知識工学と人工知能 50－10
（1987. 1. 17）

非単調論理を用いた知識ベースの形式化

侯 本慧　　富樫 敦　　野口 正一

東北大学電気通信研究所

本報告では論理の立場で不完全な知識を扱う知識ベース、即ち一階非単調推論を含む様な知識ベースNMKBの形式化を提案する。

”矛盾がない”と言う意味をするオペレーターMを一階述語言語に導入しMを含む様な論理式の出場と伴に新しい論理体系に関する様々な定義を述べる。構成的な手法でいくつかのアルゴリズムを作って、これらのアルゴリズムによってMを含む一階述語論理体系の擬モデルの構成法及び構成された擬モデルの性質、即ち、ここで議論される論理体系での擬モデルに関する妥当性と完全性をフォーマルな、且つ、サイクル無しの体系の元で示す。そして、この擬モデルを用いてNMKBの形式的な定義を与え、NMKBで”信じられる”と言う導出関係が導かれる。

ある与えられた閉じた質問に対してNMKBでの意味は真と偽の代わりに”信じられる”と”信じられない”と対応付ける。

FORMALIZATION OF A KNOWLEDGE BASE USING NON-MONOTONIC REASONING (in English)

Ben-Hui HOU　Atsushi TOGASHIand　Shoichi NOGUCHI
Research Institute of Electrical Communication — TOHOUKU UNIV.
2—1—1, KATAHIRA SENDAI 980 JAPAN

In this paper we shall give the formal definition of a knowledge base with non-monotonic reasoning called non-monotonic knowledge base, denoted by **NMKB**, such that it becomes possible to theoretically consider the knowledge base containing incomplete information with non-monotonic reasoning mechanism.

A default theory is the theory in the first-order language with a special operator **M** which means '*consistency*', informally. The algorithm to construct the *pseudo-model* for the default theory is given. The correctness and the completeness of the pseudo-model for the default theory is shown when the related default theory is assumed to be in the normal form and there is no cycle in it.

The formalization of the **NMKB** shall be carried out using the concept of a pseudo-model. The meaning of a given formula in the **NMKB** is assigned to *believable* or *doubt* instead of *true* or *false.*

1. Introduction and motivations

The motivation of this paper is shown in the following example[5].

$(\forall x)$Bird(x)$\wedge\neg$Penguin(x)$\wedge\neg$Ostrich(x)$\wedge$...
$\supset$ Fly(x) (1)

Bird(Tweety) (2)

(1) means that '*Most birds fly except for penguins, ostrich, the Maltese falcon etc.*' and (2) means that '*Tweety is a kind of bird.*'.

{(1), (2)}$\vdash$Fly(Tweety) (3)

(3) could not be concluded because we cannot make certain that $\neg$Penguin(Tweety) and $\neg$Ostrich(Tweety) *etc.* hold. However (3) is expected to hold, that is Fly(Tweety) is expected to be deduced from (1) and (2). Obviously the concept of deduction has been changed. The method [5] to treat this problem is to modify (1) and (2) as

Bird(x): ***M***Fly(x)/Fly(x) (1*)

$(\forall x)$ Penguin(x)$\supset\neg$Fly(x) (1.1*)

$(\forall x)$ Ostrich(x)$\supset\neg$Fly(x) (1.2*)

......

The intuitive explanation of the operator ***M*** is that '*it is consistent to assume*...'.

{(1*), (1.1*), (1.2*), ..., (2)}$\vdash^*$Fly(Tweety) (3*)

(3*) holds if nothing has been known further, while it would be destroyed by the addition of Penguin(Tweety). However how to define $\vdash^*$ appropriately for the above situation is still yet to be explained.

We would like to clarify in the above example:

*@Using operator **M** to represent the incomplete knowledge such as in* (1);

@Using closed world assumption to deal with negative information such as that $\neg$Penguin(Tweety) *holds if there is no* Penguin(Tweety).

This paper is motivated by the consideration of $\vdash^*$ from the model-theoretic viewpoint and the formalization of the knowledge base with the kind of reasoning such as $\vdash^*$.

The contents of this paper, in brief, enclude: an algorithm by which a pseudo-model is generated for a set of sentences in the first-order language with operator ***M***; the proof of some relative theorems about the algorithm; the formal definition of the **NMKB** using earlier results; explanation of the intuitive meaning of the **NMKB**.

2. Pseudo-model

To begin with we propose an algorithm to construct a pseudo-model for a default theory in the EFO-language detailed later. The pseudo-model for a default theory is a Herbrand model for the extension of the default theory instead of the default theory itself. The model-theoretical explanation for the believability of a formula in default theory [27] is given by the concept of the pseudo-model. A formula in the $\pounds_{efo}$ is believable in the default theory if there is a pseudo-model for the default theory, instead of an extension for the default theory, such that the formula is true in the pseudo-model.

2.1 Preliminaries

Now we shall go to the details of the EFO-language. The EFO-language, the short of Extended First-Order language, denoted by the notation $\pounds_{efo}$, consists of non-monotonic connective ***M*** in addition to the symbols contained in the function-free first-order language.

***Symbols* in $\pounds_{efo}$**

(1) Constant symbols denoted by italics $a, b, c, \ldots$;
(2) Variable symbols denoted by small letters such as $x, y, z, \ldots$;
(3) *n-ary* predicate symbols denoted by capital letters such as P, Q for each integer *n*;
(4) Connectives are

$\wedge$ (and)
$\vee$ (or)
$\neg$ (not)
$\supset$ (implies)
M (consistent)

Terminologies in $\pounds_{efo}$

The definitions of terminologies such as *term*, *formula etc.* are the same as those the first-order and logic are omitted here. We shall only define the special terminologies occurring in this paper, in the following.

