ICOT Technical Report: TR-306

TR-306

Consideration of a Hypothesis-based Reasoning System

by
S. Kunifuji (Fujitsu), K. Tsurumaki (NTT)
and K. Furukawa

October, 1987

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Consideration of a Hypothesis-based Reasoning System *

by

Susumu KUNIFUJI **

Koji TSURUMAKI ***

Koichi FURUKAWA ****

* The main part of this paper appeared in the Journal of Japanese Society for Artificial Intelligence, Vol.1 No.2, pp.228-237, Dec. 1986 (in Japanese).

** Senior Research Staff, Information Engineering Section, International Institute for Advanced Study of Social Information Science, FUJITSU LIMITED, Shizuoka 410-03, Japan

*** Researcher, Customers Equipment Department, NTT Integrated Communications Laboratories, NTT, Kanagawa 238-03, Japan

**** Deputy Director, Research Center, Institute for New Generation Computer Technology, Tokyo 108, Japan

ABSTRACT

This paper describes mechanizing abduction which means hypothesis generation and selection for given observations. It considers hypothesis generation and selection problems in a similar framework to the theory formation system, Theorist. The experimental system is implemented in DEC-10 Prolog. It generates hypotheses from a set of hypothetical formulas and selects one to form consistent explanations of given observations.

The differences between our system and Theorist are as follows:

(1) In our system, hypotheses and given knowledge are represented in an is-a hierarchical structure, and appropriate hypotheses for given observations are generated based on the structure. In Theorist,hypotheses are not supposed to have such a structure.

(2) If multiple possible hypotheses are to be generated by the system, our system asks new observations of the user in order to select one possible hypothesis. The important thing is that meta-programming enables the user easily to define the criteria for hypothesis generation and selection mechanism.

(3) Our system can make use of negative knowledge (integrity constraints) in the same way as Theorist-S. Our system has simple semantics and can be implemented easily, using meta-programming in Prolog.

1. Introduction

According to the pragmatist, C.S. Peirce [18], there are three kinds of inference processes in human problem-solving: deduction, induction, and abduction [1,8]. To build a prototype of a knowledge information processing system, the most important research theme is to build such a system with deductive, inductive and abductive inference capabilities. However, the research themes in the knowledge information processing system or in artificial intelligence are mainly related to deduction and induction, not related to abduction. Then, as one of basic researches related to abduction, we investigate a hypothesis generation and selection mechanism, which is the most typical inference pattern in human beings. We propose an experimental system for hypothesis generation and selection, implemented in logic programming language Prolog, and also clarify a framework of abdutive inference.

The essence of the human problem-solving processes consists of a hypothesis generation process and a hypothesis selection process. The former generates hypotheses which can explain given observed facts by logical implications from given hypotheses. The latter selects a candidate hypotesis which verifies the derived logical implications by some added observations (experiments) from the user.

It is well-known that, from the view of computer science, inductive inference mechanism aims at a formalization of learning from examples. Note that inductive inference consists of hypothesis generation and testing problems [2]. The abductive inference mechanism proposed by us aims at a formalization of hypothesis creation which consists of several complex problems, as described in section 2.2.

2. Abduction and Hypothesis-based Reasoning

2.1 Abduction

The most attractive theme of the human problem-solving and inference processes is mechanizing abductive inference. Generally speaking, abduction consists of complex mechanisms. We simplify and clarify abduction, as a first step toward mechanizing it, as in the following inference scheme [8,18], where the symbol $\vdash$ means deducibility in first-order logic. The scheme shows that

$$\frac{\begin{array}{l}\text{mortal(socrates)}\\ \forall X\ \text{human}(X) \vdash \forall X\ \text{mortal}(X)\end{array}}{\text{human(socrates)}} \qquad (1)$$

If the unknown fact 'socrates is mortal' is observed in a real world, and the known fact 'every human is mortal' exists in another real world, then we have evidence that the fact 'socrates is human' may be an unknown hypothesis of the observation. This statement might be abstracted to the next scheme.

$$\frac{\begin{array}{ll}B & \text{observation}\\ A \vdash B & \text{fact}\end{array}}{\begin{array}{ll}A & \text{hypothesis}\end{array}} \qquad (2)$$

The first question is whether we can easily infer the other solutions, for example, 'socrates is an ape' or ... or 'socrates is a living-thing'. By intuition, we suppose that 'socrates is a living-thing' is the most desirable hypothesis. Therefore, the question gives rise to the problem of how to generate hypotheses systematically and to select one of them efficiently in a given knowledge structure.

$$\frac{\begin{array}{l}\text{mortal(socrates)}\\ \forall X\ \text{foo}(X) \vdash \forall X\ \text{mortal}(X)\end{array}}{\text{foo(socrates)}} \tag{3}$$

$$\left.\begin{array}{l}\text{ape(socrates)}\\ \vdots\\ \text{living-thing(socrates)}\end{array}\right\}\ \text{possible hypotheses}$$

2.2 Hypothesis-based Reasoning

In this section, we propose a framework of a hypothesis-based reasoning system. The hypothesis-based reasoning system uses unknown (possible) hypotheses and known facts to form consistent explanations of observations. A hypothesis is a subset of the possible hypotheses which are consistent for given facts, and a union of the hypothesis and facts should imply given observations. Now, we can provide a theoretical framework of the hypothesis-based reasoning system, as a generalization of the theory formation system, Theorist [10].

