

## 知識ベースの段階的詳細化 に適した知識表現モデル

石田 亨 森原 一郎 古屋 博行 服部 文夫

(NTT 電気通信研究所)

### あらまし

世の中に存在する未整理な知識から段階的に知識ベースを構築するのに適した知識表現の初期検討結果を報告する。この知識表現は論理を基礎としており、以下の特徴を持つ。

- (1) 概念と関係の包含を用いて様々な詳細度の知識を表現する。
- (2) 詳細化過程で明らかとなった関係の性質を容易に追加定義できる。

本資料の中では、関係の性質として、定義域、値域、包含、リダクション規則を取上げ、その定義方法を議論する。また、この範囲での知識の表現能力とこれまでの知識表現システムの表現能力を、主として継承機能の側面から比較する。

"A Representation Model for Stepwise Knowledgebase Refinement" (in Japanese)  
by Toru Ishida, Ichirou Morihara, Hiroyuki Furuya, and Fumio Hattori  
(NTT Electrical Communication Laboratories, Yokosuka-shi, 238-03, Japan)

This paper describes a new representation model, which is suited to construct knowledgebases in step-by-step fashion. The model is logic-based and has the following advantages.

- (1) Knowledge in various refinement levels is represented by using subsumption of concepts and relations.
- (2) Features of concepts and relations can be easily added to their definitions during the process of stepwise knowledgebase refinement.

The comparison between this model and other representation models is also discussed in this paper.

## 1. まえがき

種々の概念や概念間の関係を表現するモデルとして、セマンティックネットワークが研究されている。従来から関係(リンク)の意味があいまいであるという欠点が指摘されてきたが、PSN<sup>[1]</sup>やFox<sup>[2]</sup>による継承規則の定義、及びKRYPTON<sup>[3]</sup>、DCKR<sup>[4]</sup>、IXL<sup>[5,6]</sup>等、論理に基礎を置く研究により、様々な関係を厳密に定義することが可能となりつつある。しかしながら、世の中に存在する知識は未整理かつあいまいなため、厳密に定義されたセマンティックネットワークを構築するのは容易なことではない。セマンティックネットワークの構築には、知識をまずおおまかに整理し、必要に応じて知識ベースの詳細化が行えることが要求される。筆者らは、知識ベースの段階的詳細化を可能とすることを目的に、様々な詳細化のレベルにある知識の表現方法、及び詳細化を支援する環境を検討中であるが、本資料ではその内、知識表現についての初期検討結果を報告する。

ここで報告する知識表現は1階論理のサブセットで、定義と事実とからなる。基本的な考え方を以下に示す。

(1) 概念と関係の包含を用いて様々な詳細度の知識を表現する。

概念間や関係間の包含により形成される束を用いて様々な詳細度の知識を表現する。例えば、『何かと何かの間に何らかの関係がある』というような、人間があいまいと感じる知識から、『太郎と花子は結婚している』というような明確な知識までを表わす。

(2) 詳細化過程で関係の性質を容易に追加定義できる。

詳細化の途中段階では、関係の持つ意味が完全に明らかになっているとは限らない。そのため、関係をあらかじめ厳密に定義することは困難であると考えられる。段

階的詳細化に適した関係の定義の方法としては、①関係の持つ性質が明らかとなった時点で順次その性質を追加定義できる、②関係の包含の束で上位に位置する(包含する)関係の性質が下位に位置する(包含される)関係に継承される、ことが望まれる。本資料では関係の性質として、①関係の包含と、②2個の関係から1個の関係を導く関係のリダクションをとりあげ、上記の条件を満足する定義方法を述べる。

(3) 否定知識を扱う。

詳細化過程では、全ての情報は入手済ではない。従って、『太郎と花子は親戚でない』という知識は有用である。また、この否定知識は知識の矛盾の検出に役立つ。

この他、詳細化過程で大量に生じる冗長な知識を管理(圧縮<sup>[7]</sup>あるいは展開)するアルゴリズムや、詳細化をガイダンスする支援環境が必要である。これらについては、別途報告する予定である。