Defintion1

Let F be a formula in $\pounds_{efo}$.

1) The formula F is called ***M-free*** formula if there is no occurrence of a connective ***M*** in F. Otherwise, called a ***default***;
2) The ***standard form*** of a default is F_1:***M***F_2/F_3, where F_1, F_2 and F_3 are ***M***-free formulas. It represents the formula $F_1\wedge$***M***$F_2\supset F_3$.
3) **CON**(δ)=F_3 ,
NM(δ)=$\neg F_2$,
PRE(δ)=F_1 where $\delta=F_1$: ***M*** F_2/F_3.
4) Default δ is called in ***normal form*** if
CON(δ)=$\neg$**NM**(δ)
5) Default δ is said to be a ***closed*** default if there are no occurrences of free variable in it.
6) The ***default theory*** DT in $\pounds_{efo}$ is a set of defaults denoted by $DT^{(D)}$ and ***M***-free formulas denoted by $DT^{(MF)}$, which is designated as
$DT = DT^{(D)} \cup DT^{(MF)}$.
7) DT is called a ***standard closed normal default theory*** when every default in DT is in the *standard closed normal* form.

Extension of default theory DT

An extension for a default theory is the set of ***M***-free formulas, some of them are generated by the reasoning in formal logic and the rest of them are obtained by the common reasoning based upon ' *in the absence of any information to the contrary, assume . . .* '. The definition of extension for a default theory is given below. The generation of an extension can be show, as in *Fig.1*, intuitively.

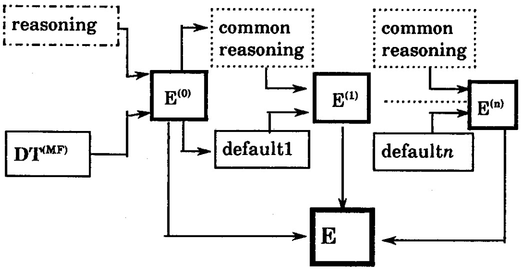

Fig.1

<u>*Definition2*</u>
(1) $E^{(0)}(DT) = Th(DT^{(MF)})$;
(2) $E^{(i+1)}(DT) = E^{(i)}(DT) \cup \{F_2 |$ if there is a default $F_1{:}MF_2/F_2$ such that $F_1 \in E^{(i)}(DT)$, $\neg F_2 \notin E(DT)\}$
$E(DT) = \cup_{i \geq 0} E^{(i)}(DT)$
$E(DT)$ is the ***extension*** for the default theory DT.
<u>*Definition3*</u>
Suppose E is an extension for the default theory DT
$GD^{(E)} = \{F_1{:}MF_2/F_2 |$ where $F_2 \in E$,
and $F_1{:}MF_2/F_2 \in DT^{(D)}\}$.
<u>*Remark*</u> Let δ_1 and δ_2 be two defaults.
$F_1{:}MF_2/F_2$ (δ_1)
$F_1'{:}MF_2/F_2$ (δ_2)
which are combined into one default δ in the normal default theory
$F_1 \vee F_1'{:}MF_2/F_2$ (δ)
<u>*Definition4*</u>
A default theory DT is ***consistent*** *if and only if* there is a consistent extension E for DT.

By definition a default theory DT is consistent if and only if $DT^{(MF)}$ is consistent if DT is in the normal form. Of course the consistent extension is not unique.

The default theory in $\pounds_{efo}$ is a set of ***M***-free formulas and defaults in the $\pounds_{efo}$. Defaults play a role in completing the world incompletely perceived by ***M***-free formulas. Thus it seems impossible to define a model for the default theory. However the extensions for the default theory are the theory completed and closed by defaults. We can see the possibility of establishing a model for the extension of a default theory. In this section we shall give an algorithm to construct a model for a consistent extension and prove that: a model for the extension can be generated by the algorithm if a set of ***M***-free formulas is a consistent extension for a consistent default theory; A default theory is consistent if a model can be generated from the set of ***M***-free formulas and defaults by the algorithm when some conditions are satisfied by the default theory.

Before presenting the algorithm we shall state some notations and definitions. The ***Herbrand interpretation*** I is simply a set of ground positive atomic formulas.

Herbrand interpretations are denoted by I or I with a subscript such as I_i, I_j, formulas by F or F_i, F_j, *etc.*, and ground atomic formula by α or α_i, α_j, *etc.*. $\textsf{Polarity}(\alpha) = +$ if α is positive and $\textsf{Polarity}(\alpha) = -$ if α is negative.

<u>*Definition5*</u>
(1) $I \models F$ if $F \in I$;
(2) $I \models \neg F$ if $F \notin I$;
(3) $I \models F$ if $I \models \neg F_1$ or $I \models F_1$ and $I \models F_2$
where $F = F_1 \supset F_2$

