle@mpi-sb.mpg.de

Abstract

We provide a resolution-based proof procedure for modal, description and hybrid logic that improves on previous proposals in important ways. It avoids translations into large undecidable logics, and works directly on modal, description or hybrid logic formulas instead. In addition, by using the hybrid machinery it avoids the complexities of earlier propositional resolution-based methods for modal logic. It combines ideas from the method of prefixes used in tableaux, and resolution ideas in such a way that some of the heuristics and optimizations devised in either field are applicable.

Keywords: direct resolution, modal logic, description logic, hybrid logic.

1 Introduction

Resolution, originally introduced for first-order logic (FO) in [36], is the most widely used reasoning method for first-order logic today: most of the available automatic theorem provers for FO are resolution based. The propositional core of the method is well-known: to check whether a propositional formula ϕ is not satisfiable, start by turning it into *clausal form*. To this end, we write ϕ in conjunctive normal form

$$\phi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)},$$

for $\psi_{(l,m)}$ a literal, and let the clause set associated with ϕ be

$$\mathit{ClSet}(\phi) = \{\{\psi_{(l,m)} \mid m \in M\} \mid l \in L\}.$$

Next, we define $\mathit{ClSet}^*(\phi)$ as the smallest set containing $\mathit{ClSet}(\phi)$ and closed under a unique, very easy to grasp rule:

$$\text{(RES)} \quad \frac{\mathit{Cl}_1 \cup \{N\} \in \mathit{ClSet}^*(\phi) \quad \mathit{Cl}_2 \cup \{\neg N\} \in \mathit{ClSet}^*(\phi)}{\mathit{Cl}_1 \cup \mathit{Cl}_2 \in \mathit{ClSet}^*(\phi)}$$

If $\{\} \in \mathit{ClSet}^*(\phi)$, then ϕ is not satisfiable. The intuition behind (RES) is as follows: given that either N or $\neg N$ is always the case in any model, they can be "cut away" if the sets of clauses are conjoined. The aim of the whole method is to "cut away everything" and arrive at the empty set.

The elegance of the resolution method for propositional logic relies mostly on its bare simplicity. The method can also be straightforwardly implemented, it seems tailored for a dumb machine able to crunch symbols quickly. The only computational cost is a search for complementary atoms in the set of clauses.

Of course, the picture for the first-order case is different (to start with, first-order resolution has to address an undecidable problem!). And actual implementations of first-order resolution systems are not "dumb" at all. In particular, during first-order resolution we have to cope with the rich structure of terms, and the unification algorithm (introduced by Robinson in [36], see [33] for a linear time version) plays a fundamental role in handling this complexity, and in using it to guide the search. The field of resolution based first-order theorem proving has developed into a community of its own, with an impressive collection of methods and optimizations [9, 34].

In contrast to the popularity of resolution-based methods in first-order logic, modern *modal* theorem provers are generally based on tableau methods [15]. Nowadays, resolution and modal languages seem to be related only when *indirect* methods are used. In translation-based resolution calculi for modal logics, one translates modal languages into a large background language (typically first-order logic), and devises strategies that guarantee termination for the fragment corresponding to the original modal language [22, 29, 17, 5]. First-order resolution provers like BLIKSEM [12] or SPASS [38] handle modal formulas in this way, in some cases using extremely optimized translations like those investigated in [32, 37]. This approach has both advantages and disadvantages with respect to the tableau approach. On the one hand we can translate many systems into the same background language and hence explore different, and also combined, systems without the need to modify the prover. But empirical tests show that the price to pay is high [28, 5]. The undecidability of the full background language shows up in degraded performance on the modal fragments, and first-order provers can hardly emulate their tableau based competitors.

Given the simplicity of propositional resolution, it is natural to wonder why *direct* resolution methods for modal languages don't figure in the picture. Designing resolution methods that can directly (without translation into large background languages) be applied to modal logics, received some attention in the late 1980s and early 1990s [20, 30, 16]. Also, the first (non-clausal) resolution methods for temporal languages go back to that period with the work of Abadi and Manna [1]. Recently, new results on clausal temporal resolution have been presented (see [18]). But even though we might sometimes think of modal languages as a "simple extension of propositional logic," direct resolution for modal languages has proved a difficult task. Intuitively, in basic modal languages the resolution rule has to operate *inside* boxes and diamonds to achieve completeness. This leads to more complex systems, less elegant results, and poorer performance, ruining the one-dumb-rule spirit of resolution.

In this paper we will show how ideas from hybrid logics can be put to work with benefit even when the subject is purely modal. In particular, aided by the notions of nominals and labeling, we will show how to define simple direct resolution methods for modal languages. This "case study" is an example of how the additional flexibility provided by the ability to name states can be used to improve reasoning methods. In addition, we can build over the basic resolution system and obtain extensions for hybrid and also description languages.

The main characteristics of the resolution method we will introduce can be summarized as follows:

- by using labeled formulas it avoids the complexity of earlier direct resolution-based methods for modal logic;
- it does not involve skolemization beyond the use of constants;
- it does not involve translation into large undecidable languages, working directly on modal, hybrid or description logic formulas instead;
- it is flexible and conservative in more than one sense: it incorporates the method of prefixes used in tableaux [23] into resolution in such a way that different heuristics and optimizations devised in either field are applicable.

The structure of the paper is as follows. In Section 2 we discuss, in some detail, the problems with direct modal resolution. We do this by discussing the system presented by Enjalbert and Fariñas del Cerro in [20]. In Section 3 we introduce *labeled resolution* for the basic multi-modal logic $\mathbf{K}_m$. Informally, the calculus uses the hybrid @ operator to "push formulas out of modalities" and in this way, feed them into a simple (RES) rule. Whereas in Section 3, we use the hybrid machinery only at the meta-logical level, the next step naturally leads us to internalized systems. We start in Section 4, by providing a resolution based method for deciding knowledge base inconsistency (with simple, acyclic T-Boxes and non-empty A-Boxes) for the description logic $\mathcal{ALCR}$; as discussed in [2], $\mathcal{ALCR}$ can be viewed as a restricted hybrid language. In Section 5 we finally move into the complete basic hybrid language, fully internalizing our use of @ and explaining how to handle nominals. We need to incorporate a form of equality reasoning into the resolution calculus, and discuss paramodulation and other techniques. In the last part of this section we show how to treat very expressive hybrid languages, by considering the $\downarrow$ hybrid binder. In Section 6 we conclude, with comments on related work and some directions for further research.

2 Direct Resolution for Modal Languages

To understand how we can use hybrid logic ideas to improve direct modal resolution, we introduce the system presented by Enjalbert and Fariñas del Cerro in [20]. Enjalbert and Fariñas del Cerro use some non-standard definitions which we introduce below and to which we adhere only in the present section; we will revert to more standard notation in the rest of the paper.

A modal formula is in *disjunctive normal form* if it is a (possibly empty) disjunction of the form

$$\phi = \bigvee L_i \vee \bigvee \Box D_j \vee \bigvee \Diamond A_k,$$

where each L_i is a literal, each D_j is in disjunctive normal form, and each A_k is in conjunctive normal form. A modal formula is in *conjunctive normal form* if it is a conjunction $\phi = \bigwedge C_i$, where each C_i is in disjunctive normal form. A formula in disjunctive normal form is called a *clause*. The empty clause is denoted as $\perp$. The conjunction $C_1 \wedge \cdots \wedge C_n$ is identified with the set $(C_1, \ldots, C_n)$. For any modal formula an equivalent clause can be obtained, so that attention can be restricted to clauses.

The following examples of applications of the resolution rule "in modal contexts" are discussed in [20] to show the intricacies of modal resolution:

$$(a)\quad \frac{\Box(p \vee q) \qquad \Box\neg p}{\Box q} \qquad\qquad (b)\quad \frac{\Box(p \vee q) \qquad \Diamond\neg p}{\Diamond(\neg p, q)}.$$

Both inferences are sound, and can be viewed as generalizations of the (RES) rule. While (a) closely follows the (RES) pattern (we resolve on p inside $\Box$ and cut it out to obtain $\Box q$), (b) is more complex: we again resolve on p but simply eliminating p from $\Box(p \vee q)$ to obtain $\Box q$ is unsound. Instead, we can soundly infer $\Diamond q$ which would somehow follow the "cutting" pattern of resolution but this is too weak; the proper inference being both $\neg p$ and q possible at the same state of the model.

