PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT. © ICOT, 1988

DECLARATIVE SEMANTICS FOR MODAL LOGIC PROGRAMS

Ph. BALBIANI, L. FARIÑAS DEL CERRO, A. HERZIG

Langages et Systèmes Informatiques, Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse Cedex

ABSTRACT

In this paper we consider modal logics as programming languages. To modal programs we associate a declarative semantics represented by a tree which will be defined as the limit of a certain transformation on modal programs. This fixpoint represents the minimal Kripke model of the program. We give a procedural semantics based in SLD-resolution.

1 INTRODUCTION

Since logic programming is a useful tool for developping AI Systems and modal logics are natural formalisms in domains as natural language understanding or problem solving (see (Konolige1986), (Conf. 1986)), we undertake to extend the classical theoretical results in logic programming of (Apt 1987) and (Lloyd 1984) to modal logic. In the last years there have been several approaches to make logic programming more powerful by extending it to intensional logics, e.g. (Abadi and Manna 1987) and (Gabbay 1987) for temporal logic, (Subrahmanian 1987) and (van Emden 1987) for quantitative and (Blair and Subrahmanian 1987) for paraconsistent reasoning, (Okada 1988) and (Akama 1986) for modal logics, (Yiang 1988) for epistemic logics, (Nute 1987) for conditional logics, (Fitting 1988) for reasoning with contradictions, (Gabbay and Reyle 1984) and (Miller 1987) for intuitionistic logic.

In previous works we introduced a system called MOLOG that allows us to consider modal logic as a programming language (Balbiani et al. 1987), (Fariñas 1986), (Fariñas and Penttonen 1987). The aim of this paper is to continue this work by giving some theoretical bases. We define a particular set of modal formulas called modal Horn clauses in analogy with classical logic (see (Colmerauer et al. 1986)). We will consider modal programs to be sets of modal Horn clauses. To them we associate a declarative semantics represented by a tree which will be defined as the limit of a certain transformation defined for the modal programs. This limit represents the minimal Kripke model of the program. We give an inference mechanism based on a resolution principle.

The paper is organized as follows: In section 2 we introduce modal logic. In section 3 we define the Herbrand Universe for modal logic. In section 4 we give the definition of modal Horn clauses. The resolution rules for an exemplary modal system are given in section 5. In section 6 we present the declarative semantics for it. In section 7 the correctness of declarative semantics with respect to possible worlds-semantics is established. The completeness theorem of the resolution method is presented in section 8. In section 9 we give the resolution rules and declarative semantics for T and S4.

We give only the mail ideas of the proof, all details and proofs and the cases of other modal systems are given in the full paper (Balbiani,and al. 1987).

2 MODAL LOGIC

2.1 Syntax. A modal language L is defined from the set of **primitive symbols** composed of the following pairwise disjoint sets: predicate names - p,q..., functions names - g, h,..., variables - x, y, ..., classical

This word has been supported by ESPRIT Project P973 ALPES.

connectives - $\neg$, $\wedge$, $\rightarrow$, $\leftrightarrow$, $\vee$, classical quantifiers - $\exists$, $\forall$, unary modal connective - $\Box$ ("necessary").

As usual $\Diamond A$ is defined by $\neg\Box\neg A$. The formation rules for **terms** and **formulas** are the classical ones with : if A is a formula $\Box A$ and $\Diamond A$ are formulas.

2.2 Semantics. The Kripke semantics with uniform domain (Hughes and Cresswell 1986) for L is defined in terms of a **model** M = (W,R,D,m) where $R \subseteq W^2$ is a relation, D is a set of objects, and m is a meaning function such that

. if x is a variable, $m(x) \in D$;

. if g is a n-ary functional symbol, $n \geq 0$, then $m(g): D^n \rightarrow D$ is a function;

. if p is an n-ary predicate symbol then $m(p) \subset W \times D^n$.

In order to simplify our proofs we consider the relation R to be total, i.e. for every w there is a w' such that $(w,w') \in R$.