$I \models F$ *if and only if* (1) or (2) or (3) can be satisfied by I and F. We say that the Herbrand interpretation I is a ***Herbrand model*** for F.
<u>*Definition6*</u>
$I_i \equiv I_j$ *iff* $S_\upsilon(I_i) = S_\upsilon(I_j)$, where $S_\upsilon(I) = \{F \mid I \models F\}$
$\equiv$ is called the ***identity*** *relation* among Herbrand interpretations.
<u>*Remark*</u> $S_v(I) = \{\perp\}$ when $I = \emptyset$.
<u>*Definition7*</u>
$I_i \preceq I_j$ *iff* $S_\upsilon(I_i) \subseteq S_\upsilon(I_j)$. $\preceq$ is called the ***ordering*** *relation* among Herbrand interpretations.
<u>*Definition8*</u>
I_i is a Herbrand ***minimal*** *model* for F *iff*
(1) $F \in S_\upsilon(I_i)$ and
(2) there is no such an interpretation I_j, in which
$F \in S_\upsilon(I_j)$, $I_i \sim\equiv I_j$ and $I_j \preceq I_i$.
If there is only one minimal model for F, this minimal model is called a ***unique*** *minimal model* for F.
<u>*Lemma1*</u>
$\preceq$ is a ***partial*** ordering relation among Herbrand interpretations.
【PROOF】 Let I_i , I_j and I_k be three Herbrand interpretations.
(1) Transitivity: $I_i \preceq I_k$ if $I_i \preceq I_j \preceq I_k$.
If $I_i \preceq I_j$ according to the definition of $\preceq$, $S_\upsilon(I_i) \subseteq S_\upsilon(I_j)$. In the same way, if $I_j \preceq I_k$ then $S_\upsilon(I_j) \subseteq S_\upsilon(I_k)$. That is, from $I_i \preceq I_j \preceq I_k$ we can get $S_\upsilon(I_i) \subseteq S_\upsilon(I_j) \subseteq S_\upsilon(I_k)$. According to the transitivity of $\subseteq$, $S_\upsilon(I_i) \subseteq S_\upsilon(I_k)$. Thus, by the definition of $\preceq$, there is $I_i \preceq I_k$;
(2) Antisymmetry: $I_i \equiv I_j$ if $I_i \preceq I_j$ and $I_j \preceq I_i$.
By the definition of $\preceq$, from $I_i \preceq I_j$, there is $S_\upsilon(I_i) \subseteq S_\upsilon(I_j)$ from $I_j \preceq I_i$, there is $S_\upsilon(I_j) \subseteq S_\upsilon(I_i)$ .By the antisymmetry of $\subseteq$, $S_\upsilon(I_i) = S_\upsilon(I_j)$. According to the definition of $\equiv$, $I_i \equiv I_j$;
(3) Reflexity : $I_i \preceq I_i$.

By the reflexity of $\subseteq$, S_υ $(I_i) \subseteq S_\upsilon$ (I_i). By the definition of $\not\prec$ $I_i \not\prec I_i$. ■

Lemma2

(1) There is not always a Herbrand model for any well-formed formula F;

(2) There are always Herbrand minimal models for F if there are Herbrand models for F;

(3) There is not always unique minimal model for any well-formed formula F.

【PROOF】

(1) can be proved by the following example :

$F = \neg F_1 \wedge F_1$ where F is a formula.There is no model for F;

(2) Let $I_1, I_2, \ldots, I_n$ be Herbrand models for formula F, and $I_i \sim\equiv I_j$ for any $I_i, I_j \in \{I_1, I_2, \ldots, I_n\}$. Prove (2) by refutation. Suppose there is no minimal model for F. For any $I_{i_j} \in \{I_1, I_2, \ldots, I_n\}$, according to the definition of minimal model, there must be an $I_{i_{j+1}} I_{i_{j+1}} \in \{I_1, I_2, \ldots, I_n\}$ such that $I_{i_{j+1}} \not\prec I_{i_j}$ Similarly, we can get a chain By *Lemma1*, $\not\prec$ has transitivity, thus $I_{i_1} \not\prec I_{i_n}$ and as shown in the chain $I_{i_n} \not\prec I_{i_1}$, $\not\prec$ has antisymmetry. Thus $I_{i_1} \equiv I_{i_n}$. $I_{i_1} \equiv I_{i_n}$ contradicts the supposition that $I_i \sim\equiv I_j$ for any $I_i, I_j \in \{I_1, I_2, \ldots, I_n\}$;

$$\not\prec I_{i_1} \not\prec I_{i_2} \not\prec \ldots \not\prec I_{i_j} \not\prec I_{i_{j+1}} \not\prec \ldots \not\prec I_{i_{n-1}} \not\prec I_{i_n}$$

(3) can be proved by the following example.

$F = F_1 \vee F_2 \subset Q$

$I_1 = \{F_1\}$

$I_2 = \{F_2\}$

$I_3 = \{F_1, F_2\}$ ....

I_1, I_2, I_3 are Herbrand models for F and I_1, I_2 are two minimal models for F. ■

Lemma3

If F is in a definite clausal form and there are Herbrand models for F, then there is a unique minimal model for F.

【PROOF】Let F be a definite clause. $P_1 \wedge P_2 \wedge \ldots \wedge P_n \supset Q$

Firstly, we can construct an interpretation I_i for F, $I_i = \{Q\}$. Now, we shall prove I_i is the unique minimal model for F.

① I_i is a model for F;

② I_i is a minimal model for F.

For any model I_j for F, $I_i \sim\equiv I_j$, $S_v(I_i) = \{F, Q, \neg P_1, \ldots, \neg P_n, \ldots\}$. If $I_j \not\prec I_i$ then $S_\upsilon(I_j) \subseteq S_\upsilon(I_i)$ because of $I_i \sim\equiv I_j$ that means there must be an element s, $s \in S_\upsilon(I_i)$ but $s \notin S_\upsilon(I_j)$. However $F \in S_\upsilon(I_j)$

(1) if there is an $\neg P_k$, $\neg P_k \in S_\upsilon(I_i)$ but $\neg P_k \notin S_\upsilon(I_j)$ then $P_k \in S_\upsilon(I_j)$ which contradicts $S_\upsilon(I_j) \subseteq S_\upsilon(I_i)$;

(2) if $Q \notin S_\upsilon(j)$ then $I_j = \varnothing$, it contradicts with the assumption that I_j is a model for F.