Moreover, an attempt to apply a similar rule to the clauses $\Diamond(p \vee q)$ and $\Diamond\neg p$ to derive $\Diamond(\neg p, q)$ does not preserve soundness. Also, inferences with only one premise seem to be needed, as for example in

$$(c)\quad \frac{\Diamond(\neg p, p \vee q)}{\Diamond(\neg p, p \vee q, q)},$$

where we resolve on p inside the *same* clause to infer $\Diamond(\neg p, q)$. Enjalbert and Fariñas del Cerro explain that, actually, the logically equivalent but more explicit formula $\Diamond(\neg p, p \vee q, q)$ needs to be retained for completeness of the resolution method.

A resolution system based on these intuitions has been introduced and proved complete for **K** in [20]. Specifically, define, by induction, two relations on clauses $\Sigma(\alpha, \beta) \to \gamma$ and $\Gamma(\alpha) \to \gamma$, as indicated in Figure 1, where α, β, κ, δ_1, δ_2 are clauses, and Ψ, Φ are sets of clauses. The relations Σ and Γ will be used to define the notion

Axioms	
(A1) $\Sigma(p, \neg p) \to \perp$ (A2) $\Sigma(\perp, \alpha) \to \perp$	
Σ-Rules	**Γ-Rules**
$(\vee)\ \dfrac{\Sigma(\alpha, \beta) \to \kappa}{\Sigma(\alpha \vee \delta_1, \beta \vee \delta_2) \to \kappa \vee \delta_1 \vee \delta_2}$	$(\Diamond 1)\ \dfrac{\Sigma(\alpha, \beta) \to \kappa}{\Gamma(\Diamond(\alpha, \beta, \Phi)) \to \Diamond(\alpha, \beta, \kappa, \Phi)}$
$(\Box\Diamond)\ \dfrac{\Sigma(\alpha, \beta) \to \kappa}{\Sigma(\Box\alpha, \Diamond(\beta, \Psi)) \to \Diamond(\beta, \kappa, \Psi)}$	$(\Diamond 2)\ \dfrac{\Gamma(\alpha) \to \beta}{\Gamma(\Diamond(\alpha, \Phi)) \to \Diamond(\beta, \alpha, \Phi)}$
$(\Box\Box)\ \dfrac{\Sigma(\alpha, \beta) \to \kappa}{\Sigma(\Box\alpha, \Box\beta) \to \Box\kappa}$	$(\vee)\ \dfrac{\Gamma(\alpha) \to \beta}{\Gamma(\alpha \vee \kappa) \to \beta \vee \kappa}$
	$(\Box)\ \dfrac{\Gamma(\alpha) \to \beta}{\Gamma(\Box\alpha) \to \Box\beta}$

Figure 1: Enjalbert and Fariñas del Cerro resolution rules.

of resolvent, i.e., of a clause obtained via resolution from a previous set of clauses. The definition is rather involved, starting from the axioms stating the cut on opposing literals and the propagation of inconsistencies ((A1) and (A2)), to the inductive steps which specify how disjunctions should be handled (the pair of ($\wedge$) rules) and how one should deal with modal contexts (($\Box\Diamond$), ($\Box\Box$), ($\Diamond 1$), ($\Diamond 2$), ($\Box$)).

The full formal definition runs as follows. Start by defining the simplification relation $A \approx B$ (perhaps better understood as a rewriting system $\rightsquigarrow$) as the least congruence containing

$$\begin{array}{rcl|rcl} \Diamond\bot & \approx & \bot & \bot \vee D & \approx & D \\ (\bot, A) & \approx & \bot & A \vee A \vee D & \approx & A \vee D. \end{array}$$

For any formula F there is a unique F' such that $F \approx F'$ and F' cannot be simplified further. This formula F' is called the normal form of F. C is a *resolvent* of A and B (respectively A) iff there is some C' such that $\Sigma(A, B) \to C'$ (respectively, $\Gamma(A) \to C'$) and C is the normal form of C'. We write $\Sigma(A, B) \Rightarrow C$ (respectively, $\Gamma(A) \Rightarrow C$) if C is a resolvent of A and B (respectively, of A).

Given a set of clauses S, let $\mathit{ClSet}^*(S)$ be the smallest set containing S that is closed under resolvents of elements in $\mathit{ClSet}^*(S)$. D is said to be a *resolution consequence* of a set of clauses S (notation $S \vdash D$) iff $D \in \mathit{ClSet}^*(S)$.

Theorem 1 ([20]) For $S \cup \{D\}$ a finite set of clauses, $S \vdash D$ iff $\models_{\mathbf{K}} \bigwedge S \to D$.

So much for the "one dumb rule spirit" of resolution. Let us go through an example to better understand how this resolution method works. As is standard, we use the resolution system to check for unsatisfiability. If starting from a clause ϕ we are able to derive the empty clause $\bot$, then ϕ is unsatisfiable.

Example 2 Consider the formula $\Diamond(p \wedge (\neg p \vee \Box r \vee q)) \wedge \Box\neg q \wedge \Box\Diamond\neg r$. In the resolution proof below we underline the literals on which resolution takes place, and simplify some steps for succinctness.

1. $(\Diamond(\underline{p}, \underline{\neg p} \vee \Box r \vee q), \Box\neg q, \Box\Diamond\neg r)$ by (A1), ($\vee$) and ($\Diamond 1$)
2. $(\Diamond(p, \neg p \vee \Box r \vee q, \Box r \vee \underline{q}), \Box\underline{\neg q}, \Box\Diamond\neg r)$ by (A1), ($\vee$) and ($\Box\Diamond$)
3. $(\Diamond(p, \neg p \vee \Box r \vee q, \Box r \vee q, \Box\underline{r}), \Box\neg q, \Box\Diamond\underline{\neg r})$ by (A1) and ($\Box\Diamond$) twice
4. $(\Diamond(p, \neg p \vee \Box r \vee q, \Box r \vee q, \Box r, \Diamond(\neg r, \bot)), \Box\neg q, \Box\Diamond\neg r)$ by (A2) and ($\Diamond 1$)
5. $\bot$

Even following this simple proof is complex. For example, line 1 should be understood as follows. Given that $(p, \neg p) \Rightarrow \bot$ by (A1), we can infer by ($\vee$) that $(p, \neg p \vee \Box r \vee q) \Rightarrow (p, \neg p \vee \Box r \vee q, \Box r \vee q)$ (this already involves some simplifications). An application of ($\Diamond 1$) allows us to perform this inference under $\Box$.

As we have just seen, the direct resolution method for modal logics presented in [20] (and, similarly, those in [21, 30, 16]) performs resolution "inside" modalities, leading to a proliferation of deductive rules. In the next sections we develop a direct resolution method for modal, description and hybrid logic that retains as much of the lean one-rule character of traditional resolution methods as possible. The key idea, from a basic modal logic perspective, is to use labels to decorate formulas with additional information. Labels allow us to make information explicit and resolution can then always be performed at the "top level." From a hybrid logic perspective, we are just taking advantage of the new expressive power that nominals and @ provide.

3 Labeled Modal Resolution

We now introduce a direct resolution proof procedure for the basic multi-modal logic $\mathbf{K}_m$. We assume a fixed modal similarity type $\mathcal{S} = \langle \mathsf{REL}, \mathsf{PROP} \rangle$ of accessibility relation and propositional symbols, together with a basic hybrid logic similarity type $\mathcal{S}' = \langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM} \rangle$, where NOM is a countably infinite set of nominals (we use ATOM to denote $\mathsf{PROP} \cup \mathsf{NOM}$).

Definition 3 (Normal Form) We define the following rewriting procedure *nf* on modal formulas:

$$\begin{array}{rcl} \neg\neg\phi & \overset{nf}{\leadsto} & \phi, \\ \langle R \rangle \phi & \overset{nf}{\leadsto} & \neg([R]\neg\phi), \\ (\phi_1 \vee \phi_2) & \overset{nf}{\leadsto} & \neg(\neg\phi_1 \wedge \neg\phi_2). \end{array}$$

For any formula ϕ, the rewriting of subformulas of ϕ by means of $\overset{nf}{\leadsto}$ converges to a unique *normal form* $nf(\phi)$ which is logically equivalent to ϕ. If we take $\vee$ and $\langle R \rangle$ as defined operators, then $nf(\phi)$ is slightly more than an expansion of definitions (see [27]).

Definition 4 (Clauses) A *clause* is a set Cl such that each element of Cl is a formula of the form $t : \phi$ or $(t_1, t_2) : R$ for $t, t_1, t_2 \in \mathsf{NOM}$, $R \in \mathsf{REL}$ and ϕ a basic multi-modal formula. Let ϕ be a basic multi-modal formula. The set S_ϕ of clauses corresponding to ϕ is simply $\{\{a : nf(\phi)\}\}$, for a an arbitrary label in NOM.

