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

KNOWLEDGE REPRESENTATION AND INFERENCE BASED ON FIRST-ORDER MODAL LOGIC

Koji Iwanuma and Masateru Harao

Dept. of Information Engineering, Yamagata University
3-16, Zyounan 4-chome, Yonezawa-shi, Yamagata 992, Japan

ABSTRACT

In this paper, we present a knowledge representation system based on a first-order modal logic and give its deductive inference system. The modal logic which we proposed here is a kind of first-order dynamic logic, and is suitable for representing both structured knowledge and metaknowledge. A possible-world model used in modal logic can be regarded as structured knowledge, and modal operators can be regarded as operators to describe various kinds of properties on a possible-world model. In order to describe knowledge structure effectively and compactly, we introduce a concept, "viewpoints of modalities". We also show that schema formulas available in this framework are useful for the description of metaknowledge such as property inheritance. According to this idea, we construct a knowledge representation system based on a subset of our first-order modal logic whose syntax is restricted to definite modal clauses, and give a complete deductive system which is as effective as SLD resolution. This fomalization offers a theoretical framework to a graph type structured knowledge like a frame system.

1 INTRODUCTION

One of the most important problems in artificial intelligence is to design a knowledge representation language in which knowledge structure and metaknowledge can be described effectively. Until now, lots of knowledge representation systems have been proposed. Roughly speaking, we can divide them into three classes: production systems; frame systems; and first-order logical systems.

A production system treats only the "if-then-else" type of procedural knowledge, therefore, the inference on a production system is simple and can be done very effectively. But its ability for knowledge representation is limited to some extent, since it is difficult to deal with complex structured knowledge and metaknowledge.

A frame system has very flexible and general expressive power. In a frame system, a knowledge structure is described as a graph where a node corresponds to a set of knowledge. The inference on a frame system can be regarded as a search over this graph; therefore, its fast processing can be available. But we should note that a frame system, as proposed by Minsky, is fundamentally a programming paradigm. Unfortunately, it has no sufficient theoretical basis, though lots of efforts have been made until now. The semantics of knowledge and the concept of inference in a frame system are not so clear.

A system based on first-order logic has a fundamental mathematical theory and clear semantics. In these days, a lot of knowledge representation systems have been developed by this approach. But, historically, first-order logic was developed in order to describe mathematics. Therefore, it is suitable for representing properties which are invariable over time and space, but it has no convenient mechanism for describing the structure and the hierarchy of knowledge. This causes inconvenience for practical knowledge representation. In order to design a knowledge representation language improving on this defect, it is necessary to introduce a mechanism of treating both structured knowledge and metaknowledge to first-order logic.

The structure of knowledge can be regarded as a kind of space. We have proposed a first-order modal logic [Iwanuma and Harao 1987] which can treat both temporal and spatial modalities. Our logic is based on first-order dynamic logic [Fischer and Ladner 1979, Harel 1979, 1984], and has some useful mechanisms, which first-order logic alone does not possess. In this paper, we will extend a knowledge representation system based on first-order logic by using a first-order spatial modal logic, which is a subsystem of our logic. At first, we show that a possible-world model can be regarded as structured knowledge and that modal operators are very useful in describing knowledge structures. We introduce a new concept "viewpoints of modalities", which corresponds to a target program of dynamic logic. This makes it easier to describe various kinds of knowledge structures. Also, we show the usefulness of schema formulas for representing such metaknowledge as property inheritance. Next, we contemplate some properties of a first-order modal logic in order to establish a effective inference rule. In accord with these considerations, we construct a knowledge representation system which is based on our spatial modal Horn logic, and give a complete deductive method which is as effective as SLD resolution.

This paper is organized as follows: In Chapter 2, we show the usefulness of: a possible-word model; viewpoints of modalities; and schema formulas. In Chapter 3, at first, we discuss some problems to establish a effective inference system to a first-order modal logic. Next, we give the formal definition of our knowledge representation system. In Chapter 4, we give a deductive inference rule, and show its completeness and soundness. Also we show some examples of question-answering computations of an inference program in Prolog. In Chapter 5, we compare our system with other knowledge representation systems based on modal logic. Chapter 6 is the conclusion.

2 POSSIBLE-WORLD MODELS AND SCHEMA FORMULAS

2.1 Possible-World Models and Structured Knowledge

A modal logic is a logic for treating two kinds of modalities: "necessity" and "possibility". In this paper, we introduce a concept, "viewpoints of modalities", and investigate its applicability to knowledge representation. Viewpoints of modalities correspond to target programs of dynamic logic. Necessity and possibility under the viewpoint 'a' are expressed by the modal operators [a] and <a>, respectively, and these have the following meanings:

[a]p <==> under the viewpoint 'a', it is inevitable that p is true.
<a>p <==> under the viewpoint 'a', it is possible that p is true.