## 2. 概念と関係の定義

知識表現の基本的な構成要素を以下に述べる。

### 定義 2.1

① 実体：定数である。E, E1, E2 …で表わす。『太郎』、『タマ』がこれにあたる。

② 概念：1変数述語である。C(x), C1(y), C2(z) …で表わす。C, C1, C2 と略記する。『人間(x)』、『ネコ(y)』がこれにあたる。

③ 関係：2変数述語である。R(x, y), R1(x, y), R2(x, y) …で表わす。R, R1, R2 と略記する。『飼う(x, y)』がこれにあたる。□

概念が登録されると、その否定が自動的に登録される。概念とその否定の名前は以下のように与える。

### 定義 2.2

いま概念の名前をC とすると、C の否定はnot-C で参照できる。厳密には、 $\forall x \{ \text{not-C}(x) \leftrightarrow \neg C(x) \}$ 。

また概念の名前、及びその否定の別名は<C, not-C の別名>なる順序対で指定する。□

関係に関しては、本資料の範囲では、PSN のように構造的属性と宣言的属性を区別することは考えていない。従って、『太郎の身長が 180cm』であることは、『身長(太郎, 180cm)』という関係の実例で表わす。関係の定義域と値域は以下のように与える。

### 定義 2.3

関係R の定義域をCdで、値域をCrで表わす。同様にR1, R2... の定義域、値域はCd1, Cr1, Cd2, Cr2... で表わす。定義域、値域を満足しない関係は成立しない。□

今後、関係を表現するには、変数に対してその領域となる概念を明示して、 $(\forall x/C1)(\exists y/C2)R(x,y)$  等と表現することにする [8]。概念と同様に、ある関係が登録されると、その関係の逆、及び否定が自動的に登録されるものとする。関係の名前は以下のように与える。

### 定義 2.4

いま、関係の名前をR とすると、R の逆、及び否定は以下の名前と呼ぶこととする。

①not-R : R の否定である。厳密には、

$$(\forall x/Cd)(\forall y/Cr) \{ \text{not-R}(x,y) \leftrightarrow \neg R(x,y) \}。$$

即ち、not-R の定義域と値域はR のそれと同じである。

②inv-R : R の逆である。厳密には、

$$(\forall x/Cd)(\forall y/Cr) \{ \text{inv-R}(y,x) \leftrightarrow R(x,y) \}。$$

即ち、inv-R の定義域、値域は、それぞれR の値域、定義域と同じである。

③not-inv-R 又は inv-not-R: inv-R の否定、又は not-Rの逆である。

関係の名前、及びその逆や否定の別名は、<R, not-Rの別名, inv-Rの別名, not-inv-Rの別名> という形式で指定する。□

(例1) <親, -, 子, ->

(例2) <家族, -, 家族, ->

例1では『親』という関係名とその別名を定義している。この例では『子』は『inv-親』の別名であって、別個に独立して存在する関係ではない。関係の呼び方は相対的である。『inv-親』をR とおくと、not-R は『not-inv-親』、即ち『not-子』である。例2では、4種の関係が2種に縮退している。これは、『家族』という関係に交換律が成立するためである。

## 3. 関係の性質の定義

この章では、関係の性質として包含とリダクションを取り上げ、その定義方法を考える。

### 3.1 関係の包含

まず概念と関係の包含を以下のように定義する。

#### 定義 3.1

$\forall x \{ C1(x) \rightarrow C2(x) \}$  であるとき、C1はC2に包含されるといい、 $C1 \rightarrow C2$ と略記する。全ての概念を包含する概念をCtで、全ての概念に包含される概念(空の概念)をCnilで表わす。□

#### 定義 3.2

$(\forall x/Cd1)(\forall y/Cr1)\{ R1(x,y) \rightarrow R2(x,y) \}$  であるとき、R1はR2に包含されるといい、 $R1 \rightarrow R2$ と略記する。また、全ての関係を包含する関係をRtで、全ての関係に包含される関係(空の関係)をRnilで表わす。Rtの定義域、値域はCt、Rnilの定義域、値域はCnilとする。□

以下は上記の定義から導びける関係に関する性質である。(概念に関する性質も同様である。)

### 性質 3.1

- ①  $R_t \leftrightarrow \text{inv-}R_t$ , ②  $R_{\text{nil}} \leftrightarrow \text{inv-}R_{\text{nil}}$