Formulas in clauses can be seen as *labeled formulas* [23, 25]; $t_1 : \phi$ specifies that the formula ϕ holds at the label t_1, and $(t_1, t_2) : R$ requires the labels t_1 and t_2 to be related by the accessibility relation R. Equivalently, a set of clauses can be seen as a set of hybrid formulas, with $t : \phi$ standing for $@_t\phi$ and $(t_1, t_2) : R$ standing for $@_{t_1}\langle R \rangle t_2$, and we can use hybrid models to define satisfiability for clauses. We recall the definition of hybrid model and satisfaction (see [2] for details).

Definition 5 (Semantics) A (hybrid) *model* $\mathcal{M}$ is a triple $\mathcal{M} = \langle M, \{R_i\}, V \rangle$ such that M is a non-empty set, $\{R_i\}$ is a set of binary relations on M, and $V : \mathsf{PROP} \cup \mathsf{NOM} \to Pow(M)$ is such that for all nominals $i \in \mathsf{NOM}$, $V(i)$ is a singleton subset of M. Let $\mathcal{M} = \langle M, \{R_i\}, V \rangle$ be a model and $m \in M$. The relevant conditions for the *satisfiability relation* are defined as follows:

$$\begin{array}{rcl} \mathcal{M}, m \Vdash a & \text{iff} & m \in V(a),\ a \in \mathsf{ATOM} \\ \mathcal{M}, m \Vdash [R]\phi & \text{iff} & \forall m'.(R(m, m') \text{ then } \mathcal{M}, m' \Vdash \phi) \\ \mathcal{M}, m \Vdash @_i\phi & \text{iff} & \mathcal{M}, m' \Vdash \phi, \text{ where } V(i) = \{m'\},\ i \in \mathsf{NOM}. \end{array}$$

If $\mathcal{M}$ is understood from the context, we simply write $m \Vdash \phi$ for $\mathcal{M}, m \Vdash \phi$. We write $\mathcal{M} \Vdash \phi$ iff for all $m \in M$, $\mathcal{M}, m \Vdash \phi$.

Definition 6 (Satisfiability of Clauses) Let Cl be a clause, and let $\mathcal{M}$ be a hybrid model. We write $\mathcal{M} \models Cl$ if $\mathcal{M} \models \bigvee Cl$. A set of clauses S is *satisfiable* if there is a model $\mathcal{M}$ such that for all $Cl \in S$, $\mathcal{M} \models Cl$.

Let ϕ be a basic multi-modal formula and S_ϕ its corresponding set of clauses. Proving that ϕ is satisfiable iff S_ϕ is satisfiable is straightforward. For the left to right implication, given $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ and $m \in M$ such that $\mathcal{M}, m \Vdash \phi$, just extend V so that $V(a) = \{m\}$ and give any interpretation to others elements in NOM. For the other direction, drop the interpretation of elements in NOM.

We have now set up the machinery to provide the appropriate set of resolution rules. As a guide, it is useful to recall that modal formulas can be seen as first-order formulas by means of the standard translation ST.

Definition 7 (Standard Translation into FO) The mutually recursive functions ST_x and ST_y map basic modal formulas into FO:

$$\begin{array}{rl|rl}
ST_x(p_j) = & P_j(x),\ p_j \in \mathsf{PROP} & ST_y(p_j) = & P_j(y),\ p_j \in \mathsf{PROP} \\
ST_x(\neg\phi) = & \neg ST_x(\phi) & ST_y(\neg\phi) = & \neg ST_y(\phi) \\
ST_x(\phi \wedge \psi) = & ST_x(\phi) \wedge ST_x(\psi) & ST_y(\phi \wedge \psi) = & ST_y(\phi) \wedge ST_y(\psi) \\
ST_x([R]\phi) = & \forall y.(R(x,y) \to ST_y(\phi)) & ST_y([R]\phi) = & \forall x.(R(y,x) \to ST_x(\phi)).
\end{array}$$

Figure 2 provides a set of rules transforming sets of clauses into sets of clauses.

$$(\wedge)\ \frac{Cl \cup \{t{:}\phi_1 \wedge \phi_2\}}{\begin{array}{c} Cl \cup \{t{:}\phi_1\} \\ Cl \cup \{t{:}\phi_2\}\end{array}} \qquad (\neg\wedge)\ \frac{Cl \cup \{t{:}\neg(\phi_1 \wedge \phi_2)\}}{Cl \cup \{t{:}\mathit{nf}(\neg\phi_1), t{:}\mathit{nf}(\neg\phi_2)\}}$$

$$(\mathrm{RES})\ \frac{Cl_1 \cup \{t{:}\phi\} \quad Cl_2 \cup \{t{:}\neg\phi\}}{Cl_1 \cup Cl_2}$$

$$([R])\ \frac{Cl_1 \cup \{t_1{:}[R]\phi\} \quad Cl_2 \cup \{(t_1,t_2){:}R\}}{Cl_1 \cup Cl_2 \cup \{t_2{:}\phi\}}$$

$$(\neg[R])\ \frac{Cl \cup \{t{:}\neg[R]\phi\}}{\begin{array}{c} Cl \cup \{(t,n){:}R\} \\ Cl \cup \{n{:}\mathit{nf}(\neg\phi)\}\end{array}}, \text{ where } n \text{ is new.}$$

Figure 2: Labeled resolution rules.

If you read the rules with ST in the back of your mind, the meaning of $([R])$ and $(\neg[R])$ will be immediately clear. $([R])$ is needed to account for the "hidden" negation arising from the implication in the translation of $[R]$, and in that sense it is indeed a standard resolution rule which cuts away the two complementary binary literals $\neg R(t_1, x)$ and $R(t_1, t_2)$ and unifies x to t_2. On the other hand, $(\neg[R])$ can be seen as a mild kind of skolemization which only involves the introduction of constants. From this perspective, we can view the rules $(\wedge)$, $(\neg\wedge)$ and $(\neg[R])$ as preparing the input formula and feeding it into the resolution rules (RES) and $([R])$. In other words, the system interleaves the reduction towards a standard clausal form with the resolution steps as in [24]. An immediate advantage of this method is that resolution can be performed not only on literals, but on complex formulas.

Before moving on, let's redo Example 2 in the new system. As before, we underline the part of the formula where a rule applies.

Example 8 Consider again the formula $\phi = \Diamond(p \land (\neg p \lor \Box r \lor q)) \land \Box\neg q \land \Box\Diamond\neg r$, which in the new notation is written as $\langle R\rangle(p \land (\neg p \lor [R]r \lor q)) \land [R]\neg q \land [R]\langle R\rangle\neg r$. $nf(\phi)$ is the singleton $\{i : \neg[R]\neg(p \land \neg(p \land \neg[R]r \land \neg q)) \land [R]\neg q \land [R]\neg[R]\neg r\}$. In each line we only show the newly generated clauses and those which will still be required in the successive steps.

1. $\{i : \neg[R]\neg(p \land \neg(p \land \neg[R]r \land \neg q))\underline{\triangle}[R]\neg q\underline{\triangle}[R]\neg[R]\neg r\}$, by $((\land)$ twice)
2. $\{i : \underline{\neg[R]}\neg(p \land \neg(p \land \neg[R]r \land \neg q))\}, \{i : [R]\neg q\}, \ \{i : [R]\neg[R]r\}$, by $(\neg[R])$
3. $\{R(i,j)\}, \{j : (p\underline{\triangle}\neg(p \land \neg[R]r \land \neg q))\}, \{i : [R]\neg q\}, \ \{i : [R]\neg[R]r\}$, by $(\land)$
4. $\{R(i,j)\}, \{j : p\}, \{j : \underline{\neg}(p\underline{\triangle}\neg[R]r\underline{\triangle}\neg q)\}, \{i : [R]\neg q\}, \ \{i : [R]\neg[R]r\}$, by $(\neg\land)$
5. $\{R(i,j)\}, \{j : \underline{p}\}, \{j : \underline{\neg p}, j : [R]r, j : q\}, \{i : [R]\neg q\}, \ \{i : [R]\neg[R]r\}$, by (RES)
6. $\{R(i,j)\}, \{j : [R]r, j : q\}, \{i : \underline{[R]}\neg q\}, \ \{i : [R]\neg[R]r\}$, by $([R])$
7. $\{j : [R]r, j : \underline{q}\}, \{j : \underline{\neg q}\}, \ \{j : \neg[R]r\}$, by (RES)
8. $\{j : \underline{[R]r}\}, \ \{j : \underline{\neg[R]r}\}$, by (RES)
9. $\{\}$.