Modal formulas are interpreted in a possible-world model, which takes the form of a directed graph. One example of possible-world models is shown in Fig.1, where each node is called a possible world. Formally, a possible world is an assignment of truth values to modal formulas. In Fig.1, P is true in the world W_1 and both P and Q are true in the world W_2. Directed arcs represent relations, called accessibility relations. The labels of these arcs are names of these accessibilities, that is, they represent viewpoints of modalities. We say the world w is accessible from the world y under the viewpoint 'a' if there is a direct arc labeled by 'a' from y to w. In Fig.1, the worlds W_1 and W_2 are accessible from the world W_0 under the viewpoint 'A', and W_2 and W_3 are accessible from W_0 under the viewpoint 'B'.

The truth values of modal formulas are defined in the possible worlds. We write w |= p if the formula p is true in the world w. The truth values of the formulas [a]p and <a>p are defined respectively as follows:

w |= [a]p <==> p is true in all worlds which are accessible from w under the viewpoint 'a'.
w |= <a>p <==> there is a world y such that y is accessible form w under the viewpoint 'a' and p is true in y.

In Fig.1, the formulas [A]P, [B]Q, <A>P and <B>P are true in the world W_0. On the other hand, [A]Q and [B]P are false in W_0.

A possible-world model with viewpoints of modalities resembles a frame system. A possible world can be regarded as a module of knowledge which corresponds to the concept of a frame. The accessibility relations and the viewpoints of modalities correspond to connections between frames and names of these connections, respectively. Therefore, a possible-world model can be regarded as a frame system, where slot values are represented by logical formulas. In addition, inference rules of modal logic can be regarded as those of a frame system. The computation of the truth value of a formula including modal operators involves a search over a possible-world model, where the viewpoints in these operators represent search paths in the model.

We show an example of mechanical theorem proving on a possible-world model, which can be considered as a question-answering computation

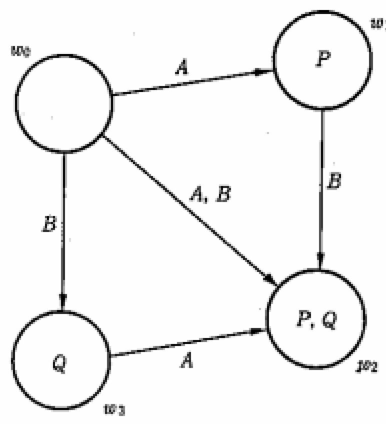

Fig. 1 Possible-world model.

for a frame system. Consider the possible-world model in Fig.2. All of the formulas holding in this model have the forms of definite clauses, so inference like SLD resolution is possible. At first, we ask "How many legs does taro have ?" to the possible world (that is, the frame) "animal" by using the following question clause:

?- animal !! number_legs(taro,y).

The underlined body of this clause can be unified with the head of the following clauses in "animal":

number_legs(x,y)<--<ako$^-$>num_legs(x,y).

After the unification, the body of the above clause changes to "<ako$^-$>num_legs(taro,y)", which should be tested at the next step. This unified body is the procedure which examines whether "num_legs(taro,y)" holds in some world which is accessible from "animal" under the viewpoint "ako$^-$". "ako$^-$" is the converse viewpoint of "ako", that is, the direction of accessibility of "ako$^-$ is the opposite one of "ako". The clause

num_legs(x,4)<--<isa$^-$>monkey(x)

holds in the world "monkey" which is accessible from "animal" under "ako$^-$", and the head can be unified with "num_legs(taro,y)". Therefore, the next step is to determine the truth value of "<isa$^-$>monkey(taro)". It is enough to examine whether "monkey(taro)" holds in some frame which is accessible from "monkey" under the viewpoint "isa$^-$". In the model, the following clause holds in "taro".

monkey(taro) <--

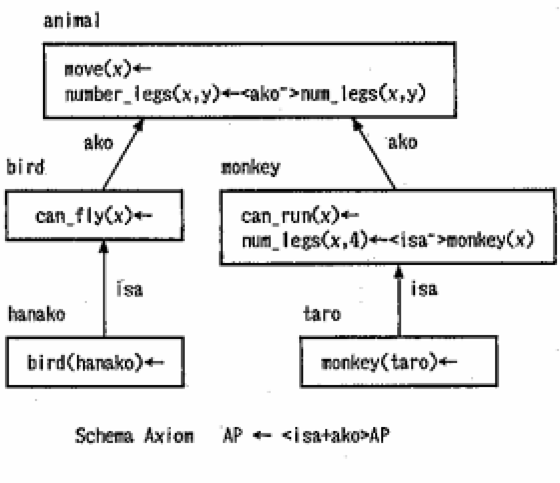

Fig. 2 Possible-world model.