Let O, H and F be a set of given observations, a set of possible hypotheses and a set of given facts, respectively, expressed in the well-formed formulae of the clausal form of first order logic. We say a hypothesis h is explainable iff there is a ground instantiation H' via unifier θ of some subset H' of H such that

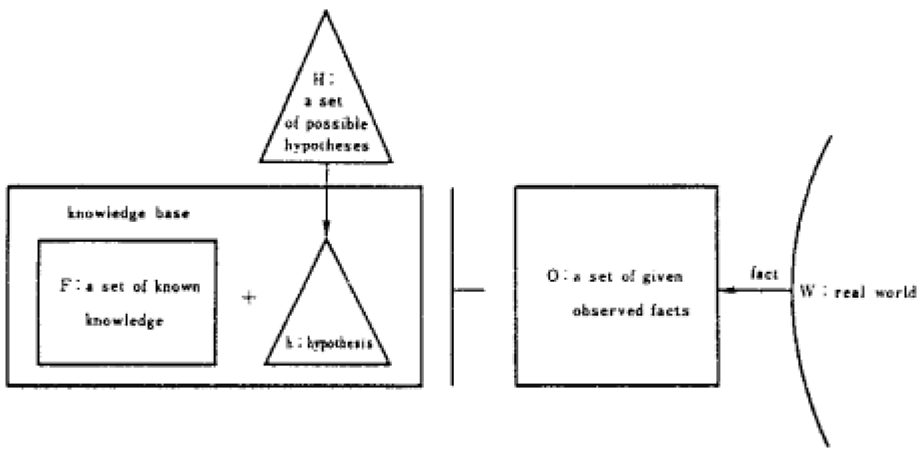

Fig. 1 A framework of hypothesis-based reasoning

$F \nvdash O$ (where $F \nvdash \Box$), and (4)

$h = H'\theta$, $H' \subseteq H$, $F + h \vdash O$ (where $F + h \nvdash \Box$) (5)

Here, the symbol $\nvdash$ means undeducibility, $\Box$ means contradiction, $\subseteq$ means a subset, and + means a set union.

For this scheme shown in Fig. 1, we propose several problems on hypothesis-based reasoning as follows:

(1) Consistency Maintenance Problem: How should the consistency of knowledge base KB ($\triangleq$ F+h) be maintained?

(2) Hypotheses Generation Problem: What is the criterion of hypotheses generation to generate adequate hypotheses from possible hypotheses set?

(3) Conflict Hypotheses Resolution Problem: What is the criterion to decide what hypothesis is true, when competetive multiple hypotheses are generated, because given knowledge is incomplete?

(4) Hypothesis Selection Problem: What is the criterion of hypothesis selection to identify a unique hypothesis among generated tentative hypotheses using new observed facts set, if multiple hypotheses are generated?

(5) Inductive Inference Problem: How is it possible to infer inductively possible hypotheses set H for given observed facts set? In this case, it is supposed that F+H is consistent. Note that, if F+H is inconsistent, the problem becomes non-monotonic.

(6) Frame Selection Problem: First of all, what F and H are selected to explain observed facts set O? Note that we often make use of analogical reasoning to find such F and H.

In this paper, we propose a solution of the above problems, (2), (3), and (4). Later, we will discuss other problems in chapter 5.

2.3 Relationship to Knowledge Acquisition Functions

This section describes the relationship between the hypothesis-based reasoning system and the knowledge acquisition support system [5,6,7] which were investigated in ICOT, as a basic function of the problem-solving and inference system and the knowledge base management system in the fifth generation computer systems. The systems are implemented in DEC-10 Prolog on the DEC2060. To support deductive and/or inductive inference-based knowledge acquisition, the system fundamentally employs three kinds of knowledge acquisition functions: knowledge assimilation, knowledge accommodation, and knowledge transaction control [5,6,7,9]. As shown in Fig. 2, knowledge assimilation [7,9] means adding new facts or rules to the knowledge base, without violating its consistency. Knowledge accommodation [5,6,7] means consistently modifying the knowledge base, using Shapiro's model inference system [15], when adding new correct facts or rules to it. In the knowledge accommodation process, given added facts or rules are supposed to be absolutely true, while, in the assimilation process, a given knowledge base is supposed to be consistent. The essential problem in assimilation and accommodation is to keep the entire knowledge base consistent and non-redundant. Knowledge transaction control [6] means an adjustment of knowledge assimilation and knowledge accommodation in a given transaction span. If one knowledge assimilation process is triggered, then often another related knowledge accommodation process is to be applied, because they are mutually dependent in the transaction span. Note that all functions are implemented by using meta-programming techniques in Prolog [7].