The satisfiability relation is defined as usual by induction on the structure of the formulas. Here we give only the case of the modal operator :

M,w sat $\Box A$ iff for every $w' \in W$ if wRw' then M,w' sat A.

We say that M **satisfies** A or M **is a model for** A if there is a w in W such that M,w sat A.

3 MODAL HERBRAND UNIVERSE

In what follows we shall consider a subset of formulas called universal formulas. A **universal formula** is a closed formula of the form $\forall x_1 \ldots \forall x_n$ M where M is a formula in which no quantifiers appear.

Definition. The H-domain for L (denoted by H_L) is the set of all ground terms, which can be formed out of the constants and functions appearing in L. (In the case that L has no constants we add some constant, say "a", to form ground terms).

Definition. Let M be a model. M is an H-interpretation iff $D = H_L$, m(c) = c for every constant c, $m(g)(t_1,\ldots,t_n) = g(t_1,\ldots,t_n)$ for every functional symbol g and $t_1,\ldots,t_n \in H_L$.

Lemma 3.1 A is satisfiable iff A is satisfied in an H-interpretation.

Proof. The proof is a trivial extension of the classical case.

Note that we can say now that M,w sat $\forall x\, A(x)$ iff M,w sat A(t) for every $t \in H_L$. So from now on we shall consider only H-interpretations.

4 MODAL HORN CLAUSES

4.1 Since $\Diamond(A \wedge B)$ and $\Diamond A \wedge \Diamond B$ are not equivalent for every definition which does not identify $\Diamond$ and $\Box$, any notion of modal Horn clause must allow the $\wedge$-operator in the scope of the $\Diamond$-operator. This makes things rather complicated because (in opposition to classical logic) one modal Horn clause may be unsatisfiable alone, as for example $\Diamond(p \wedge \neg p)$. Our aim is to avoid this undesirable feature. Therefore we use a technique similar to the Skolem technique for eliminating existential quantifiers and extend the language L in order to translate formulas into the extended language. This idea appeared already in (Fariñas and Penttonen 1987) and (Chan 1987).

We add to L the new set of modal operators $\{\langle t \rangle : t$ is a term$\}$. If A is a formula then $\langle t \rangle A$ is a formula, too. The new language is called L°. Take some enumeration $\{t_1,t_2,\ldots\}$ of the ground terms. The semantics for L° is defined by a model M° = (W,(R,F),D,m) where F is a set $\{f_1,f_2,\ldots\}$ such that each $f_i: W \rightarrow W$ is a function and $f_i \subseteq R$ for every i. The satisfiability relation is defined as above where M°,w sat $\langle t_i \rangle A$ iff M,$f_i(w)$ sat A. The definitions of satisfiability and of an H-interpretation for L° are the same as for L.

Now we define a translation * that allows us to embed the universal formulas of L in L°. Without loss of generality we suppose from now on that all formulas contain only operators $\neg$, $\wedge$, $\vee$, and that the negation appears immediately before the atomic formulas.

Definition. We map a formula A of L into a formula A* of L° by replacing each occurence of "◊" in A by $<g(x_1,...,x_n)>$ where g is a new function name, and each x_i is a variable such that $\forall x_i$ governs the ◊-occurrence in A. (So in A* every new function name occurs only once).

Example. If A is $\forall x \Box(\Diamond p(x) \vee \Diamond(q \wedge \neg t))$ then A* will be $\forall x \Box(<g(x)>p(x) \vee <h(x)>(q \wedge \neg t))$.

Lemma 4.1 Let $A = \forall x_1 ... \forall x_n$ C be a universal formula of L in which negations occur immediately before atomic formulas. A is satisfiable in an H-interpretation of L iff A* is satisfied in an H-interpretation of L°.