The further computation is not necessary, because this is a unit clause. Thus, the computation is finished. Note it is possible to give the answer "y = 4" to the first question, by using the unifiers constructed in this computation.

In general, it is very difficult and (has not yet been achieved) to furnish an effective inference system for full first-order modal logic; but, we think, it is possible for that subsystem all of whose modal formulas take the forms of definite clauses.

2.2 Schema formulas and Meta Knowledge

Metaknowledge is knowledge which deals with knowledge. Therefore, if knowledge is expressed by a formula at a certain level of fomalization, then its metaknowledge should be expressed by a higher-level logical formula. Higher-order can deals with different level formulas uniformly, thus it is a very natural framework for treating metaknowledge. But, unfortunately, it is very difficult to construct its symbolic computation rules. For example, it is impossible to construct any complete axiom systems [Miller 1983]; also even the unification problem of second-order formulas is not decidable [Goldfarb 1981].

In this paper, we use schema formulas to describe metaknowledge instead of higher-order formulas. Schema formulas are regarded as the knowledge for formulas, that is, metaknowledge. Therefore, a knowledge representation system based on first-order logic acquires the ability for treating some metaknowledge, by introducing a framework for treating schema formulas. In order to treat schema formulas, we use three kinds of syntactical variables: function, predicate and atom variables. These variables express symbols which are names of functions, predicates or atoms. Note they don't directly express functions or predicates themselves. They are used only for syntactical matching between the sequences of alphabets.

We give an example to show the usefulness of schema formulas. Consider the following axiom schema in Fig.2.

AP <-- <isa+ako>AP

This formula implies that the property AP must hold on the frame w_0 if AP holds in a frame which is accessible from w_0 under the viewpoints "isa" or "ako". Therefore, this expresses the property of inheritance. For example, in Fig.2, the formula "move(x)" is false in the frame "bird", but is true in the frame "animal" which is accessible from "bird" under "ako". If the above schema formula is assumed to be an axiom, then "move(x)" should hold in "bird"; that is, "bird" should inherit "move(x)" from "animal".

. Mechanical theorem proving with schema formulas can also be easily established. The following inference is possible for the question "how many legs does taro have ?" to the frame "taro". At first, the question clause is given as follows:

?- taro || num_legs(taro,y)

The body, which is underlined, can be unified with the head of the above axiom schema. After the unification, the body of the axiom schema changes into

<isa+ako>num_legs(taro,y).

The next step is to examine this body, which is true if "num_legs(taro,y)" holds in a frame which is accessible from "taro" under "isa" or "ako". In the frame "monkey", there is the clause

num_legs(x,4)<--<isa⁻>monkey(x),

whose head can be unified with the above body. The unfinished tasks of this inference are the same as those of Section 2.1. As a result, this inference will succeed, and the unifiers through this computation will give the answer "y=4" to the first question.

3. A KNOWLEDGE REPRESENTATION SYSTEM BASED ON A MODAL LOGIC

3.1 System Architecture of a Knowledge Representation Language

In this chapter, we construct a knowledge representation system based on a first-order modal logic. But, it is very difficult to determine the truth values of modal formulas. For example, in principle, it is impossible to construct complete deductive systems for either first-order dynamic logic [Harel 1979, 1984] and first-order temporal logic with equality, ( ) (next) and <> (eventual) operators [Abadi and Manna 1986a, Iwanuma and Harao 1987]. Also, many researchers have investigated the applicability of the resolution principle to various kinds of first-order modal logic [Abadi and Manna 1986a, 1984b, Ventatesh 1985]; but until now, no fruitful result has been achieved.

Considering these difficulties, some restrictions to the base logic of our system are necessary in order to establish effective inference rules. A simple but useful idea is to restrict the syntax of formulas to definite (modal) clauses. This makes it possible to construct an effective inference rule. We can also observe that complex modal formulas aren't so meaningful in practical knowledge presentation. One requirement of our system is to have the ability for the natural description of structured knowledge, so we introduce a framework for describing knowledge structure, that is, finite skeleton of a possible-world model which is defined by a finite set of possible worlds and finite accessibility relations. Improvement on inference speed can be expected as this consequence, because the system knows the finite skeleton of the model in advance.

Also, the domain for variables is assumed to be common over all possible worlds, and the interpretations of function constant symbols and variables are independent of possible worlds. These restrictions have been assumed in the examples of inference shown in the previous chapter; therefore, we think they are not too severe for practical knowledge representation. The unary operator "(v_i)", which denotes the worlds v_i where its argument formula must hold, is introduced into our system, besides the operators used in the previous chapter. Regular expressions over the set of basic viewpoints are used in order to express various kinds of relations over possible-worlds .

