論　文

知識構造に基づく機械的推論規則について*

大　須　賀　節　雄**

Abstrct

The representation of knowledge is the problem becoming very important for the natural language understanding systems. The form of representation must be such as capable of representing various concepts containd in everyday conversation and as well suitable for internal processing to achieve a number of faculties required for such systems. Among many faculties required, the one concerning the deductive inference is of vital importance becauce it is closely related with the function of retrieving relevant knowledge from the data base, and consequently, is the basic function of all.

In this paper, the author will describe a deductive inference rule and its algorithm based on a knowledge structure. In this system, the concept of set is regarded as the pivotal concept of all and the representation of knowledge, the knowledge structure and the deductive inference rule are all based on this concept.

1. まえがき

計算機による自然言語処理をはじめとして，多くの分野で計算機内での知識の表現形式とその処理に関する問題が重要視されるようになっている．この論文もこの知識表現の問題に関連するが，特にある特定の知識表現形式と，その特徴を利用した知識構造を想定したとき，この構造に基づく演繹処理の規則と方法を述べる．

我々の目的は簡単な自然語で会話ができ，かつ質問等に応答することによって人間を援助できるような計算機システムを構成することである．システムはこれらの仕事をすでに与えられている知識に基づいて処理することとする．

このようなシステムには様々な形式の質問が提示され，その各々が計算機に異なった能力を要求するであろう．たとえばいくつかの知識の組から特定の結論を導く能力，二種の事実間の因果関係を見出す能力，指定された最終状態に達するために最良の動作列を見出す能力その他である．

しかしこれらのうちで演繹処理の能力が最も基本的なものとして重要である．それはこれが知識検索機能と密接に結び付き，したがって知識を利用する他のすべての処理も元来はこの存在を前提とするからである．これらの処理がすべて計算機内に貯えられた知識に基づいて行われるので，この有効性は知識表現の形式に依存することはいうまでもないが，多くの知識や質問が自然語で与えられることを意図する以上，知識表現の形式は少なくとも使用の許された言葉が含むすべての概念を表現できなくてはならない．特に言葉が表わす概念には物事の論理的な関係と同時に数値的な関係が含まれているので，この表現形式は論理と数量の概念を同一形式で表現でき，さらにそれらが各種処理に有効に用いられなければならない．

従来，論理処理に適した表現形式としては一階述語が多く用いられているが，数値の表現には適せず，知識表現形式として適切とはいえない．

このような考察から，我々は知識表現の基本に集合の概念を採用し，これを一階述語表現と組合わせることにする．集合の概念は論理的な性質と同時に数値化

---

* A Mechanical Deductive Inference Rule Based on a Knowledge Structure by Setsuo OHSUGA (Institute of Space and Aeronautical Science, University of Tokyo)

** 東京大学宇宙航空研究所

され得る性質を併せ持つからである．

このような知識は集合の階層的性質を利用して構造化することができ，それによって知識間の連想関係を効率的に表わすことができる．

一方，前述のように演繹処理能力がすべての処理の基本となるから，このような新らしい表現形式を採用する際にはまずそれに適した効率の良い演繹規則を見出す必要がある．以下では主として集合の概念に基づく知識表現形式を用いた時の演繹規則を述べるが，その準備としてこのような知識表現形式について，後の記述に必要な部分のみをまず簡単に述べる．知識表現形式そのものについての詳細は別の機会に述べたい．

2. 基本的考察

まず従来の一階述語表現を例にとる．たとえば“人はすべて死す”は一階述語では通常 $(\forall x)[\mathrm{MAN}(x)\Rightarrow\mathrm{MORT}(x)]$ と表わす．この時記述される対象の全体 (universe of discourse) は一定で，変数 x はこの範囲内で任意の値をとることができる．しかし上の表現は $\mathrm{MORT}(x)$ が真であることが確実なのは $\mathrm{MAN}(x)$ を真とする x に限られることを述べている．そこでこれを少しかえて $(\forall x/\mathrm{MAN})\mathrm{MORT}(x)$ のように表現してみる．ここで MAN はもはや述語記号ではなく，上述の表現で $\mathrm{MAN}(x)$ を真にする x の集合を表わす．この表現は $x\in\mathrm{MAN}$ なるすべての x について $\mathrm{MORT}(x)$ は真なることを表わし，意味的にはこの両者の間に差はない．

この表現形式を用いると，一般に $(\forall x)[X(x)\Rightarrow F(x)]$ において，$X(x)$ が $X\ni x$ を表わすもの，すなわち X が元来，実体の集まりを表わし，F がこのような実体 x についての性質とか関係を表わす概念であるとした場合，この表現が $(\forall x/X)F(x)$ と表わされる．説明の便宜上，前者の概念を実体概念と呼んで，X, Y, Z 等で示し，後者を関係概念と呼んで F, G, H 等で表わすことにする．またこのような諸概念は語を通して導入されるが，これらに対応する語をそれぞれ実体語および関係語と呼ぶ．実体語は名詞，関係語は動詞，形容詞，前置詞，一部の名詞などである．

こうすると，集合を用いる知識表現形式では述語記号になるのは関係概念のみであるということになる．したがって従来の表現で $(\exists x)[X(x)\cap F(x)]$ は $(\exists x/X)F(x)$ ($F(x)$ を真にする x が X 内に少くとも一つ存在する) となる．$(\forall x)[X(x)\Rightarrow Y(x)]$ のようなものは $X\subset Y$ なる集合関係を示しており，後に述べるように知識構造そのもので表現されるので，述語としての表現は不要であるまた，$(\forall x)[F(x)\Rightarrow G(x)]$ は $(\forall x/U)[F(x)\Rightarrow G(x)]$ に，$(\exists x)[F(x)\cap G(x)]$ は $(\exists x/U)(F(x)\cap G(x))$ になる．ここで U はすべての実体の集合を示す．なお $(\forall x)X(x)$ は $X=U$ なることを示すから，$(\forall x)[X(x)\cap F(x)]$ は $(X=U)\cap(\forall x/U)F(x)$ を表わし，また，$(\exists x)[X(x)\Rightarrow F(x)]$ のような表現は（含意関係の定義 $(A\Rightarrow B)\equiv(\sim A\cup B)$ から，あまり意味のない表現であるが）$(X\subset U)\cup(\exists x/U)F(x)$ なることを表わしているので，いずれも前半の部分は知識構造によって表現され，$X=U$ の時のみ，それぞれ $(\forall x/X)F(x)$ および $(\exists x/U)F(x)$ が意味のある表現となる．