Therefore we can say that

$I_i = \{Q\}$ is a minimal model for F ;

③ $I_i = \{Q\}$ is a unique minimal model for F. This is trivial by the above proof. ■

2.2 Algorithm to construct pseudo-model

The purpose of this section is to establish an algorithm to construct an interpretation for the standard closed normal default theory, which is actually a model for its consistent extension shown in the theorem1. Preparatory to the presentation of the main algorithm, the algorithm to generate an interpretation for a set of ***M***-free formulas is proposed firstly.

Some of the notations used are explained below.

Char(*Algo-Name*, S_{in}, S_{out}) means that the set of S_{out} is the output set obtained by applying the algorithm *Algo-Name* on the input set of S_{in}.

Output(*Algo*−*Name*(S_{in})) $= S_{out}$

Im(F) $= \bigwedge_{1 \le i \le n} P_i$

Imd(F) $= \bigvee_{1 \le j \le n} Q_j$ where $F = \bigwedge_{1 \le i \le n} P_i \supset \bigvee_{1 \le j \le n} Q_j$.

<u>Algorithm to generate an interpretation for a set of *M*-free formulas:</u>

The operator **Generator1** generates the ground instances of the non−ground atomic formula. **Generator1**(P) is the set of the ground atomic formulas obtained by applying **Generator1** on the non-ground atomic formula P. We shall not enter into the details of **Generator1**.

We shall present the **generator21 − Algorithm** shorted by **G21 − A**. Firstly we shall define a table called **search table**.

$F = P_1, P_2, \ldots, P_n$				$S = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$			
F△S	P_1	P_2	…	P_j	…	P_{n-1}	P_n
α_1	θ_{11}	$P_2\theta_{11}$	…	$P_j\theta_{11}$	…	P_{n-1} θ_{11}	$P_n\theta_{11}$
α_2	θ_{21}	$P_2\theta_{21}$	…	$P_j\theta_{21}$	…	P_{n-1} θ_{21}	$P_n\theta_{21}$
…	…	…	…	…	…	…	…
α_k	θ_{k1}	$P_2\theta_{k1}$	…	$P_j\theta_{k1}$	…	P_{n-1} θ_{k1}	$P_n\theta_{k1}$
…	…	…	…	…	…	…	…
α_{m-1}	$\theta_{m-1,1}$	P_2 $\theta_{m-1,1}$	…	P_j $\theta_{m-1,1}$	…	P_{n-1} $\theta_{m-1,1}$	P_n $\theta_{m-1,1}$
α_m	$\theta_{m,1}$	P_2 $\theta_{m,1}$	…	$P_j\theta_{m,1}$	…	P_{n-1} $\theta_{m,1}$	P_n $\theta_{m,1}$

Where the P_i are atomic formula and S is the set of ground atomic formulas. θ_{k1} is the unifier of P_1

and α_k. $P_j\theta_{k1}$ is the result obtained by applying θ_{k1} on P_j.

The first column in the table is called **base column** and the second row in the table is called the **base row**; **F△S** is a search table with the base column **S** and base row **F**. The first element in the base row is called the **current element** and is denoted by **CurrE(F△S)**. The column headed by **CurrE(F△S)** is called **substitution column** denoted by **Sub(F△S)**; The column headed by P_j is denoted by **Col**(P_j), the row haded by α_i **Row**(α_i) and the *i*th row in the table **F△S** by the notation **Row(F△S**, i); The *i*th element in a set of S is denoted by **Ele**(S, i).

Generator21 – Algorithm

Char(G21 – A, S_{in}, S_{out})
  $S_{in} = <F, S>$
    $F = \wedge_{1 \leqq i \leqq r} P_i$, P_i is an atomic formula
      Polarity(P_i) = +
    $S = \{\alpha_i \mid$ **Polarity**$(\alpha_i) = +\}$
  $S_{out} = \{\theta \mid$ if $P_i\theta \in S$, for all P_i in F$\}$

Step1 $F_1 \Leftarrow \{\varnothing, \varnothing\} \cup \{P_i \mid$ for all P_i in F and $P_i \notin S$, $i \geqq 3\}$
  $S_1 \Leftarrow S$
  goto *Step2*;

Step2 If $\|$**Col(G – T**(F_1, S_1))$\| = 2$
  then $\psi \Leftarrow \{$**Sub(G – T**(F_1, S_1))$\} \cup \{\Psi\}$
    Cancel **Row(G – T**(F_1, S_1), k)
  else if **Ele(Row(G – T**(S_1, F_1), k), j) is ground
    and **Ele(Row(G – T**(S_1, F_1), k), j)$\notin$S
      then Cancel **Row(G – T**(S_1,F_1),k)
      else if $\|$**Row(G – T**(S_1, F_1))$\| > 2$
        then £ ⇐ **G – T**(S_1, F_1) ∪ £
          goto *Step3*
        else goto *Step3*;

Step3 If £ ≠ ∅ and Table ∈ £
  then for each **Row**(Table, i) do
    $F_1 \Leftarrow$ **Row**(Table, i)
    $S_1 \Leftarrow S$
    goto *Step2*
  else goto *End*;

End ▲

In this algorithm **G – T** is an operator to generate the new search table when the current element is unifiable with each element in the base column. **G – T**(F, S) is a search table that consists of the rows **Row**(α_i), for each α_i, $\alpha_i \in S$.

Row(α_i) = {α_i,**Ele**(F,2)$\theta_{i,1}$,**Ele**(F,4)$\theta_{i,1}$,...,**Ele**(F,n)$\theta_{i,1}$}

Lemma4

For any θ, $\theta \in$ ***Output(G21 – A**(F, S))* *iff* $P_i\theta \in S$, $1 \leqq i \leqq r$, *where* $F = \wedge_{1 \leqq i \leqq r} P_i$.