3.2 Syntax

Atomic symbols consist of: frame name symbols v_1, v_2, ... ; link name symbols A_1, A_2, function constants f^n, g^n, ... ; n-ary predicate constants p^n, q^n,... ; n-ary function variables F^n, G^n, ... ; n-ary predicate variables P^n, Q^n, ... ; and atom variables AP, AQ, ... Also, we use {, }, <, >, [,], (,), ;, *, +, -, <--, || , "any" and "Frame_structure" as subsidiary symbols.

Definition.1 Links and frame structures are inductively defined as follows:
1) If v_i and v_j are frame names and A_k is a link name, then an expression $A_k(v_i,v_j)$ is a link.
2) If L_1, ... ,L_n are links, then an expression Frame_structure { $L_1,...,L_n$ } is a frame structure.

Definition.2 A viewpoint over a set of link names is a regular expression over the set of link names with the operators ;, *, +, and the unary operator "-".

Definition.3 A knowledge is inductively defined as follows:
1) A term is an expression defined as usual from individual variables, function constants and function variables.
2) An atom is an atom variable, or an expression defined as usual from terms, predicate constants and predicate variables.
3) If A is an atom, v_i is a frame name and 'a' is a viewpoint, then
A, [a]A, <a>A, (v_i)A, (v_i)[a]A, (v_i)<a>A
are modal atoms.
4) If A is an atom and $MA_1,...,MA_n$ are modal

atoms, then an expression

$A \leftarrow\!- MA_1, \ldots, MA_n$

is a definite clause.

5) If v_i is a frame name and $C_1, \ldots, C_n$ are definite clauses, then an expression

$v_i \{ C_1, \ldots, C_n \}$

is knowledge of the frame v_i. Also,

$any \{ C_1, \ldots, C_n \}$

is knowledge of the special frame "any". They are denoted by K_{vi} and K_{any}, respectively.

6) If $K_{v1}, \ldots, K_{vn}$ are knowledges of the frames $v_1, \ldots, v_n$ appearing in a frame structure F, respectively, and K_{any} is a knowledge of the frame "any", then an expression

$\{ K_{v1}, \ldots, K_{vn}, K_{any} \}$

is a knowledge of F, and is denoted by K_F.

7) If v_i is a frame name and $MA_1, \ldots, MA_n$ are modal atoms, then an expression

$?- v_i \,||\, MA_1, \ldots, MA_n$

is a goal clause for v_i, and is denoted by G_{vi}.

An expression is called a schema expression if it includes some of function variables, predicate variables or atom variables. An expression is called a ground expression if it isn't a schema expression and doesn't include any individual variables.

We give an example of a knowledge representation corresponding to the possible-world model of Fig.2.

```
Frame_structure (
        ako(bird,animal), ako(monkey,animal),
        isa(taro,monkey), isa(hanako,bird)
    )

animal { move(X),
         number_legs(X,Y)<--<ako⁻>num_legs(X,Y)
    }
bird   {    can_fly(X)      }
monkey {    can_run(X),
            num_legs(X,4) <-- <isa⁻>monkey(X)
      }
hanako {     bird(hanako)     }
taro {     monkey(taro)      }
any  {     AP <-- <isa+ako>AP    }
```

We have constructed a simple inference system in Prolog with the deductive inference rules stated in Section 4.1. We will show some examples of its question-answering computations in Section 4.2.

3.3 Semantics

<u>Definition.4</u> A model of a frame structure F is a directed graph $G_F = \langle V_F, L_F, E_F \rangle$ with labeled arcs, where V_F is the set of all frame names appearing in F, L_F is the set of all link names appearing in F, and E_F: $\Sigma_F \rightarrow 2^{VF/VF}$ is a function satisfying the following conditions:

1) Σ_F is the set of viewpoints over L_F,
2) $E_F(A_k) = \{ \langle v_i, v_j \rangle \mid A_k(v_i, v_j) \in F \}$
3) $E_F(a;b) = E_F(a) \circ E_F(b)$ (composition of relations)
4) $E_F(a+b) = E_F(a) + E_F(b)$ (union of relations)
5) $E_F(a^*) = E_F(a)^*$ (reflex and transitive closure of a relation)
6) $E_F(a^-) = \{ \langle v_i, v_j \rangle \mid \langle v_j, v_i \rangle \in E_F(a) \}$ (converse of a relation)

The frame structure F defines the skeleton of a possible-world model. G_F is abbreviated as F if no confusion arises.

We consider a kind of Herbrand interpretation in order to simplify our discussion. We refer the reader to [Abadi and Manna 1986b, Harel 1979, Lloyd 1984] for other definitions of interpretations. As stated in Section 3.1, the interpretation of variables and function constants is assumed to be independent of possible worlds. The interpretation of predicate constants must be dependent on possible worlds.

<u>Definition.5</u> Let K_F be a knowledge of a frame structure F. The Herbrand universe H_{KF} for K_F is the set of all ground terms which can be constructed with the function constants appearing in K_F. Also, the Herbrand base B_{KF} for K_F is the set of all modal atoms of the form $(v_i)p^n(t_1, \ldots, t_n)$ such that $t_1, \ldots, t_n \in H_{KF}$ and both symbols v_i and p^n appear in F or K_F.