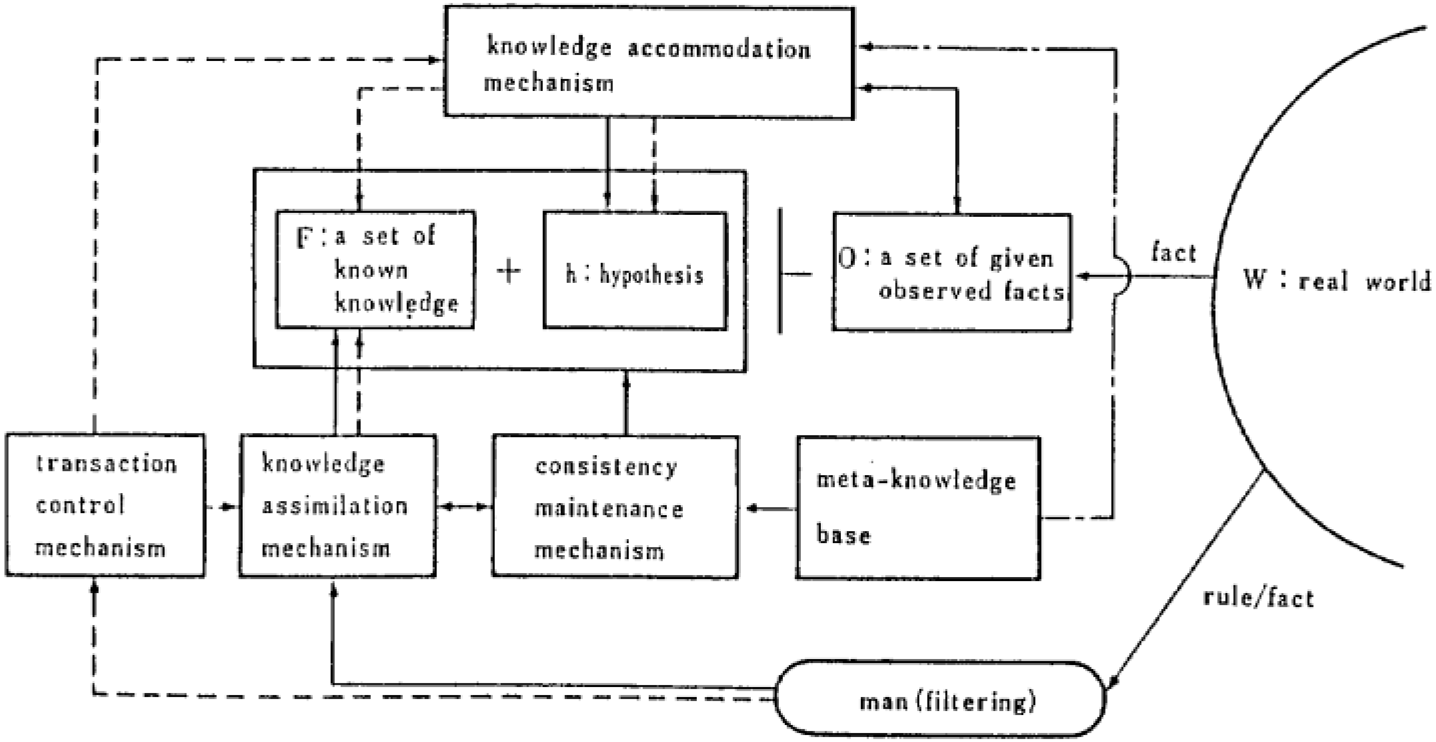

Fig. 2 Hypothesis-based reasoning and knowledge acquisition

3. Knowledge Representation

3.1 Characteristics

We implemented the following hypothesis generation and selectionsystem in DEC-10 Prolog.Itscharacteristics are summarized as follows, compared with a similar theory formation system, Theorist [10]. Fig. 3 shows a framework of our system.

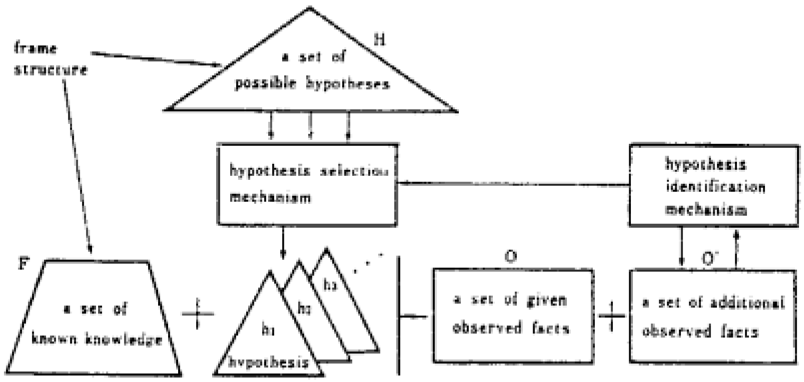

Fig.3 A framework of hypothesis selection mechanism

(1) In our system, hypotheses and given knowledge are represented in an is-a hierarchical structure, and appropriate hypotheses for given observations are generated based on the structure. In Theorist, hypotheses and given knowledge are not supposed to have such a structure, that is, they have a flat structure.

