An SLD-Resolution Calculus for Basic Serial Multimodal Logics

Linh Anh Nguyen

Institute of Informatics, University of Warsaw
ul. Banacha 2, 02-097 Warsaw, Poland
nguyen@mimuw.edu.pl

ICTAC'2005

Why Modal Logic Programming?

- Classical logic programming is very useful in practice.

- Multimodal logics are useful in many areas of computer science.

  For example, multimodal logics are used in knowledge representation and multi-agent systems by interpreting $\Box_i\varphi$ as "agent i knows/believes that φ is true".

  As another example, description logics (cousins of multimodal logics) can be used to reason about concepts, objects, and their relationships.

History of Modal Logic Programming

The translational approach translates modal logic programs to classical logic programs.

– Debart, Enjalbert, and Lescot, 1992: functional translation,
multimodal logics using axioms D, T, 4, F, I.

– Nonnengart, 1994: semi-functional translation,
basic serial monomodal logics.

The direct approach directly uses modalities.

– Balbiani, Fariñas del Cerro, and Herzig, 1988: KD, T, $S4$,
$\Box$ is not allowed to occur in bodies of program clauses and goals.

– Baldoni, Giordano, and Martelli, 1996: grammar logics,
existential modal operators are disallowed in programs and goals.

– Nguyen, 2003: serial/almost serial monomodal logics
no restrictions

Results of this Work

an SLD-resolution calculus for basic serial multimodal logics, which are parameterized by an arbitrary combination of generalized versions of axioms T, B, 4, 5, and I.

The calculus

- uses the direct approach,
- assumes no special restrictions for the form of programs and goals,
- is accompanied with fixpoint semantics and least model semantics,
- is sound and complete.

Outline

- Basic Serial Multimodal Logics
- Modal Logic Programs
- Labeled Modal Operators
- Model Generators
- Fixpoint Semantics
- SLD-Resolution

First-Order Multimodal Logics

Language: the language of predicate logic plus m pairs of modal operators $\Box_i$ and $\Diamond_i$.

Semantics: Kripke semantics with fixed-domain and rigid terms

Kripke model: $\langle D, W, \tau, R_1, \ldots, R_m, \pi \rangle$

D : the domain,

W : the set of possible worlds,

$\tau \in W$: the actual world,

$R_1, \ldots, R_m \subseteq W \times W$: the accessibility relations,

π : the interpretation of symbols.

Satisfaction relation:

$M, V, w \models \Box_i \varphi$ iff $\forall v \in W.\ R_i(w, v) \rightarrow M, V, v \models \varphi$

. . .

$M \models \varphi$ iff $M, V, \tau \models \varphi$ for every variable assignment V

Basic Serial Multimodal (BSMM) Logics

A $BSMM$ logic is parameterized by relations AD, AT, AI, AB, $A4$, $A5$, where AD is full. These relations specify the following axioms:

Axiom		Corresponding Frame Restriction
$\Box_i\varphi \rightarrow \Diamond_i\varphi$	if $AD(i)$	$\forall u\ \exists v\ R_i(u,v)$
$\Box_i\varphi \rightarrow \varphi$	if $AT(i)$	$\forall u\ R_i(u,u)$
$\Box_i\varphi \rightarrow \Box_j\varphi$	if $AI(i,j)$	$R_j \subseteq R_i$
$\varphi \rightarrow \Box_i\Diamond_j\varphi$	if $AB(i,j)$	$\forall u,v\ (R_i(u,v) \rightarrow R_j(v,u))$
$\Box_i\varphi \rightarrow \Box_j\Box_k\varphi$	if $A4(i,j,k)$	$\forall u,v,w\ (R_j(u,v) \wedge R_k(v,w) \rightarrow R_i(u,w))$
$\Diamond_i\varphi \rightarrow \Box_j\Diamond_k\varphi$	if $A5(i,j,k)$	$\forall u,v,w\ (R_i(u,v) \wedge R_j(u,w) \rightarrow R_k(w,v))$

The MProlog Language

We use
- $\boxdot$ to denote a universal modality (a seq. of univ. modal operators),
- E to denote a classical atom,
- A, B to denote simple atoms of the form E, $\Box_i E$, or $\Diamond_i E$,
- $\boxdot(\varphi \leftarrow \psi_1, \ldots, \psi_n)$ to denote the formula $\forall(\boxdot(\varphi \vee \neg\psi_1 \ldots \vee \neg\psi_n))$.