Proof. The proof is done by induction on the structure of C. The main case in the induction step is when the formula is of the form ◊ B. Suppose M,w sat ◊B. Hence for some w' such that wRw' we have M,w' sat B. By induction hypothesis there exists a model M'=(W,(R,F),D,m) such that M',w' sat B*. Now as in classical logic we extend the language by a new skolem function g, and we build a new model $M''=(W,(R,F \cup \{f_j\}),D,m)$ such that j is the index of $g(t_1,...,t_n)$ and $f_j(w) = w'$. Consequently

M'',w sat $<g(t_1,...,t_n)>$ B*.

Remark. The translation and the lemma are identical for all modal systems based in a language with the two classical modal operators □, ◊ and possessing a Kripke style semantics.

4.2 In analogy with classical logic we give the following definitions:

Goal clauses (GC):atomic formulas ⊆ GC

F, G ∈ GC implies ◊F, <t>F, F ∧ G ∈ GC

Definite clauses (DC): atomic formulas ⊆ DC

F ∈ DC, G ∈ GC implies F ← G, □F, <t>F ∈ DC

A **Horn clause** (HC) of L° will be either a goal or a definite clause. As in classical logic we suppose that every variable in the clauses is bound by a universal quantifier, i.e. a set of clauses is considered to be a universal formula. In what follows sets of definite clauses will be called **programs.**

Using Lemma 4.1 we will consider a resolution principle for formulas whose translation is a set of Horn clauses (where a conjunction of clauses is considered as a set). In particular this will be the case for every formula which after deletion of the modal operators gives a set of classical Horn clauses.

Example. The formula $\Box\Diamond((p(x) \vee \neg q(x)) \wedge p(a))$ becomes after skolemization a set of Horn clauses $\{\Box<f(x)> (p(x) \leftarrow q(x)) , \Box <f(x)> p(a) \}$.

5 RESOLUTION

Now we shall study in an exemplary way a particular modal system, quantificational Q. For the other classical modal systems T and S4 see section 9 or the full paper (Balbiani, Fariñas and Herzig 1987). The models of Q are caracterized by a serial accessibility relation R. The modal operator can be interpreted as a temporal operator able to describe from each state the next state. The intuitive meaning of □A is that in every next state we will have A, and of ◊A that there is a next state in which A is true. We shall describe SLD-resolution for the modal Horn clauses of Q. We start by defining how to produce a new goal clause from an old goal and a definite clause. For it we give a formal system composed of rules for computing resolvents and of simplification rules.

5.1 Rules for computing resolvents. We define the relation on clauses "G_1 is a **direct resolvent** of A and G" where A is a definite clause and G and G_1 are goal clauses, in symbols $A,G \Rightarrow G_1$, by the following formal system :

Axiom: p, q ⇒ T(σ) if there is a most general unifier σ such that pσ = qσ, where p and q are atomic formulas and

T denotes the empty question ("True").

Classical rule:

($\leftarrow$ rule): from $A, G \Rightarrow G_1(\sigma)$

infer $A \leftarrow G_2\ , G \wedge G_3 \Rightarrow G_1 \wedge G_2 \wedge G_3(\sigma)$

Modal rules:

($\Box\langle t\rangle$-rule): from $A, G \Rightarrow G_1(\sigma)$

infer $\Box A, \langle t\rangle G \Rightarrow \langle t\rangle G_1(\sigma)$

($\langle t\rangle\langle t'\rangle$-rule): from $A, G \Rightarrow G_1(\sigma)$

infer $\langle t\rangle A, \langle t'\rangle G \Rightarrow \langle t\rangle G_1(\tau\sigma)$ if $\tau(\sigma(t)) = \tau(\sigma(t'))$

($\Diamond$-rule): from $A, \langle t\rangle G \Rightarrow G_1(\sigma)$

infer $A, \Diamond G \Rightarrow G_1(\sigma)$

t and t' denote classical terms, and we use again the classical definition of a most general unifier (Lassez et al. 1987). By a goal clause we denote some particular permutation of it, in other words for each permutation we have a particular selection function.

The procedural interpretation of the axiom and the rules is as follows.

Axiom: p is an answer to the question q if p and q are unifiables.