このようにして従来の一変数表現はすべてこの新らしい表現形式で表現できる．多変数表現も全く同様であり，各変数ごとに上述のように表現すればよい．紙数の都合でこのすべてを示すことができぬので一例を示すと $(\forall x/X)(\exists z/Z)(\forall y/Y)F(xyz)$ は $(\forall x)(\exists z)(\forall y)[X(x)\Rightarrow Z(z)\cap\{Y(y)\Rightarrow F(xyz)\}]$ に対応する．この変換手順は単純で，接頭部の限量記号を右から順に上述の一変数の場合の手順を用いて変換すればよい．

実際には知識表現において必要なのは x, y, z のような変数記号でなく，この変域である X, Y, Z である．したがって $(\forall x/X)F(x)$, $(\exists x/X)F(x)$ の代りに計算機内では単に $(\forall X)F(X)$, $(\exists X)F(X)$ のように表わす．多変数の場合も同様で，たとえば上例は $(\forall X)(\exists Z)(\forall Y)F(XYZ)$ のように表わせば十分である．以下では便宜上 x, y, z を含めた表現を用いるが，これは説明のための便法である．この表現法のもとでは論理処理を進める上で必要なのは二変数の変域間の関係であり，これが見出されるものなら個々の変域はどのような集合であってもよい．特にこの変域として実数区間上で定義される変数は数量を示す変数であるが，これも一般の論理変数と同様に扱うことができる．このように変形はしてもこの表現法は一階述語としての性質を多分に備えている．したがって以下ではこれをもやはり一階述語と呼び，これにもとづいてリテラルとかフォーミュラなど，一般の定理証明に関連して定義された概念を定義することにする．

このような表現法は以下のような特徴を有している．

第一に言語表現から知識表現への変換に適してい

る．この表現で（∀MAN)MORT(MAN) が単独で，“人はすべて死す”を表わすように，一つの述語は最小の文章すなわち最小単位の知識を表現している．関係語は基本知識単一のものと，基本知識の複合形を導入するものに分けられるが，いずれにしろ前もって各関係語に上述の表現形式あるいは後述するようなその複合形をそれら各語の意味表現として与えておく．この際，各変数の変域はその語が合理的に適用され得る最大の集合とする．たとえば MORT(x) は x∈生物 (LT) なる集合 LT に適用できるが，LT を含むさらに大きな集合，たとえば物体 (PO) 全体には適用できない．このような関係語が文章中で用いられた時，文中の実体語の示す集合が意味表現内の変域集合の部分集合である時にのみ有意な文と判断され，この変域にこの部分集合が代入されて特定の知識が生成される．

第2に後述するように，知識を階層的な構造に組むことが容易である．

第3に集合を通して物事の論理的な面と数量的な面を同一形式に表現できる．“遠い”，“長い”，“早い”などの諸語が数量的な“程度”の概念を含むことは明らかであり，またすべての知識に密接に関連する“時間”の概念を論理的に同じレベルで処理することができる．

第4にこれと関連して，自然言語の表現で最も重要な動詞の表現を基本形の複合として表現でき，単一の論理処理のアルゴリズムの使用を可能にする．

これらの特徴の詳細については別の機会に述べたい．

3. 基本知識表現

知識表現形式には基本となるいくつかの基本知識のセットがあり，それらは形式により三種に分類できる．時間変数はいずれにも含めることとする．時間変数も他の変数と同様，変域で示されるが，これは一般に実数区間またはその集まりである．時間以外にも連続な区間を変域とする変数が存在し，さらにある変数が他の変数の関数でもあり得る．後述のように論理処理に必要なのはその変数の変域もしくは関数の値域であり，変数の性質にかかわらない．特にある変数が時間変数の関数である時，この述語は時間的に変化する事象を表わす．“離れる”，“成長する”，“縮む”など．このような事象を表わす言葉が多数あるのでこれは重要である．以下では，x_1, x_2 などは論理変数，n_1, n_2 などは数値変数を，t は時間変数を表わす。数値変数の変域は実数区間とする．論理変数，数値変数を問わず，変域が一点にまで縮小したものをもって特定の値を表わす．

（1） $F(t, x_1, x_2, \cdots, x_n)$；この型は t 以外に数値変数を含まず，純粋に論理的な性質，関係等を表わす．この例は AT(t, a, b)，PARENT(t, a, b) のようなもので，たとえば t の変域として仮りに前者に $T_1=(t_1, t_2)$，後者に $T_2=(t_1, \infty)$ が与えられたとすると，これらの表現の意味はそれぞれ，“a は t_1 から t_2 まで b に居た”，および“a は t_1 以来 b の親である”を表わす．

（2） $F(t_1, x_1, x_2, \cdots, x_n, n_1, n_2, \cdots, n_k)$；この型は t 以外にも数値変数を有するが，関数関係は存在しない．この例は HEIGHT(t, a, n) (“a の高さは n である”)，DISTANCE(t, a, b, n) (“a, b 間の距離は n である”) のようなものである．

（3） $F(t, x_1, x_2, \cdots, x_n, n_1(*), n_2(*), \cdots, n_k(*))$；この型は t 以外に数値変数を含み，かつ t を含めてこの間に関数関係のあるもので，この例は LENGTH($t, a, n(t)$) のようなものである．これは $n(t)$ が t の増加（減少）関数の時，“a は伸びる（縮む）”を表わす．t 以外の変数の関数の場合も多い．

4. 複合表現

基本表現以外の知識はこれらの組合せで表現する．このためにいくつかの接続記号が準備されている．これは通常の合接 (Conjunctron)，離接 (Disjunction) 以外に，因果関係や行為の表現に使われるものを含む．

一般に複文や重文で表わされた知識は複合表現となる他，動作表現の多くが複合表現になる．このような知識表現形式の詳細についてはここでは述べないが，知識の多くが複合形式で表わされ，多種の接続子が用いられていても，演繹という目的にたいしてはこの表現形式はフォーミュラの組，すなわち述語記号と変数（ここでは変域）と，合接と離接のみを含む接続子から構成されるとして扱える．このような知識は各リテラルを端点とする AND—OR 木構造で表わされている．