Definition 9 (Deduction) A *deduction* of a clause Cl from a set of clauses S is a finite sequence $S_1, \ldots, S_n$ of sets of clauses such that $S = S_1$, $Cl \in S_n$ and each S_i (for $i > 1$) is obtained from S_{i-1} by adding the consequent clauses of the application of one of the resolution rules in Figure 2 to clauses in S_{i-1}. Cl is a *consequence* of S if there is a deduction of Cl from S. A deduction of $\{\}$ from S is a *refutation* of S, and we say that S is *refutable.*

The set $ClSet^*(S)$, defined as the smallest set containing S and all its consequences, need not be finite because the rule $(\neg[R])$ can introduce infinitely many clauses which only differ on the label. By restricting $(\neg[R])$ to be "fired only once" as we will describe in the next section, we can ensure finiteness of $ClSet^*(S)$, and hence termination of the search for consequences.

It is straightforward to prove that the resolution rules in Figure 2 preserve satisfiability. That is, given a rule, if the premises are satisfiable, then so are the conclusions. In Section 4, we will extend the system to deal with knowledge bases in the description language $\mathcal{ALCR}$, and prove there, in detail, soundness, completeness and termination.

3.1 Modal Extensions

From a traditional modal logic point of view we often want to consider systems above $\mathbf{K}_m$. Here we choose systems **T**, **D**, and **4** as examples. Each system is axiomatically defined as an extension of the basic system **K** by the addition of an axiom scheme which characterizes certain property of the accessibility relation.

Name	Axiom Scheme	Accessibility Relation
T	$p \to \langle R\rangle p$	reflexivity: $\forall x.R(x,x)$
D	$[R]p \to \langle R\rangle p$	seriality: $\forall x\exists y.R(x,y)$
4	$\langle R\rangle\langle R\rangle p \to \langle R\rangle p$	transitivity: $\forall xyz.(R(x,y) \land R(y,z) \to R(x,z))$

Corresponding to each of the axioms we add a new resolution rule:

$$(\mathbf{T})\quad \frac{Cl \cup \{t\!:\![R]\phi\}}{Cl \cup \{t\!:\!\phi\}}$$

$$(\mathbf{D})\quad \frac{Cl \cup \{t\!:\![R]\phi\}}{Cl \cup \{t\!:\!\neg[R]nf(\neg\phi)\}}$$

$$(\mathbf{4})\quad \frac{Cl_1 \cup \{t_1\!:\![R]\phi\} \quad Cl_2 \cup \{(t_1,t_2)\!:\!R\}}{Cl_1 \cup Cl_2 \cup \{t_2 : [R]\phi\}}.$$

Soundness for these systems is immediate. For completeness and termination we should modify the constructions in Section 4 (in particular (**4**) needs a mechanism of cycle detection); this can be done using methods from [20].

4 Description Logic

We will now discuss resolution based decision methods for description logics (DL's), a family of specialized languages for the representation and structuring of knowledge, related to the KL-ONE system of Brachman and Schmolze [13]. The connections between DL's and hybrid logics are strong (see [2, 3]), as we will clearly see in what follows. We spell out the details of a labeled resolution system to decide inconsistency of knowledge bases with so-called simple, acyclic T-Boxes and non-empty A-Boxes in the description logic $\mathcal{ALCR}$.

We assume fixed a description logic signature $\mathcal{S} = \langle \mathsf{CON}, \mathsf{ROL}, \mathsf{IND}\rangle$ together with an additional countable set of labels LAB. CON is the set of atomic concepts, ROL the set of atomic roles and IND the set of individuals.

Definition 10 (Interpretation) An *interpretation* $\mathcal{I}$ for $\mathcal{S}$ is a tuple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set, and $\cdot^{\mathcal{I}}$ is a function assigning an element $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each individual a_i; a subset $C_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each atomic concept C_i; and a relation $R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each atomic role R_i.

In other words, a description logic interpretation is just a model for a particular kind of first-order signature, where only unary and binary predicate symbols are allowed and the set of function symbols is empty.

The atomic symbols in a description logic signature can be combined by means of *concept* and *role constructors*, to form complex concept and role expressions. Each description logic is characterized by the set of concept and roles constructors it allows. Figure 3 defines the roles and concepts constructors for the description logic $\mathcal{ALCR}$, together with their semantics.

In description logics we are interested in performing inferences given certain background knowledge.

Definition 11 (Knowledge Bases and Inference) A *knowledge base* is a pair $\Sigma = \langle T, A\rangle$ such that T is the T(erminological)-Box, a (possibly empty) set of *terminological axioms* of the form $C_1 \sqsubseteq C_2$ or $R_1 \sqsubseteq R_2$, and A is the A(ssertional)-Box, a (possibly empty) set of *assertions* of the form $a\!:\!C$ or $(a,b)\!:\!R$, where C, C_1, C_2 are complex concepts, R, R_1, R_2 are complex roles, and a, b are individuals.

For $\mathcal{I}$ an interpretation and ϕ a terminological axiom or an assertion, we define the relation $\mathcal{I} \models \phi$ as follows

Constructor	Syntax	Semantics
concept name	C	$C^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
universal quant.	$\forall R.C$	$\{d_1 \mid \forall d_2 \in \Delta^{\mathcal{I}}.(R^{\mathcal{I}}(d_1, d_2) \to d_2 \in C^{\mathcal{I}})\}$
existential quant.	$\exists R.C$	$\{d_1 \mid \exists d_2 \in \Delta^{\mathcal{I}}.(R^{\mathcal{I}}(d_1, d_2) \wedge d_2 \in C^{\mathcal{I}})\}$
role name	R	$R^{\mathcal{I}}$
role conjunction	$R_1 \sqcap R_2$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}}$

Figure 3: Operators of $\mathcal{ALCR}$.

$$\begin{array}{rcl} \mathcal{I} \models C_1 \sqsubseteq C_2 & \text{iff} & C_1^{\mathcal{I}} \subseteq {C_2}^{\mathcal{I}} \\ \mathcal{I} \models R_1 \sqsubseteq R_2 & \text{iff} & R_1^{\mathcal{I}} \subseteq {R_2}^{\mathcal{I}} \\ \mathcal{I} \models a : C & \text{iff} & a^{\mathcal{I}} \in C^{\mathcal{I}} \\ \mathcal{I} \models (a, b) : R & \text{iff} & (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}. \end{array}$$

Let $\Sigma = \langle T, A \rangle$ be a knowledge base and $\mathcal{I}$ an interpretation, then $\mathcal{I} \models \Sigma$ if for all $\phi \in T \cup A, \mathcal{I} \models \phi$. We say that Σ is *satisfiable* if there exists a model $\mathcal{I}$ such that $\mathcal{I} \models \Sigma$; it is *unsatisfiable* otherwise.

Let $\Sigma = \langle T, A \rangle$ be a knowledge base (with so-called simple, non-cyclic definitions, see [31]) in $\mathcal{ALCR}$. Σ can be transformed into an "unfolded" equivalent knowledge base $\Sigma' = \langle \emptyset, A' \rangle$ where all concept and role assertions are of the form $\bar{t} : \bigsqcap_i N_i$ [19]. Hence, from now on we will only consider knowledge bases with empty T-Boxes.

The definition of the normal form *nf* for DL assertions is a notational variation of Definition 3, obtained by exchanging $\vee$ by $\sqcup$, $\wedge$ by $\sqcap$, etc; and setting $nf(\bar{t} : N) = \bar{t} : nf(N)$. Again, for any assertion $\bar{t} : N$, *nf* always converges to a unique normal form.

Definition 12 (Clauses) A *clause* is a set Cl such that each element of Cl is either a concept assertion of the form $t : C$ or a role assertion of the form $(t_1, t_2) : R$, where $t, t_1, t_2 \in \mathsf{IND} \cup \mathsf{LAB}$. A formula in a clause is a *literal* if it is either a role assertion, a concept or negated concept assertion on an atomic concept, or a universal or negated universal concept assertion. The set S_Σ of clauses corresponding to a knowledge base $\Sigma = \langle \emptyset, A \rangle$ is the smallest set such that

- if $a : \bigsqcap_i C_i = nf(a : C)$ for $a : C \in A$ then $\{a : C_i\} \in S_\Sigma$,
- if $(a, b) : \bigsqcap_i R_i \in A$ then $\{(a, b) : R_i\} \in S_\Sigma$.

Formulas in a clause are simply assertions over an expanded set of individuals. Let Cl be a clause, and $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ a model over the expanded signature $\langle \mathsf{CON}, \mathsf{ROL}, \mathsf{IND} \cup \mathsf{LAB} \rangle$; we put $\mathcal{I} \models Cl$ if $\mathcal{I} \models \bigvee Cl$. A set of clauses S *has a model* if there is model $\mathcal{I}$ such that for all $Cl \in S$, $\mathcal{I} \models Cl$. Clearly, Σ is satisfiable iff S_Σ has a model.