(2) If multiple possible hypotheses are to be given by the system, our system asks new observations of the user in order to select one possible hypothesis.

(3) Our system can make use of negative knowledge (integrity

constraints) in the same way as Theorist-S [3]. Our system has simle semantics and is implemented easily, using meta-programming in Prolog.

3.2 Frame-type Knowledge Representation

To handle a given problem in a framework of the first order logic, we introduce such frame-type knowledge representation with reference to [4,12] as follows:

$$
\begin{array}{l}
[\text{FRAME} \ \langle\text{frame-name}\rangle \\
\quad \text{hypothesis\_of} : \langle\text{hypothesis-name}\rangle \ ; \\
\quad \langle\text{SLOT-NAME}\rangle_1 : \text{slot-value}_1 \ ; \\
\qquad \vdots \\
\quad \langle\text{SLOT-NAME}\rangle_n : \text{slot-value}_n]
\end{array}
\tag{6}
$$

where we suppose that this hierarchical frame structure has no loop. In this frame scheme, we introduce a new relation, hypothesis-of relation which gives a unique name for the frame. There are two types of slots. One is called an is-a relation. The other is called a property-of relation. The starred property-of relation has a default which value can be assumed until or unless other information is available. The non-starred property-of relation is a normal property-of relation. The above frame scheme can be interpreted as the first order formula for each i.

$$
\begin{array}{l}
\forall X \, (\text{hypothesis\_of} \, (X, \langle\text{hypothesis-name}\rangle) \\
\qquad \equiv \text{is\_a} \, (X, \langle\text{frame-name}\rangle))
\end{array}
\tag{7}
$$

$$
\begin{array}{l}
\forall X \, (\text{is\_a} \, (X, \langle\text{frame-name}\rangle) \supset \\
\qquad \langle\text{SLOT-NAME}\rangle_i \, (X, \text{slot-value}_i))
\end{array}
\tag{8}
$$

where $\equiv$ means "if and only if".

Example 1 : Consider the next frame-type knowledge.

```
[FRAME  alkaline_earth
  hypothesis_of : "alkaline-earth-ion" ;
  is_a : metal ;
  precipitate : 'SO4-ion']
```
(9)

This frame are translated into the following clausal formulas.

$$\forall X\,(hypothesis\_of\,(X,\ \text{"alkaline-earth-ion"}) \equiv is\_a\,(X, alkaline\_earth\,)) \quad (10)$$

$$\forall X\,(is\_a\,(X, alkaline\_earth\,) \supset is\_a\,(X, metal)) \quad (11)$$

$$\forall X\,(is\_a\,(X, alkaline\_earth\,) \supset precipitate\,(X, \text{'}SO_4\text{-ion'})) \quad (12)$$

3.3 Relationship to Default Theory

As known knowledge set F and possible hypotheses set H have a frame-like structure, we want to describe defaults with exceptions to the structure. Therefore, the starred property-of relation can be interpreted as follows:

$$\frac{is\_a\,(X, \langle frame\text{-}name\rangle) \;:M\,(\langle SLOT\text{-}NAME\rangle_i\,(X,\ slot\text{-}value_i))}{\langle SLOT\text{-}NAME\rangle_i\,(X,\ slot\text{-}value_i)} \quad (13)$$

where a meta-operator M is a Reiter's normal default operator [11,13].

Example 2 : Consider the statement "birds fly". This can be expressed as the default:

$$\forall X\,(\mathrm{is\_a}(X,\ \mathrm{bird}) \wedge M\,(\mathrm{can\_fly}(X,\mathrm{sky})) \supset \mathrm{can\_fly}(X,\mathrm{sky})) \tag{14}$$

It is difficult to handle the M operator, as it is a non-monotonic operator [11,13]. However, if we restrict its semantics as shown below, we can easily implement it, using Prolog's negation as failure [7]. This idea does not give rise to block problems of inheritance in semi-normal defaults[14].

```
can_fly (X, sky) :-is_a (X, bird),
                  \+ (cannot_fly (X, sky)).          (15)

                                                     (16)
cannot_fly (ostrich, sky).
```

4. Hypothesis Generation and Selection Problems

4.1 Hypothesis Generation Problem

We assume the frame-like knowledge representation introduced in the preceding chapters. Let F_h, F_i and F_p be a set of hypothesis-of relations, a set of is-a relations, and a set of property-of relations. Then, the following relationships hold:

$$F \triangleq F_h + F_i + F_p \equiv F_h + F_{ip} \equiv F_{hi} + F_p \tag{17}$$

$$F_{ip} \triangleq F_i + F_p \ (\triangleq F_{pi}) \tag{18}$$

$$F_{hi} \triangleq F_h + F_i \ (\triangleq F_{ih}) \tag{19}$$

Now, we can introduce observed facts expressed as the conjunctions of property-of relations.

$$\bigwedge_{i=1}^{n} \exists X^{i} \langle \text{SLOT-NAME} \rangle_{i} (X^{i}, \text{slot-value}_{i}) \tag{20}$$