<u>Definition.6</u> Let K_F be a knowledge of a frame structure F. An interpretation I for K_F is a subset of the Herbrand base B_{KF} for K_F.

$(v_i)p^n(t_1, \ldots, t_n) \in I$ means that $p^n(t_1, \ldots, t_n)$ is true in the frame v_i under the interpretation I. We will extend the usual definition of substitutions in order to deal with function, predicate and atom variables.

<u>Definition.7</u> A substitution θ is a finite set of the form $\{ \alpha_1/\beta_1, \ldots, \alpha_n/\beta_n \}$, where each pair of α_i/β_i satisfies the following conditions:

1) α_i is an individual variable and β_i is a term distinct from α_i, or
2) α_i is a n-ary function variable and β_i is an n-ary function constant or an n-ary function variable distinct from α_i, or
3) α_i is an n-ary predicate variable and β_i is a n-ary predicate constant or an n-ary predicate variable distinct from α_i, or
4) α_i is an atom variable and β_i is an atom distinct from α_i.

If $E\theta$ is a ground expression, then θ is called a ground substitution of E. Composition of substitutions and the most general unifier (mgu), etc. are defined as usual [Lloyd 1984]. A substitution θ is called a substitution over a knowledge K_F if all constant symbols appearing in θ also appear in K_F. All substitutions used in this paper are assumed to be substitutions over K_F.

Definition.8 We write $I,v_i \models_F E$ if the expression E is true in the frame v_i in the interpretation I for the knowledge K_F. The truth values of ground modal atoms are inductively defined as follows:

1) $I,v_i \models_F p^n(t_1,\ldots,t_n)$ iff $(v_i)p^n(t_1,\ldots,t_n) \in I$

2) $I,v_i \models_F [a]A$ iff $I,v_k \models_F A$ for all v_k such that $\langle v_i,v_k\rangle \in E_F(a)$

3) $I,v_i \models_F \langle a\rangle A$ iff there is a v_k such that $\langle v_i,v_k\rangle \in E_F(a)$ and $I,v_k \models_F A$

4) $I,v_i \models_F (v_j)A$ iff $I,v_j \models_F A$

5) $I,v_i \models_F (v_j)[a]A$ iff $I,v_k \models_F A$ for all v_k such that $\langle v_j,v_k\rangle \in E_F(a)$

6) $I,v_i \models_F (v_j)\langle a\rangle A$ iff there is a v_k such that $\langle v_j,v_k\rangle \in E_F(a)$ and $I,v_k \models_F A$

The truth values of definite clauses $C = A \leftarrow MA_1,\ldots,MA_n$ are defined as follows:

7) $I,v_i \models_F A \leftarrow MA_1,\ldots,MA_n$ iff for all ground substitutions θ of C, if $I,v_i \models_F MA_1\theta, \ldots, I,v_i \models_F MA_n\theta$, then $I,v_i \models_F A\theta$

We write $I \models_F K_{v_i}$ if the knowledge $K_{v_i} = v_i\ (C_1,\ldots,C_n)$ is true in the interpretation I.

8) $I \models_F K_{v_i}$ iff $I,v_i \models_F C_j$ for all $C_j \in K_{v_i}$

Particularly, we define the truth value of the knowledge K_{any} as follows:

9) $I \models_F K_{any}$ iff $I,v_i \models_F C_j$ for all $v_i \in V_F$ and $C_j \in K_{any}$

We write $I \models_F K_F$ if the knowledge $K_F = \{K_{v_1},\ldots,K_{v_n}, K_{any}\}$ of the frame structure F is true in the interpretation I.

10) $I \models_F K_F$ iff $I \models_F K_{v_i}$ for all $K_{v_i} \in K_F$, and $I \models_F K_{any}$

Function, predicate and atom variables are treated in Definition 8-7). The values of these variables are restricted to the constant symbols appearing in K_F, because all substitutions in this paper have been assumed to be substitutions over K_F.

Definition.9 Let K_F be a knowledge of F and $G_{v_i} = ?\text{-}\ v_i \parallel MA_1,\ldots,MA_n$ be a goal clause for v_i. G_{v_i} is said to be true in the model I on F for K_F if G_{v_i} satisfies the following condition:

For every ground substitution θ of G_{v_i}, $I,v_i \models_F MA_1\theta, \ldots, I,v_i \models_F MA_n\theta$

G_{v_i} is said to be valid on K_F if G_{v_i} is true in all models on F for K_F.

Definition.10 Let K_F be a knowledge and G_{v_i} be a goal clause. A correct answer substitution θ for G_{v_i} on K_F is a substitution for variables of G_{v_i} such that $G_{v_i}\theta$ is valid on K_F.