Figure 4 shows the labeled resolution rules recast for the language $\mathcal{ALCR}$. The only differences with the rules in Figure 2 are that ($\sqcap$) handles both concept and role conjunction, and that labels are now part of the language as the calculus directly manipulates concept and role assertions. Before proving soundness, completeness and termination we present a simple example of resolution in our system.

$$
\begin{array}{llll}
(\sqcap) & \dfrac{Cl \cup \{\bar{t}\!:\!N_1 \sqcap N_2\}}{\begin{array}{c}Cl \cup \{\bar{t}\!:\!N_1\} \\ Cl \cup \{\bar{t}\!:\!N_2\}\end{array}} & (\neg\sqcap) & \dfrac{Cl \cup \{t\!:\!\neg(C_1 \sqcap C_2)\}}{Cl \cup \{t\!:\!\mathit{nf}(\neg C_1), t\!:\!\mathit{nf}(\neg C_2)\}}
\end{array}
$$

$$
(\mathrm{RES}) \qquad \frac{Cl_1 \cup \{\bar{t}\!:\!N\} \quad Cl_2 \cup \{\bar{t}\!:\!\neg N\}}{Cl_1 \cup Cl_2}
$$

$$
(\forall) \qquad \frac{Cl_1 \cup \{t_1\!:\!\forall R.C\} \quad Cl_2 \cup \{(t_1, t_2)\!:\!R\}}{Cl_1 \cup Cl_2 \cup \{t_2\!:\!C\}}
$$

$$
(\neg\forall) \qquad \frac{Cl \cup \{t\!:\!\neg\forall R.C\}}{\begin{array}{c}Cl \cup \{(t,n)\!:\!R\} \\ Cl \cup \{n\!:\!\mathit{nf}(\neg C)\}\end{array}}, \text{ where } n \text{ is new.}
$$

Figure 4: Labeled resolution rules for $\mathcal{ALCR}$.

Example 13 Consider the following description. Suppose that children of tall people are blond (1). Furthermore, all Tom's daughters are tall (2), but he has a non-blond grandchild (3). Can we infer that Tom has a son (4)?

(0) $\mathtt{FEMALE} \doteq \neg\mathtt{MALE}$
(1) $\mathtt{TALL} \sqsubseteq \forall\mathtt{Child}.\mathtt{BLOND}$
(2) $\mathtt{tom}\!:\!\forall\mathtt{Child}.(\neg\mathtt{FEMALE} \sqcup \mathtt{TALL})$
(3) $\mathtt{tom}\!:\!\exists\mathtt{Child}.\exists\mathtt{Child}.\neg\mathtt{BLOND}$

---

(4) $\mathtt{tom}\!:\!\exists\mathtt{Child}.\mathtt{MALE}$.

As is standard in DL, we use a new proposition letter REST-TALL to complete the partial definition in (1) to (1') $\mathtt{TALL} \doteq \forall\mathtt{Child}.\mathtt{BLOND} \sqcap \mathtt{REST\text{-}TALL}$ (any partial definition $A \sqsubseteq B$ can be *completed* to an equivalent full definition $A = B \sqcup C$ for C a new concept, see [31]) and we resolve with the negation of the formula we want to infer (as a knowledge base Σ entails ϕ iff $\Sigma \cup \{\phi\}$ is inconsistent). After unfolding definitions (0) and (1') in (2) and applying *nf* we obtain the following three clauses:

1. $\{\mathtt{tom}\!:\!\forall\mathtt{Child}.\neg(\neg\mathtt{MALE} \sqcap \neg((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL}))\}$
2. $\{\mathtt{tom}\!:\!\neg\forall\mathtt{Child}.\forall\mathtt{Child}.\mathtt{BLOND}\}$
3. $\{\mathtt{tom}\!:\!\forall\mathtt{Child}.\neg\mathtt{MALE}\}$.

Now we start resolving:

4. $\{\mathtt{s}\!:\!\neg\forall\mathtt{Child}.\mathtt{BLOND}\}$	by $(\neg\forall)$ in 2
5. $\{(\mathtt{tom}, \mathtt{s})\!:\!\mathtt{Child}\}$	by $(\neg\forall)$ in 2
6. $\{\mathtt{s}\!:\!\neg\mathtt{MALE}\}$	by $(\forall)$ in 3
7. $\{\mathtt{s}\!:\!\neg(\neg\mathtt{MALE} \sqcap \neg((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL})\}$	by $(\forall)$ in 1
8. $\{\mathtt{s}\!:\!\mathtt{MALE}, \mathtt{s}\!:\!((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL})\}$	by $(\neg\sqcap)$ in 7
9. $\{\mathtt{s}\!:\!((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL})\}$	by (RES) in 6 and 8
10. $\{\mathtt{s}\!:\!\forall\mathtt{Child}.\mathtt{BLOND}\}$	by $(\sqcap)$ in 9
11. $\{\mathtt{s}\!:\!\mathtt{REST\text{-}TALL}\}$	by $(\sqcap)$ in 9
12. $\{\}$	by (RES) in 4 and 10.

Thus, indeed, Tom has a son.

4.1 Soundness, Completeness and Termination

We will now prove that the resolution calculus for deciding knowledge base inconsistency in $\mathcal{ALCR}$ is sound and complete, and that a procedure for ensuring termination exists. The proofs can easily be adapted to prove completeness and termination for the resolution calculus presented in Section 3.

Given a set of clauses S_Σ corresponding to a knowledge base Σ, the notion of a refutation for S_Σ is as in Definition 9.

Theorem 14 (Soundness) The rules described in Figure 4 are sound. That is, if Σ is a knowledge base, then S_Σ has a refutation only if Σ is inconsistent.

Proof We prove that labeled resolution rules preserve satisfiability. We only discuss $(\neg\forall)$. Let $\mathcal{I}$ be a model of the premise. If $\mathcal{I}$ is a model of Cl we are done. If $\mathcal{I}$ is a model of $t:\neg\forall R.C$, then there exists d in the domain, such that $(t^{\mathcal{I}}, d) \in R^{\mathcal{I}}$ and $d \in \neg C^{\mathcal{I}}$. Let $\mathcal{I}'$ be like $\mathcal{I}$ except perhaps in the interpretation of n, where $n^{\mathcal{I}'} = d$. As n is new, $\mathcal{I}' \models t:\neg\forall R.C$. But now $\mathcal{I}' \models Cl \cup \{(t,n):R\}$ and $\mathcal{I}' \models Cl \cup \{n: nf(\neg C)\}$.

Next, we prove completeness. We follow the approach used in [20]: given a set of clauses S we aim to define a structure T_S such that

($\dagger$) if S is satisfiable, a model can be effectively constructed from T_S; and

($\dagger\dagger$) if S is unsatisfiable, a refutation can be effectively constructed from T_S.

In our case we also have to deal with A-Box information, that is, with named objects (concept assertions) and fixed constraints on relations (role assertions). We will proceed in stages. To begin, we will obtain a first structure to account for named states and their fixed relation constraints. After that we can use a simple generalization of results in [20]. We base our construction on trees which will help in guiding the construction of the corresponding refutation proof.

Let Σ be a knowledge base and S_Σ its corresponding set of clauses. In S_Σ we can identify a (possibly empty) subset of clauses RA of the form $\{(a,b):R\}$, and for each label a a (possibly empty) subset CA_a of clauses of the form $\{a:C\}$. Notice that S_Σ has no *mixed* clauses containing both concept and role assertions (in a single disjunction). Also, there are no disjunctive concept assertions on different labels, i.e., there is no clause Cl in S_Σ such that $Cl = Cl' \cup \{a:C_1\} \cup \{b:C_2\}$ for $a \neq b$. We will take advantage of these properties in the first steps of the completeness proof.

For each label a appearing in Σ, construct inductively a binary tree T_a whose nodes will be sets of clauses. Let the original tree T_a consist of the single node CA_a and repeat in alternation the following operations:

Operation A1. Repeat the following steps as long as possible:

- choose a leaf w. Replace in w any clause of the form $\{a:\neg(C_1 \sqcap C_2)\}$ by $\{a: nf(\neg C_1),\ a: nf(\neg C_2)\}$; and any clause of the form $\{a: C_1 \sqcap C_2\}$ by $\{a:C_1\}$ and $\{a:C_2\}$.

Operation A2. Repeat the following steps as long as possible:

- choose a leaf w and a clause Cl in w of the form $Cl = \{a:C_1, a:C_2\} \cup Cl'$;
- add two children w_1 and w_2 to w, where $w_1 = w \backslash \{Cl\} \cup \{\{a:C_1\}\}$ and $w_2 = w \backslash \{Cl\} \cup \{\{a:C_2\} \cup Cl'\}$.