また、 $R_1 \rightarrow R_2$  である時、以下のことがいえる。

- ③  $\text{inv-}R_1 \rightarrow \text{inv-}R_2$   
 ④  $R \rightarrow R_1$  ならば  $R \rightarrow R_2$   
 ⑤  $R_2 \rightarrow R$  ならば  $R_1 \rightarrow R$

定義 3.2 より、 $\text{not-}R_2 \rightarrow \text{not-}R_1$  は、 $R_1$  と  $R_2$  の定義域、値域が同一の時にのみ成立する。□

具体的な例をあげる。いま、『親 → 家族』であれば、

- ③  $\text{inv-}親(即ち子) \rightarrow \text{inv-}家族(即ち家族)$   
 ④ 父 → 親 であれば 父 → 家族  
 ⑥ 家族 → 親族 であれば 親 → 親族 が成立する。

関係  $R_1$  を包含する関係の集合を、

$$\text{super}(R_1) = \{R_2 \mid R_1 \rightarrow R_2\}$$

で表わすこととする。(概念の包含も同様に  $\text{super}$  で定義する。)  $\text{super}$  は集合であるので、詳細化過程での定義の追加が容易である。また以下のことがいえる。

- (1) 性質 3.1④ より、 $R \rightarrow R_1$  ならば  $\text{super}(R) \supseteq \text{super}(R_1)$  である。従って、包含の束で下位の関係(包含される関係)が、上位の関係(包含する関係)の  $\text{super}$  の要素を継承する。  
 (2) 性質 3.1⑤ より、 $R_2 \rightarrow R$ 、 $R_2 \in \text{super}(R_1)$  ならば  $R \in \text{super}(R_1)$  である。従って、 $\text{super}$  には下位の関係だけを指定すればよい。  
 (3) 性質 3.1③ より、 $R_2 \in \text{super}(R_1)$  ならば  $\text{inv-}R_2 \in \text{super}(\text{inv-}R_1)$  である。これらを重複定義する必要はない。

### 3.2 関係のリダクション

2個の関係から1個の関係を導くリダクション規則を以下のように定義する。

#### 定義 3.3

( $\forall x/Cd1$ ) ( $\forall y/Cr1 \wedge Cd2$ ) ( $\forall z/Cr2$ ) ( $R_1(x,y) \wedge R_2(y,z) \rightarrow R_3(x,z)$ ) であるとき、 $R_1$  と  $R_2$  のリダクション結果が  $R_3$  であるといい、 $R_1 \wedge R_2 \rightarrow R_3$  と略記する。□

上記の定義から以下の性質が導びける。

#### 性質 3.2

- ①  $R_t \wedge R_t \rightarrow R_t$ , ②  $R_{\text{nil}} \wedge R_{\text{nil}} \rightarrow R_{\text{nil}}$

また、 $R_1 \wedge R_2 \rightarrow R_3$  である時、以下のことがいえる。

- ③  $R \rightarrow R_1$  ならば  $R \wedge R_2 \rightarrow R_3$   
 ④  $R \rightarrow R_2$  ならば  $R_1 \wedge R \rightarrow R_3$   
 ⑤  $R_3 \rightarrow R$  ならば  $R_1 \wedge R_2 \rightarrow R$  □

例えば、「兄弟  $\wedge$  親  $\rightarrow$  親」であれば、以下が成立する。

- ③ 兄  $\rightarrow$  兄弟 ならば 兄  $\wedge$  親  $\rightarrow$  親  
 ④ 父  $\rightarrow$  親 ならば 兄弟  $\wedge$  父  $\rightarrow$  親  
 ⑤ 親  $\rightarrow$  家族 ならば 兄弟  $\wedge$  親  $\rightarrow$  家族

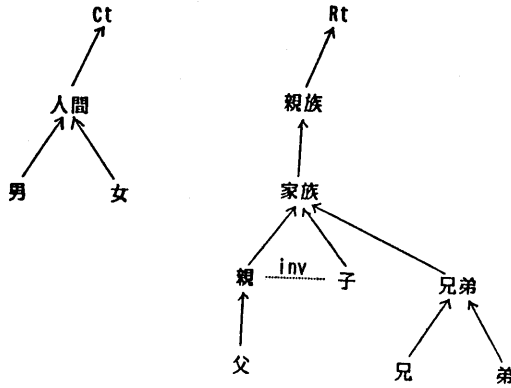
③~⑤の性質を利用して、段階的詳細化に適したリダクション規則の定義方法を考える。