Let $N_{\mathfrak{F}}$ and $N_{\mathcal{H}}$ be a finite set of all frame names and a finite set of all hypotheses names in a given frame. In general, hypothesis-of relation "hypothesis_of(X,Y)" ($X \in N_{\mathfrak{F}}$, $Y \in N_{\mathcal{H}}$) is one-to-one mapping from X to Y, because X determines Y. Then, possible hypotheses set H is a finite set as follows:

$$H = \{\text{hypothesis\_of}(x_1, y_1), \cdots, \text{hypothesis\_of}(x_m, y_m)\} \tag{21}$$

where $|H| = |N_{\mathfrak{F}}| = |N_{\mathcal{H}}| = m$.

For a given frame, any selectable hypothesis is defined as an element of a powerset of H.

$$h \in 2^{H} \tag{22}$$

If the cardinality of h is equal to 1, then we call such h a single hypothesis. If the cardinality of h is larger than and equal to 1, then we call such h a multiple hypothesis. Therefore, the final goal is to find a ground substitution of logical variable $X \doteq (X^1, X^2, \ldots, X^n)$ that explains observation (20).

To clarify rational criteria for hypothesis selection and hypothesis identification, we introduce an ordered relation $>$ based on hypothesis_of relations and is_a relations among possible hypotheses set 2^{H}.

$$h_1 > h_2 \triangleq \text{"} h_1 \in 2^{H}, h_2 \in 2^{H}, F_{k,i} + h_2 \vdash h_1 \text{"} \tag{23}$$

These ordered relations constitute a partially-ordered lattice. If one hypothesis is larger than another hypothesis in the meaning of this order, then the larger hypothesis is called an upper hypothesis. Our search strategy is to generate tentative hypotheses systematically along with the frame structure, because this strategy preserves the above order. This reduces the search cost.

Observations O are given by the facts of property-of relations. We introduce a concept of "more general" hypothesis by the following definition:

$$h_1 \in 2^H,\ F+h_1 \vdash O \quad (24)$$

$$h_2 \in 2^H,\ F+h_2 \vdash O \quad (25)$$

$$h_1 > h_2 \quad (26)$$

If a hypothesis h_1 is more general than another hypothesis h_2, h_1 is not an unnecessarily concrete hypothesis. Then, our problem is finding the most general hypothesis which explains observations.

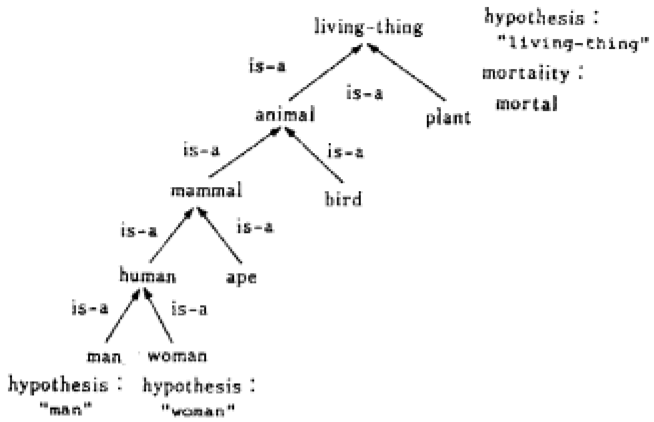

Fig. 4 Knowledge representation of "living-thing" frame

Example 3 : When the knowledge representation in Fig. 4 is given, suppose that a fact "mortality(socrates,mortal)" is observed. As a so-called property inheritance is supposed on this frame, this frame has nine possible hypotheses. Our system is expected to select the most general hypothesis. In this case, it is "hypothesis_of(socrates, living_thing)" for the observation "mortality(socrates,mortal)".

4.2 Hypothesis Selection Problem

We can introduce a criterion for hypothesis identification with ease. Suppose that both hypotheses h_1 and h_2 explain the same observations O.

$$F + h_1 \vdash O \tag{27}$$

$$F + h_2 \vdash O \tag{28}$$

If h_1 explains new observations O', and h_2 does not explain the same observations O', then h_1 is a more realistic candidate for the hypotheses than h_2.

$$F + h_1 \vdash O' \tag{29}$$

$$F + h_2 \nvdash O' \tag{30}$$

If a proprty-of relation has the above properties, (29) and (30), obtaining new observations with such properties can be considered as an experimental test for hypothesis identification. Otherwise, obtaining new observations without such properties can be considered as a belief test for certainty factor, which gives more evidence that given hypotheses, h_1 and h_2, are true after the test.

4.3 Hypothesis Selection Procedure

We implemented a hypothesis-based reasoning system. Fig. 5 shows the main flowchart of our hypothesis selection procedure.