Classical rule: from the fact $A \leftarrow G_2$ and goal $G \wedge G_3$ we infer the new goal $G_1 \wedge G_2 \wedge G_3$ if G_1 is the goal obtained from A and G.

Modal rules: We consider only the $\Box\langle t\rangle$-rule, for the another rules the interpretation is identical. A fact $\Box A$ and a goal $\langle t\rangle G$ produce the new goal $\langle t\rangle G_1$ if G_1 is obtained from the fact A and the goal G.

5.2. Simplification rules. The role of the simplification rules is to eliminate the **T** symbol after application of the formal system rules. The relation "A is similar to B", noted $A \approx B$, is the least congruence relation containing $\Delta T = T$, where $\Delta = \Box, \Diamond, \langle t\rangle$, and $T \wedge D \approx D$.

Definition. We say that G_1 is a **resolvent** of A and G if there is some G_1' such that $A, G \Rightarrow G_1'$ and $G_1' \approx G_1$. We write $A, G \Rrightarrow G_1$ for "G_1 is a resolvent of A and G".

Example. The definite clause $\langle f(x)\rangle\ (p(x) \leftarrow q)$ and the goal clause $\Diamond p(a)$ have as a resolvent $\langle f(x)\rangle\ q\ \{x\backslash a\}$.

From the axiom $p(x), p(a) \Rightarrow T\{x\backslash a\}$ we infer $p(x) \leftarrow q, p(a) \Rightarrow T \wedge q\{x\backslash a\}$ by the classical rule. From the latter we infer $\langle f(x)\rangle(p(x) \leftarrow q),\ \Diamond p(a) \Rightarrow \langle f(x)\rangle(T \wedge q)\ \{x\backslash a\}$ by $\Diamond$-rule and $\langle t\rangle\langle t'\rangle$-rule. As $\langle f(x)\rangle(T \wedge q)\ \{x\backslash a\} \approx \langle f(x)\rangle\ q\ \{x\backslash a\}$ the new goal will be $\langle f(a)\rangle\ q$.

5.3 Resolution rule

$$\frac{C \quad G}{G_1(\sigma)} \qquad \text{if } C, G \Rrightarrow G_1(\sigma)$$

where C is a definite clause and G, G_1 are goal clauses.

SLD-resolution is defined using the resolution rule as follows. Given a program P and a goal clause G we say that G is derivable from P, and we note it $P \vdash G$, if there is a sequence $G_1, \ldots, G_n$ of goal clauses where G_1 is G and G_n is **T**, and each G_{i+1} is obtained from G_i against a clause of P using the resolution rule.

Example. Let $P = \{\Box q,\ \langle f(x)\rangle(p(x) \leftarrow q)\}$ and $G = \Diamond p(a)$. We prove that G is derivable from P:

$$\frac{\dfrac{\langle f(x)\rangle(p(x) \leftarrow q) \quad \Diamond p(a)}{\Box q \qquad \langle f(a)\rangle q}}{T}$$

- $\Box q \quad \langle f(a)\rangle q$: by means of $\Diamond$-rule, $\langle t\rangle\langle t'\rangle$-rule and $\leftarrow$-rule
- T: by means of the $\Box\langle t\rangle$-rule.

5.4 Consistence of Resolution. The proof is by induction on the number of classical and modal rules.

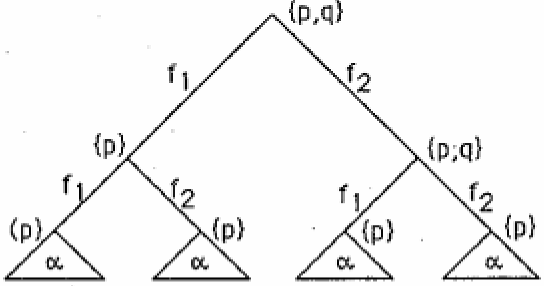
6 DECLARATIVE SEMANTICS FOR MODAL LOGIC

In this section we will associate to modal logics, as in classical logic, a declarative semantics that allows us to consider modal Horn clauses as programs. The base of L° denoted by $\mathfrak{B}_{L°}$ is defined as in the classical case as the set of ground instances of the atomic formulas appearing in L° obtained using the domain $H_{L°}$. In order to simplify the notation, we drop the index L° from $\mathfrak{B}_{L°}$ and $H_{L°}$.