関係  $R_1$  に関するリダクション規則を、

$$\text{reduc}(R_1) = \{ \langle R_2, R_3 \rangle \mid R_1 \wedge R_2 \rightarrow R_3 \}$$

で表現する。 $\text{reduc}$  は集合であるので詳細化過程での定義の追加が容易である。また、以下のことがいえる。

- (1) 性質 3.2③ より、 $R \rightarrow R_1$  ならば、 $\text{reduc}(R) \supseteq \text{reduc}(R_1)$  である。従って、包含の束で下位の関係が上位の関係の  $\text{reduc}$  の要素を継承する。(注：関係の推移律 (transitivity) は上位の関係で成立しても、下位の関係で成立するとは限らない。そこで推移律はリダクション規則 ( $\langle R, R \rangle \in \text{reduc}(R)$ ) で定義させることにしている。)  
 (2) 性質 3.2④ より、 $R \rightarrow R_2$ 、 $\langle R_2, R_3 \rangle \in \text{reduc}(R_1)$  ならば、 $\langle R, R_3 \rangle \in \text{reduc}(R_1)$  である。従って、リダクション規則の第1要素には、上位の関係だけで指定すればよい。  
 (3) 性質 3.2⑤ より、 $R_3 \rightarrow R$  かつ  $\langle R_2, R_3 \rangle \in \text{reduc}(R_1)$



(1) 概念の登録

<人間, - >  
 <男, - >  
 <女, - >

(2) 関係の登録

<親族, -, 親族, - >  
 <家族, -, 家族, - >  
 <親, -, 子, - >  
 <兄弟, -, 兄弟, - >  
 <父, -, -, - >  
 <兄, -, -, - >  
 <弟, -, -, - >

(3) 概念の包含の定義

Super (男) = (人間, not-女)  
 Super (女) = (人間, not-男)

(4) 関係の包含の定義

Super (家族) = (親族)  
 Super (親) = (家族, not-子, not-兄弟)  
 Super (子) = (家族, not-親, not-兄弟)  
 Super (兄弟) = (家族, not-親, not-子)  
 Super (父) = (親)  
 Super (兄) = (兄弟, not-弟)  
 Super (弟) = (兄弟, not-兄)

(5) 関係のリダクションの定義

reduc (親族) = (<親族, 親族>)  
 reduc (家族) = (<家族, 家族>)  
 reduc (親) = (), reduc (子) = ()  
 reduc (兄弟) = (<兄弟, 兄弟>, <親, 親>, <父, 父>)  
 reduc (父) = ()  
 reduc (兄) = (), reduc (弟) = ()

(6) 関係の定義域と値域

domain (親族) = (人間), range (親族) = (人間)  
 domain (家族) = (), range (家族) = ()  
 domain (親) = (), range (親) = ()  
 domain (子) = (), range (子) = ()  
 domain (兄弟) = (), range (兄弟) = (男)  
 domain (父) = (), range (父) = (男)  
 domain (兄) = (), range (兄) = ()  
 domain (弟) = (), range (弟) = ()

図1 概念と関係の定義例

ならば、 $\langle R2, R \rangle \in \text{reduc}(R1)$  である。従って、リダクション規則の第2 要素は下位の関係だけを指定すればよい。

### 3.3 関係の定義域と値域

再度、関係の定義域と値域について考察する。詳細化過程では、関係の定義域、値域についても、予め明らかになっているとは限らない。概念の階層が段階的に詳細化されるのであるから、関係の定義域、値域もまた詳細化の対象であると考えられる。定義 2.3、定義 3.2より定義域、値域について以下の性質が導ける。

#### 性質 3.3

$R \rightarrow R1$ ならば、 $Cd \rightarrow Cd1$ , かつ  $Cr \rightarrow Cr1$ 、さらに、

$R \rightarrow R2$ ならば、 $Cd \rightarrow Cd1 \wedge Cd2$ , かつ  $Cr \rightarrow Cr1 \wedge Cr2$  □

