RESOLUTION MODAL LOGIC

Luis FARIÑAS-del-CERRO

Abstract

In this paper we describe a general way to define a resolution method in the framework of non classical logic.

1. *Introduction*

What does resolution mean in the case of Modal Logic? This question looks natural if we consider the fruitfulness of the resolution principle from both the theoretical and the practical point of view. This paper is aimed at supplying some answers to this question.

We bear in mind the following resolution principle: for any two clauses C_1 and C_2, if there is a literal L_1 in C_1 that is complementary to a literal L_2 in C_2, then delete L_1 and L_2 from C_1 and C_2 respectively, and construct the disjunction of the remaining clauses.

We now consider literals governed by the modal operators $[\,]$ (necessary) or $<>$ (possible), that are related by: $[\,]A =_{def} \sim <> \sim A$, (by A we note a formula). Consider the two clauses $[\,]p \vee C_1$ and $<> \sim p \vee C_2$ (where p is a propositional variable); we have that $<> \sim p$ is equivalent to $\sim [\,]p$. This suggests the following modal resolution rule:

$$\frac{[\,]p \vee C_1 \qquad <> \sim p \vee C_2}{C_1 \vee C_2}$$

The upper formulas of the rule will be called premises. In the same way we consider the clauses $[\,]p \vee C_1$ and $[\,](\sim p \vee q) \vee C_2$ and the axiom $[\,](A \vee B) \rightarrow <> A \vee [\,]B$ we can state a resolution rule such as:

$$\frac{[\,]p \vee C_1 \qquad [\,](\sim p \vee q) \vee C_2}{[\,]q \vee C_1 \vee C_2}$$

However if we consider the system with axioms $[\,](A \vee B) \rightarrow <> A \vee [\,] B$ and $[\,] A \rightarrow <> A$, and the clauses $<> p \vee C_1$ and $<> \sim p \vee C_2$. We can't have the rule:

$$\frac{<> p \vee C_1 \qquad <> \sim p \vee C_2}{C_1 \vee C_2}$$

As the relationship between $<> p$ and $[\,] p$ in this system is given by $[\,] p \rightarrow <> p$ and not the converse, the rule is not justified.

Everything suggests that the definition of resolution modal rules is linked to the relationship between the modal operators. We find the same idea in methods like the ones developed by Carnap [CR] and Lemmon [LE] and in the truth-table methods of Anderson [AR] and Bayart [BA]. However it appears only explicitly as a basis to define resolution methods in Shimura [SM], Orlowska [OE] and Fariñas [FC1].

This kind of decision method may be called syntactical, since semantics is not explicitly mentioned.

A first step for defining resolution decision methods is to define the notion of a normal form. It is necessary because the elements of the normal form are a set of expressions that is closed under resolution rules. In other words, if we consider a conjunctive normal form ($F = \overset{m}{\underset{i=1}{}} C_i$, where each C_i is a clause) and R as a rule with n arguments, then:

$$R(C_1, \ldots, C_n) = C$$

if R is defined for $C_1, \ldots, C_n$, and we note that C must be a clause. The formulation of R, as seen above, is subordinate to the characterization of elementary inconsistency i.e. inconsistencies between disjuncts of the clauses.

In what follows we will see precisely what this means. To this end we consider a particular modal system, the system Q, from which examples will be taken.

The paper will be organised as follows: in section 2, we describe the system Q (syntax and semantics), in section 3, we define a particular normal form for Q, which is the same for every normal modal logic with only monadic modal operators. In section 4, we deal with the problem of modal resolution; we will define what resolvent clause means in modal logic and then the resolution rule will be defined as in classical logic. The completeness theorem will be presented in section 5. In section 6, a refinement of the resolution rule will be given as well as the completeness theorem. Finally, in section 7 some applications of modal resolution will be sketched.

2. *The system Q*

The modal formulas of the system Q are expressions of the form (A & B), (A ∨ B), (A → B), ~A, [] A, or <> A where A and B are modal formulas. We introduce the constant symbol ⊥ to be read "the false". The system Q is a set of formulas obtained from the axiom schemas:

1. A → (B → A)
2. (A → (B → C)) → ((A → B) → (A → C))
3. ((A → ⊥) → ⊥) → A
4. [] (A → B) → ([] A → [] B)
5. [] A → <> A

and rules:

R1. Modus ponens $\dfrac{A,\ A \rightarrow B}{B}$

R2. Necessitation $\dfrac{A}{[\,]\,A}$

We define the notions of proof and theorem in the usual way. A proof of a formula A from a set S of formulas is a finite sequence of formulas each of which is either an axiom or an element of set S or a formula obtainable from earlier formulas by a rule of inference. A formula A is derivable from a set S(S ⊢ A) iff it has a proof from set S. A formula A is a theorem of Q(⊢ A) iff it is only derivable from the axioms. A set S of formulas is consistent if no formula of the form A & ~ A is derivable from S.