4 A DEDUCTIVE SYSTEM

4.1 A Deductive System and Its Soundness and Completeness

We give a deductive system for our language and show its soundness and completeness. Though it is possible to translate our knowledge representation to an equivalent many-sorted Prolog program, we will investigate a variant of SLD resolution here, because it is directly applicable to our language.

Definition.11 Let v_i be a frame name and MA be a modal atom. Then we define the modal atom $\overline{(v_i)MA}$ as follows:

$$\overline{(v_i)MA} = \begin{cases} (v_j)MB & \text{if MA has the form } (v_j)MB \\ (v_i)MA & \text{otherwise.} \end{cases}$$

Lemma.1 Let K_F be a knowledge, v_i and v_k be the frame names appearing in F and MA be a modal atom. Then the following statements are equivalent:

1) A goal $?\text{-}\ v_k \parallel \overline{(v_i)MA}$ is valid on K_F,

2) A goal $?\text{-}\ v_i \parallel MA$ is valid on K_F.

Definition.12 Let K_F be a knowledge, $G^i_{v_i} = ?\text{-}\ v_i \parallel MA_1,\ldots, MA_m,\ldots, MA_k$ be a goal and CS_{i+1} be a set of definite clauses. Then the goal $G^{i+1}_{v_i}$ satisfying the following conditions is said to be the resolvent obtained from $G^i_{v_i}$ and CS_{i+1} using the mgu θ_{i+1} on the selected atom MA_m:

1) If MA_m is an atom B, then
   a) CS_{i+1} is a set consisting of exactly one clause $A \leftarrow MB_1,\ldots,MB_n$ of K_{v_i} or K_{any},
   b) θ_{i+1} is a mgu such that $B\theta_{i+1} = A\theta_{i+1}$.
   c) $G^{i+1}_{v_i} = ?\text{-}v_i \parallel (MA_1,\ldots,MA_{m-1}, MB_1,\ldots, MB_n, MA_{m+1},\ldots,MA_k)\ \theta_{i+1}$.

2) If MA_m is a modal atom $(v_j)B$, then
   a) CS_{i+1} is a set consisting of exactly one clause $A \leftarrow MB_1,\ldots,MB_n$ of K_{v_j} or K_{any},
   b) θ_{i+1} is a mgu such that $B\theta_{i+1} = A\theta_{i+1}$.
   c) $G^{i+1}_{v_i} = ?\text{-}v_i \parallel (MA_1,\ldots,MA_{m-1}, \overline{(v_j)MB_1}, \ldots, \overline{(v_j)MB_n}, MA_{m+1},\ldots,MA_k)\ \theta_{i+1}$.

3) If MA_m is a modal atom $\langle a\rangle B$ (or $(v_j)\langle a\rangle B$) and v_r is a frame such that $\langle v_i,v_r\rangle \in E_F(a)$ (respectively, $\langle v_j,v_r\rangle \in E_F(a)$), then
   a) CS_{i+1} is a set consisting of exactly one clause $A \leftarrow MB_1,\ldots,MB_n$ of K_{v_r} or K_{any},
   b) θ_{i+1} is a mgu such that $B\theta_{i+1} = A\theta_{i+1}$.
   c) $G^{i+1}_{v_i} = ?\text{-}v_i \parallel (MA_1,\ldots,MA_{m-1}, \overline{(v_r)MB_1}, \ldots, \overline{(v_r)MB_n}, MA_{m+1},\ldots,MA_k)\ \theta_{i+1}$.

4-1) If MA_m is a modal atom $[a]B$ (or $(v_j)[a]B$) and there is no frame v_r such that $\langle v_i,v_r\rangle \in E_F(a)$ (respectively, $\langle v_j,v_r\rangle \in E_F(a)$), then
   a) CS_{i+1} is the empty set
   b) θ_{i+1} is the identity substitution
   c) $G^{i+1}_{v_i} = ?\text{-}\ v_i \parallel (MA_1,\ldots,MA_{m-1}, MA_{m+1},\ldots,MA_k)$

4-2) If MA_m is a modal atom $[a]B$ (or $(v_j)[a]B$) and $v_1,\ldots,v_h$ are all frames v_r satisfying the condition $\langle v_i,v_r\rangle \in E_F(a)$ (respectively,

$\langle v_j, v_r\rangle \in E_F(a))$, then

a) CS_{i+1} is a set $\{C_{v1},\dots,C_{vh}\}$ such that each C_{vr} belongs to K_{vr} or K_{any}.