【PROOF】 Firstly, the *only–if–half* of the lemma, if $\theta \in$ **Output(G21 – A**(F, S)) then $P_i\theta \in S$, which is trivial by algorithm itself; Next, we shall prove the *if–half* of the lemma for any P_i, if $P_i\theta \in S$ then $\theta \in$ **Output(G21 – A**(F, S)).

① When i = 1, that is $F = P_1$, $F_1 = <\varnothing, \varnothing, P_1>$, $S_1 = S$ by *Step1*. The table has been created by **G – T**(F_1, S_1) in *Step2*, $\|$**Col(G – T**(F_1, S_1))$\| = 2$ and **Output(G21 – A**(F_1, S_1)) can be returned only from one step. Thus the *if–half* of the lemma is true according to the generation method in **G – T**;

② Let F be $P_1 \wedge P_2$. We shall prove **Output(G21 – A**(F_1, S)) = **Output(G21 – A**(F_2, S)), where $F_1 = <P_1, P_2>$ and $F_2 = <P_2, P_1>$. Both **Output(G21 – A**(F_1, S)) and **Output(G21 – A**(F_2, S)) are obtained from two steps in *Step2*. Now we assume that the substitutions obtained by the first step and the second step are $\theta_1^{(1)}$, $\theta_1^{(2)}$ and $\theta_2^{(1)}$, $\theta_2^{(2)}$ for F_1, and F_2 respectively. We shall prove $\theta_1^{(2)} = \theta_2^{(2)}$. For any $\theta_i \in \theta_1^{(2)}$, there are $P_1\theta_i \in S$ and $P_2\theta_i \in S$. For any $\theta_j \in \theta_2^{(2)}$, there are $P_1\theta_j \in S$ and $P_2\theta_j \in S$. If there is an θ, $\theta \in \theta_1^{(2)}$ but $\theta \notin \theta_2^{(2)}$, then there must be $P_1\theta \notin S$ or $P_2\theta \notin S$. It contradicts the previous conclusion. Thus $\theta_1^{(2)} \subseteq \theta_2^{(2)}$. The converse can be proved in the same way. Therefore $\theta_1^{(2)} = \theta_2^{(2)}$;

③ Now we suppose that the *if–half* of the lemma is true when $F = <P_1, P_2, \ldots, P_n>$;

④ We shall prove the *if–half* of the lemma is also true for $F = <P_1, P_2, \ldots, P_n, P_{n+1}>$. By the result proven in ② the order F can be changed into the form of $F = <P_{n+1}, P_1, P_2, \ldots, P_n>$ without affecting the results obtained by the algorithm. For the sake of convenience, we assume $F' = <P_1, P_2, \ldots, P_n>$. Then for any P_i in F' $\theta \in$ **Output(G21 – A**(F′, S)) if $P_i\theta \in S$. Therefore it is impossible that there is an θ, such that for any P_i, $1 \leqq i \leqq n+1$, $P_i\theta \in S$ but $\theta \notin$ **Output(G21 – A**(F, S)) by ③ Together with ①, the *if–half* of the lemma has been proven. ■

Generator2 – Algorithm

Char(G2 – A, S_{in}, S_{out})
  $S_{in} = <F, S>$
    $F = \wedge_{1 \leq i \leq n} P_i \supset \vee_{1 \leq j \leq m} Q_j$
      where P_i and Q_j are atomic formulas;
        Polarity(P_i) = +
        Polarity(Q_j) = +.
    $S = \{\alpha_i \mid$ **Polarity**$(\alpha_i) = +\}$
  $S_{out} = \{S_i \mid S_i = \{\alpha_j \mid$ **Polarity**$(\alpha_j) = +\}\}$

Step1 $\Omega \Leftarrow$ **Output(G21 – A(Im**(F), S));

Step2 Choose one Q_j in $\vee_{1 \leq j \leq m} Q_j$ do
  Step2.1 if there is a S_l, $S_l \in S$ such that $Q_j\theta_k \in S_l$,
      for all j, $1 \leqq j \leqq m$, for any $\theta_k \in \Omega$
    then $S_{new} \Leftarrow \{S_l\} \cup S_{new}$
      $S \Leftarrow S - S_l$
      if $S = \varnothing$
        then goto *End*
        else goto *Step2*
    else goto *Step2.2*;
  Step2.2 for all $Q_j\theta_k \notin S_l$ do
      $S_l \Leftarrow (Q_j\theta_k)^* \cup S_l$
      $S_{new} \Leftarrow \{S_l\} \cup S_{new}$
      $S \Leftarrow S - S_l$
      goto *Step2*;   *End* ▲

Lemma5

S_i is a Herbrand model for F if

*$S_i \in$ **Output(G2 - A(F, S))**, where* F is the set of formulas with the form $\wedge_{1 \le i \le n} P_i \supset \vee_{1 \le j \le m} Q_j$, and S is the set of positive ground atomic formulas.

*S = **Generator1(Im(F))***

【PROOF】 It is trivial from algorithm. ■

Lemma6

S_i is a minimal Herbrand model for F iff

*$S_i \in$ **Output(G2 - A(F, S))**, where*

$F = \wedge_{1 \le i \le n} P_i \supset Q$,

*S = **Generator1(Im(F))***

【PROOF】 The proof of the *if−half* of the lemma, similar to the proof ② of the lemma3, is omitted here; The *only−if−half* of the lemma can be proved by the uniqueness of the minimal model for the set of the definite clauses according to the lemma3. ■

Definition9

Let P and Q be two atomic formulas. P and Q are called ***polymorphic*** *iff* there are occurrences of the same predicate symbols in P and Q.