5. 知識構造[2),5),6),7)]

すべての知識は構造化される．構造化の方法は各種あるが，我々は集合間の関係に基づく階層構造を作

る．

まず実体概念について考えよう．実体の集合である実体概念のあるものは直接実体語で参照できる．これを基本集合と呼ぶ．複数要素を含む集合は普通名詞に，単一要素集合は固有名詞に対応する．基本集合以外のものは基本集合から集合演算等により指定される．実体概念は計算機内では固有の情報を含む一定語長のメモリ・ブロックで表わされる．

U以外の任意の集合は他の集合に特定の性質を導入したり，集合演算を施こすことにより定義できる．Y_1 を X のある部分集合とし，Y_1 に固有の特定の性質で特徴づけられているとする．この性質を属性一値の対により$A_Y(V_{Y1})$と表わすことにする．すると Y_1 は X と $A_Y(V_{Y1})$ とにより定義できる．これを $Y_1=X*A_Y(V_{Y1})$ と書き，"Y_1 は X のうち $A_Y(V_{Y1})$ を満たすもの" を表わすことにする．次に $Y_2 \neq Y_1$ かつ $Y_2 \subset X$ なる Y_2 が Y_1 と同じ属性によって X から定義され，その値が V_{Y2} とする．$Y_2=X*A_Y(V_{Y2})$．もし V_{Y1} と V_{Y2} が共通の要素を含まないなら $Y_1 \wedge Y_2=\phi$ である．$\wedge$ は積集合記号である．V は数値区間，論理値あるいは一般の集合のいずれかである．

この例は MALE=PERSON*SEX(0), FEMALE =PERSON*SEX(1) のようなものである．

このような Y_1, Y_2 を "X から A_Y により定義された" と言うことにする．X にいくつかの異なった属性を導入できる場合，いく組かの部分集合の組が X から定義される．たとえば A_Y の他に A_Z が X に適用できるとすると，$Z_1=X*A_Z(V_{Z1})$, $Z_2=X*A_Z(V_{Z2})$ $\cdots Z_i \wedge Z_j=\phi$, $(i \neq j)$ が得られる．この時一般に任意の i, j について $Y_i \wedge Z_j \neq \phi$ である．たとえば $X=$ PERSON, A_Y=SEX, A_Z=AGE, Y_1=MALE, Y_2 =FEMALE, Z_1=ADULT, Z_2=CHILD などである．

次に集合 W が Y_1 から $W=Y_1*A_W(V_W)$ で定義されたとする．すると Y_1 の定義から，W は X 内で $A_Y(V_{Y1})$ と $A_W(V_W)$ の両性質を備えているものとなり，

$$W=X*A_Y(V_{Y1}) \cap A_W(V_W) \qquad (1)$$

と表わされる．

次にある集合 W' が他の二つの集合の積集合として定義されたとしよう．たとえば W'=BOY とすると，$W'=Y_1 \wedge Z_2$．この場合も Y_1, Z_2 に前記表現を用いて $W'=X*A_Y(V_{Y1}) \cap A_Z(V_{Z2})$ となって(1)式の表現と形式的に一致する．一般に X より性質を一つ余分に有する集合 Y を X より一段低位にあると言うことにすると，N 段低位にある W は $W=X*\bigcap_{i=1}^{N} A_i(V^i)$ のように表わされる．またこの集合一部分集合の関係は **Fig. 1** (基本構造要素) のように構造化される．

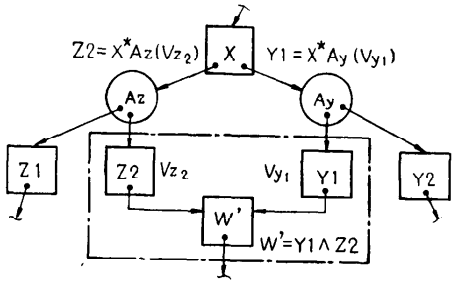

Fig. 1 Basic structure element.

次に任意の二つの集合 X, Y 間に成立する集合関係を後の論理処理に用いられる次の五種に分類してみる．

① $X=Y$,
② $X \supset Y$,
③ $X \subset Y$,
④ $X \wedge Y=\phi$,
⑤ $X \in Y$, (①～④以外のものすべて)

たとえば PERSON⊃MALE⊃BOY PERSON⊃FEMALE⊃GIRL PERSON⊃CHILD⊃BOY CHILD ⊃GIRL MALE∧FEMAE=ϕ MALE∈CHILD etc.

二つの集合 X, Y が与えられた時，これが上記五種のどの関係に相当するかは次のように見出される．

Y と Z の共通の先祖（上位）集合を X とする．Y と Z から X までの道を上方に辿ることにより

$$Y=X*\bigcap_{i}^{m} A_Y{}^i(V_Y{}^i),\quad Z=X*\bigcap_{j}^{k} A_Z{}^j(V_Z{}^j)$$

が得られ，次の関係が成立する．

①② & ③ $Y \supseteq Z$ if $\boldsymbol{A}_Y \supseteq \boldsymbol{A}_Z$

④ $Y \wedge Z=\phi$ if $A_Y{}^i=A_Z{}^j$ and $V_Y{}^i \wedge V_Z{}^j=\phi$ for some (ij) ⑤ 残りの case

ただし，

$$\boldsymbol{A}_Y=\{A_Y{}^i(V_Y{}^i)/i=1,2,\cdots,m\},$$
$$\boldsymbol{A}_Z=\{A_Z{}^j(V_Z{}^j)/j=1,2,\cdots,k\},$$

集合 U は明らかにすべての集合の先祖集合である．これ以外に先祖集合を見出す方法もあるが，この論文の範囲外であるので省略する．

実体概念の構造ができたらすべての知識はこれに関

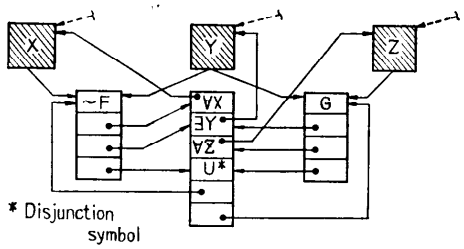

Fig. 2 Anexample of knowledge structure.