The meaning of formulas is defined using the notion of a model. For us a model is a triple:

M = <G, R, m>

where G is a non-empty set of states. R is a binary relation on G such that $\forall K \in G$, $\exists K' \in G$, $(K, K') \in R$, and m is a meaning function that assigns to each propositional variable p a subset m(p) of G.

Given a model M we say that a formula A is satisfied by a state K in model M(M, K sat A) iff the following conditions are satisfied:

M, K sat p iff $K \in m(p)$ where p is a propositional variable
M, K sat $\sim A$ iff not M, K sat A
M, K sat $A \vee B$ iff M, K sat A or M, K sat B
M, K sat A & B iff M, K sat A and M, K sat B
M, K sat $A \rightarrow B$ iff M, K sat $(\sim A \vee B)$
M, K sat $A \leftrightarrow B$ iff M, K sat $(A \rightarrow B)$ & $(B \rightarrow A)$
M, K sat [ ] A iff all $K' \in G$ if $(K, K') \in R$ then M, K′ sat A
M, K sat <> A iff there is a $K' \in G$ and such $(K, K') \in R$ and M, K′ sat A.

Given a model M, to each formula A of the language we assign a set of states called the extension of A in model M (ext_M A):

$$ext_M\ A = \{K \in G\colon M, K \text{ sat } A\}$$

We admit the usual notions of truth and validity of formulas. A formula A is true in a model M ($\models_M$ A) iff ext_M A = G. A formula A is valid in T($\models$A) iff A is true in every model for T. A formula A is a semantic consequence of a set S of formulas ($S \models A$) iff for any model M the formula A is true in M whenever all formulas in S are true in M. A formula A is satisfiable iff M, K sat A for some model M and state K. A set S of formulas is satisfied in a model M by a state K (M, K sat S) iff M, K sat A for all $A \in S$. A set S is satisfiable iff M, K sat S for some model M and state K.

3. *Conjunctive Normal Form (C.N.F.)*

Let F be a formula, we shall say that F is in C.N.F. if it is of the form:

$$F = C_1 \& \ldots \& C_m$$

where $m \geq 1$ and each C_i (clause) is a disjunction (perhaps with only one disjunct) of the general form:

$$C_i = L_i \vee \ldots \vee L_{n_1} \vee [\,]D_1 \vee \ldots \vee [\,]D_{n_2} \vee$$
$$<> A_1 \vee \ldots \vee <> A_{n_3}$$

where each L_i is a literal; each D_i is a disjunction that possesses the general form of the clauses, and each A_i is a conjunction, where each conjunct possesses the general form of the clauses. Each disjunct in C_i will be called element.

We note by E(E′) that E′ is a subformula of E.

Examples:

The following formulas are in conjunctive normal forms:

- [] (p ∨ q ∨ <> (r & t))
- <> ((p ∨ q) & t) & ~ p
- ~ p ∨ p ∨ [] (r ∨ s) ∨ <> ((p ∨ [] r) & e)
- ~ p ∨ [] ([] p ∨ (<> (q & [] r)) ∨ <> ([] (<> ([] q ∨ <> t) & r) ∨ [] t) & p)

The degree d(A) of a formula A is defined in the following way:

- if A is a literal, d(A) = 0
- if d(A) = n and d(B) = m, d(A △ B) = max(m,n) provided that △ is & or ∨.
- if d(A) = n, d(~ A) = n
- if d(A) = n, d(△ A) = n+1 where △ = [] or <>.

3.1. There is an effective procedure for constructing, for any given formula F in Q, an equivalent formula F′ in conjunctive normal form.

The proof is obtained by induction on the degree of the formula F.

The following example is given to illustrate this effective procedure. Take F as the formula [] (p & <> (q ∨ [] (r & t)) & (p→[] (q & <> t))). We obtain the conjunctive normal form by the following transformations:

1. [] p & []<> (q ∨ ([] r & [] t)) & (p→[] (q & <> t)) using the fact that [] (A & B) ↔ [] A & [] B
2. [] p & []<> ((q ∨ [] r) & (q ∨ [] t)) & (p→([] (q & <> t))) using propositional methods

3. $[\,]p \,\&\, [\,]<>((q \vee [\,]r) \,\&\, (q \vee [\,]t) \,\&\, (\sim p \vee [\,](q \,\&\, <>t))$
   using propositional methods
4. $[\,]p \,\&\, [\,]<>((q \vee [\,]r) \,\&\, (q \vee [\,]t) \,\&\, (\sim p \vee ([\,]q \,\&\, [\,]<>t))$
   using the fact that $[\,](A \,\&\, B) \leftrightarrow [\,]A \,\&\, [\,]B$

In the same way we have:

5. $[\,]p \,\&\, [\,]<>((q \vee [\,]r) \,\&\, (q \vee [\,]t)) \,\&\, (\sim p \vee [\,]q) \,\&\, (\sim p \vee [\,]<>t)$

4. *Modal Resolution*

The aim of classical resolution is to delete two inconsistent literals from two given clauses, i.e. consider the clauses $p \vee C_1$ and $\sim p \vee C_2$. Then:

$$\frac{p \vee C_1 \qquad \sim p \vee C_2}{\emptyset \vee C_1 \vee C_2}$$

where $\emptyset$ denotes the empty symbol. The intuitive justification of this rule is that the set $\{p, \sim p\}$ is inconsistent and no proper subset of it is inconsistent.