Definition10

$$\wedge_{1 \le i1 \le n1} P_{i_1}^{(1)} \supset \vee_{1 \le j1 \le m1} Q_{j_1}^{(1)},$$
$$\wedge_{1 \le i2 \le n2} P_{i_2}^{(2)} \supset \vee_{1 \le j2 \le m2} Q_{j_2}^{(2)},$$
$$\cdots\cdots$$
$$\wedge_{1 \le ir \le nr} P_{i_r}^{(r)} \supset \vee_{1 \le jr \le mr} Q_{j_r}^{(r)}$$

is a ***cycle*** *iff* for each $\vee_{1 \le jk \le mk} Q_{j_k}^{(k)}$ and $\wedge_{1 \le i_{K+1} \le n_{k+1}} P_{i_{k+1}}^{(k+1)}$, $1 \leqq k \leqq r$, there is at least one $Q_{j_s}^{(s)}$ in $\vee_{1 \le jk \le mk} Q_{j_k}^{(k)}$, at least one $P_{i_t}^{(t)}$ in $\wedge_{1 \le iK+1 \le nk+1} P_{i_{k+1}}^{(k+1)}$ is polymorphic, $1 \leqq s \leqq n_k$, $1 \leqq t \leqq n_{k+1}$,and at least one $P_u^{(1)}$ in $\wedge_{1 \le i1 \le n1} P_{i_1}^{(1)}$, $1 \leqq u \leqq n_1$, at least one $Q_v^{(r)}$ in $\vee_{1 \le jr \le mr} Q_{j_r}^{(r)}$, $1 \leqq v \leqq m_r$, $P_u^{(1)}$ and $Q_v^{(r)}$ are polymorphic.

Selector(S) = $\wedge_{1 \le ik \le nk} P_{ik}^{(k)} \supset \vee_{1 \le jk \le mk} Q_{j_k}^{(k)}$, if there is no polymorphic atomic formulas in $\wedge_{1 \le ik \le nk} P_{i_k}^{(k)}$ and $\vee_{1 \le jt \le mt} Q_{j_t}^{(t)}$ for each t, $1 \leqq t \leqq r$;
∅, otherwise.

where $S = \{\wedge_{1 \le ik \le nk} P_{i_k}^{(k)} \supset \vee_{1 \le jk \le mk} Q_{j_k}^{(k)} \mid 1 \leqq k \leqq r\}$

Lemma 7

***Selector**(S) ≠ ∅ if there is no cycle in S and S ≠ ∅.*

Sub-Algorithm:

Char(S - A, S_{in}, S_{out})

S_{in} = <A_1, A_2, R>

$A_1 = \{a_i \mid$ **Polarity**$(a_i) = +\}$

A2 is the set of non-ground atomic formulas.

R is the set of formulas of the form

$\wedge_{1 \le i \le n} P_i \supset \vee_{1 \le j \le m} Q_j$ and

Polarity$(P_i) = +$, **Polarity**$(Q) = +$

$S_{out} = \{I_j \mid I_j = \{a_i \mid$ **Polarity**$(a_i) = +\}\}$

Step1 I ⇐ A_1

goto *Step2*

Step2 for all $a \in A_2$ do

I ⇐ I ∪ (**Generator1**(a) − I)

$A_2 \Leftarrow A_2 - a$

if $A_2 \neq \emptyset$ then goto *Step2*

else goto *Step3*

Step3 F⇐**Selector**(R)

if F = ∅

then return **Fail**

else do

for any $I_i \in I$, $1 \leqq i \leqq \|I\|$

for each $S_k \in$ **Output(Q2 - A**(F,{I_i})),

$1 \leqq k \leqq \|$**Output(Q2 - A**(F,{I_i}))$\|$

$I_i^{(k)} \Leftarrow \{S_k - I_i\} \cup \{I_i\}$

$I \Leftarrow \cup_{k,i} \{I_i^{(k)}\}$

R ⇐ R − F

if R ≠ ∅

then goto *Step3*

else goto *End*

End ▲

Lemma8

*For any I_i, $I_i \in$ **Output(S - A(V))**, and I_i is a minimal Herbrand model for V, where $V = A_1 \cup A_2 \cup R$, if there is no cycle in V and R is a set of definite clauses.*

【PROOF】 It follows *Lemma6* and *Lemma7*. ■

Restrictor - Algorithm

Char(R - A, S_{in}, S_{out})

S_{in} = <F, S>

$F = \wedge_{1 \le i \le n} P_i \supset Q$

where P_i and Q are atomic formulas;

Polarity$(P_i) = +$, **Polarity**$(Q) = -$.

$S = \{a_i \mid$ **Polarity**$(a_i) = +\}$

S_{out} = S or **Fail**

Step1 Ω ⇐ **Output(G21 - A**($\wedge_{1 \le i \le n} P_i$, S));

Step2 For each P_i, $1 \leqq i \leqq n$, $\theta \in \Omega$

if $P_i\theta \in S$

then if $(\neg Q)\theta \in S$

then return **Fail**

else $S \Leftarrow S \cup Q\theta$

goto *Step3*

Step3 Ω ⇐ Ω − θ

if Ω ≠ ∅

then goto *Step2*

else goto *End*

else goto *Step3*

End Return S ▲

Lemma9

***Output(R - A(F, S))** is a Herbrand model for F.*

OP$_m$(S, F) = T if ***Output(G2 - A(**F, S**)** =

⊥ otherwise.

OP$_{nm}$(S, F) = T if ***Output(G2 - A(**F, S**)** ⊈ S

⊥ otherwise. Where F is a ***M***-free formula and S is a set of ground atomic formulas.