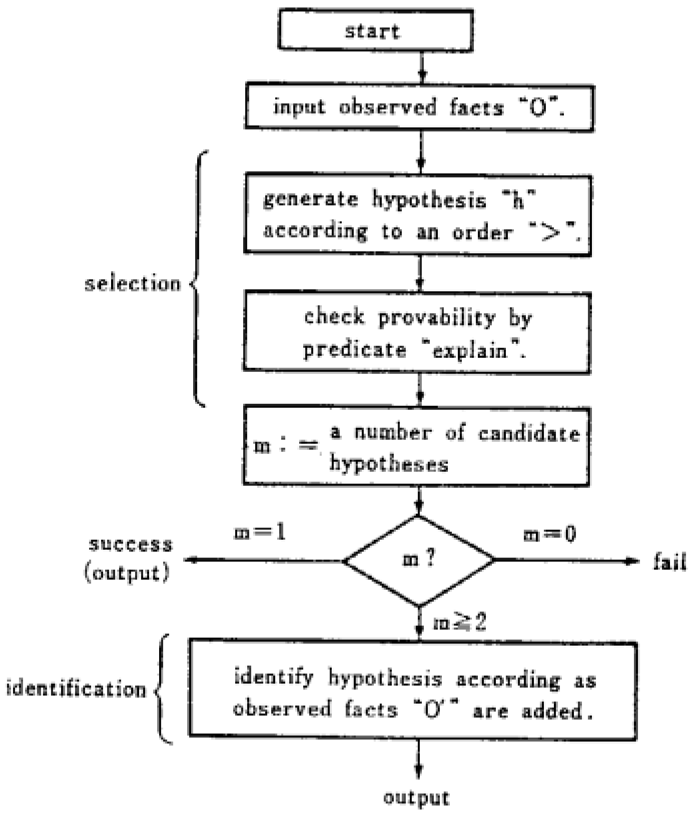

Fig. 5 A flowchart for hypothesis selection

5. Implementation

5.1 Knowledge Representation for Prolog

As our implementation language is DEC-10 Prolog, we adopt a knowledge representation for meta-programming [6,7] in Prolog. The hypothesis_of relation, is_a relation, and property_of relation are described as follows:

hypothesis_of (FrameName, Hypothesis). (31)

is_a (FrameName, SuperFrameName). (32)

property_of (FrameName, HornClause). (33)

where FrameName is the name of the frame in frame structure F, and Hypothesis is an atomic formula corresponding to the name of the hypothesis.

Example 4 : Consider an identification problem of the solution of an unknown ion. For example, these are described as

hypothesis_of ('Zn' ,zn (X)). (34)

is_a ('Zn', transition_element). (35)

property_of('Zn',(precipitate_S_ion(X):- zn (X), acidity (X))). (36)

Expression (36) means "if there exists Zn-ion in any solution X and X is acid, then X causes a precipitate with S_ion".

5.2 Main Prolog Program

The next program gives a shell meta-interpreter [6,7,15] of the system. The first argument of the program is the name of a given world, KB. The second argument is a list of given observations, i.e. goals to be explained by the system. The third argument is a work variable to handle exception knowledge. The fourth argument is a pair of the current FrameName and HypothesisName. The fifth argument is binding informaton to explain the goals.

```
explain(KB, true, Except, X, X) :- !, true.

explain(KB, except(Goal), Except, X, Y) :- !,
        add_list(Goal, Except),
        ( explain(KB, Goal, Except, X, X),
          !, fail
        ; Y=X ).

explain(KB, (P,Q), Except, X, Z) :- !,
        explain(KB, P, Except, X, Y),
        explain(KB, Q, Except, Y, Z).

explain(KB, (P;Q), Except, X, Y) :-
        ( explain(KB, P, Except, X, Y) ;
          explain(KB, Q, Except, X, Y) ).

explain(KB, Goal, Except, X, X) :-
        default_with(Goal, X), !.

explain(KB, Goal, Except, X, Y) :-
        system(Goal), !, Goal ;
        clause_of(KB, (Goal:-Body)),
        explain(KB, Body, Except, X, Y),
        check_exception(KB, Y, Except).

explain(KB, Goal, Except, (Class,X), Y) :-
        is_a(Low, Class),
        hypothesis_of(Low, Z),
        explain(KB, Goal, Except, (Low,Z), Y).
```

Fig. 6 "explain" program in Prolog

The first clause is a termination condition. The second clause is an exception handling part. The third clause is a procedure for conjunctive goals. The fourth clause is a procedure for disjunctive goals. The fifth clause is the case in which the current hypothesis explains goals. The sixth clause is the case in which a normal explanation succeeds without exceptions. The seventh clause is the case in which a lower level hypothesis tries to explain observations along with the frame structure. Note that this program is a revised version of our system [17].

6. Examples

Our system can solve an identification problem of the solution of an unknown ion. The problem is characterized as a compound hypotheses generation and selection problem. When a solution with unknown compound ions and a solution with a known ion are mixed, then it is supposed that the user can observe a result whether the action causes a precipitate or not. Now, the problem is the solution of an unknown ion, X.