By analogy the aim of modal resolution will be to delete two (or more) elements at each step of a resolution proof [OE] [FC1] [SM].

Therefore from a family of sets INC of elements (disjunct of clauses) such that each set of the family is inconsistent and no proper subset of it is inconsistent. Then a set of resolution rules (one for each set of the family) is defined.

For example consider the three clauses:

$[\,](p \vee q) \vee C_1$
$<>\sim p \vee C_2$
$[\,]\sim q \vee C_3$

Then the rule:

$$\frac{[\,](p \vee q) \vee C_1 \qquad <>\sim p \vee C_2 \qquad [\,]\sim q \vee C_3}{C_1 \vee C_2 \vee C_3} \quad (*)$$

is a resolution rule for the system Q, because the set $\{[\,](p \vee q), <>\sim p, [\,]\sim q\}$ is inconsistent in Q.

Since $[\,](A \vee B) \nleftrightarrow [\,]A \vee [\,]B$ in Q, it is easy to see that for each formula of the form $[\,](A \vee B \vee \ldots)$ we must define a new family of sets in INC. This leads us to stress the following two problems.

1° The caracterization of the set INC can be as complex as a decision method for the system Q

2° The number of premises in the resolution rules is variable.

Therefore this kind of method will be interesting only for a subset of formulas [OE] [FC1] [FC2] [FC3] [FC8] or for the systems where INC is simple [OE] [SM].

In order to solve these problems we will consider a new method which we think is closer to the idea of the classical resolution principle [RJ]. Since the set INC will be reduced to the classical one (INC = $\{\sim p, p\}$); we will obtain from this a set of operations, whose purpose is to find this classical inconsistency.

Before going on we return to an example. Consider again the set of clauses:

$$[\,](p \vee q)$$
$$<>\sim p$$
$$[\,]\sim q$$

The set of these three clauses is inconsistent. Now we consider the two clauses $[\,](p \vee q)$ and $<>\sim p$. Since $[\,]A \,\&\, <>B \rightarrow <>(A \,\&\, B)$, it is possible to deduce in an informal way that the set $\{[\,](p \vee q), <>\sim p\}$ is satisfiable, because there is a state, where the set $\{p \vee q, \sim p\}$ is satisfiable. Then from $[\,](p \vee q)$ and $<>\sim p$ we can obtain $<>q$ and the two clauses $<>q$ and $[\,]\sim q$ are inconsistent. From this we can deduce that the set INC can be simplified by breaking down the rule (*) into two new rules. To explain this more precisely, we define a set of operations and properties.

Let C_1 and C_2 be two clauses. We define the operations: $\Sigma(C_1, C_2)$ and $\Gamma(C_1)$, and the properties: (C_1, C_2) is resolvable (i.e. C_1 and C_2 are resolvable) and (C_1) is resolvable, recursively as follows:

Classical operations

a) $\Sigma(p, \sim p) = \emptyset$
and $(p, \sim p)$ is resolvable. p will be called a resolved literal.

b) $\Sigma((D_1 \vee D_2), F) = \Sigma(D_1, F) \vee D_2$
and if (D_1, F) is resolvable, then $((D_1 \vee D_2), F)$ is resolvable.
c) $\Sigma(D_1 \,\&\, F_1 \,\&\, D_2 \,\&\, F_2) = \Sigma(D_1, D_2) \,\&\, F_1 \,\&\, F_2$
and if (D_1, D_2) is resolvable, then $(D_1 \,\&\, F_1 \,\&\, D_2 \,\&\, F_2)$ is resolvable.

Modal operations

a) $\Sigma([\,]E, [\,]F) = [\,]\Sigma(E, F)$
And if (E, F) is resolvable, then $([\,]E, [\,]F)$ is resolvable.
b) $\Sigma([\,]E, <>F) = <>(\Sigma(E, F) \,\&\, F)$
And if (E, F) is resolvable, then $([\,]E, <>F)$ is resolvable.
c) $\Sigma([\,]E, F) = \Sigma(E, F)$
And if (E, F) is resolvable, then $([\,]E, F)$ is resolvable.
d) $\Gamma(E(\Diamond(D \,\&\, D' \,\&\, F)) = E(<>(\Sigma(D, D') \,\&\, F \,\&\, D \,\&\, D'))$
And if (D, D') is resolvable, then $(E(<>((D \,\&\, D') \,\&\, F))$ is resolvable.

