

三 中 穂 積

(東京工業大学)

1. 言語理解システムと知識そして認知科学

筆者が行って来た(自然)言語理解システムの研究に関連して、知識と推論の問題について述べてみたい。

言語理解のための知識として、我々は言語学的な知識(文法等)とそれ以外の知識(常識等)とがある事を知っている。前者については言語学者が様々な角度から研究している。ところが後者については、それがどの様なものであるかについて、哲学者、心理学者、言語学者等によって断片的な研究はなされてはいるものの、体系的な研究がなされているとは言い難い。

頭脳の中から知識を取り出して眺めることができれば、我々は知識の姿について、なにがしかのコンセンサスを得ることができるかもしれない。そのためにたとえば、頭脳の機構を神経細胞のレベルから解明することが考えられる。知識の素性に対するいわばボトムアップ的な迫り方である。しかし、微細な神経細胞に関するこれまでの研究成果を利用した言語理解システムはまだ実現していない。かなり先の事になるだろう。

知識の研究をトップダウン的に進めることもできる。言語理解システムの研究者が採用しているアプローチがそれである。このアプローチの特徴は、ひとまず頭脳の神経細胞レベルでのミクロな研究に目をつぶり、言語理解システムの作成を試み、システムがどの様な姿態から構成されるべきかを研究し、その中で知識がどの様な形式である事が望ましいか、またそれらがどの様に使われるかを研究するのである。この場合、知識はシステム内で何等かの実体として表現されるが、この実体は我々から見ると、記号の塊としてみえ

る。それ以上に詳細なもの、例えば電子のレベルにまで分解して観察しても意味がない。記号の塊としての知識は実際取り出して眺めることができるだけでなく、頭脳の中の知識と異なりその挙動は、適当なプローブを差込むことで比較的詳細に追跡することができる。言語理解システムについても同様である。

このような実験により言語理解に関する新しい知見が得られれば、それを我々の頭脳が行っている言語理解の機構に対する一つのモデルないしは予測として用い、神経細胞レベルのボトムアップ的な研究をも効率良く進めることができる。今世紀最後の科学であるといわれる認知科学は、コンピュータ上にこのようなシステムを実現し、それを実験道具として我々の認知の機構に迫る事が一つの眼目であったと思われる。

2. 知識表現と推論

ここで言語理解システムを作成する上での望ましい知識表現とはどの様なものかを考えてみたい。筆者は知識は推論と背中合わせでなければならないと考えている。推論機構の裏付けのない知識は不完全な知識であると言って良い。この考え方は近似的に次の等式で表される。

$$\text{知識表現} = \text{表現形式} + \text{推論機構}$$

図1 知識表現の等式

知識表現で良く使われるセマンティック・ネットワークやフレームをこの立場から見るとどうなるか。この種の知識表現では、それを解釈するためのインタープリタを別途必ず作成しなければならないことに注意しよう[Nilsson 80]。推論はインタープリタを介して行われる。したがって、フレームやセマンティック・ネットワークで表現された知識は、使用者個々が適当な推論機構をインター

ブリタに組み込まない限り、推論機構の裏付けを持たない不完全な知識である。このような知識の問題点は、それをどの様なものとしてみて使うかは、使用者個々の作成するインタプリタがどの様なものであるかに依存することである。そのため表現されたものの形式は同じでも、その意味が使用者に依存し（外部からみて）不明確になる。

以上の説明を図示すれば次のようになる。

- (a) セマンティック・ネットワーク、
フレームによる知識表現

- (b) セマンティック・ネットワーク、フレームに対する、プログラミング言語L1で記述されたインタプリタ
=====
- (c) プログラミング言語L1のインタプリタ

図2 セマンティック・ネットワーク、フレームによる知識表現

(b)の部分は、セマンティック・ネットワーク、フレームによる知識表現を使うために個々のユーザが作成するインタプリタである。(b)を作成する労力は無視できない。(c)はプログラミング言語L1に対するインタプリタであるから、システムに組み込まれている。

図1の知識表現に対する等式では、(b)+(c)の部分が推論機構に対応している。(b)の部分を個々のユーザが作らなければ働かない知識表現は、不完全な知識表現である。筆者は[Barwise 83]の状況意味論で提案されている知識表現も、推論機構の裏付けをまだ持っていないため、現状では不完全な知識表現であるといわざるをえないと考えている。その点に関して、[Mukai 85]の研究が注目される。

先に述べた知識表現に対する等式からす

れば、形式的な推論機構の裏付けのある述語論理による知識表現は、望ましい知識表現の一つであるといえる。特にロジック・プログラミングの枠内で知識表現の問題を考えることは意味がある。なぜなら、一定の形式の述語論理で表現された知識に関する推論は、全てロジック・プログラミング・システムに組み込みの機能を利用して行うことが可能になるからである[Lloyd 84]。