Suppose that each C_{vr} is A_{vr}<--$MB_{1_{vr}},\dots,MB_{n_{vr}}$. Then

b) θ_{i+1} is a mgu such that

$$B\theta_{i+1} = A_{v1}\theta_{i+1} = \dots = A_{vh}\theta_{i+1}$$

c) G_{vi}^{i+1} = ?- v_i || $(MA_1,\dots,MA_{m-1}, [\overline{(v_1)MB_{1_{v1}}},\dots,\overline{(v_1)MB_{n_{v1}}}],\dots, [\overline{(v_h)MB_{1_{vh}}},\dots,\overline{(v_h)MB_{n_{vh}}}], MA_{m+1},\dots,MA_k)\,\theta_{i+1}$.

Definition.13 Let K_F be a knowledge and G_{vi} be a goal. A derivation of G_{vi} on K_F consists of a sequence $G_{vi}^0 = G_{vi}, G_{vi}^1, G_{vi}^2, \dots$ of goals, a sequence $CS_1, CS_2, \dots$ of sets of clauses and a sequence $\theta_1, \theta_2, \dots$ of mgu's such that G_{vi}^{i+1} is a resolvent obtained from G_{vi}^i and CS_{i+1} using θ_{i+1} on a selected atom in G_{vi}^i. A refutation of G_{vi} on K_F is a finite derivation of G_{vi} on K_F which has the empty clause as the last goal in the derivation.

Definition.14 Let K_F be a knowledge and G_{vi} be a goal. A computed answer substitution for G_{vi} on K_F is the substitution obtained by restricting the composition $\theta_1\dots\theta_n$ to the variables of G_{vi}, where $\theta_1,\dots,\theta_n$ is the sequence of mgu's in a refutation for G_{vi} on K_F.

We can prove the followings by Lemma.1, though their proofs are omitted here. We refer the reader to [Iwanuma and Harao 1988, Lloyd 1984] for detailed proofs.

Theorem.1 (Soundness of computed answer substitutions) Let K_F be a knowledge and G_{vi} be a goal. Then every computed substitution for G_{vi} on K_F is a correct answer substitution.

Theorem.2 (Completeness of computed answer substitutions) Let K_F be a knowledge and G_{vi} be a goal. For every correct answer substitution θ for G_{vi} on K_F, there is a computed answer substitution σ for G_{vi} on K_F and a substitution γ such that $\theta = \sigma\gamma$.

4.2 An Inference System in Prolog

We have constructed a simple inference system in Prolog. We demonstrate some examples of question-answering computations of this system over the knowledge shown in Section 3.2.

```
?- animal || number_legs(taro,Y).       ----- (1)
Y = 4;
no

?- taro || number_legs(taro,Y).         ----- (2)
Y = 4;
no

?- bird || AP.                          ----- (3)
AP = can_fly(X)
AP = move(X);
AP = number_legs(taro,4);
no

?- taro || [(isa+ako)*]AP.              ----- (4)
AP = move(X);
AP = number_legs(taro,4);
no

?- FX || can_fly(hanako).               ----- (5)
FX = bird;
FX = hanako;
no
```

The first two questions were explained in Chapter 2. The 3rd question is what holds in the frame "bird". The answers are "can_fly(x)", "move(x)" and "number_legs(taro,4)". The last two answers are inherited from "animal", by the schema formula "AP <--<isa+ako>AP". The 4th question is what holds in all frames which are super-concept frames of "taro". The answers are ones which holds in the top frame "animal". In the 5th question, we ask where the atom "can_fly(hanako)" holds, by using a frame name variable FX, which is not treated in this paper in order to simplify our discussion.

The above computations are faithful to the deductive method stated in Section 4.1. Our system has some modes which control the search space of inference. Formal discussion about those modes are omitted to simplify our discussion, though they are possible. We show some examples. The command "inf_mode" instructs the system to restrict the search space for solving a goal "?- v_i || Body" to a set of possible worlds whose distance from v_i is equal or less than 1.

```
?- inf_mode(restrict,1).
yes

?- bird || AP.                          ----- (6)
AP = can_fly(X);
AP = move(X);
no

?- taro || [(isa+ako)*]AP.              ----- (7)
P = num_legs(taro,4);
no
```

Compare the answers of the 6th question with those of the 3rd question. "number_legs(taro,4)" is dropped here, because, in order to infer "number_legs(taro,4)", it is necessary to search the worlds "monkey" and "taro" whose distance from "bird" is more than 1. Also, the answers of the 7th question are ones which hold in "monkey", and are different from those of the 4th question. Note "animal" is at a distance of 2 from "taro".

5 MODAL LOGIC AND KNOWLEDGE REPRESENTATION

Roughly speaking, there are two kinds of applications of modal logic to knowledge representation. One of them is represented by the logic of belief and knowledge [Fagin et al. 1984, Levesque 1984], where metaknowledge such as "believe" or "know" are represented by modal operators. In this case, the concept of possible worlds is hidden from knowledge representation. The other one is the research such as [Cerro 1985, Nakazima 1985, Sakakibara 1986], which introduce the concept of possible worlds into knowledge representation systems in order to describe knowledge structures. Modal operators are used for describing properties over the structure. Our research belongs to the latter category.