4.1. If C_1 and C_2 are unit clauses (clauses with only one disjunct) and C_1 and C_2 are resolvable (or C_1 is resolvable), then a clause is called resolvent of C_1 and C_2 (C_1) if it is the result of substituting:

$\emptyset$ for every occurrence of $(\emptyset \,\&\, E)$
E for every occurrence of $(\emptyset \vee E)$
$\emptyset$ for every occurrence of $\triangle\,\emptyset$, where $\triangle$ is $[\,]$, or $<>$ in $\Sigma(C_1, C_2)(\Gamma(C_1))$, as many times as necessary.

We note by $R(C_1, C_2)$ (or $R(C_1)$) a resolvent of C_1 and C_2 (C_1).

4.2. Let $C_1 \vee C$ and $C_2 \vee C'$ be two clauses. The resolution rules:

$$1)\ \frac{C_1 \vee C \qquad C_2 \vee C'}{R(C_1, C_2) \vee C \vee C'}$$

is applied if C_1 and C_2 are resolvable.

And the rule:

$$2)\ \frac{C_1 \vee C}{R(C_1) \vee C}$$

is applied if C_1 is resolvable.

4.3. Let $E(D \vee D \vee F)$ be a clause. The following rule will then be applied

$$3)\ \frac{E(D \vee D \vee F)}{E(D \vee F)}$$

4.4. Let S be a set of clauses. A deduction of C from S is a finite sequence $C_1, \ldots, C_n$ such that:

C_n is C, and
C_i $(1 \leqslant i \leqslant n)$ is:

a clause of S, or
a clause obtained from C_j, $j < i$ using the inference rules 2) or 3) or a clause obtained from C_j and C_k, $j, k < i$, using the inference rule 1).

4.5. A deduction of the empty clause is called a refutation.

We give an elementary example to illustrate the method. Given the two unit clauses $[\,]p$ and $<>(\sim p \vee q)$, a set of operations and properties corresponding to this set will be:

1) $\Sigma([\,]p, <>(p \vee q)) = <>\Sigma(p, \sim p \vee q)$.
   And if $(p, \sim p \vee q)$ is resolvable, then $([\,]p, <>(\sim p \vee q))$ is resolvable.
2) $\Sigma(p, \sim p \vee q) = \Sigma(p, \sim p) \vee q$
   And if $(p, \sim p)$ is resolvable then $(p, \sim p \vee q)$ is resolvable.
3) $\Sigma(p, \sim p) = \emptyset$
   And $(p, \sim p)$ is resolvable. Therefore $(p, \sim p \vee q)$ and $([\,]p, <>(\sim p \vee q)$ are resolvable and the inference rule 1) can be applied as follows:

$$\frac{[\,]p \quad <>(\sim p \vee q)}{<>q}$$

because $<>q$ is the result of substituting q for $(\emptyset \vee q)$ in $<>(\emptyset \vee q)$.

In classical logic only one operation, $\Sigma(p, \sim p)$, is required to apply the resolution rule, while in modal logic more than one operation may be needed to do it.

Intuitively we say that if a set of clauses is inconsistent, then there is a state which is classically inconsistent. And the aim of the operations is to find such a state.

We return again to the example. Given the three clauses $[\,](p \vee q)$, $<> \sim p$ and $[\,] \sim q$ we obtain the following refutation:

$$\frac{[\,](p \vee q) \qquad <> \sim p}{}$$

using that:
$\Sigma([\,](p \vee q), <> p) = <> (p \vee q, \sim p)$
And $\Sigma(p \vee q, \sim p) = \emptyset \vee q$

$$\frac{<> q \qquad [\,] \sim q}{\emptyset}$$

using that:
$\Sigma(<> q, [\,] \sim q) = <> \Sigma(q, \sim q)$
And $\Sigma(q, \sim q) = \emptyset$

5. *Completeness*

5.1. A set of clauses S is unsatisfiable iff S is refutable.
To prove this theorem the following two lemmas are necessary.

5.2. The set $S = \{L_1, \ldots, L_{n_1}, [\,] A_1, \ldots, [\,] A_{n_2}, <> P_1, \ldots, <> P_{n_3}\}$ of unit clauses is unsatisfiable iff either the set $\{L_1, \ldots, L_{n_1}\}$ is unsatisfiable or $\exists P_i \;\; 1 \leqslant i \leqslant n_3$ for which $S_i = \{A_1, \ldots, A_{n_2}, P_i\}$ is unsatisfiable.

Proof.

a) suppose that $\{L_1, \ldots, L_{n_1}\}$ and $S_i = \{A_1, \ldots, A_{n_2}, P_i\}$ $1 \leqslant i \leqslant n_3$ are satisfiable. Then we can construct a model M satisfying S, from models M_i of S_i and a maximal consistent extension O of $\{L_1, \ldots, L_{n_1}\}$ that is different from the states of M_i. M is obtained by union of the M_i and O where the accessibility relation has been extended by a set of pairs (O, O_i) where O_i is a state in M_i such that: M_i, O_i sat S_i. Therefore it is easy to see that S is satisfiable in M.

b) The converse statement is obtained using the fact that $[\,](A \to B) \to ([\,] A \to [\,] B)$, the necessitation rule and propositional reasoning.