Definition. A tree is a structure $t=\langle W,F,m\rangle$. W is the set of nodes, and F is a set of functions from W into W standing for the arrows of the tree. $m: W \rightarrow 2^{\mathfrak{B}}$ is a meaning function.

Remark. As for the translation of section 4 we note that the definition of tree structure is independent of the modal logic.

Example.



It is a tree if α is a tree, for a language possessing two terms f_1 and f_2.

6.1. The lattice of trees. Let Γ be the set of trees and $t = \langle W,F,m\rangle$ and $t' = \langle W,F,m'\rangle$ two trees in Γ. We define the **intersection** and the **union** operations between trees:

$t \cap t' = \langle W,F,m''\rangle$ where $m''(w)=m(w) \cap m'(w)$ for every $w \in W$.

$t \cup t' = \langle W,F,m''\rangle$ where $m''(w)=m(w) \cup m'(w)$ for every $w \in W$.

As m(w) and m'(w) are sets of ground atomic formulas, ∩ and ∪ are the standard set intersection and union.

We define the relation ⊆ by : $t \subseteq t'$ if and only if $t \cap t' = t$.

The minimal element of Γ is noted ⊥. ⊥ is the tree $\langle W,F,m\rangle$ where m(w) is the empty set {} for every element w of W.

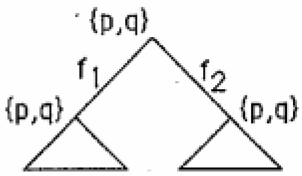
Fact. Since $2^{\mathfrak{B}}$ is a complete lattice Γ will be a complete lattice, too.

6.2. The transformation Tp for Q. In order to give a declarative semantics to a program P we associate to it a transformation T_P depending on the particular modal logic. In what follows we shall give the transformation for Q.

Definition. Let a program P, a Horn clause G, a tree t, a node w in t. By induction on the structure of G we define "G **appears in** w ", denoted by $G \in (t,w)$, as follows :

1. $p \in t,w$ iff $p \in m(w)$ if p is a ground atomic formula
2. $A \wedge B \in (t,w)$ iff $A \in (t,w)$ and $B \in (t,w)$
3. $B \leftarrow A \in (t,w)$ iff **if** $A \in (t,w)$ **then** $B \in (t,w)$
4. $\langle t_i \rangle A \in (t,w)$ iff $A \in (t,f_i(w))$
5. $\Diamond A \in (t,w)$ iff there is an i such that $\langle t_i \rangle A \in (t,w)$

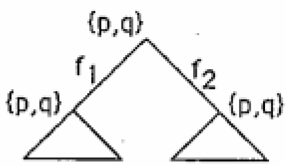
Example. In the tree below $\Diamond p$ and $\langle t_2 \rangle q$ appear in the root, and $\Diamond s$, $\Diamond m$ don't appear in the root :



Definition. We define a function $+: \Gamma \times W \times DC \rightarrow \Gamma$ which gives a new tree $+(t,w,C)$ from t,w and a ground definite clause C. In the following, in order to simplify our notation we will write $(t,w) +C$ instead of $+(t,w,C)$. The function + is defined by the following recursive algorithm :

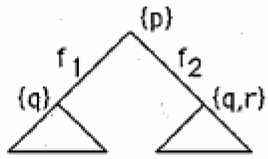
1. $(t,w) +p = t'$ where t' is identical to t, except $m'(w) = m(w) \cup \{p\}$
2. $(t,w)+(A \leftarrow B) =$ **if** $B \in (t,w)$ **then** $(t,w) +A$ **else** t
3. $(t,w) +\langle t_i \rangle A = (t,f_i(w)) + A)$
4. $(t,w) + \Box A = \cup_i ((t,f_i(w)) + A)$