その時点で分っている関係R の定義域、値域を、

$$\text{domain}(R) = \{C \mid Cd \rightarrow C\}$$

$$\text{range}(R) = \{C \mid Cr \rightarrow C\}$$

で表現することとする。ここでCd, Cr はR の真の定義域、値域である。domain, range は集合であるので詳細化過程での定義の追加が容易である。また、性質 3.3より以下のことがいえる。

(1)  $R \rightarrow R1$ ならば、 $\text{domain}(R) \supseteq \text{domain}(R1)$  である。

従って、包含の束で下位の関係が上位の関係のdomainの要素を継承する。range についても同様である。

(2)  $C \rightarrow C1$ ,  $C \in \text{domain}(R)$ ならば $C1 \in \text{domain}(R)$  である。従って、domainには下位の概念だけを指定すればよい。range についても同様である。

概念や関係の定義の例を図1に示す。図1では互いに独立な最小限の定義だけを記述している。

### 4. 事実知識の表現

3章で定義した概念と関係を用いて事実知識を表現する方法を述べる。事実には以下の2種がある。

① C(E)

②  $(\forall x/C1)(\exists y/C2)R(x,y)$ 、又は  
 $(\exists y/C2)(\forall x/C1)R(x,y)$

①②はそれぞれ概念と関係の外延を表わしている。ここでQ1, Q2 は、全称(∀)又は、存在限量子(∃)である。x, y の代わりに実体が指定されてもよい。但し、関係の定義域と値域を満足していることが前提となる。以下では、事実知識間の包含とリダクションについて述べる。

#### 4.1 事実知識の包含

##### (1) 包含の判定

関係の事実知識の包含を判定するアルゴリズムを述べるために、まず関係をデータ構造、

$\langle [Q1Q2], N1, N2, R \rangle$

の形で扱うこととする。ここで、N1, N2 は概念 (C1, C2)、又は実体 (E1, E2) を表わす。[Q1Q2]は限量子を組合わせたもので、VV、V∃、∃∃、∃Vのように表わす。V∃、∃Vは限量子の出現順がQ2, Q1 の順(即ち、逆順)であることを示す。N1, N2 の一方、又は両方が実体である場合には、∃e、ve、e∃、ev、eeで表わす。

##### アルゴリズム 4.1

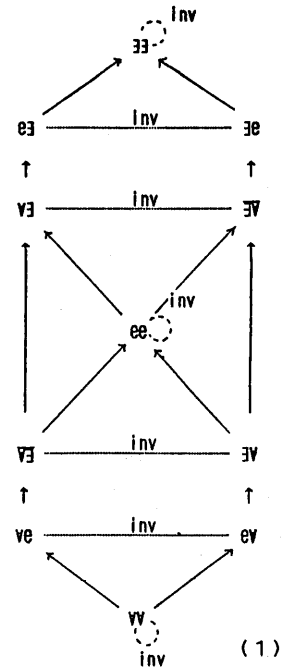
2個の事実知識f1, f2 を、 $\langle [Q11Q12], N11, N12, R1 \rangle$  と  $\langle [Q21Q22], N21, N22, R2 \rangle$  と表わすと、 $f1 \rightarrow f2$ であることは以下の手順で判定できる。

step1:  $R1 \rightarrow R2$ であればstep2へ。そうでなければstep4へ。

step2:  $[Q11Q12] \rightarrow [Q21Q22]$  (図2(1)参照)であればstep3へ。

step3: N11 とN21、及びN12 とN22 がそれぞれ図2(2)を満足すれば、 $f1 \rightarrow f2$ 。

step4:  $R1 \rightarrow \text{inv-}R2$ であればstep5へ。



Q2j \ Q1i	∃	e	∀
∃	C1i → C2j	-	-
e	C2j (E1j)	E1i = E2j	-
∀	C1i → C2j 又は C2j → C1i	C1j (E2j)	C2j → C1i

(2)

図2 事実知識の包含判定

step5:  $[Q11Q12] \rightarrow \text{inv-}[Q21Q22]$  (図2(1)参照)であればstep6へ。

step6: N11 とN22、及びN12 とN21 がそれぞれ図2(2)を満足すれば、 $f1 \rightarrow f2$ 。□

step1 ~3 によって、例えば、 $(\exists y/\text{女})(\forall x/\text{男})$  夫婦(x, y)  $\rightarrow$   $(\forall x/\text{男})(\exists y/\text{人間})$  家族(x, y) が、step4 ~6 によって、 $(\exists y/\text{男})(\forall x/\text{学生})$  親(x, y)  $\rightarrow$   $(\exists x/\text{人間})(\forall y/\text{小学生})$  子(x, y) が判定できる。