ALGORITHM

Char(A, S_{in}, S_{out})

$S_{in} = <W, W', \delta>$

W' is the set of formulas of the form $\wedge_{1\le i\le n} P_i \supset \neg Q\}$

W is the set of formulas of the form

$\wedge_{1\le i\le n} P_i \supset \vee_{1\le j\le m} Q_j\}$

δ is the set of defaults of the form

$F_1{:}\boldsymbol{M}F_2/F_3$ where F_1, F_2 and F_3 are of the form

$\wedge_{1\le i\le n} P_i \supset \vee_{1\le j\le m} Q_j\}$

$S_{out} = \{ M_i \mid M_i$ is a set of ground positive atomic formulas$\}$

Step1 $M^{(0)} \Leftarrow$ **Output(R – A(**W'**, Output(S – A(**W**)))**

$R^{(0)} \Leftarrow \delta$;

Step2 For each $M_j \in M^{(i)}$ do

Begin

Step2.1 $M_j^{(k)} \Leftarrow M_i \cup (S_k - M_j)^{**}$ where

$1 \leqq k \leqq \|$**Output(Q2 – A(CON(**δ_{i+1}**),** M_i**))**$\|$

$S_k \in$ **Output(Q2 – A(CON(**δ_{i+1}**),** M_i**))**

if there is a default δ_{i+1} in $R^{(i)}$ such that

OPm(M_i**, PRE(**δ_{i+1}**))**

$\wedge$**OPnm(**M_i**, NM(**δ_{i+1}**))**

$\wedge$**OPnm(**M_i**, CON(**δ_{i+1}**))**

$M_j \Leftarrow \cup_k M_j^{(k)}$

Step2.2 $M_j \Leftarrow M_j$

if **OPm(**M_i**, PRE(**δ_{i+1}**))**

$\wedge$**OPnm(**M_i**, NM(**δ_{i+1}**))**

$\wedge$**OPm(**M_i**, CON(**δ_{i+1}**))**

Step2.3 Cancel M_j

if **OPm(**M_i**, PRE(**δ_{i+1}**))**

$\wedge$**OPm(**M_i**, NM(**δ_{i+1}**))**

$M^{(i+1)} \Leftarrow \cup_j M_j$

$R^{(i+1)} \Leftarrow R^{(i)} - \delta_{i+1}$ **End**

Step3 Return M if $R^{(i)} = \varnothing$ ▲

The following theorem shows that the model for the consistent extension can be generated by the above algorithm.

<u>*Lemma*</u>

*If DT is a consistent default theory and E is a consistent extension for DT, then $M \models E$, where $M \in$ **Output(A(**$DT^{(MF)}$, $GD^{(E)}$**))** and there is no cycle in DT.*

【PROOF】 Assume that

$W = DT^{(MF)}$,

$\delta = GD^{(E)}$ in the algorithm .

For any $F \in E$

(1) If $F \in Th(W)$

by the step (1) in the algorithm

$M^{(0)} \models F$ then

$M \models F$;

(2) Suppose for all $F \in E_i$

$M^{(i)} \models F$ then

$M \models F$;

(3) $E_{i+1} = E_i \cup \{F_2 | F_1{:}\boldsymbol{M}F_2/F_2$, where $F_1 \in E_i$ and $\neg F_2 \notin E\}$

for all $F \in E_{i+1}$

if $F \in E_i$ by (2) $M^{(i)} \models F$

if $F \notin E_i$ by the definition of E_{i+1}

there is a default δ in the $GD^{(E)}$ $F_1{:}\boldsymbol{M}F/F$ where $F_1 \in E_i$ means that $M^{(i)} \models F_1$; $\neg F \in E$ means that there is no i, such that $M^{(i)} \models \neg F$, thus $M^{(i)} \models \neg F$. Therefore by algorithm we get

$M^{(i+1)} \Leftarrow M^{(i)} \cup S_k$

$S_k \in$ **Output(G2 – A(CON(**δ**),** $M^{(i)}$**))**

$R^{(i+1)} \Leftarrow R^{(i)} - \delta$

$S_k \models F$ by lemma5

Thus $M^{(i+1)} \models F$. ■

W is a subset of the $DT^{(MF)}$ and D is a subset of the $DT^{(D)}$.

$S^{(0)}(W, D) = DT^{(MF)}$;

$S^{(i+1)}(W, D) = S^{(i)} \cup \{$**CON(**$\delta$**)**$|$if there is a default δ in D such that $S^{(i)} \vdash$ **PRE(**δ**)**, and $S^{(i)} \sim\vdash$ **NM(**δ**)**$\}$

$S^{(W,D)} = \cup_{0 \leqq j \leqq n} S^{(j)}(W, D)$.

<u>*Theorem*</u>

DT is a consistent normal default theory if and only if there is a $S^{(W,D)}$ such that

$E = Th(S^{(W,D)})$ and

*$S_k \in$ **Output(A(**W, D**))**, $S_k \models E$*

where E is a consistent extension for DT and there is no cycle in DT.

【PROOF】 The *if – half* is trivial by the definition of the consistency from the default theory; The *only – if – half* follows the above lemma. ■

<u>*Definition11*</u>

Let **DT** be a consistent normal default theory. S_k is called a ***pseudo – model*** for **DT**, denoted by the notation **Mps(DT)**, where $S_k \in$ **Output(A(**W, D**))**.

Suppose **DT** is a default theory, in which all defaults in $DT^{(D)}$ are of the form:

$:\boldsymbol{M}\neg P/\neg P$,

This is called the *closed world assumption.*

Now we can reason out a conclusion from the theorem that the model for a set W of ***M***–free formulas is identical to the pseudo–model for the default theory composed of W and $DT^{(D)}$ in which each default is a closed world assumption. This is stated in the following corollary.