連づけられて貯えられる．この方法は各知識内リテラルがその中の変数の変域集合に連結されるとするもので，**Fig. 2**（知識構造例）は $(\forall x/X)(\exists y/Y)(\forall z/Z)\{\sim F(xy)\cup G(yz)\}$ が X, Y および Z に接続されている状況を示す．各知識は複合表現としてAND—OR木構造を作り，この端末リテラルが変数の変域に接続される．一つの集合に複数個のリテラルが接続される時はこれらがチェインで連なる．これをリテラル・チェインと呼ぶ．

以上は一般の知識の表現形式であるが，知識の中には特定の実体のみに関するもの，すなわちすべての変数の変域が一要素集合のものが多いと期待される．これらは表現形式にまとめた方が記憶および検索効率が良い．この表は述語記号をヘッダに持つ．

6. 演 繹 規 則

知識表現形式に適した効率の良い演繹機能を有することはシステムにとって不可欠である．以下では集合関係に基づく演繹規則を述べる．

これまで機械的な演繹の手法として，J. Robinsonの導出原理[3]は前提となるフォーミュラの組と，導びかれる結論が与えられている時の演繹手順を与えるという点では非常に優れたものであった．それでもこれが問題解決において人間に遠く及ばないのは，多くの問題で必要なのが単に演繹の手続きのみでなく，知識の有効利用の方法を確立することだからであろう．

以下で述べる方法も単に演繹手続きとしての効率のみでなく，前述の知識の利用を可能にすること，および大量の知識の中から問題に応じて必要な知識を検索しながら推論を進めてゆくという方法を可能にすることにより，総合的な知識利用に役立ち得ることを意図して開発されたものである．

6.1 含意判定規則

前章では任意の二集合間の関係を構造関係を利用して判定する方法を示した．一方論理的な含意関係の処理は演繹の基本であるが，集合を基礎にした知識表現のもとでは，この含意関係は変域集合間の関係と密接な関連がある．すなわち，一変数述語の場合で示すと，X, Y およびある要素 a の間に $X\supset Y\ni a$ の関係があれば

$$(\forall x/X)F(x)\Rightarrow(\forall x/Y)F(x)\Rightarrow F(a)$$
$$\Rightarrow(\exists x/Y)F(x)\Rightarrow(\exists x/X)F(x)$$

なる含意関係が成立し，また $X\wedge Z\neq\phi$ の時

$$(\forall x/X)F(x)\Rightarrow(\exists x/Z)F(x)$$

が成り立つことが容易に判る．

一般の場合はもうすこし複雑である．ある前提 P と結論 C の間に，$P\Rightarrow C$ が成立するか否かを調べるものとする．P, C が次のように表わされているとする．

$$P: (Q_{P1}x_{P1}/X_{P1})(Q_{P2}x_{P2}/X_{P2})\cdots$$
$$(Q_{Pn}x_{Pn}/X_{Pn})F(x_{j1}, x_{j2}, \cdots, x_{jn})$$
$$C: (Q_{C1}x_{C1}/X_{C1})(Q_{C2}x_{C2}/X_{C2})\cdots$$
$$(Q_{Cn}x_{Cn}/X_{Cn})F(x_{j1}, x_{j2}, \cdots, x_{jn})$$

ここで Q は“∃ もしくは ∀”を表わす．

$P\Rightarrow C$ の判定に関係する要因は次の三種である．

（1） 各変数 x_{ji} について P, C の限量記号の関係

（2） P, C の接頭部内の限量記号順序

（3） 各変数の P, C における変域の関係

である．これらについて，$P\Rightarrow C$ の条件は

（1） すべての変数について，その限量記号が P では∃，C では ∀ なるものがあってはならない．

（2） 限量記号が P では ∀，C では ∃ なる変数すべてにつき P 内で限量記号を ∀ から ∃ に変えると仮定する．この後，限量記号が∃なる任意の変数 x_k について，接頭部内で x_k の限量記号より前方(左)にある全称記号に対応する変数の集合を P では S_k，C では T_k とした時 $S_k\supset T_k$ なる x_k が存在しないこと．

（3） 変数の変域が P および C において，限量記号の組合せにより **Table 1**（$P\Rightarrow C$ の条件）の条件を満たすこと．この証明は以下の置換規則に含まれている．

Table 1 The conditions of $P\Rightarrow C$.

P	C	CONDITION
∀	∀	$Xp_k\supset Xc_k$
∀	∃	$Xp\wedge Xc_k\neq\phi$
∃	∃	$Xp_k\subset Xc_k$

Xp_k: The domain of x_k in P.
Xc_k: The domain of x_k in C.

6.2 置 換 規 則

与えられた質問 C を導びく知識の集合を $\boldsymbol{P}_c=$

$\{P_i/i=1,2,\cdots,n\}$ とする，すなわち

$$P_1\cap P_2\cap\cdots\cap P_n\Rightarrow C \text{ あるいは } \bigcap_i P_i\Rightarrow C \quad (2)$$

ここで次のように定義する．

$$\boldsymbol{P}_c{}^0=P_1\cap P_2\cap\cdots\cap P_n=\bigcap_i P_i$$

$$\boldsymbol{P}_c{}^j=P_1\cap P_2\cap\cdots\cap P_{j-1}\cap P_{j+1}\cap\cdots\cap P_n=\bigcap_{i\neq j} P_i$$

$$\boldsymbol{P}_c{}^{jk\cdots m}=\bigcap_{i\neq j,k,\cdots,m} P_i$$

すると(2)式は $\boldsymbol{P}_c{}^0=\boldsymbol{P}_c{}^j\cap P_j\Rightarrow C$ と表わされる．

$\boldsymbol{P}_c$ は全知識の部分集合であり，C に依存して定まるので最初は未だ見出されていないものである．したがって $\boldsymbol{P}_c$ に関する情報なしに C を証明せねばならない．

ここで次のような置換規則を定義する．

置換規則：あるフォーミュラ P_j' が与えられた時，以下の定義から定まる新らしいフォーミュラ $R_j{}^*$ を "C の P_j' による置換" と呼ぶ．

$\boldsymbol{R}_j$ を $P_j'\cap R_j{}'\Rightarrow C$ なる関係をみたす全ての有意なフォーミュラ $R_j{}'$ の集合とする．すなわち

$$\boldsymbol{R}_j=\{R_j{}'\mid P_j'\cap R_j{}'\Rightarrow C,\ i=1,2,\cdots\}$$

この時，$R_j{}^*\subset\boldsymbol{R}_j$ であり，任意のフォーミュラ S について，もし $S\Rightarrow R_j{}'$ なら $S\Rightarrow R_j{}^*$ である．