Example. Assume that t has root w and is



then we have $(t,w) + \langle t_2\rangle(r \leftarrow s)$

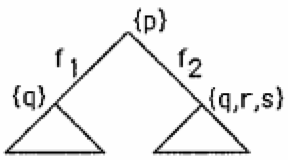
where w is the root



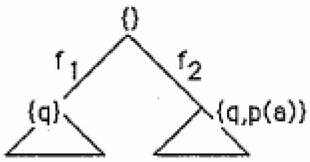
Definition. Now we introduce the transformation T_P. T_P is a mapping $T_P : \Gamma \rightarrow \Gamma$ such that $T_P(t) = \cup C$ $\{(t,w)+C$: C is a ground instance of a clause of P$\}$, where w is the root of t.

Example. Let $H = \{a,b\}$, $P = \{\Box q\ ,\ \langle g(x)\rangle(p(x) \leftarrow q)\}$.

$T_P(\perp)$ is



and $T_P{}^2(\perp)$ is



Proposition 6.1. T_P is continuous.

Given a program P, since T_P is continuous and Γ is a complete lattice, we have that T_P possesses a least fixpoint $lfp(T_P) = \cup_{n \in N} T_P{}^n(\perp)$.

Example. In the example above we have $lfp(T_P) = T_P{}^2(\perp)$.

7 POSSIBLE WORLDS AND DECLARATIVE SEMANTICS

Now we establish the connection between classical modal semantics and declarative semantics.

Fact. Each tree defines a model.

Proof. Given a tree (W,F,m) we put $R = \cup_{fi \in F} \{f_i\}$.

Theorem 7.1 Let P be a program, G a goal, and $\forall P$ the universal closure of P and $\exists G$ the existential closure of G. If $\forall P \rightarrow \exists G$ is valid then there is a ground instance $G\sigma$ of G such that $G\sigma \in (lfp(T_P),w)$, where w is the root of $lfp(T_P)$.

Proof. Assume on the contrary that $G\sigma \notin (lfp(T_P),w)$. Then for any σ there is no model obtained from $lfp(T_P)$ in which $G\sigma$ is satisfied. However $P\sigma$ is satisfiable for every σ because $P\sigma \in (lfp(T_P),w)$. So it is in contradiction with the fact that $\forall P \rightarrow \exists G$ is valid.

8 RESOLUTION COMPLETENESS BY FIXPOINT

In order to prove the completeness of the SLD-relsolution some lemmas and theorems are necessary.

(Upward) Lemma 8.1 Let $A_1,\ldots,A_n,Q_1,\ldots,Q_r$ be definite clauses and G a goal.

(i) If $A_1,\ldots,A_n \vdash G$ then $\Box A_1,\ldots,\Box A_n \vdash \Box G$.

(ii) If $A_1,\ldots,A_n,Q_1,\ldots,Q_r \vdash G$ then

$$\Box A_1,\ldots,\Box A_n,\langle t\rangle Q_1,\ldots,\langle t\rangle Q_r \vdash \langle t\rangle G.$$

Proof. We only prove (ii), the proof of (i) is similar. Let there be k inferences in the derivation $A_1,\ldots,A_n,Q_1,\ldots,Q_r \vdash G$. Each inference comes from a rule $C_i, G_i \Rightarrow G_{i+1}$, for $1 \le i < k$. We replace C_i by $\Box C_i$ if C_i is in $\{A_1,\ldots,A_n\}$ and by $\langle t\rangle C_i$ else. The correctness of the resulting inference is warranted by the $\Box\langle t\rangle$-rule or by the $\langle t\rangle\langle t\rangle$-rule, and thus we have build the required derivation.

Theorem 8.1 Let A be a ground goal. $A \in lfp(T_P)$ implies $P \vdash A$.