Suppose $\{A_1, \ldots, A_{n_2}, P_i\}$ is unsatisfiable. Then the following formulas are theorems:

1. $\vdash \sim A_1 \vee \ldots \vee \sim A_{n_2} \vee \sim P_i$
2. $\vdash [\,]\,(A_1 \,\&\ldots\&\, A_{n_2} \to \sim P_i)$
3. $\vdash [\,]\,(A_1 \,\&\ldots\&\, A_{n_2}) \to [\,] \sim P_i$
4. $\vdash \sim [\,]\, A_1 \vee \ldots \vee \sim [\,]\, A_{n_2} \vee \sim <> P_i$

Hence $[\,]\, A_1 \,\&\ldots\&\, [\,]\, A_{n_2} \,\&\, <> P_i$ is unsatisfiable.

The proof for the other case is trivial.

On the light of this lemma the modal operations in the resolution can be interpreted as a tool which prevents us from using formulas belonging to the sets S_i, S_j, if $i \neq j$ $i, j = 0, \ldots, n_3$ to obtain inconsistency for set S.

For example for any formula $<> P_i$ $1 \leqslant n \leqslant n_3$ in S we obtain a set $S_i = \{A_1, \ldots, A_{n_2}, P_i\}$; from the modal operations point of view, this means that we can't define a rule as $\Sigma\,(<> P_i, <> P_j)$. And since the A_j, $j = 1, \ldots, n_2$ appears in S_i, the rule $\Sigma\,([\,]\, A_j, <> P_i)$ must be defined.

5.3. Given the two sets of unit clauses $S = \{L_i, \ldots, L_{n_1}, [\,]\, A_1, \ldots, [\,]\, A_{n_2}, <> (C_1 \,\&\ldots\&\, C_{n_3}), \ldots, <> P_{n_4}\}$ and $S' = \{A_1, \ldots, A_{n_2}, C_1, \ldots, C_{n_3}\}$. If R is a refutation of S′ then R can be transformed into a refutation of S.

Remark: to transform a refutation of S′ into a refutation of S, it is sometimes necessary to modify the order in the application of rules in the refutation of S. We give the following two examples:

Example 1. Consider the set of clauses $S = \{[\,]\,(p \vee q), [\,]\,(\sim p \vee q), <> \sim q\}$. And $S' = \{p \vee q, \sim p \vee q, \sim q\}$. The refutation of S′:

$$\dfrac{\dfrac{p \vee q \qquad \sim q}{p} \qquad \dfrac{\sim p \vee q \qquad \sim q}{\sim p}}{\emptyset}$$

must be reorganized to

$$\dfrac{\dfrac{p \vee q \qquad \sim p \vee q}{q} \qquad \sim q}{\emptyset}$$

Because it is only in this way that we can obtain directly a refutation of S:

$$\frac{\dfrac{[\,](p \vee q) \qquad [\,](\sim p \vee q)}{[\,]q} \qquad <>\sim q}{\emptyset}$$

Example 2. Consider the set of clauses S = {[ ] (p ∨ q), [ ] (~ p ∨ t), <> (~ q & ~ t)}. And S′ = {p ∨ q, ~ p ∨ t, ~ q, ~ t}. The refutation of S′:

$$\frac{\dfrac{p \vee q \qquad \sim q}{p} \qquad \dfrac{\sim p \vee t \qquad \sim t}{\sim p}}{\emptyset}$$

must be reorganized to

$$\frac{\dfrac{p \vee q \qquad \dfrac{\sim p \vee t \qquad \sim t}{\sim p}}{q} \qquad \sim q}{\emptyset}$$

Because it is only in this way that we can obtain directly a refutation of S:

$$\frac{\dfrac{[\,](p \vee q) \qquad \dfrac{[\,](\sim p \vee t) \qquad <>(\sim q \,\&\, \sim t)}{<>(\sim q \,\&\, \sim p)}}{<>(q \,\&\, \sim q)}}{\emptyset}$$

Consequently the proof of this lemma must proceed by induction on the number of inferences which use two formulas that belong to a conjunct governed by a <>.

Then an inference in a refutation R of S′ with two premises D and D′ is called critical iff some C_i (in the element <> (C_1 & . . . & C_{n_3}) of S) are used to derive D and D′.

The proof of the lemma is obtained by induction on the number c(R) of critical inferences in R.

Proof of theorem 5.1.

The proof of the theorem 5.1. is obtained by induction on the degree of S.

If d(S) = 0 then the theorem is proved by propositional methods [CHL].

Assume that the theorem holds when $0 \leq d(S) \leq n$. To complete the induction we consider $d(S) = n+1$. The proof is obtained by induction on the number of $\vee$ governing the disjuncts of the clauses in S (noted by $\nu(S)$).