ここで $P_j'=P_j\in\boldsymbol{P}_c$ であったとしてみる．P_j は真であるから，もし $R_j{}^*$ が真であれば C も真である．また上の定義で $S=\boldsymbol{P}_c{}^j$ とおくと，$\boldsymbol{P}_c{}^j\Rightarrow R_j{}^*$ である．すなわち，$\boldsymbol{P}_c{}^0\Rightarrow C$ は $\boldsymbol{P}_c{}^j\Rightarrow R_j{}^*$ と等価である．したがって $R_j{}^*$ を新たに C の如くみなすと上述の手順が再帰的に繰り返されることが判る．すなわち，

$$\boldsymbol{P}_c{}^{jk}\Rightarrow R_{jk}{}^*,\quad \boldsymbol{P}_c{}^{jkl}\Rightarrow R_{jkl}{}^*\cdots$$

ここで $R_{jk}{}^*$ は $\boldsymbol{P}_c{}^j$ の P_k による置換である．$R_{jkl}{}^*$ 等も以下同様とする．このような置換列を生成してゆき，最後に置換の一つが離接を含まない知識により含意されたら，すべての置換および C が真なることが示される．

この手順において，置換生成のために選ばれる P_j' 等を "置換生成の候補" と呼び，これが満たすべき一定の条件を定める，すなわち，上述のように置換生成の候補がすべて $\boldsymbol{P}_c$ に含まれるなら，最も効果的に置換系列を生成できるが，$\boldsymbol{P}_c$ が不明のため P_j' はこの保証はなしに選ばれることになる．そこで置換生成候補の選定のための条件を定め，かつ知識構造を有効に利用して，この探索範囲を出来る限り限定することにより，無駄な置換の生成を少なくする．これについては後に述べることにし，まず前述の置換規則をみたす置換生成の方法を示すことにする．

6.3 置換生成手順

質問 C と，この C 内に含まれるものと共通の述語記号 G をもつリテラルを含む知識 P が次のように与えられたとする．

$$P;\ (Q_1x_1/X_1)(Q_2x_2/X_2)\cdots(Q_nx_n/X_n)$$
$$[\sim F(x_{i1},x_{i2},\cdots,x_{im})\cup G(x_{j1},x_{j2},\cdots,x_{jp})]$$

$$C;\ (Q_{C1}x_{C1}/X_{C1})(Q_{C2}x_{C2}/X_{C2})\cdots(Q_{Cl}x_{Cl}/X_{Cl})$$
$$[H(x_{k1},x_{k2},\cdots,x_{ks})\circ G(x_{j1},x_{j2},\cdots,x_{jp})]$$

ここで$\circ$は "離接または合接" を示す．また次のような変数の集合を定義しておく．

$$\boldsymbol{X}_0=\{x_{j1},x_{j2},\cdots,x_{jp}\},\ \boldsymbol{X}_1=\{x_{i1},x_{i2},\cdots,x_{im}\},$$
$$\boldsymbol{X}_2=\{x_{k1},x_{k2},\cdots,x_{ks}\},$$
$$\boldsymbol{X}_3=\boldsymbol{X}_0\cap(\boldsymbol{X}_1\cup\boldsymbol{X}_2),\ \boldsymbol{X}_4=\boldsymbol{X}_0-\boldsymbol{X}_1-\boldsymbol{X}_2,$$
$$\boldsymbol{X}_5=\boldsymbol{X}_1-\boldsymbol{X}_0,\ \boldsymbol{X}_6=\boldsymbol{X}_2-\boldsymbol{X}_0.$$

なおここで $\boldsymbol{X}_0\wedge\boldsymbol{X}_2\neq\phi$，$\boldsymbol{X}_0\wedge\boldsymbol{X}_3\neq\phi$ とする．これはもし P において F と G あるいは C において H と G とが共通の変数を有さない場合にはこれらを別個の二つの表現に分けて扱えるからである．

C の P による置換 R の基本形は

$$R:\ (Q_{ir_1}x_{ir_1}/X_{ir_1})(Q_{ir_2}x_{ir_2}/X_{ir_2})\cdots$$
$$(Q_{irm}x_{irm}/X_{irm})$$
$$[F(x_{i1},x_{i2},\cdots,x_{im})\circ H(x_{k1},x_{k2},\cdots,x_{ks})]$$

である．$\circ$は C 内のものと同一の接続子である．この R について，まず(1)各変数の限量記号，(2)限量記号の順序，(3)変数の変域，を定めれば最終的な形が決定される．このうち変域については後に述べることとし限量記号に関する部分の決定手順を示す．この決定には導出原理を利用することができる．すなわち置換生成の条件のうち $P\cap R\Rightarrow C$ なる R は $P\cap R\cap\sim C$ を unsatisfiable にするものとして求められる．P,R,C を通常の一階述語形式に変換し，Skölem 関数を使って標準形に直したとする．以下では C は否定形 $\sim C$ で扱うので，$\exists$ と $\forall$ はもとのものの逆になることに注意されたい．まず R の変数 x_k の限量記号を定めるために，P と $\sim C$ における F,G,H の各リテラル内の $\exists$ 変数を先行 $\forall$ 変数の関数で表わしたものを **Fig. 3** ((次頁参照)（限量記号の条件)) のように表わしてみる．この図で⊙は $\forall$ 変数を表わし，△は $\exists$ 変数，したがって先行する $\forall$ 変数の関数もしくは定数である．導出原理では代入によって各リテラルを unify するが，実線矢印はこの代入の操作を示す．この時の条件より以下の結果が得られる．

(1) P,C 共通のリテラル G は unifiable でなく

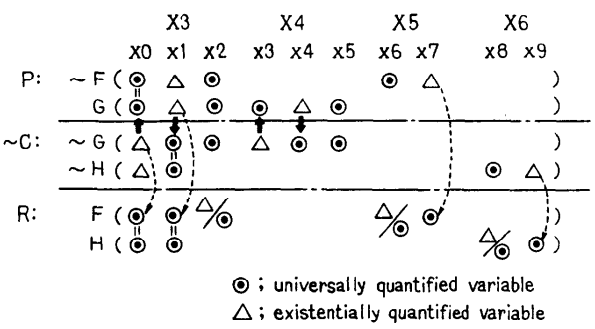

Fig. 3 The condition of each variable being either universally or existentially quantified variable in the replacement.