Proof. Suppose $A \in lfp(T_P)$. As the sequence $\{T_P{}^n(\bot)\}_n$ is a directed set there exists n such that $A \in T_P{}^n(\bot)$. Adding subformulas of A to the nodes in the tree whenever we do a +-operation it is possible to obtain a tableau-like tree. Now by induction on its depth and using the Upward Lemma 8.1 we can show that $P \vdash A$.

(Lifting) Lemma 8.2 Let P be a program, G a goal and σ a substitution such that Gσ is ground. $P \vdash G$ if $P \vdash G\sigma$.

Proof. The proof is as for the classical case, by transforming each step of the proof $P \vdash G\sigma$ into a new step of the proof $P \vdash G$. This is done by induction on the number of operations which are necessary in each step of the proof.

Theorem 8.2 Let P be a program. Let $lfp(T_P)$ be its fixpoint and w the root of $lfp(T_P)$. If $G\sigma \in (lfp(T_P),w)$ for some ground instance Gσ of G then $P \vdash G$.

Proof. Assume that $G\sigma \in lfp(T_P,w)$, then $P \vdash G\sigma$ by Theorem 8.1 and using the Lifting Lemma we have $P \vdash G$.

9 MODAL SYSTEMS T AND S4

In this section we present schematically the declarative semantics for the systems T and S4 .

9.1 System T. The semantics of T is characterized by models having a reflexive accesibility relation. Its resolution rules are those of Q plus:

T-rules: from $A,G \Rightarrow G_1(\sigma)$ infer $A, \Diamond G \Rightarrow G_1(\sigma)$

from $A,G \Rightarrow G_1(\sigma)$ infer $\Box A, G \Rightarrow G_1(\sigma)$

In what concerns the declarative semantics, we just redefine the following two operations on trees:

5'. $\Diamond A \in (t,w)$ iff $A \in (t,w)$ or there is an i such that $A \in (t,f_i(w))$

4'. $(t,w) + \Box A = \cup_i \{(t,f_i(w)) + A\} \cup \{(t,w)+A\}$

9.2 System S4. Models for S4 have a reflexive and transitive accesibility relation. The resolution rules are those of T,where we replace the □<t>-rule by:

S4-rule: from $\Box A,G \Rightarrow G_1(\sigma)$
infer $\Box A,\langle t\rangle G \Rightarrow \langle t\rangle G_1(\sigma)$

Declarative semantics for S4 is more complexe than that for Q and T. As the accesibility relation is transitive, in order to check if a goal ◊G appears in a node w of a tree we must test if it appears in a descendant of w. The depth of the tree being unbound, in order to remain decidable it is sufficent to consider only descendants whose distance from w is less or equal than $\nu(P)*deg(G)$, where $\nu(P)$ is the cardinal of the set of subformulas of P, and deg(G) is the modal degree of G.

5'. $\Diamond A \in (t,w)$ iff there is $k \geq 0$ and $i_1,...i_k$, such that $A \in (t,(f_{ik}(...(f_{i1}(w)...)))$

4'. $(t,w)+\Box A=\cup_{n\in n} (\cup_{i1,...in} ((t,f_{in}(...f_{i1}(w))+A)))$

10 CONCLUSION

We have defined a declarative semantics that allows us to consider modal Horn clauses as programs using Skolem techniques. This semantics was used to obtain the completeness of a resolution inference mechanism. We consider restricted class of formulas, prenex formulas; however the method can be extended as in (Fariñas and Herzig 1988) to general quantified formulas, in which the □-operator is also transformed using the same techniques, but we must note that the readibility of the modal expressions decreases. Using a closely related methodology Wallen (1987), Ohlbach (1988) and Auffray and Enjalbert (1988) obtained similar results.

Throughout the paper there is a strong analogy with the classical case, where sets of classical Horn clauses are considered as logic programs. The gist of this work is the generalization of the least Herbrand model to the modal case. It was achieved considering each model as a tree whose nodes are classical Herbrand model. The associated complete lattice is obtained by a generalization of the classical Herbrand interpretation. Every nice classical property of classical Horn clauses can be extended to the modal case. Resolution rules, declarative semantics, and the completeness proof have been given only for the modal system Q, in order to facilitate understanding. For the multi-modal versions of Q, T or S4 and systems with further restrictions on the accessibility relation, the methodology given in this paper can be used.