A program clause is a formula of the form $\boxdot(A \leftarrow B_1, \ldots, B_n)$.

An MProlog program is a finite set of program clauses.

An MProlog goal atom is a formula of the form $\boxdot E$ or $\boxdot\Diamond_i E$.

An MProlog goal is a formula written in the clausal form $\leftarrow \alpha_1, \ldots, \alpha_k$, where each α_i is an MProlog goal atom.

Correct Answers

Let P be an MProlog program
and $G = \leftarrow \alpha_1, \ldots, \alpha_k$ be an MProlog goal.

An answer θ for $P \cup \{G\}$ is a substitution whose domain is a set of variables of G.

We say that θ is a correct answer in L for $P \cup \{G\}$ if θ is an answer for $P \cup \{G\}$ and

$$P \models_L \forall((\alpha_1 \wedge \ldots \wedge \alpha_k)\theta)$$

Labeled Modal Operators

The forward labeled form of $\Diamond_i E$ is $\langle E \rangle_i E$, where E is a classical atom.

The intuition is that: To satisfy a formula $\Diamond_i E$ at a world w we connect w to a new world u via R_i and add E to the content of u. The world u can be denoted by the composition of w and $\langle E \rangle_i$.

The backward labeled form of $\Diamond_i E$ is $\langle X \rangle_i E$, where X is a fresh variable for classical atoms.

We cannot replace $\leftarrow \Diamond_i(A \wedge B)$ by $\leftarrow \Diamond_i A, \Diamond_i B$.
But we can break $\leftarrow \langle S \rangle_i(A \wedge B)$ into $\leftarrow \langle S \rangle_i A, \langle S \rangle_i B$.

Semantics of Labeled Modal Operators

Denote $EdgeLabels = \{\langle E\rangle_i \mid E \in \mathcal{B} \cup \{\top\} \text{ and } 1 \leq i \leq m\}$,
where $\mathcal{B}$ is the Herbrand base.

Let $M = \langle D, W, \tau, R_1, \ldots, R_m, \pi\rangle$ be a Kripke model.
A $\Diamond$-realization function on M is a partial function

$$\sigma : W \times EdgeLabels \rightarrow W$$

such that if $\sigma(w, \langle E\rangle_i) = u$ then $R_i(w, u)$ holds and $M, u \models E$.

$M, \sigma, w \models \langle E\rangle_i\psi$ iff $\sigma(w, \langle E\rangle_i)$ is defined and $M, \sigma, \sigma(w, \langle E\rangle_i) \models \psi$.

For a set I of ground atoms, we write $M \models I$ if $M, \sigma, \tau \models I$ for *some* σ.

Definitions

A modality is a sequence of modal operators (which may be labeled). A modality is in labeled form if it does not contain unlabeled existential modal operators.

We use $\triangle$ to denote a modality and E to denote a classical atom.

An atom is a formula of the form $\triangle E$. It is in labeled form if $\triangle$ is in labeled form.

An atom is in almost labeled form if it is of the form $\triangle A$ with $\triangle$ in labeled form and A being a simple atom of the form E, $\Box_i E$, $\Diamond_i E$, or $\langle S \rangle_i E$.

Model Generators

A model generator is a set of ground atoms not containing $\Diamond_i$, $\langle\top\rangle_i$, $\top$.

A model generator I represents an L-model (see the paper for the construction) which is called the *standard L-model of I* and has the property that:

The standard L-model of a model generator I is a least L-model of I.

(A model M is *less than or equal to* a model M' if for every positive ground formula φ without labeled operators, if $M \models \varphi$ then $M' \models \varphi$.)

Fixpoint Semantics of MProlog Programs (1)

Fixpoint semantics of *L*-MProlog programs are defined using a direct consequence operator denoted by $T_{L,P}$.

Given a model generator I, how can $T_{L,P}(I)$ be defined?

- Basing on the axioms of *L*, I is first extended to the *L*-saturation of I, denoted by $Sat_L(I)$, which is a set of atoms.
- Then *L*-instances of program clauses of P are applied to the atoms of $Sat_L(I)$. This is done by the operator $T_0{}_{L,P}$.
- $T_{L,P}(I)$ is defined as $T_0{}_{L,P}(Sat_L(I))$.