Three exapmles are given. Fig. 7 shows given facts and possible hypotheses represented in the is_a hierarchical structure. Given facts consist of both is_a hierarchical

```
%----------------------------------------------
%       is_a
%----------------------------------------------
is_a( hydro , cation ).
is_a( metal , cation ).

 is_a( alkali_metal , metal ).
 is_a( alkaline_earth  , metal ).
 is_a( transition_element , metal ).
                  :

%----------------------------------------------
%       hypothesis
%----------------------------------------------
hypothesis_of( cation, exist_ion(cation) ).

hypothesis_of( hydro , exist_ion(hydro) ).
hypothesis_of( metal , exist_ion(metal) ).

hypothesis_of( alkali_metal , exist_ion(alkali_metal) ).
hypothesis_of( alkaline_earth  , exist_ion(alkaline_earth ) ).
hypothesis_of( transition_element , exist_ion(transition_element) ).
                  :

%----------------------------------------------
%       property
%----------------------------------------------
kb( cation, { conductive :- exist_ion(cation) }).

kb( hydro, { acidity :- exist_ion(hydro) }).
kb( metal, { precip_PO4_ion :- exist_ion(metal) }).
kb( metal, { precip_CO3_ion :- exist_ion(metal),
                        except(exist_ion(alkali_metal)) }).
kb( metal, { precip_OH_ion  :- exist_ion(metal),
                        except(exist_ion(alkali_metal)),
                        except(exist_ion('Ba')) }).
                  :
kb( 'Pb', { precip_SO4_ion :- exist_ion('Pb') }).
kb( 'Ag', { precip_Cl_ion  :- exist_ion('Ag') }).
kb( 'Ag', { precip_halogen_ion :- exist_ion('Ag') }).
kb( 'Pb', { precip_halogen_ion :- exist_ion('Pb') }).
kb( 'Na', { flame_yellow :- exist_ion('Na') }).
kb( 'K' , { flame_violet :- exist_ion('K')  }).
kb( 'Ca', { flame_orange :- exist_ion('Ca') }).
kb( 'Ba', { flame_green  :- exist_ion('Ba') }).
kb( 'Cu', { flame_green  :- exist_ion('Cu') }).
```

Fig. 7 A part of knowledge base F

knowledge and property_of rule knowledge. Note that a property-inheritance is assumed based on the is_a hierarchy.

Example 5 : The first execution example is shown in Fig. 8. For given observations, the system generates the simplest tentative possible explanations. In this case, the generated multiple hypotheses are two single hypotheses. Then, the system asks new observations of the user to select one of such possible hypotheses.

```
| ?- start.
observations : precip_SO4_ion.

Tentative hypotheses for precip_SO4_ion
    alkaline_earth :exist_ion(alkaline_earth )
    Pb:exist_ion(Pb)

Is precip_S_ion right ? y.

Rejected hypotheses ...
    alkaline_earth :exist_ion(alkaline_earth )

verified hypothesis for precip_SO4_ion
    Pb:exist_ion(Pb)

yes
```

Fig.8 An example 1

Example 6 : The second execution example is shown in Fig. 9. For two given observations, the system tries to plan a hypothesis, exist_ion(Ba), which explains them. However, this hypothesis should be back-tracking, because it deduces exceptional knowledge, exist_ion(Ba). The only hypothesis that explains these observations is the hypothesis, exist_ion(Cu).

```
| ?- start.
observations : precip_OH_ion.
observations : flame_green.

hypothesis exist_ion(Ba) deduces an exceptional knowledge exist_ion(Ba)
should be back_tracking ...
only one hypothesis for precip_OH_ion,flame_green .

    Cu:exist_ion(Cu)

yes
```

Fig.9 An example 2

Example 7 : The third execution example is also shown in Fig. 10. For two given observations, the system generates three tentative possible explanations. Fortunately, new observed input by the user rejects these two single hypotheses. After the man-machine interactions to the system, the system selects a rational compound hypothesis, that is, "there exist alkaline_earth ion and transition_element ion in a given solution, X".

```
| ?- start.
observation : precip_SO4_ion(x).
observation : precip_OH_ion(x).

diagnosing...
Tentative hypotheses for precip_SO4_ion(x),precip_OH_ion(x) .....

    exist_Pb(x)                                                   ----- *
    exist_alkaline_earths(x) & exist_transition_element(x)  ----- *
    exist_Ca(x)                                                   ----- *

Is "precip_S_ion(x)" right ?
>> y

Rejected Hypotheses...
   Ca exist_Ca(x)

Is "precip_halogen_ion(x)" right ?
>> n

Rejected Hypotheses...
   Pb exist_Pb(x)

Verified hypothesis .....
[exist_alkaline_earth (x),exist_transition_element(x)]
yes
| ?-
```

Fig. 10 An example 3

7. Conclusion

We described the hypothesis-based reasoning system which generates tentative hypotheses from a set of hypothetical formulae and selects one to form consistent explanations of given observations. The characteristics of our system are as follows:

(1) In our system, hypotheses and given knowledge are represented in an is-a hierarchical structure and appropriate hypotheses for given observations are generated based on the partial order $\succ$ which is the criterion of hypohesis generation. In Theorist, hypotheses are not supposed to have such a structure. Its structure of hypotheses is flat.