11 ACKNOWLEDGMENTS

We thank P. Bieber and S. Passy for their helpful criticisms.

12 REFERENCES

Apt Kr., Introduction to Logic Programming. Texas University Reports, TR-87-35, 1987.

Akama S., A Proposal of Modal Logic Programming. Procc. of 6th Canadian Artf. Intell. Conf. 99-102 Montreal 1986.

Abadi M., Manna Z., Temporal Logic Programming. 4th IEEE Symposium on Logic Programming, San Francisco, Sept. 87.

Auffray Y., Enjalbert P., Demonstration de théorèmes en logique modale: un point de vue équationnel. JELIA,Roscoff, 1988.

Balbiani et al., Molog User Manual. Report Université Paul Sabatier Toulouse, May 1987.

Balbiani Ph., Fariñas del Cerro L., Herzig A., Theoretical Foundations for Modal Logic Programs. Report Université Paul Sabatier, June 1987.

Blair, H., Subrahmanian, V.S., Paraconsistent Logic Programming (Draft). Syracuse University Reports, 1987.

Chan, M., The Recursive Method for Modal Logic. New Generation Computing, 5 (1987), 155-183.

Colmerauer A. et al., Un système de communication homme-machine en Français. Report GRIA, Marseille 1973.

Conference on Theoretical Aspects of Reasoning about Knowledge (Halpern J.Y. ed.), Morgan Kaufmann, Los Altos, 1986.

Fariñas del Cerro L., Molog: A System that Extends Prolog with Modal Logic. The New Generation Computer Journal, Feb.1986.

Fariñas del Cerro L., Herzig A., Automated Quantified Modal Logics. Machine Learning Meta-Reasoning and Logics, (Ed. Brazdil and Konologe), Kluwer Acad. Publ., to appear.

Fariñas del Cerro L., Penttonen M., A Note on the Complexity of the Satisfiability of Modal Horn Clauses. J. of Logic Programming, March 1987.

Fitting M., Logic Programming on a Topological Bilattice. Fundamenta Informaticae XI (1988), 209-218.

Gabbay D.M., Reyle U., N-Prolog: An Extension of Prolog with Hypothetical Implications. I, Journal of Logic Programming 1:319-355 (1984).

Hughes G.E., Cresswell M.J., A Companion to Modal Logic. Methuen &Co. Ltd., London 1986.

Konolige K., A Deductive Model of Belief. Morgan Kaufmann, Los Altos, 1986.

Lloyd J.M., Foundations of Logic Programming. Springer-Verlag, 1984.

Lassez J.L., Maher M.J., Marriott K., Unification Revisited. Foundations of Deductive Databases and Logic Programming, (Minker J. ed.), Morgan Kaufmann, Los Altos, 1987.

Miller D., A logical analysis of modules in logic. Technical Report MS-CIS-87-96, University of Pennsylvania, October 1987.

Nute D., LDR, A logic for defeasible reasoning, ACM Research Report University of Georgia, 1986.

Ohlbach, H.J., A resolution calculus for modal logics. Proc. of the9th Int. Conf. on Automated Deduction, (Lusk E., Overbeek R. eds.), Springer Verlag, May 1988.

Okada M., Mathematical Basis of Modal Logic Programming. JELIA, Roscoff, 1988.

Subrahmanian V.S., On the Semantics of Quantitative Logic Programs. 4th IEEE Symposium on Logic Programming, San Francisco, Sep. 1987.

Van Emden M., Quantitative Deduction and its Fixpoint Theory. J. of Logic Programming, 4, 1, pps 37-53.

Wallen L., Matrix Proof Methods for Modal Logics. IJCAI 1987.

Yiang Y.L., Intension, Quantified Beliefs and Epistemic Resolution. Third Int. Conf. on Methodology for Intelligent Systems ICMIS, Torino, Italy, 1988.