Fixpoint Semantics of MProlog Programs (2)

The saturation operator Sat_L is specified by the following rules:

$$\triangle\langle E\rangle_i\alpha \rightarrow \triangle\Diamond_i\alpha \quad (1)$$
$$\triangle\Box_i\alpha \rightarrow \triangle\Diamond_i\alpha \quad (2)$$
$$\triangle\Box_i\alpha \rightarrow \triangle\alpha \text{ if } AT(i) \quad (3)$$
$$\triangle\alpha \rightarrow \triangle\Diamond_i\alpha \text{ if } AT(i) \quad (4)$$
$$\triangle\Box_i\alpha \rightarrow \triangle\Box_j\alpha \text{ if } AI(i,j) \quad (5)$$
$$\triangle\Diamond_j\alpha \rightarrow \triangle\Diamond_i\alpha \text{ if } AI(i,j) \quad (6)$$
$$\triangle\alpha \rightarrow \triangle\Box_i\Diamond_j\alpha \text{ if } AB(i,j) \quad (7)$$
$$\triangle\Diamond_i\Box_j\alpha \rightarrow \triangle\alpha \text{ if } AB(i,j) \quad (8)$$
$$\triangle\Box_i\alpha \rightarrow \triangle\Box_j\Box_k\alpha \text{ if } A4(i,j,k) \quad (9)$$
$$\triangle\Diamond_j\Diamond_k\alpha \rightarrow \triangle\Diamond_i\alpha \text{ if } A4(i,j,k) \quad (10)$$
$$\triangle\Diamond_i\alpha \rightarrow \triangle\Box_j\Diamond_k\alpha \text{ if } A5(i,j,k) \quad (11)$$
$$\triangle\Diamond_j\Box_k\alpha \rightarrow \triangle\Box_i\alpha \text{ if } A5(i,j,k) \quad (12)$$

Given a model generator I, $Sat_L(I)$ is the least extension of I that contains all ground atoms in almost labeled form that are derivable from some atom in I using the rules specifying Sat_L.

Fixpoint Semantics of MProlog Programs (3)

For defining the operator $T_{0L,P}$, we need some definitions:

Define that $\Diamond_i \preceq \langle S \rangle_i \preceq \Box_i$. This order can be extended for comparing modalities and we can specify when an atom is an instance of another atom.

Let $\boxdot$ and $\boxdot'$ be universal modalities. We say that $\boxdot'$ is an *L*-context instance of $\boxdot$ if $\boxdot\varphi \rightarrow \boxdot'\varphi$ is *L*-valid for any formula φ.

Let φ and φ' be program clauses with empty modal context. We say that $\boxdot\varphi$ is an *L*-instance of a program clause $\boxdot'\varphi'$ if $\boxdot$ is an *L*-context instance of $\boxdot'$ and there exists a substitution θ such that $\varphi = \varphi'\theta$.

The forward labeled form of an atom α is the atom α' such that if α is of the form $\triangle\Diamond_i E$ then $\alpha' = \triangle\langle E \rangle_i E$, else $\alpha' = \alpha$.

Fixpoint Semantics of MProlog Programs (4)

Let I be a set of ground atoms in almost labeled form.

$T_{0L,P}(I)$ is defined to be the least set of atoms such that, if

- $\boxdot(A \leftarrow B_1, \ldots, B_n)$ is a ground instance of a program clause of P

- and $\triangle$ is a maximally general ground modality in labeled form s.t.

    * $\triangle$ is an L-instance of $\boxdot$

    * and $\triangle B_i$ is an instance of some atom of I (for every $1 \leq i \leq n$),

then the forward labeled form of $\triangle A$ belongs to $T_{0L,P}(I)$,

Fixpoint Semantics of MProlog Programs (5)

Recall that $T_{L,P}(I)$ is defined as $T_{0L,P}(Sat_L(I))$.

This operator $T_{L,P}$ is monotonic, continuous and has the least fixpoint

$$T_{L,P}\uparrow\omega = \bigcup_{n=0}^{\omega} T_{L,P}\uparrow n$$

where $T_{L,P}\uparrow 0 = \emptyset$ and $T_{L,P}\uparrow n = T_{L,P}(T_{L,P}\uparrow(n-1))$ for $n > 0$.

It is proved in another paper that

The standard *L*-model of $T_{L,P}\uparrow\omega$ is a least *L*-model of P.