If $\nu(S) = 0$, then using lemma 5.2., the induction hypothesis and lemma 5.3. the result will be established. The induction step is proved as usual [CHL]. Let $C = E_1 \vee C_1$ be a clause in S. Construct separately refutations R_2 from $(S\text{-}\{C\}) \cup E_1)$ and R_1 from $(S\text{-}\{C\}) \cup C_1$ respectively.

R_2 [E_1 ⋮ empty clause]

$C_1 \vee E_1$ ⋮

R_1 [C_1 ⋮ empty clause]

and put R_2 on the top of R_1 after adding C_1 to all clauses in R_2 (figure 2).

This kind of method can be extended easily to other modal logics. In this way a completeness theorem has been obtained for the propositional calculus K [FC4], S4 [FC6], S5 [FC5], for linear temporal logic of programs [CF2] and for a mutual belief logic [FS]. M. Cialdea [CM1] obtained the same result for a modal translation of the propositional intuitionistic calculus.

It is easy to see that for the set of prenex formulas of the first order modal calculus the method can be extended [FC1], [FC6], [FC7], [CF1]; on the contrary when we consider the complete first order calculus some new problems arise, since permutation of quantifiers and modal operators don't hold, for example: $\exists x\, [\,]\, p(x) \not\leftrightarrow [\,]\, \exists x p(x)$. To solve this kind of problems it is necessary to distinguish the skolem constant introduced by $\exists x$ in $\exists x\, [\,]\, p(x)$ from the one, introduced by

$\exists x$ in [] $\exists x\ p(x)$ which can be different in each state [CM2]. Comprehensive work on this subject has been presented by K. Konolige [KK] who obtains a Herbrand theorem and a refutation procedure in the lemma 5.2. style for belief logic.

6. *Deduction in normal form*

By analogy with classical logic we can easily define a refinement of modal resolution i.e. normal form of deduction. We will present an extension of linear resolution to the modal operators. The definitions are essentially the same as for classical resolution. But each time a subformula of the modal formula is resolved, a resolved expression [CHL] is generated.

In what follows we suppose that clauses are ordered; in other words the logical operators & and $\vee$ are treated as non-commutative. We also have a mechanism for recording the information about resolved clauses. In this way the expression in a clause can be enclosed in $\Box$. $\boxed{E}$ means that the expression E has been resolved.

We call a sequence an ordered clause with possible resolved expressions.

6.1. First formula of a sequence

Let S be a sequence. We call a first formula of S the subformula of S defined recursively as follows:

- S is a first formula
- If F is a first subformula of S, and F is of the form $F_1 \vee F_2 \vee \ldots \vee F_n$, then F_1 is a first subformula
- If F is a first subformula of S and F is the form F_1 & F_2 & ... & F_n, then each F_i is a first subformula
- If F is a first subformula of S and F is of the forms $[\,]\,F_1$ or $<>F_1$, then F_1 is a first subformula.

6.2. Linear operations

Let C_1 and C_2 be two sequences. We define the operations

- $\Sigma(C_1, C_2)$
- $\Gamma(C_1)$

And the properties

- (C_1, C_2) is resolvable
- (C_1) is resolvable

recursively as follows:

– classical operations

Since the disjuncted clauses can possess the disjunction and conjunction symbols, it is necessary to introduce operations to manipulate these symbols by opposition to classical linear resolution where rules for the $\vee$-operator only are necessary.

- $\Sigma(A_1 \& \ldots \& A_n, A) = \Sigma(A_i, A) \& A_1 \& \ldots \& A_n$

And if (A_i, A) is resolvable, then $(A_1 \& \ldots \& A_n, A)$ is resolvable.

- $\Sigma(A, A_1 \& \ldots \& A_n) = \Sigma(A, A_i) \& A_1 \& \ldots \& A_n$

And if (A, A_i) is resolvable, then $(A, A_1 \& \ldots \& A_n)$ is resolvable

- $\Sigma(A_1 \vee \ldots \vee A_n, A) = \Sigma(A_1, A) \vee A_2 \vee \ldots \vee A_n$

And if (A_1, A) is resolvable then $(A_1 \vee \ldots \vee A_n)$ is resolvable

- $\Sigma(A, A_1 \vee \ldots \vee A_n) = A_1 \vee \ldots \vee A_{i-1} \vee \Sigma(A, A_i) \vee A_{i+1} \vee \ldots \vee A_n \vee \boxed{A}$

And if (A, A_i) is resolvable then $(A, A_1 \vee \ldots \vee A_n)$ is resolvable.

In this operation we express the fact that the expression A has been resolved, if (A, A_i) is resolvable.

- $\Sigma(p, \sim p) = \emptyset$

And $(p, \sim p)$ is resolvable in p.

– Reduction operations

- $\Gamma(E(\boxed{A} \vee A_1)) = E(A_1)$
- $\Gamma(E(A_1 \vee \ldots \vee \boxed{A_i} \vee \ldots \vee A_n)) = E(\Sigma(A_1, A_i) \vee A_2 \vee \ldots \vee \boxed{A_i} \vee \ldots \vee A_n)$

And if (A_1, A_i) is resolvable then $E(A_1 \vee \ldots \vee A_n)$ is resolvable.