The leaves of T_a give us the possibilities for "named states" in our model. We can view each leaf as a set S_a^j, representing a possible configuration for state a.

Proposition 15 Operation A (the combination of A1 and A2) terminates, and upon termination

1. all the leaves $S_a^1, \ldots, S_a^n$ of the tree are singleton sets of literal clauses,
2. if all $S_a^1, \ldots, S_a^n$ are refutable, then CA_a is refutable,
3. if one S_a^j is satisfiable, then CA_a is satisfiable.

Proof Termination is trivial. Item 1 holds by virtue of the construction, and 2 is proved by induction on the depth of the tree. We need only realize that by simple propositional resolution, if the two children of a node w are refutable, then so is w. Item 3 is also easy. Informally, Operation A "splits" disjunctions and "carries along" conjunctions. Hence if some S_a^j has a model we have a model satisfying all conjuncts in CA_a and at least one of each disjunct.

We now consider the set RA of role assertions. Let NAMES be the set of labels which appear in Σ. If a is in NAMES but CA_a is empty in S_Σ, define $S_a^1 = \{\{a\!:\!C, a\!:\!\neg C\}\}$ for some concept C. We define the set of sets of nodes $\mathcal{N} = \{N_i \mid N_i$ contains exactly one leaf of each $T_a\}$; each N_i is a possible set of constraints for the named worlds in a model of S_Σ.

For all i, we will now extend each set in N_i with further constraints. For each $S_a \in N_i$, start with a node w_a labeled by S_a.

Operation B1. Equal to *Operation A1.*

Operation B2. Repeat the following steps as long as possible:

- choose nodes w_a, w_b such that $\{(a,b)\!:\!R\}$ in RA, $\{a\!:\!\forall R_i.C_i\} \in w_a$, $\{b\!:\!C_i\} \notin w_b$, where w_b is without children;
- add a child to w_b, $w_b' = w_b \cup \{\{b\!:\!C_i\}\}$.

Let N_i^* be the set of all leaves obtained from the forest constructed by applying Operation B (the combination of B1 and B2).

Proposition 16 Operation B terminates, and upon termination

1. all nodes created are derivable from $\bigcup N_i \cup RA$, and hence if a leaf is refutable so is $\bigcup N_i \cup RA$, and hence S_Σ too.
2. if some $\bigcup N_i^*$ is satisfiable, then S_Σ is satisfiable.

Proof To prove termination, notice that in each cycle the quantifier depth of the formulas considered decreases. Furthermore, it is not possible to apply twice the operation to nodes named by a and b and a formula $a\!:\!\forall R_i.C_i$.

As to item 1, each node is created by an application of the ($\forall$) rule to members of $N_i \cup RA$ or clauses previously derived by such applications. Hence either already the original $\bigcup N_i^*$ is refutable and we can build a refutation for S_Σ as indicated in Proposition 15, or $\{\}$ can be derived from a $\bigcup N_i$ by simple application of ($\forall$).

To prove item 2, let $\mathcal{I}$ be a model of N_i^*. Define $\mathcal{I}' = \langle \Delta', \cdot^{\mathcal{I}'} \rangle$ as $\Delta' = \Delta$, $a^{\mathcal{I}'} = a^{\mathcal{I}}$ for all labels a, $C^{\mathcal{I}'} = C^{\mathcal{I}}$ for all atomic concepts C, and $R^{\mathcal{I}'} = R^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \{(a,b):R\} \in RA\}$.

Observe that $\mathcal{I}'$ differs from $\mathcal{I}$ only in an extended interpretation of role symbols. By definition, $\mathcal{I}' \models RA$. It remains to prove that $\mathcal{I}' \models CA$. By Proposition 15, we are done if we prove that $\mathcal{I}' \models \bigcup N_i^*$. Since we only expanded the interpretation of relations, $\mathcal{I}$ and $\mathcal{I}'$ can only disagree on universal concepts of the form $a:\forall R.C$. By induction on the quantifier depth we prove this to be false.

Assume that $\mathcal{I}$ and $\mathcal{I}'$ agree on all formulas of quantifier depth less than n, and let $a:\forall R.C$ be of quantifier depth n, for $\{a:\forall R.C\} \in S_a^*$. Suppose $\mathcal{I}' \not\models \forall R.C$. This holds iff there exists b such that $(a^{\mathcal{I}'}, b^{\mathcal{I}'}) \in R^{\mathcal{I}'}$ and $\mathcal{I}' \not\models b:C$. By the inductive hypothesis, $\mathcal{I} \not\models b:C$. Now, if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ we are done. Otherwise, by definition $\{(a,b):R\} \in RA$. But then $\{b:C\} \in S_b^*$ by construction and as $\mathcal{I} \models S_b^*$, we also have $\mathcal{I} \models b:C$ — a contradiction.

Each N_i^* represents the "named core" of a model of S. The final step is to define the non-named part of the model. The following operations are performed to each set in each of the N_i^* obtaining in such a way a forest F_i.

Fix N_i^*, and a. We construct a tree "hanging" from the corresponding $S_a^* \in N_i^*$. The condition that each node of the tree is named by either an individual or a new label (that is, all the formulas in a node have the same label) will be preserved as an invariant during the construction. Set the original tree u to S_a^* and repeat the following operations C1, C2 and C3 in succession until the end-condition holds.

Operation C1. Equal to *Operation A1.*

Operation C2. Equal to *Operation A2.*

Operation C3. For each leaf w of u,

- if for some concept we have $\{C\}, \{\neg C\} \in w$, do nothing;
- otherwise, since w is a set of unit clauses, we can write $w = \{\{t:C_1\}, \ldots, \{t:C_m\}, \{t:\forall R_{k_1}.A_1\}, \ldots, \{t:\forall R_{k_n}.A_n\}, \{t:\neg\forall R_{l_1}.P_1\}, \ldots, \{t:\neg\forall R_{l_q}.P_q\}\}$. Form the sets $w_i = \{\{nf(t':\neg P_i)\}\} \cup S_i$, where t' is a new label, and $S_i = \{\{t':A_h\} \mid \{t:\forall R_i.A_h\} \in w\}$, and append each of them to w as children marking the edges as R_i links. The nodes w_i are called the *projections* of w.

End-condition. Operation C3 is inapplicable.

Proposition 17 Operation C cannot be applied indefinitely.

Definition 18 We call nodes to which Operation C1 or C2 has been applied of type 1, and those to which Operation C3 has been applied of type 2. The set of *closed nodes* is recursively defined as follows,

- if for some concept $\{t:C\}, \{t:\neg C\}$ are in w then w is closed,
- if w is of type 1 and all its children are closed, w is closed,
- if w is of type 2 and some of its children is closed, w is closed.

Let F_i be a forest that is obtained by applying Operations C1, C2, and C3 to N_i^* as often as possible. Then F_i is *closed* if any of its roots is closed.

Lemma 19 If one of the forests F_i is not closed, then S_Σ has a model.

Proof Let F_i be a non-closed forest. By a simple generalization of the results in [20, Lemma 2.7] we can obtain a model $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ of all roots S_a^* in F_i, from the trees "hanging" from them, i.e., a model of $\bigcup N_i^*$. By Proposition 16, S_Σ has a model.

Lemma 19 establishes the property (†) we wanted in our structure T_S. To establish (††) we need a further auxiliary result.

Proposition 20 Let w be a node of type 2. If one of its projections w_i is refutable, then so is w.

Proof Let w be a set of unit clauses $w = \{\{t : C_1\}, \ldots, \{t : C_m\}, \{t : \forall R_{k_1}.A_1\}, \ldots, \{t : \forall R_{k_n}.A_n\}, \{t : \neg\forall R_{l_1}.P_1\}, \ldots, \{t : \neg\forall R_{l_q}.P_q\}\}$. And let w_i be its refutable projection: $w_i = \{\{nf(t' : \neg P_i)\}\} \cup S_i$, where t' is a new label, and $S_i = \{\{t' : A_h\} \mid \{t : \forall R_i.A_h\} \in w\}$. We use resolution on w to arrive at the clauses in w_i from which the refutation can be carried out: Apply $(\neg\forall)$ to $\{t : \neg\forall R_i.P_i\}$ in w to obtain $\{t' : nf(t' : \neg P_i)\}$ and $\{(t, t') : R_i\}$. Now apply $(\forall)$ to all the clauses $\{t : \forall R_i.A_h\}$ in w to obtain $\{t' : A_h\}$.

Lemma 21 In a forest F_i, every closed node is refutable.

Proof For w a node in F_i, let $d(w)$ be the largest distance from w to a leaf.

If $d(w) = 0$, then w is a leaf, thus for some concept C, $\{t : C\}$ and $\{t : \neg C\}$ are in w. Using (RES) we immediately derive $\{\}$.