Example

Consider the multimodal logic L specified by $m = 2$, $AD = \{1, 2\}$, $AT = \{1\}$, $AI = \{(2, 1)\}$, and $AB = A4 = A5 = \emptyset$. In other words, the logic is characterized by the axioms: $\Box_1\varphi \rightarrow \Diamond_1\varphi$; $\Box_2\varphi \rightarrow \Diamond_2\varphi$; $\Box_1\varphi \rightarrow \varphi$; and $\Box_2\varphi \rightarrow \Box_1\varphi$. Consider the following program P:

$$\varphi_1 = \Diamond_2 p(a) \leftarrow$$
$$\varphi_2 = \Box_2(\Box_1 q(x) \leftarrow \Diamond_2 p(x))$$
$$\varphi_3 = \Box_2(r(x) \leftarrow p(x), q(x))$$

We have $T_{L,P}\uparrow 1 = \{\langle p(a)\rangle_2 p(a)\}$ and

$$Sat_L(T_{L,P}\uparrow 1) = \{\langle p(a)\rangle_2 p(a), \langle p(a)\rangle_2 \Diamond_1 p(a), \langle p(a)\rangle_2 \Diamond_2 p(a)\}$$

Applying φ_2 and its L-instance $\Box_1 q(x) \leftarrow \Diamond_2 p(x)$ to $Sat_L(T_{L,P}\uparrow 1)$, we obtain $T_{L,P}\uparrow 2 = T_{L,P}\uparrow 1 \cup \{\langle p(a)\rangle_2 \Box_1 q(a),\ \Box_1 q(a)\}$.

The set $Sat_L(T_{L,P}\uparrow 2)$ contains both $\langle p(a)\rangle_2 p(a)$ and $\langle p(a)\rangle_2 q(a)$.

Applying φ_3, we have $\langle p(a)\rangle_2 r(a) \in T_{L,P}\uparrow 3$ and arrive at

$$T_{L,P}\uparrow\omega = T_{L,P}\uparrow 3 = \{\langle p(a)\rangle_2 p(a), \langle p(a)\rangle_2 \Box_1 q(a), \Box_1 q(a), \langle p(a)\rangle_2 r(a)\}$$

SLD-Resolution for MProlog

A goal is a clause of the form $\leftarrow \alpha_1, \ldots, \alpha_k$, where α_i are atoms.

SLD-derivation is defined using two kinds of steps:

- resolving a goal with a program clause,
- resolving a goal with an $rSat_L$ rule.

Then SLD-refutation and computed answer are defined as usual.

Resolving a Goal with a Program Clause

Let $G = \leftarrow \alpha_1, \ldots, \alpha_i, \ldots, \alpha_k$ be a goal and $\varphi = \boxdot(A \leftarrow B_1, \ldots, B_n)$ a program clause. Then G' is derived from G and φ in L using mgu θ, and called an *L-resolvent* of G and φ, if the following conditions hold:

- $\alpha_i = \triangle' A'$, with $\triangle'$ in labeled form, is called the *selected atom*.
- $\triangle'$ is an L-instance of $\boxdot'$ which is an L-context instance of $\boxdot$.
- θ is an mgu of A' and the forward labeled form of A.
- G' is the goal $\leftarrow (\alpha_1, \ldots, \alpha_{i-1}, \triangle' B_1, \ldots, \triangle' B_n, \alpha_{i+1}, \ldots, \alpha_k)\theta$.

Rules as Meta-Clauses

$rSat_L$ (reverse saturation) rules:

(1) $\triangle\Diamond_i\alpha \leftarrow \triangle\langle X\rangle_i\alpha$ where X is a fresh atom variable

(2) $\triangle\nabla_i\alpha \leftarrow \triangle\Box_i\alpha$ where ∇_i is a modal operator with subscript i

(3)..(12) a rule $\alpha \leftarrow \beta$ for each k-th rule $\beta \rightarrow \alpha$ specifying Sat_L, $3 \leq k \leq 12$, with the same accompanying condition.

Resolving a goal with an $rSat_L$ rule is similar to deriving a resolvent in classical logic programming.

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2 (r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2(r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$
$\leftarrow \langle p(a) \rangle_2 q(a)$	$\Diamond_2 p(a) \leftarrow$	$\{X/p(a), x/a\}$

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2 (r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$
$\leftarrow \langle p(a) \rangle_2 q(a)$	$\Diamond_2 p(a) \leftarrow$	$\{X/p(a), x/a\}$
$\leftarrow \langle p(a) \rangle_2 \Box_1 q(a)$	$\triangle \alpha \leftarrow \triangle \Box_1 \alpha$	