- $\Gamma(E(A_1 \vee A_2 \vee A_1 \vee A_3)) = E(A_2 \vee A_1 \vee A_3)$

– Modal operations

• Σ ([ ] E, $\triangle$ F) = Σ (E, F) provided that $\triangle$ is [ ], or <>.

And if (E,F) is resolvable, then ([ ] E, $\triangle$ F) is resolvable

It must be noted that the symmetrical operations Σ ($\triangle$ F, [ ] E), when $\triangle$ is [ ] or <>, and Σ (F, [ ] E) must be considered.

• Γ (E (<> (A_1 & . . . & A_n) = E (<> (Σ (A_i, A_j) & A_1 & . . . & A_n)))

And if (A_i, A_j) is resolvable, then (E (<> (A_1 & . . . & A_n)) is resolvable.

– Simplifications

• Γ (E (A_1 & $\emptyset$ & A) = Γ (E ($\emptyset$))
• Γ (E (A_1 $\vee$ $\emptyset$ $\vee$ A) = E (A_1 $\vee$ A)
• Γ (E $\triangle$ ($\emptyset$)) = Γ (E ($\emptyset$)) where $\triangle$ = [ ] or <>.

6.3. Let C_1 and C_2 be two sequences and p a first literal of C_1. If (C_1, C_2)(C_1) is resolvable in p, then a remaining sequence obtained from Σ (C_1, C_2)(Γ (C_1)) after simplification and reduction will be called a resolvent.

6.4. Let S be a set of ordered clauses. A linear deduction from S of C_n, with a top ordered sequence C in S is a sequence $C_0, \ldots, C_n$ such that each C_i ($1 \leqslant i \leqslant n$) is either a resolvent of C_{i-1}, or a resolvent of C_{i-1} against an ordered clause of S.

We give a modal version of a usual example for propositional calculus. Let us give the inconsistent set of clauses:

[ ] (p $\vee$ q)
[ ] ($\sim$ p $\vee$ q)
[ ] (p $\vee$ $\sim$ q)
<> ($\sim$ p $\vee$ $\sim$ q)

Its refutation will be:

$$
\begin{array}{l}
[\,](p \vee q) \qquad [\,](\sim p \vee q) \\
\hline
[\,](q \vee \boxed{p} \vee q) \vee \boxed{[\,](p \vee q)} \\
\hline
[\,](\boxed{p} \vee q) \vee \boxed{[\,](p \vee q)} \\
\hline
[\,]q \vee \boxed{[\,](p \vee q)} \qquad [\,](p \vee \sim q) \\
\hline
[\,](p \vee \boxed{q}) \vee \boxed{[\,](p \vee q)} \vee \boxed{[\,](q)} \qquad <>(\sim p \vee \sim q) \\
\hline
<>(\sim q \vee \boxed{q} \vee \boxed{p}) \vee \boxed{[\,](p \vee q)} \vee \boxed{[\,]q} \vee \boxed{[\,](p \vee q)} \\
\hline
<>\boxed{p} \vee \boxed{[\,](p \vee q)} \vee \boxed{[\,]q} \vee \boxed{[\,](p \vee q)} \\
\hline
\qquad\qquad\qquad \emptyset
\end{array}
$$

where we have considered only the main steps in the proof.

6.5. Completeness

Let S be a set of ground sequences, and C a sequence in S. If S is unsatisfiable and S – {C} is satisfiable, then there is a linear refutation from S with top sequence C.

Proof. The proof proceeds by induction on the degree of S, using a reasoning similar to the one used in the case of modal resolution given in 5.

7. *Applications and Implementations*

Several implementations of this method have been done. For propositional S5, O. Galion [GO] has implemented a linear refinement in Pascal on a APPLE II. Rychlik [RP] and Kochut [KR] made a program which is a linear refinement, for a prenex subset of formulas of quantificational S5. It has been applied in an expert system in medecine. In [LE] and [FL] an implementation of linear refinement for the prenex formulas of linear temporal logic of programs has been realized in Prolog on DEC LSI-11-23 and on DPS8. These programs have been used to prove properties of programs [CF2]. In the same way P. Combe and V. Leford [CL] devised a program for a subset of linear temporal formulas and used it for proving properties of network protocols.

One of the most important problems of these realisations is the difficulty in manipulating realistic sets of data. To solve this problem J. Henry [HJ] has implemented the modal resolution on a subset of prenex temporal formulas, that we will call Horn modal clauses, clauses with at most one of its propositional variables which is not negated. The programs have been realized in Prolog on a DPS8, and control facilities of Prolog can be used.

8. *Acknowledgements*

I wish to thank G. Minc and A. Raggio for their stimulating correspondance and encouragement, and E. Audureau, D. Laurier and S. Soulhi for their comments. We also thank very much P. Enjalbert for pointing out errors in the definition of modal operations.