For the induction step, suppose the lemma holds for all w' such that $d(w') < n$ and that $d(w) = n$. If w is of type 1, let $w_1 = w \setminus \{Cl\} \cup \{Cl_1\}$ and $w_2 = w \setminus \{Cl\} \cup \{Cl_2\}$ be its children. By the inductive hypothesis there is a refutation for w_1 and w_2. By propositional resolution there is a refutation of w: repeat the refutation proof for w_2 but starting with w, instead of the empty clause we should obtain a derivation of Cl_2; now use the refutation of w_2. Suppose w is of type 2. Because w is closed, one of its projections is closed. Hence, by the inductive hypothesis it has a refutation. By Proposition 20, w itself has a refutation.

Theorem 22 (Completeness) The resolution method described above is complete: if Σ is a knowledge base, then S_Σ is refutable whenever Σ is inconsistent.

Proof We only need to put together the previous pieces. If Σ does not have a model then neither does S_Σ. By Lemma 19 all the forests F_i obtained from S_Σ are closed, and by Lemma 21, for each N_i^*, one of the sets $S_{a_j}^*$ is refutable. By Proposition 16, for all i, $\bigcup N_i \cup RA$ is refutable, and hence S_Σ is refutable.

Because we have shown how to *effectively* obtain a refutation from an unsatisfiable set of clauses we have also established termination. Notice that during the completeness proof we have used a *specific strategy* in the application of the resolution rules (crucially, the $(\neg\forall)$ rule is never applied twice to the same formula). By means of this strategy, we can guarantee termination of labeled modal resolution when verifying the consistency of any knowledge base in $\mathcal{ALCR}$.

Theorem 23 (Termination) Labeled resolution can effectively decide consistency of simple, acyclic knowledge bases in $\mathcal{ALCR}$.

We have spelled out in detail our resolution method for the basic description logic $\mathcal{ALCR}$, and we could naturally consider extensions. For instance, in [14] some attention has been given to n-ary roles (in modal logic terms, n-ary modal operators). Our approach generalizes to this case without further problems.

An attractive idea which matches nicely with the resolution approach is to incorporate a limited kind of unification on "universal labels" of the form $x : C$, to account for on the fly unfolding of definitions and more general T-Boxes. The use of such universal labels would make it unnecessary to perform a complete unfolding of the knowledge base as a pre-processing step. The leitmotif would be "to do expansion by definitions only when needed in deduction." On the fly unfolding has already been implemented in tableaux based systems like KRIS [7].

5 Hybrid Logic

It is the turn of hybrid languages now. Of course, we have already been dealing with hybrid languages throughout the previous section: formulas in the A-Box are actually a restricted form of @ formulas. To handle the full basic hybrid language $\mathcal{H}(@)$, we need to provide rules for nominals and @. As before, we can get a hint of what is needed from the standard translation of this language into FO. The new clauses of ST are as follows:

$$\begin{array}{rcl|rcl} ST_x(i) & = & (x = i),\ i \in \mathsf{NOM} & ST_y(i) & = & (y = i),\ i \in \mathsf{NOM} \\ ST_x(@_i\phi) & = & \exists x.(x = i \wedge ST_x(\phi)) & ST_y(@_i\phi) & = & \exists y.(y = i \wedge ST_y(\phi)). \end{array}$$

Nominals and @ introduce a limited form of equational reasoning on labels. Indeed, a formula like $@_i j$ is true in a model iff i and j label the same state. Moreover, in the tableau treatment of hybrid languages, nominals and the @ operator have been considered as the mechanisms needed to formulate modal theories of state equality and state succession [10].

In the resolution tradition, Robinson and Wos [35] introduced a rule called *paramodulation* to improve previous accounts of equational reasoning in FO:

$$(\text{PARAM}) \quad \frac{Cl_1 \cup \{t = s\} \qquad Cl_2 \cup \{\phi(u)\}}{(Cl_1 \cup Cl_2 \cup \{\phi(u/s)\})\sigma},$$

where σ is the most general unifier of t and u. In descriptions of paramodulation calculi one usually identifies the symmetric equations $s = t$ and $t = s$, includes (positive) factoring, and an inference rule that encodes resolution with the reflexivity axiom:

$$(\text{REF}) \quad \frac{Cl \cup \{t \neq s\}}{Cl\sigma},$$

where σ is the most general unifier of s and t. For a complete discussion of equational reasoning in first-order logics, we refer to [8]. We can introduce paramodulation in our direct resolution calculus for the basic hybrid language as follows:

$$(\text{PARAM}) \quad \frac{Cl_1 \cup \{@_t s\} \qquad Cl_2 \cup \{\phi(t)\}}{Cl_1 \cup Cl_2 \cup \{\phi(t/s)\}}$$

$$\text{(SYM)} \quad \frac{Cl \cup \{@_t s\}}{Cl \cup \{@_s t\}} \qquad \text{(REF)} \quad \frac{Cl \cup \{@_t \neg t\}}{Cl}$$

We need only one further rule to handle formulas of the form $@_s @_t \phi$ (see (@) below). Figure 5 shows the complete set.

$$(\wedge) \quad \frac{Cl \cup \{@_t(\phi_1 \wedge \phi_2)\}}{\begin{array}{c} Cl \cup \{@_t \phi_1\} \\ Cl \cup \{@_t \phi_2\} \end{array}} \qquad (\neg\wedge) \quad \frac{Cl \cup \{@_t \neg(\phi_1 \wedge \phi_2)\}}{Cl \cup \{@_t \mathit{nf}(\neg\phi_1), @_t \mathit{nf}(\neg\phi_2)\}}$$

$$\text{(RES)} \quad \frac{Cl_1 \cup \{@_t \phi\} \quad Cl_2 \cup \{@_t \neg\phi\}}{Cl_1 \cup Cl_2}$$

$$([R]) \quad \frac{Cl_1 \cup \{@_t [R]\phi\} \quad Cl_2 \cup \{@_t \neg[R]\neg s\}}{Cl_1 \cup Cl_2 \cup \{@_s \phi\}}$$

$$(\neg[R]) \quad \frac{Cl \cup \{@_t \neg[R]\phi\}}{\begin{array}{c} Cl \cup \{@_t \neg[R]\neg n\} \\ Cl \cup \{@_n \mathit{nf}(\neg\phi)\} \end{array}}, \text{ where } n \text{ is new.}$$

$$(@) \quad \frac{Cl \cup \{@_t @_s \phi\}}{Cl \cup \{@_s \phi\}}$$

$$\text{(PARAM)} \quad \frac{Cl_1 \cup \{@_t s\} \quad Cl_2 \cup \{\phi(t)\}}{Cl_1 \cup Cl_2 \cup \{\phi(t/s)\}}$$

$$\text{(SYM)} \quad \frac{Cl \cup \{@_t s\}}{Cl \cup \{@_s t\}} \qquad \text{(REF)} \quad \frac{Cl \cup \{@_t \neg t\}}{Cl}$$

Figure 5: Labeled resolution rules for $\mathcal{H}(@)$.

Two remarks on the rules above. Given that @ is self dual, we can define $\mathit{nf}(\neg @_t \phi) = @_t \mathit{nf}(\neg\phi)$; and for any formula ϕ in $\mathcal{H}(@)$, ϕ is satisfiable iff $@_t \phi$ is satisfiable, for a nominal t not appearing in ϕ. Hence, we can define the set S_ϕ of clauses corresponding to ϕ to be $\{\{@_t \mathit{nf}(\phi)\}\}$, where t does not appear in ϕ.

The soundness of the calculus is straightforward and the proof of Theorem 22 can be adapted in a way similar to the standard first-order case, to prove that paramodulation can handle @ and nominals (see [8]).

Theorem 24 The resolution calculus introduced in Figure 5 is sound and complete for $\mathcal{H}(@)$.

It is interesting to point out that optimizations and alternatives to paramodulation such as those discussed in [8], can now be investigated in the hybrid setting. In particular, we conjecture that heuristics can be devised to make the resolution calculus above terminating.

What about binders? Extending the system to account for hybrid sentences using $\downarrow$ is fairly straightforward. Consider the rule $(\downarrow)$ below (again $\downarrow$ is self dual, so we don't need a rule for its negation):

$$(\downarrow) \quad \frac{Cl \cup \{@_t{\downarrow}x.\phi\}}{Cl \cup \{@_t\phi(x/t)\}}.$$

Notice that the rule transforms hybrid sentences into hybrid sentences. The full set of rules gives us a complete calculus for sentences in $\mathcal{H}(@, \downarrow)$. Let's go through a short example.