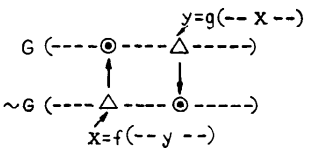

Fig. 4 Circuit condition

てはならない．すなわち P と $\sim C$ に関して，G および $\sim G$ は符号を除き，これらを同一のものにする代入が可能でなければならない．このための必要条件は G および $\sim G$ の同一変数位置に共に関数もしくは定数が生じてはならないことで，このために G の変数の限量記号が Table 1 の条件を満たさねばならない．これは含意規則の(1)に相当する．さらに各関数はいくつかの変数をアーギュメントとして含むが，**Fig. 4**（サーキット条件）のような状況において任意の関数の組について，G 内のある変数 y の Skölem 関数が変数 x をアーギュメントとして含みかつ $\sim G$ 内の変数 x の関数が y をアーギュメントとして含む場合は unifiable でない．変数が二つ以上でも同様で，これは変数をノード，関数関係をアークとして定義できるグラフ内にサーキットができることを示すので，サーキット条件と呼ぶ．含意規則の(2)はこの条件の禁止に相当する．

この結果は P が置換生成の候補となる条件を示している．この条件は，“P からリテラル G の部分を変数の限量記号順序を変えずに抽出したものが，C から同様に限量記号順序を変えずに抽出したリテラル G を含意すること” である．

(2) R 内変数の限量記号についてはまず P と $\sim C$ の間で代入が行われ，$\forall$ 変数が関数または定数で置きかえられる場合，すなわちある変数の限量記号が P で $\forall$，$\sim C$ では $\exists$（したがって C では $\forall$），あるいはこの逆の場合（P で $\exists$，$\sim C$ で $\forall$，すなわち C では $\exists$），対応する R 内変数は代入を可能とするために $\forall$ 変数でなくてはならない．これにたいし，P と $\sim C$ で共に限量記号が $\forall$ である変数（したがって C では $\exists$）については R は $\forall$，$\exists$ いずれの限量記号であってもよい．しかし置換規則より，この場合には $\exists$ 変数とすべきことが判る．他のすべての場合が **Table 2**（置換内変数の変域と限量記号）に示されている．

(3) R の変数の限量記号順序は前記サーキットを形成しないように定める．これについても置換規則より各 $\exists$ 変数を Skölem 関数で表わした時，この条件内で最大数のアーギュメントを含むように限量記号順序を定めることが必要である．

$P, \sim C, R$ 間で unification を行う時，R 内の $\forall$ 変数には P および $\sim C$ 間で代入された結果の関数または定数が代入される（**Fig. 3** 点線）ので，上述の条件は R 内の $\exists$ 変数を Skölem 関数で表わす時の関数形のみで調整される．この意味で $\forall$ 変数に代入されるものを “所与の関数” の意で G 関数，$\exists$ 変数に与えられるものを “可制御の関数” の意で C 関数と呼ぶ．すると限量記号順序は G 関数が与えられたとき，各 $\exists$ 変数について最大数の変数をアーギュメントとして含み，かつサーキット条件を満たさないように C 関数を定めることによって求まる．

G 関数により導入される変数間の順序関係にもとづいて R の全変数は半順序集合を形成するが一般には線形順序列は形成しない．C 関数はこれを線形順序化するものであるが，この線形順序化は必ずしも一通り

Table 2 Domains and quantifieres of the variables in the replacement.

	Qpi	Qci	Qri	Xri	CONDITION
$xi \in X3$	$\forall$	$\forall$	$\forall$	Xci	$Xpi \supset Xci$
	$\forall$	$\exists$	$\exists$	$Xpi \wedge Xci$	$Xpi \wedge Xci \neq \phi$
	$\exists$	$\exists$	$\forall$	Xpi	$Xpi \subset Xci$
$xi \in X4$	$\forall$	$\forall$	—	Xci	$Xpi \wedge Xci$
	$\forall$	$\exists$	—	$Xpi \supset Xci$	$Xpi \wedge Xci \neq \phi$
	$\exists$	$\exists$	—	Xpi	$Xpi \subset Xci$
$xi \in X5$	$\forall$	—	$\exists$	Xpi	
	$\exists$	—	$\forall$	Xpi	
$xi \in X6$	$\forall$	—	$\forall$	Xci	
	—	$\exists$	$\exists$	Xci	

Xpi: The domain of xi in P. Qpi: The quantifier of xi in P.
Xci: The domain of xi in C. Qci: The quantifier of xi in C.
Xri: The domain of xi in R. Qri: The quantifier of xi in R.
( ): not included in the replacement but only saved.

ではない．これが複数個ある場合，同一の P,C から置換が複数個生成されることになる．

しかし置換は演繹過程における中間生成物であるから，置換列を形成するためだけに用いられるものについては必ずしもその表現形式を一般の述語と同形式とする必要はない．この場合，置換に要求されることは，さらに次の置換を生ずるための候補を判定し得ること，すなわち含意判定が正しく行われることで，この目的には変数順序を線形化する必要はなく，G関数を求めておけば十分である．この時の問題は含意規則(2)をいかに判定するかであるが，これをサーキット条件から行う．すなわち，P 内のヨ変数 x_i に先行する∀変数 x_j がある時，この x_j が C 内に G 関数を有し，かつそのアーギュメントに x_i を含む場合サーキットが形成される．この判定は極めて容易である．

G関数を求める段階に処理をとどめたものも置換と定義するなら，P,C から生成される置換は唯一である．G 関数は R 内の各∀変数ごとにそれに先行する変数集合で与えればよく，これは Table 2 にもとづき R 内に∀変数をもたらしたもとの変数が，(1) P 内でヨ変数だった場合，(P 内で) 先行する∀変数の集合として，(2) C 内で∀変数だった場合，(C 内で) 先行するヨ変数の集合として，求められる．引き続いて置換が生成されている場合，(2)の変数については前段の置換ですでにこの集合は求まっているから，(1)の変数についてのみ G 関数を付加すればよい．