(2) If multiple possible hypotheses are to be generated by the system, our system asks new observations of the user to select one possible hypotheses based on the hypothesis selection criterion "the most general". Note that meta-programming methodology enables the user easily to define the criteria for hypothesis generation and selection mechanism.

(3) Our system can make use of negative knowledge (integrity constraints) in the same way as Theorist-S. Our system has simple semantics and can be implemented esily, using meta-programming in Prolog.

The subjects for future study are as follows:

(a) Establishing a logical approach to the problem of hypothesis selection such as an algorithm for finding a query which discriminates competing hypotheses [16];

(b) Integrating a truth maintenance system which enlarges the knowledge acquisition capability in the system.

ACKNOWLEDGEMENTS

We are indebted to Dr. Tosio Kitagawa, the president of IIAS-SIS, and Dr. Hajime Enomoto, the director of IIAS-SIS, for their useful suggestion and warm encouragement.

We are also grateful to their colleagues, Prof. Randy Goebel (Univ. of Waterloo), Mr. Hirohisa Seki and Mr. Toshiaki Takewaki (ICOT), Prof. Setsuo Arikawa (Kyushu University), and Dr. Takashi Yokomori [IIAS-SIS], who contributed the paper for their comments and suggestions.

This is a part of the work in the major R & D of the Fifth Generation Computer Project, conducted under the program set up by MITI, when the first author was the chief researcher of the ICOT Research Center in 1985.

REFERENCES

[1] Arikawa, S. : Inductive Inference and Analogical Reasoning - Theory and Application, in Learning Mechanism on Knowledge (Furukawa, K. and Mizoguchi, F. (eds.)), Kyoritsu Shuppan (1986) [in Japanese]

[2] Blum, L. and Blum, M. : Toward a Mathematical Theory of Inductive Inference, Information and Control, Vol. 28, pp.125-155 (1975)

[3] Goebel, R., Furukawa, K., and Poole, D. : Using Definite Clauses and Integrity Constraints as the Basis for a Theory Formation Approach to Diagnostic Reasoning, Proc. of Third International Conference on Logic Programming, Imperial College, London (1986)

[4] Hayes, P. : The Logic of Frames, in Reading in Artificial Intelligence (Webber, B. and Nilsson, N. (eds.)), Tioga (1979)

[5] Kitakami, H., Kunifuji, S., Miyachi, T. and Furukawa, K. : A Methodology for Implementation of a Knowledge Acquisition System, Proc. of the 1984 International Symposium on Logic Programming, Atlantic City, U.S.A., Feb.6-9 (1985)

[6] Kitakami, H., Kunifuji, S., Miyachi, T. and Furukawa, K. : Knowledge Base Management System in Logic Programming Language Prolog, Journal of Information Processing Society of Japan, Vol.26 No.11, pp.1283-1295 (Nov. 1985) [in Japanese]

[7] Kunifuji, S., Kitakami, H., Miyachi, T., Furukawa, K. : Fundamentals and Applications of Knowledge Engineering (4th)- knowledge base management in PROLOG, Journal of the Society of

Instrument and Control Engineers, Vol. 24 No. 6, pp.539-548 (June 1985) [in Japanese]

[8] Kunifuji, S. : Toward Mechanization of Deductive, Inductive and Abductive Inference, in Learning Mechanism on Knowledge (Furukawa, K. and Mizoguchi, F. (eds.)), Kyoritsu Shuppan (1986) [in Japanese]

[9] Miyachi, T., Kunifuji, S., Kitakami, H. and Furukawa, K. : A Knowlege Assimilation Method for Logic Databases, New Generation Computing, Vol. 2 No. 4, pp.385-404 (1984)

[10] Poole, D., Aleliunus, R. and Goebel, R. : Theorist - a logical reasoning system for defaults and diagnosis, in Knowledge Representation (Cercone, N.J. and McCalla, G. (eds.)), IEEE Press (1985)

[11] Poole, D. : Default Reasoning and Diagnosis as Theory Formation, University of Waterloo, Technical Report CS-86-08 (1986)

[12] Poole, D. and Goebel, R. : Frame Selection based on Defaults and Theory Formation, Private Communication (1986)

[13] Reiter, R. : A Logic for Default Reasoning, Artificial Intelligence, Vol. 13, pp.81-132 (1980)

[14] Reiter, R. and Griscuolo, G.: On Interacting Defaults, IJCAI-81, pp.270-276 (1981)

[15] Seki, H. and Takeuchi, A. : An Algorithm for Finding a Query which Discriminates Competing Hypotheses, ICOT TR-143 (1985)

[16] Shapiro, E.Y. : Algorithmic Program Debugging, The MIT Press (1982)

[17] Turumaki, K., Kunifuji, S. and Furukawa, K. : Hypothesis

Generating System using Meta-programming Techniques, Proc. of Second Conference of Japan Society of Software Science and Technology, Tokyo (Dec. 1985)
[18] Yonemori, Y. : Semiotics of C.S. Peirce, Keiso Shobo (1981) [in Japanese]