In other systems, the concept corresponding to our "viewpoints" is not investigated. Consequently, our system furnishes a stronger capability for describing knowledge structures than do others. Also, there are few research efforts concerning: the metaknowledge over knowledge structure; formal inference rules; and their soundness, completeness and effectiveness.

Sakakibara (1986) presented programming language based on a modal logic. His idea is similar to ours. But he proposed only an outline of his language and its programming style. He neither discussed the formal semantics of his language nor gave any formal inference system. In his language, the names of accessibility relations are assumed to be unique, so the viewpoints of modalities proposed in this paper can be regarded as its extension. Also, the modal operators [] (necessity) and <> (possibility) are allowed to be at the heads of definite clauses in his language. They are used to express property inheritance over possible worlds. But the operator <> at the head doesn't guarantee the uniqueness of models of knowledge expressions; that is, it has the effect that some knowledge expressions may have some minimal models instead of exactly one least model. Perhaps we can construct a linear resolution method that refutes a goal iff there is a minimal model making the goal valid. But we conjecture that its computation will become very complex, therefore, this approach seems to be impractical. In this paper, we don't allow the operator <> at the head of a definite clause in order to achieve effectiveness of inference.

6 CONCLUSION

In this paper, we have shown that structured knowledge and metaknowledge can be naturally described in our knowledge representation system, which is based on a first-order modal logic. Also, we have given a complete deductive system which is as effective as SLD resolution. Its simple inference system has been constructed in Prolog.

In the future, we will investigate the applicability of higher-order types to metaknowledge representations. Also, we have a plan to investigate the possibility of parallel processing in a processor network like a cellular machine, in which each processor corresponds to one possible world.

REFERENCES

[Abadi and Manna 1986a] M.Abadi and Z.Manna; "A Timely Resolution", Proc. of the Symp. on Logic in Computer Science, Cambridge, (1986)
[Abadi and Manna 1986b] M.Abadi and Z.Manna; "Modal Theorem Proving", 8th Int. Conf. on Automated Deduction, LNCS.230, (1986), pp.172-189.
[Cerro 1985] F.del Cerro.L; "MOLOG: A System That Extends PROLOG with Modal Logic", New Generation Computing, Vol. 3 (1985), pp.359-383
[Fagin et al. 1984] R.Fagin, J.Y.Halpern and M.Y.Yardi; "A Model Theoretical Analysis of Knowledge", Proc. of 25th Foundation of Computer Science, (1984), pp.268-278.
[Fischer and Ladner 1979] M.J.Fischer and R.E.Ladner; "Propositional Dynamic Logic of Regular Programs", J. Comput. Syst. Sci., Vol. 18, (1979), pp.194-211.
[Goldfarb 1981] W.D.Goldfarb; "The Undecidability of the Second-Order Unification Problem", Theor. Comput. Sci., Vol.13 (1981), pp.225-230.
[Harel 1979] D.Harel; "First-Order Dynamic Logic", LNCS.68 (1979)
[Harel 1984] D.Harel; "Dynamic Logic", Handbook of Philosophical Logic, Vol.II (Gabbay and Guenther, eds.), D.Reidel Publishing Co., (1984), pp.497-604.
[Iwanuma and Harao 1987] K.Iwanuma and M.Harao; "First-Order Modal Logic for Time and Space and Its Incompleteness and Relative Completeness", J. Inst. Electronics & Inf. Comm. Eng. Japan, Vol.J70-D, No.5 (1987), pp.859-869.
[Iwanuma and Harao 1988] K.Iwanuma and M.Harao; "Knowledge Representation Based on First-Order Modal Logic", J. Japan Soc. Artif. Intell., Vol.3, No.3 (1988), pp.350-358.
[Levesque 1984] H.J.Levesque; "A Logic of Implicit and Explicit Belief", Proc. of Nat. Conf. of Artificial Intelligence, (1984), pp198-202.
[Lloyd 1984] J.W.Lloyd; Foundations of Logic Programming, Springer-Verlag, (1984)
[Miller 1983] D.A.Miller; "Proofs in Higher-Order Logic", Ph.D.Thesis, Carnegie-mellon Univ. (1983)
[Nakazima 1985] H.Nakazima; Knowledge Representation and Prolog/KR, Sangyou-Tosyo Co., (1985)
[Sakakibara 1986] H.Sakakibara; "Programming in Modal Logic", Proc. of the Logic Programming '86, (1986), pp.119-126.
[Ventatesh 1985] G.Ventatesh; "A Decision Method for Temporal Logic Based on Resolution", Found. of Soft. Tech. & Theor. Comput. Sci., LNCS.206, (1985), pp.273-288.