Example 25 We prove that $\downarrow x.[R](x \wedge p) \rightarrow p$ is a tautology. Consider the negation of the formula in clausal form

1.	$\{@_i\underline{\downarrow x}.\neg[R]\neg(x \wedge p)\}, \{@_i\neg p\}$	by $(\downarrow)$
2.	$\{@_i\underline{\neg[R]}\neg(i \wedge p)\}, \{@_i\neg p\}$	by $(\neg[R])$
3.	$\{@_i\neg[R]\neg j\}, \{@_j(i\underline{\wedge}p)\}, \{i:\neg p\}$	by $(\wedge)$
4.	$\{@_j\underline{i}\}, \{@_{\underline{j}}p\}, \{@_i\neg p\}$	by (PARAM)
5.	$\{@_i\underline{p}\}, \{@_i\underline{\neg p}\}$	by (RES)
6.	$\{\}$.	

Of course, we cannot expect a heuristic ensuring termination of our calculus for $\mathcal{H}(@, \downarrow)$, as the satisfiability problem for $\mathcal{H}(\downarrow)$ is already undecidable (see [11]).

6 Conclusions

Blackburn [10] argues that hybrid languages can be used to internalize labeled deduction. Similar ideas play a fundamental role in the labeled resolution systems we introduced in this paper. Once again, nominals/labels together with the satisfiability operator : or @ are the key to achieving smooth and well behaved reasoning methods. The systems we introduced make clear that labeled resolution has many advantages in comparison with direct resolution proposals, thus supporting our claim that hybrid logic ideas can indeed be used to improve reasoning methods. We conclude the paper with a discussion of a number of independent directions for future research.

Once labels are introduced, the resolution method is very close to the tableaux approach, but we are still doing resolution. As we said, the rules $(\wedge)$, $(\neg\wedge)$ and $(\neg[R])$, prepare formulas to be fed into the resolution rules (RES) and $([R])$. And the aim is still to derive the empty clause instead of finding a model by exhausting a branch. But, is this method any better than tableaux? We don't think this is the correct question to ask. We believe that we learn different things from studying different methods. For example, Horrocks and Patel-Schneider [27] study a number of interesting optimizations of the tableaux implementation which were tested on the tableaux based theorem prover DLP. Some of their ideas can immediately be (or have already been) incorporated in our resolution method (lexical normalization and early detection of clashes, for instance), and others might be used in implementations of our method. At the same time, optimizations for direct resolution such as those discussed in [6] can also be exploited. For example, in implementations of the resolution algorithm, strategies for selecting the resolving pairs are critical. This kind of heuristics has been investigated by Auffray et al. [6] and some of their results easily extend to our framework. In certain cases, establishing completeness of these heuristics is even simpler because of our explicit use of resolution via nominals and @.

The issue of heuristics is very much connected to complexity. The basic heuristic used in the proof of Theorem 22 keeps the complete clause set "in memory" at all times and hence requires non-polynomial space. A similar situation occurs in clausal propositional resolution where the translation into clausal form can introduce an exponential blow up. We conjecture that a PSPACE heuristic for labeled resolution can be obtained by exploiting further the presence of labels (and given that we don't force a translation into full clausal form). Notice that nominals and @ let us keep track of the accessibility relation and we can define the notion of "being a member of a branch." Now we can attempt to use the tree property of modal languages to guide resolution. We used similar ideas in [5] to improve the performance of translation based resolution provers; see also [26] for translation based resolution methods which are able to polynomially simulate PSPACE tableaux.

The ideas behind labeled resolution are simple enough so that adapting available provers should not turn to be a very difficult task. It would be interesting to perform empirical testing on the performance of this resolution prover along the lines of, for example [28], both in comparison with translation based resolution provers and those based on tableaux.

Finally, our completeness proof is constructive: if a refutation cannot be found, we can actually define a model for the formula or knowledge base. Hence, our methods can also be used for model extraction. How does this method perform in comparison with traditional model extraction from tableaux systems?

Acknowledgments

We want to thank Maarten Marx and an anonymous referee for useful comments and corrections. Carlos Areces was supported by a grant from the Netherlands Organization for Scientific Research (NWO) under project number 612.069.006. Maarten de Rijke was supported by the Spinoza project 'Logic in Action' and by a grant from the Netherlands Organization for Scientific Research (NWO) under project number 365-20-005.

References

[1] M. Abadi and Z. Manna. Nonclausal temporal deduction. *Lecture Notes in Computer Science*, 193:1–15, 1985.

[2] C. Areces. *Logic Engineering. The Case of Description and Hybrid Logics.* PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, October 2000.

[3] C. Areces and M. de Rijke. From description to hybrid logics, and back. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 3.* CSLI Publications, 2001.

[4] C. Areces, E. Franconi, R. Goré, M. de Rijke, and H. Schlingloff, editors. *Methods for Modalities 1*, volume 8(3). *Logic Journal of the IGPL*, 2000.

[5] C. Areces, J. Heguiabehere R. Gennari, and M. de Rijke. Tree-based heuristics in modal theorem proving. In *Proceedings of ECAI'2000*, pages 199–203, Berlin, Germany, 2000.

[6] Y. Auffray, P. Enjalbert, and J. Hebrard. Strategies for modal resolution: results and problems. *Journal of Automated Reasoning*, 6(1):1–38, 1990.

[7] F. Baader, B. Hollunder, B. Nebel, H. Profitlich, and E. Franconi. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. *Journal of Applied Intelligence*, 4:109–132, 1994.

[8] L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In Bibel and Schmitt [9], chapter 11, pages 353–397.

[9] W. Bibel and P. Schmitt, editors. *Automated Deduction: A Basis for Applications*. Kluwer Academic Publisher, Dordrecht, 1998.

[10] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10(1):137–168, 2000.

[11] 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.

[12] Bliksem Version 1.12. URL: `http://www.mpi-sb.mpg.de/~bliksem/`. Accessed Apr. 8, 2001.

[13] R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[14] D. Calvanese, G. De Giacomo, and M. Lenzerini. Conjunctive query containment in description logics with n-ary relations. In *Proceedings of DL-97*, pages 5–9, June 1997.

[15] M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors. *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.

[16] H. de Nivelle. *Ordering Refinements of Resolution*. PhD thesis, Technische Universiteit Delft, Delft, The Netherlands, October 1994.

[17] H. de Nivelle, R. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. In Areces et al. [4], pages 239–264.

[18] C. Dixon, M. Fisher, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic*, 2(1), January 2001.

[19] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Deduction in concept languages: from subsumption to instance checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

[20] P. Enjalbert and L. Fariñas del Cerro. Modal resolution in clausal form. *Theoretical Computer Science*, 65(1):1–33, 1989.

[21] L. Fariñas del Cerro. A simple deduction method for modal logic. *Information Processing Letters*, 14:49–51, April 1982.

[22] C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution Methods for the Decision Problem*. Springer-Verlag, Berlin, 1993. Lecture Notes in Artificial Intelligence.

[23] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dordrecht, 1983.

[24] M. Fitting. Destructive modal resolution. *Journal of Logic and Computation*, 1(1):83–97, 1990.

[25] D. Gabbay. *Labelled Deductive Systems. Vol. 1.* The Clarendon Press Oxford University Press, New York, 1996. Oxford Science Publications.

[26] L. Georgieva, U. Hustadt, and R. A. Schmidt. Hyperresolution for guarded formulae. Submitted to *Journal of Symbolic Computation*, 2000.

[27] I. Horrocks and P. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.

[28] I. Horrocks, P. Patel-Schneider, and R. Sebastiani. An analysis of empirical testing for modal decision procedures. In Areces et al. [4], pages 293–324.

[29] U. Hustadt. *Resolution-Based Decision Procedures for Subclasses of First-Order Logic*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1999.

[30] G. Mints. Resolution calculi for modal logics. *amst*, 143:1–14, 1989.

[31] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43(2):235–249, 1990.

[32] H. Ohlbach. *A Resolution Calculus for Modal Logics*. PhD thesis, Universität Kaiserslautern, 1988.

[33] M. Paterson and M. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16(2):158–167, 1978.

[34] A. Robinson and A. Voronkov, editors. *Handbook of Automated Reasoning*. Elsevier Science Publishers B. V., 2001. To appear.

[35] G. Robinson and L. Wos. Paramodulation and theorem-proving in first-order theories with equality. In *Machine Intelligence, 4*, pages 135–150. American Elsevier, New York, 1969.

[36] J. Robinson. A machine-oriented logic based on the resolution principle. *jacm*, 12(1):23–41, January 1965.

[37] R. Schmidt. *Optimised Modal Translation and Resolution*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.

[38] SPASS Version 1.0.3. URL: `http://spass.mpi-sb.mpg.de/`. Accessed Apr. 8, 2001.