最後に変数の変域は P および C における変数 x_i の変域をそれぞれ X_{Pi}, X_{Ci} とすると，限量記号の組合せに応じて Table 2 のように定められる．この表には R の限量記号および置換生成を可能にする条件(含意条件と同じ)を併記してある．この変域の定め方を簡単な一変数の場合について示してみる．まず P, C, R をそれぞれ

$$P: (\forall x/X_p)[\sim F(x) \cup G(x)]$$
$$C: (\forall x/X_c)[H(x) \cap G(x)]$$
$$R: (\forall x/X_r)[H(x) \cap F(x)]$$

とする．これを従来の表現に変換した後 C の否定形をとり，標準形で表わす，結局これらから作られる set of clauses は

$$P \quad \sim X_p(x), \sim F(x), G(x),$$
$$\sim C \begin{cases} X_c(a), \\ \sim H(a), \sim G(a), \end{cases}$$
$$R \begin{cases} \sim X_r(x), F(x) \\ \sim X_r(x), H(x) \end{cases}$$

である．置換の限量記号に関する前述の条件が満たされていれば，上例のように F,G,H の各リテラルは unifiable であることが常に保証される．これは多変数の場合でも全く同様である．残りは変域に関する部分で，この部分の resolvent を空にすることができれば全体が unsatisfiable であることを証明できる．上例ではまず含意規則から集合 X_c と X_p の間には $X_p \supset X_c$ の関係があり，論理表現では $(\sim X_c(x), X_p(x))$ となる．これにさらに X_r に関する性質が必要であるが，$X_r \supseteq X_c(X_r(x), \sim X_c(x))$ であれば unsatisfiablity の条件が満たされることが分る．これと置換規則の条件を考え合わせると $X_r = X_c$ とすべきことが明らかである．他のすべての場合もこれと同様にして求めることができ，この結果を表にしたものが Table 2 である．このようにして得られた変域は質問にたいしてこれを真にする答の内容をも表わしている．

このようにして生成された R が真に置換であることの証明は生成手続きからほぼ自明である．R が $P \cap R \Rightarrow C$ の条件を満たすことは明らかであり，R の各変数について限量記号，その順序，変域のいずれかを変えたものを R' とすると，R' は $R' \cap P \Rightarrow C$ であるか，$R' \Rightarrow R$ であるかのいずれかであることは容易に証明できる．また置換は P,C が共通の述語記号を有するリテラルを共に含む時，これらが含意条件を満たす限り必ず生成される．したがって C を導く知識が存在する時は必らず C は証明される．

以上の議論において H,F は単一のリテラルである必要はない．一般に P を離接標準形で表わし，

$$P: (Q_1x_1/X_1)(Q_2x_2/X_2)\cdots(Q_nx_n/X_n)$$
$$[F_1 \cup F_2 \cup \cdots \cup F_i \cup \cdots F_n]$$

とする．ここで各 $F_j (j=1,2,\cdots,n)$ は一般にリテラルの合接である．ここで F_i がリテラル G を含むとすると，以上の議論において $\sim F(x_{i_1}, \cdots, x_{i_m})$ は $(F_1 \cup F_2 \cup \cdots \cup F_{i-1} \cup F_{i+1} \cup \cdots F_n)$ に対応する．F_i 内の G 以外のリテラルは置換生成に関係ないので G を含む任意の合接は単独の G として扱われる．同様に C は一般に AND-OR 木構造で表わされ，G はその木構造の端末の一つである．H はこの木構造の G 以外のすべての部分構造を示す．すると置換 R は上記 $\sim \bigcup_{j \neq i} F_j$ で C の AND-OR 構造の端末 G を置きかえたものに相当する．

時として同一 C にたいし複数個の置換候補が存在する．たとえば $P_i = \sim A_i \cup B_i$, $(i=1,2,\cdots,N)$ なる N 個の知識が存在し，各 B_i が C 内のリテラル G

を含意する場合，すべての A_i が置換の候補となり得る．全知識は個々の知識の合接であり，$\bigcap_i(A_i \Rightarrow G) \Leftrightarrow (\bigcup_i A_i \Rightarrow G)$ あるから，この場合 $\bigcup_i A_i$ を置換要素とする必要がある．このように複数個の置換候補が存在する場合，この個々の置換の離接をもって全体の置換とする．

7. 演繹の手順

以上の演繹規則を具体化するためのアルゴリズムは比較的単純である．演繹処理の手順は，(1)知識構造を探索し，置換候補を見出すこと，(2)置換 R の生成，(3) P と C の構造を併合し，R の構造を作ること，(4) R 構造の整形と処理の四段階から成る．以下これを簡単に記す．

〔1〕 **関連知識の検索**

質問 C が与えられた時，まずこれを手掛りとして，置換生成の候補となる知識を知識構造から探し出すことが必要である．C 内のリテラルの述語記号が通常この手掛りの一つとなるが，全知識をこれだけを手掛りに検索するのは効率が良くない．探索時間を早めるには探索範囲をできるだけ絞る必要がある．もとより，この時必要な知識が落ちてしまってはならない．

Table 2 より分るように，C 内のあるリテラル G が∀変数または定数（以下定数は∀定数と同様にみなす）を含む時，この G に関して C の置換生成の候補 P がもし存在するなら，上記∀変数に対応する P 内の変数も∀変数で，かつ変域 X_p が C 内のもの X_c より大なるもの ($X_c \subset X_p$) でなくてはならない．このような P は知識構造において，X_c より上位の集合に接続されている筈である．すなわち，置換の候補知識の存在範囲は X_c の上位集合に限られ，これらは X_c から知識構造を上方に辿ることにより，有限時間内にすべて探索できる．上位集合を辿りながら，各集合に述語記号 G のリテラルが接続されているか否かを調べ，存在したら(2)の処理に進む．任意の集合について，その上位集合の全体は全知識の集合より小さな集合であるから，探索範囲は著るしく限定される．また G が二つ以上の変数∀を含む場合は全体集合 U に最も近いものを選ぶことにより探索範囲を最小にすることができる．

C が∀変数も定数も含まない場合，たとえば $(\exists x/PS)(\exists y/PS)\mathrm{LOVE}(xy)$ (“誰が誰を愛しているか”-PS: PERSON) のような場合，P の存在範囲は $X_c \supset X_p$ または $X_c \cap X_p \neq 0$ なる領域に広がってしまう．しかしこの場合でも質問が特定の要素のみに関する知識により含意される場合はこのような知識は表形式にまとめられているので，述語記号を索引にしてこの表を探すことにより解が見出される．これ以外は全知識の探索を必要とする．