<u>*Corollary*</u>

*The model for a set W of **M**-free formulas is a pseudo – model for DT , where*

$DT^{(MF)} = W$

*$DT^{(D)} = \{:\boldsymbol{M}\neg F/\neg F \mid F$ is any **M**-free formula$\}$*

【*PROOF*】 It is trivial by the algorithm. ■

3. Formal definition of the NMKB

An **NMKB** can be informally defined as a set of default theories which are changing with the knowledge assimilation. The formal definition of **NMKB** will be given as follows:

NMKB $= <U, TRG, \underline{\models}>$

$U = \{DT_1, DT_2, \ldots, DT_i, \ldots\}$

U is called the universe of the NMKB;

TRG = {<i, j> | where DT_i, $DT_j \in U$ and for any formula F, if $DT_j \models F$ then $DT_i \models F$}

TRG describes the relationships among all elements DT_i in the universe U.

$\models$ represents for any pseudo-model $M_{ps}(DT_i)$ for DT_i, $DT_i \in U$.

$M_{ps}(DT_i) \models F$ *iff* $F \in M_{ps}(DT_i)$;

$M_{ps}(DT_i) \models \neg F$ *iff* $M_{ps}(DT_i) \sim\models F$;

$M_{ps}(DT_i) \models F_1 \vee F_2$ *iff* $M_{ps}(DT_i) \models F_1$ or $M_{ps}(DT_i) \models F_2$;

$M_{ps}(DT_i) \models F_1 \wedge F_2$ *iff* $M_{ps}(DT_i) \models F_1$ and $M_{ps}(DT_i) \models F_2$;

$M_{ps}(DT_i) \models F_1 \supset F_2$ *iff* $M_{ps}(DT_i) \models \neg F_1$ or $M_{ps}(DT_i) \models F_2$;

$M_{ps}(DT_i) \models \exists x F(x)$ *iff* there is a constant c such that $M_{ps}(DT_i) \models F(c)$;

$M_{ps}(DT_i) \models \forall x F(x)$ *iff* for all constants c_i, $i = 1, \ldots, n$, $M_{ps}(DT_i) \models F(c_i)$;

$M_{ps}(DT_i) \models MF$ *iff* $M_{ps}(DT_i) \models F$

Where U is a set of default theories and TRG is a transformation graph among the default theories in U. In other words, the set of pairs <i, j>indicates that DT_j is a default theory accessible from DT_i. That is , DT_i and DT_j are two default theories in the U, $M_{ps}(DT_i)$ and $M_{ps}(DT_j)$ are two pseudo-models for DT_i and DT_j respectively such that if $M_{ps}(DT_j) \models F$ then $M_{ps}(DT_i) \models F$.

F is *believable* in the NMKB, denoted by NMKB$\blacktriangleright$F *if and only if* there is a $DT_i \in U$ such that $M_{ps}(DT_i) \models F$, or there is a pair <i, j>$\in$TRG such that $M_{ps}(DT_j) \models F$. Otherwise F is *doubtful* in the NMKB. The elements occurring in the universe U of NMKB could be explained as the contexts varying with the change of the time. As shown in the *Fig.2* the original context of the given knowledge base KB is KB_{t_0}, U = <KB_{t_0}, KB_{t_1}, KB_{t_2},...,KB_{t_i},...>, where KB_{t_i} is the context of the KB at the time point t_i.

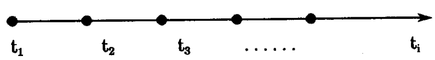

Fig.2

It is reasonable to explain the elements of the universe U in NMKB as different databases or knowledge bases in different frameworks such as relation, first-order logic *etc.*, when we assume the derivation relation $\blacktriangleright$ as a set of derivation relations corresponding to each element in U in spite of that the definition of NMKB has been given under the hypotheses that each element in U is a default theory and the derivation relation $\blacktriangleright$ is defined based upon the pseudo-model.

4. Conclusion

The problem of non-monotonic reasoning in the deductive knowledge bases is caused by the incompletely perceived properties used to classify the concepts. Furthermore knowledge assimilation is difficult when some new discoveries have been made. It seems that an object-oriented language is appropriate for the knowledge representation in order to solve this problem.

in logic language	in object-oriented language	concept-relationship model
predicate	method selector *or* class name	attribute *or* concept name
argument	class	concept *or* relationship
predicate bird	class name Bird	concept BIRD
predicate penguin	class name Penguin	concept PENGUIN
predicate canFly	method seletor CANFLY	attribute canFly
variable *x	the instance of Bird or Penguin	entity of the concepts BIRD or PENGUIN
canFly(*x)← bird(*x). ¬canFly(*x)← penguin(*x)	Bird *superclass* Object CANFLY ↑ true Penguin *superclass* Bird CANFLY ↑ false	There is a *is-a* relationship from PENGUIN to BIRD.

REFERENCES

[1] CHIN-LIANG CHANG, RICHARD CHAR-TUNG LEE Symbolic Logic and Mechanical Theorem Proving. ACADEMIC PRESS New York and London.

[2] Drew McDermott, Jon Doyle Non-monotonic Logic I *Artificial Intelligence* 13 (1980), 41-72.

[3] Joseph R. SHOENFIELD Mathematical Logic.

[4] Keith L. Clark [1978] Negation As Failure. In *logic and Databases*, H.Gallaire and J. Minker, Eds. Plenum, New York, PP 293-322.

[5] REITER R. [1980] A Logic for Default Reasoning *Artificial Intelligence* 13, 1, 2 (1980), 81-132.