知識の詳細化の過程は、より下位の(包含される)知識に置き換える過程であると考えられる。例え

ば、『花子と太郎が夫婦である』という関係、夫婦（花子，太郎）は、『何かと何かの間に何らかの関係が成り立つ』という関係、 $(\exists x/Ct)(\exists y/Ct) R(x,y)$  を詳細化した結果であると考えることができる。

## (2) 包含による継承の表現

関係の包含は各種の知識表現モデルで扱われている is-a, instance-of による継承の 1 部を表現している。

### ① is-a による継承

$C2 \rightarrow C1$  であれば、 $(\forall x/C1)(\forall y/C) R(x,y) \rightarrow (\forall x/C2)(\forall y/C) R(x,y)$  である。これは、概念  $C1$  の実体に共通の情報が下位の概念  $C2$  に継承されること (is-a による継承) を意味している。

また、 $C2 \rightarrow C1$  であれば、 $(\exists x/C2)(\forall y/C) R(x,y) \rightarrow (\exists x/C1)(\forall y/C) R(x,y)$  である。これは、概念  $C2$  のある実体が持つ情報が上位の概念  $C1$  に継承されること (is-a とは逆の方向での継承) を意味している (文献 [6])。

### ② instance-of による継承

$C1(E)$  であれば、 $(\forall x/C1)(\forall y/C) R(x,y) \rightarrow (\forall y/C) R(E,y)$  である。これは、概念  $C1$  から実体  $E$  への継承 (instance-of による継承) を表わしている。

また、 $C1(E)$  であれば、 $(\forall y/C) R(E,y) \rightarrow (\exists x/C1)(\forall y/C) R(x,y)$  である。これは、実体  $E$  から概念  $C1$  への継承 (instance-of とは逆の方向での継承) を表わしている。

## 4.2 事実知識のリダクション

### (1) リダクションの実行

事実知識のリダクションアルゴリズムを以下に示す。

#### アルゴリズム 4.2

3 個の関係  $f1, f2, f3$  を、 $\langle [Q11Q12], N11, N12, R1 \rangle$ 、 $\langle [Q21Q22], N21, N22, R2 \rangle$ 、 $\langle [Q31Q32], N31, N32, R3 \rangle$  と表わ

す。 $f1, f2$  から、 $f1 \wedge f2 \rightarrow f3$  なる  $f3$  を生成する手順は以下のとおりである。

step1:  $\langle Ri, Rj \rangle \in \text{reduc}(Rk)$ 、 $R2 \rightarrow Ri$ 、 $R1 \rightarrow Rk$  なる  $Rj$  が存在すれば、その内最も下位の関係を  $R3$  として step2 へ。 $Rj$  が存在しなければ、リダクションはできない。