〔2〕 **含意判定および置換の生成**

すべての知識および質問は一般に複合知識であるが，接頭部の情報はまとめて表形式で表わしておくのが具合が良い．この表には各変数の限量記号，その線形または半順序，変域へのポインタまたは実数区間の場合その区間などの情報を含む．置換生成の候補知識が見出された際，P の接頭部表が取り出されれば，以後〔2〕の処理の大半は P と C の接頭部表のみの処理で済むからである．

(a) 含意テスト；Table 1 の条件で含意関係のテストを行い，P が含意条件をみたさなければこれを捨て(1)に戻る．条件を満たしたら変数の限量記号，変域を Table 2 のように定める．

(b) 置換の接頭部表の生成；置換の各変数の変域を Table 2 に基づいて定め，P および C の限量記号順序から R の変数についての限量記号の半順序情報を各∀変数ごとに先行する変数集合の形式で定める．一般に一段前の置換 R_i と P_i とから新しい置換 R_{i+1} を作る時，R_{i+1} の半順序情報は R_i のものに P_i から来る新変数に関するもののみを付加すれば良い．

〔3〕 **P と C の併合**

$P=\bigcup_j F_j$ で F_i がリテラル G を含む時 $\sim\bigcup_{j\neq i} F_j$ の AND-OR 構造を作って C の AND-OR 構造内リテラル G の部分を置きかえるという形で併合したものが R の構造となる．

〔4〕 **AND-OR 構造の整形と処理**

置換生成の際，空なる置換が生じたら元のリテラル G は真なることが証されたので，これに Y なるレーベルを付す．逆にもし G と矛盾する知識が見出されたらレーベル N を，また置換が一つも見出せないまま，〔1〕の知識構造探索が最上位に至って終了した場合，レーベル K を付す，すると

(a) AND-OR 構造において AND 節点に接続されたレーベル Y 付きの節点は除去される．一方 N または K を有する節点があったら，この AND 節点自身にレーベル N または K を付し，AND 節点に接続されたすべての下位構造を削除する．

(b) OR 節点にレーベル Y 付きの節点があった場合，この OR 節点自身にレーベル Y を与え，このOR 節点に接続されたすべての下位構造を削除する．一方 N または K 付きの節点があったら，これらの節点以下を削除する．

以上の処理と，AND-OR 構造の整形を組合せ，最終的に最上位節点に Y, N または K のいずれかのレーベルが付せられるまで上記手順を繰り返す．

8. 結　論

新らしい知識表現形式を用いる時，それに適した演繹処理方式を確立しておくことは不可欠である．同時に，いかに優れた演繹能力を有しても知識表現が知識の総合的な利用に適さなければ無用のものとなる．この理由でこの両者は不可分な関係にある．本論文は集合の概念を中心においた知識表現形式を用いた場合の演繹処理の方法について述べたものである．

この方法の開発には二つの前提条件があった．一つは上述の知識表現形式に適応するものであること，もう一つは質問から出発して知識を検索しながら演繹処理を進め得るものであることである．演繹とは元来所与の前提から結論を導くプロセスであるから，これはむしろ演繹的検索とでも呼んだ方がふさわしいものである．

この演繹処理のアルゴリズムは比較的単純であり，扱う知識表現形式が冗長性の少ないこと，構造を利用して知識探索の範囲を大幅に狭め得ることなど，これ自体としてもいくつかの特徴をもつものである．ただ現状ではこれらの特徴がもたらすと期待される利点を定量的に証することは難かしい．それはこれらの特徴が大量の知識が存在する場合に効果をあげる性質のものだからであり，現在はまだそのような段階にとても達し得ないからである．知識が少量の場合にこの効果はあまり発揮されないのは知識構造そのものの特徴でもある．

知識を階層構造化することの利点は知識間のある種の関係は遷移的であり，任意の組合せの知識の対について連想関係を表現しなくても構造内のリンクを次々に辿ることによって長い連想の筋道が見出されることにある．これは構造のもつ大きな利点であるが，同時にこの関連を見出すのに時間を使ってリンクを辿らねばならないのは欠点にもなり，また構造を作るための初期投資が必要になる．しかし構造化により知識表現の数が増大しても複雑さはほぼその数に比例する程度であるのに，たとえば任意の組の連想関係をすべて記憶するとこの数の自乗の程度で記憶量が増大する．したがって知識の総量がある一定限度を越した場合は構造の有する利点が欠点を越えると予想されるのである．

本論文においては知識表現そのものや実例については述べられなかったが，別の機会に述べたいと思う．

最後に本論文の御検討を頂だいた東京大学宇宙航空研究所，穂坂衛教授に厚く感謝の意を表す次第である．

参 考 文 献

1) Kowalski, R.: "AND/OR Graphs, theorem-proving graphs and bi-directional search". Machine Intelligence 7, pp. 167～194 (1972).
2) Ohsuga, S.: "A Knowledge structure and a deductive inference rule based on it". Second USA-Japan Computer Conference, 1975 (to be published).
3) Quillian, M.R.: "The teachable language comprehender: a simulation program and theory of language". Comm. ACM Vol. 12, No. 8, pp. 459～476 (1969).
4) Robinson, J.A.: "A machine oriented logic based on the resolution principle". J. ACM Vol. 12, pp. 23～41 (1965).
5) Schwarcz, R.M., Burger, J.F. and Simmons, R.F.: "A deductive logic for answering english question" Comm. ACM Vol. 13, pp. 167～183 (1970).
6) Schank, R.C.: "Finding the conceptual content and intention in an utterance in natural language conversation". Second Int-l Joint Conf. on Artificial Intelligence, pp. 444～454 (1971).
7) Shapiro, S.C.: "A net structure for semantic information storage, deduction and retrieval". Second Int'l Joint Conf. of Artificial Intelligence, pp. 512～523 (1971).
8) Simmons, R.F. and Bruce, B.C.: "Some relation betweenpredicate calculus and semantic net representation of discourse". Second Int'l Joint Conf. of Artificial Intelligence, pp. 524～530 (1971).
9) Slagle, J.R. and Koniver, D.A.: "Finding resolution proofs and using duplicate goals in AND/OR trees" Information Science 3, pp. 315～342 (1971).

(昭和50年6月16日受付)
(昭和50年8月23日再受付)