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2(r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$
$\leftarrow \langle p(a) \rangle_2 q(a)$	$\Diamond_2 p(a) \leftarrow$	$\{X/p(a), x/a\}$
$\leftarrow \langle p(a) \rangle_2 \Box_1 q(a)$	$\triangle \alpha \leftarrow \triangle \Box_1 \alpha$	
$\leftarrow \langle p(a) \rangle_2 \Diamond_2 p(a)$	$\Box_2(\Box_1 q(x) \leftarrow \Diamond_2 p(x))$	$\{x_5/a\}$

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2(r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$
$\leftarrow \langle p(a) \rangle_2 q(a)$	$\Diamond_2 p(a) \leftarrow$	$\{X/p(a), x/a\}$
$\leftarrow \langle p(a) \rangle_2 \Box_1 q(a)$	$\triangle \alpha \leftarrow \triangle \Box_1 \alpha$	
$\leftarrow \langle p(a) \rangle_2 \Diamond_2 p(a)$	$\Box_2(\Box_1 q(x) \leftarrow \Diamond_2 p(x))$	$\{x_5/a\}$
$\leftarrow \langle p(a) \rangle_2 \Diamond_1 p(a)$	$\triangle \Diamond_2 \alpha \leftarrow \triangle \Diamond_1 \alpha$	

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2(r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$
$\leftarrow \langle p(a) \rangle_2 q(a)$	$\Diamond_2 p(a) \leftarrow$	$\{X/p(a), x/a\}$
$\leftarrow \langle p(a) \rangle_2 \Box_1 q(a)$	$\triangle \alpha \leftarrow \triangle \Box_1 \alpha$	
$\leftarrow \langle p(a) \rangle_2 \Diamond_2 p(a)$	$\Box_2(\Box_1 q(x) \leftarrow \Diamond_2 p(x))$	$\{x_5/a\}$
$\leftarrow \langle p(a) \rangle_2 \Diamond_1 p(a)$	$\triangle \Diamond_2 \alpha \leftarrow \triangle \Diamond_1 \alpha$	
$\leftarrow \langle p(a) \rangle_2 p(a)$	$\triangle \Diamond_1 \alpha \leftarrow \triangle \alpha$	

Example

Reconsider the modal logic L and the program P given earlier. Let $G = \leftarrow \Diamond_2 r(x)$. We give an SLD-refutation of $P \cup \{G\}$ in L with computed answer $\{x/a\}$.

Goals	Input clauses/rules	MGUs
$\leftarrow \Diamond_2 r(x)$		
$\leftarrow \langle X \rangle_2 r(x)$	$\triangle \Diamond_i \alpha \leftarrow \triangle \langle X \rangle_i \alpha$	
$\leftarrow \langle X \rangle_2 p(x), \langle X \rangle_2 q(x)$	$\Box_2(r(x) \leftarrow p(x), q(x))$	$\{x_2/x\}$
$\leftarrow \langle p(a) \rangle_2 q(a)$	$\Diamond_2 p(a) \leftarrow$	$\{X/p(a), x/a\}$
$\leftarrow \langle p(a) \rangle_2 \Box_1 q(a)$	$\triangle \alpha \leftarrow \triangle \Box_1 \alpha$	
$\leftarrow \langle p(a) \rangle_2 \Diamond_2 p(a)$	$\Box_2(\Box_1 q(x) \leftarrow \Diamond_2 p(x))$	$\{x_5/a\}$
$\leftarrow \langle p(a) \rangle_2 \Diamond_1 p(a)$	$\triangle \Diamond_2 \alpha \leftarrow \triangle \Diamond_1 \alpha$	
$\leftarrow \langle p(a) \rangle_2 p(a)$	$\triangle \Diamond_1 \alpha \leftarrow \triangle \alpha$	
the empty clause	$\Diamond_2 p(a) \leftarrow$	

Soundness and Completeness of SLD-Resolution

The presented SLD-resolution calculus is sound and complete.

That is:

Let L be a BSMM logic, P an MProlog program, G an MProlog goal.

Then every computed answer in L of $P \cup \{G\}$ is a correct answer in L of $P \cup \{G\}$.

Conversely, for every correct answer θ in L of $P \cup \{G\}$, there exists a computed answer γ in L of $P \cup \{G\}$ which is more general than θ.

Conclusions

We have successfully applied the direct approach for modal logic programming in a large class of basic multimodal logics, while not assuming any special restriction on the form of logic programs and goals.

The class BSMM is large and not studied before for MLP.

The clarity of the $Sat_L/rSat_L$ rules suggests that our methods can be extended for other multimodal logics.

The given SLD-resolution calculus for MProlog is not very efficient. We have implemented MProlog and used optimizations for more specific (multi)modal logics.