step2: ①  $Q12=Q21=\exists$ 、②  $Q12=e$ 、 $Q21=\exists$

③  $Q12=\exists$ 、 $Q21=e$

上記の場合には、リダクションはできない。そうでなければ、step 3 へ。

step3: ①  $Q12=Q21=e$ 、 $E12=E21$

②  $Q12=e$ 、 $Q21=\forall$ 、 $C21(E12)$

③  $Q12=\forall$ 、 $Q21=e$ 、 $C12(E21)$

④  $Q12=\exists$ 、 $Q21=\forall$ 、 $C12 \rightarrow C21$

⑤  $Q12=\forall$ 、 $Q21=\exists$ 、 $C21 \rightarrow C12$

⑥  $Q12=\forall$ 、 $Q21=\forall$ 、 $C12 \rightarrow C21$

又は  $C21 \rightarrow C12$

上記の場合には、 $Q31=Q11$ 、 $Q32=Q22$  とする。但し以下の場合には、 $[Q31Q32]$  の限量子の出現順は逆順となる。

①  $Q11=\exists$ 、 $Q22=\forall$  でかつ、

$[Q11Q12]=\exists\forall$ 、 $[Q21Q22]=\exists\forall$  なら

$[Q31Q32]=\exists\forall$

②  $Q11=\forall$ 、 $Q22=\exists$  でかつ、

$[Q11Q12]=\forall\exists$ 、 $[Q21Q22]=\forall\exists$  でないなら

$[Q31Q32]=\forall\exists\Box$

事実知識のリダクションの例を以下に示す。いま、

「兄弟  $\wedge$  親  $\rightarrow$  親」とすると、

①  $(\forall x/\text{小学生})(\exists y/\text{中学生})$  兄弟  $(x,y)$  と

$(\forall x/\text{中学生})(\exists y/\text{大人})$  親  $(x,y)$  から

( $\forall x/\text{小学生}$ )( $\exists y/\text{大人}$ ) 親( $x, y$ ) が得られる。

- ② ( $\forall x/\text{小学生}$ )( $\exists y/\text{中学生}$ ) 兄弟( $x, y$ ) と  
( $\exists x/\text{中学生}$ )( $\exists y/\text{大人}$ ) 親( $x, y$ ) とは  
リダクションできない。

## (2) リダクションによる継承の表現

関係のリダクションは、関係毎に固有の継承を表わす。

### 性質 4.1

$x$  と  $y$  の間に  $R1$  という関係がある時、即ち  $R1(x, y)$  の時、以下のことがいえる。

- ①  $y$  に関する情報  $R2(y, z)$  が、 $x$  に継承される (即ち、 $R2(x, z)$ ) ことは、 $\langle R2, R2 \rangle \in \text{reduc}(R1)$  によって表現される。
- ②  $y$  に関する情報  $R2(y, z)$  が、 $R3$  に変更されて  $x$  に継承される (即ち、 $R3(x, z)$ ) ことは、 $\langle R2, R3 \rangle \in \text{reduc}(R1)$  によって表現される。□

Fox [2] の分類した4種の関係固有の継承と、上記の性質との対比を以下に述べる。

(1)  $A, B$  間の関係によって、 $A$  から  $B$  へ情報が変更されずに継承される機能 (PASS) は、性質 4.1①によって表わされる。

(2)  $A$  から  $B$  へ情報が加工されて継承される機能 (SUBSTITUTE)、 $B$  に無条件に新しい情報が追加される機能 (ADD)、及び、 $A$  から  $B$  への情報が継承されないことを表わす機能 (EXCLUDE) は、この範囲では表現できない。

(3) 性質 4.1の②は種類の違うスロットに値が継承されることを示している。

## 5. むすび

知識ベースの段階的詳細化を実現するための1歩として、1階論理のサブセットによる知識表現方法を述べた。この方法は、概念と関係の包含を用いて様々な詳細化の

レベルにある知識を表現しようとするものである。

今後は処理系の開発と適用を通じて、関係の定義等、知識表現の評価と拡充を行う予定である。最後に本研究の機会を与えて戴いた堀内敬之知識ベース研究室長、御助言、御討論を戴いた、電総研IXプロジェクトの方々、ICOT国藤進氏、NTT 基礎研究所後藤滋樹氏、並びに知識ベース研究室の諸氏に感謝します。

## 参考文献

- [1] Levesque, H. J. and Mylopoulos, J. : A Precedural Semantics for Semantic Networks, in Associative Networks, Academic Press, 1979.
- [2] Fox, M. S. : On Inheritance in Knowledge Representation, 6th IJCAI, 1979.
- [3] Brachman, R. J., Fikes, R. E. and Levesque, H. J. : Krypton: A Functional Approach to Knowledge Representation, Computer, 16, 10, 1983.
- [4] 田中穂積、小山晴生、奥村学: 知識表現形式DC KRとその応用, ソフトウェア科学会、論理と自然言語ワークショップ, 1986.
- [5] 半田剣一、樋口哲也、古谷立美、国分明男: 意味記憶システムIX-IXLによる知識表現一、知識工学と人工知能研究会, 39-7, 1985.
- [6] 半田剣一、樋口哲也、古谷立美、国分明男: 概念間の関係の論理的意味について、日本ソフトウェア科学会第2回大会論文集, 1985.
- [7] Miyachi, T., Kunifuji, S., et. al. : A Knowledge Assimilation Method for Logic Databases, Proc. on Logic Programming, 1984.
- [8] 国藤進: 1階多類論理の推論方式について、情報処理学会第23回全国大会, 1981.