Université Paul Sabatier Luis FARIÑAS-DEL-CERRO
Langages et Systèmes Informatiques
118, route de Narbonne
F - 31062 Toulouse Cedex

BIBLIOGRAPHY

[AR] ANDERSON, R., Improved decision procedures for Lewis's calculus S4 and von Wright's calculus. *J.S.L., Vol. 19*, 3, (1953), pp. 101-214.

[BA] BAYART, A., On truth-tables for M, B, S4 and S5. *Logique et Analyse*, Sept. 1970, pp. 335-375.

[CR] CARNAP, R., Modalities and quantification. *J.S.L., vol. 11*, 2, (1946), pp. 33-64.

[CF1] CAVALLI, A.R., FARIÑAS DEL CERRO, L., Specification and verification of networks protocols using temporal logic. Proceedings of the Sixième Colloque International sur la programmation, Springer-Verlag, *LNCS n° 167*, (1984).

[CF2] Cavalli, A.R., Fariñas delCerro, L., A decision method for linear temporal logic. Proceedings of the Seventh International Conference on Automata Deduction, Springer-Verlag, *LNCS n° 170*, (1984).

[CHL] Chang, C., Lee, R., Symbolical logic and mechanical theorem proving. *Academic Press*, New-York, (1973).

[CM1] Cialdea, M., Une méthode de résolution pour la logique intuitionniste propositionnelle. Rapport LSI, Université Paul Sabatier, Toulouse, (1983).

[CM2] Cialdea, M., Ph.d. Thesis Université Toulouse III, *in preparation*.

[CL] Combes, P., Lefort, V., GENTIANE: un outil de preuve de cohérence de formules en logique temporelle. *Note technique, CNET IPAA ILLC*.

[FC1] Fariñas del Cerro, L., Déduction automatique et logique modale. *Thèse d'Etat*, Université de Paris VII, 1981.

[FC2] Fariñas del Cerro, L., Prolegomena for programming in modal logic. 6th European Meeting on Cybernetics and Systems Research, Vienne, Avril 1982, North-Holland, (1982).

[FC3] Fariñas del Cerro, L., Logique modale et processus communicants. Colloque AFCET "*Les mathématiques de l'Informatique*", Paris, (1982).

[FC4] Fariñas del Cerro, L., A simple deduction method for modal logic. *Information Processing Letters, Vol. 14*, n° 2, (1982).

[FC5] Fariñas del Cerro, L., A deduction method for modal logic. *Europeaan IA Conference*, 11-14 July 1982, Orsay.

[FC6] Fariñas del Cerro, L., Les modalités de la correction totale. RAIRO *Informatique Théorique, Vol. 16*, n° 4, (1982), pp. 349-363.

[FC7] Fariñas delCerro, L., Temporal reasoning and termination of programs. *Proceedings IJCAI* 1983.

[FC8] Fariñas delCerro, L., Un principe de résolution en logique modale. *RAIRO Informatique Théorique, Vol. 18*, n° 2, (1984).

[FL] Fariñas delCerro, L., Lauth, E., Raisonnement temporel: une méthode de déduction. *Rapport LSI, Université Paul Sabatier*, Toulouse, Novembre 1982.

[FS] Fariñas delCerro, L., Soulhi, S., Mutual belief logic for processing definite reference. *Proceedings of International Workshop on Natural Language Understanding and Logic Programming*, Rennes, 18-20 Septembre 1984, North-Holland.

[GO] Gallion, O., Implémentation d'un démonstrateur de théorèmes. *Rapport LSI, Université Paul Sabatier*, Toulouse, Novembre 1981.

[HJ] Henry, J., Implémentation d'un démonstrateur pour des clauses d'Horn en logique modale. *Rapport LSI, Université Paul Sabatier*, en préparation.

[KR] Kochut, E., Rychlink, P., Applications of modal logic in information systems, in Bolc, L., (ed.), *Translating natural language into logical form, Springer, Symbolic Computation serie, to appear*.

[KK] Konolige, K., A Deduction Model of Belief and its Logics. *Ph.D. Thesis, Stanford University, 1984*.

[LE] Lauth, E., Une méthode de résolution linéaire ordonnées pour la logique temporelle linéaire. *Rapport LSI-ENSEEITH*, Université Paul Sabatier, Toulouse, (1983).

[LE] LEMMON, E., An introduction to modal logic. *Amer. Philo. Generaly Monograph*, (1977).

[SM] SHIMURA, M., Resolution in a new modal logic. *Proceedings IJCAI 79*, pp. 809-814.

[OE] ORLOWSKA, E., Resolution systems and their applications I & II, *Fundamenta Informaticae, Vol. III*, n° 2, (1980), pp. 235-267 & 333-362.

[RJ] ROBINSON, J., A machine oriented logic based on the resolution principle. *J. ACM, 12*, (1965), pp. 23-41.

[RP] RYCHLINK, P., The use of modal default reasoning in information system. *To appears in The Int. Journal of Man-Machine Studies*.