後述するが筆者等の最近の研究によれば、セマンティック・ネットワークを自在に辿る機能、上位にある知識を下位からアクセスする機能（知識継承機能）、上位のメソッドを使う機能等は全て、ロジック・プログラミング（我々の場合Prolog[Clocksin 81]）の枠内のユニフィケーション機構で代用可能なことが明らかにされている。これはDCKR(Definite Clause KnowledgeRepresentation)と呼ばれる知識表現形式であるが[田中 86]、DCKRを用いれば先の図2に対応して図3を得る。

- (a) DCKRによる知識表現

- (b) なし
=====
- (c) Prolog のインタプリタ

図3 DCKRによる知識表現

DCKRは、ある定まった形式のPrologプログラムであるから、図2に示したような(b)に相当するプログラム（インタプリタ）を個々のユーザが作成する必要はない。それは直接Prologによってインタープリトされる。この部分はシステムに組み込まれているから、ユーザはDCKRによる知識表現を行うだけで良いことになる。

ここで次の事を注意しておきたい。セマンティック・ネットワークやフレームによる

知識表現が不完全であるからと言って、それ

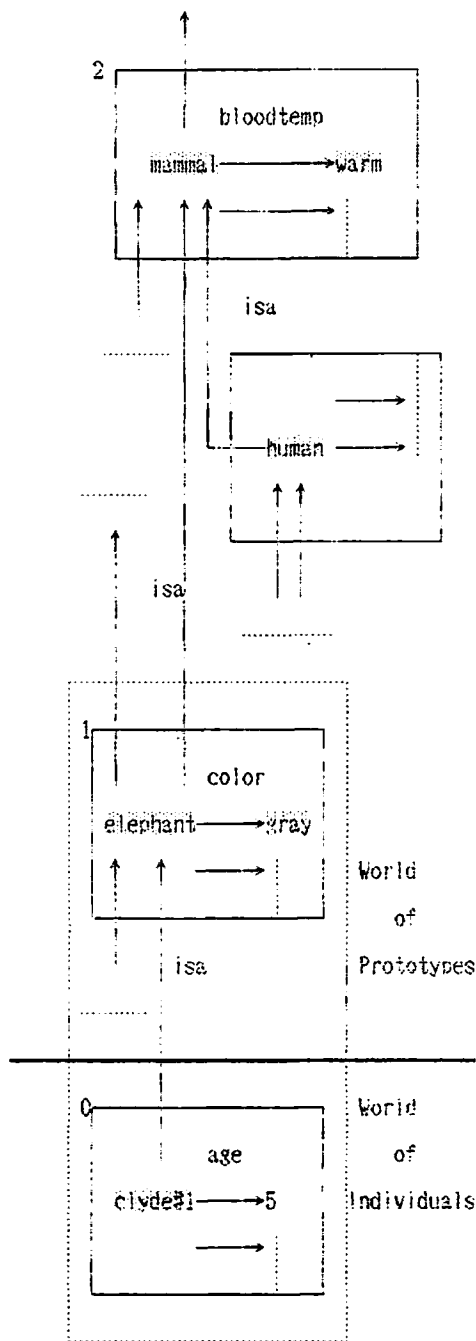


図4 セマンティック・ネットワークによる知識表現例。

らを無視して良いと主張している訳ではないということである。セマンティック・ネットワークやフレーム形式の知識表現の良さは、表現されたものの意味が直感的に理解できるという点にある。もちろん我々が直感的に理解できることが、プログラムとして簡単に実

現できるということを意味している訳ではないのであるが。

図4にセマンティック・ネットワークによる知識表現の例を示す。図中の矩形で囲まれた部分は一つのフレームに対応している。図4から我々は少なくとも次の事実が読み取れる。

- (1) clyde#1のageが5である、
- (2) clyde#1はelephantである、
- (3) elephantのcolorはgrayである、
- (4) elephantはmammalである、
- (5) mammalのbloodtempはwarmである、
- (6) humanはmammalである。

このネットワークを使って例えば「clyde#1のbloodtempは何か」を知りたいければ、isaリンクを上に通じ、パターン照合を繰り返しbloodtempリンクを発見し、その先の値を取り出すプログラムを作成する必要がある。

「何がmammalですか」という質問に答えなければ、mammalノードを発見してそこから図4のネットワークを下に通じプログラムを作る必要がある。したがって図4のネットワークには、両方向に通じるための特別な仕掛けをあらかじめ用意しておかなければならない。このようなプログラムと仕掛けとが図2の(b)の部分に対応している。

セマンティック・ネットワークによる知識表現は直感的に分かり易いということを既に述べた。DCKRによる知識表現は、セマンティック・ネットワークの良さを保存しつつ、セマンティック・ネットワーク以上の記述能力を持たせることを目標に設計された。しかも推論機構としてPrologに内蔵された読解能を直接利用することができる。これを示すために、図4の知識をDCKRで表現してみると図5のようになる【小山 85】、【田中 86】。

```

sem(clyde#1,age:5).
sem(clyde#1,P) :- isa(elephant,P).
. . . . .
sem(elephant,color:gray).
sem(elephant,P) :- isa(mammal,P).
. . . . .
sem(mammal,bloodtemp:warm).
. . . . .
sem(human,P) :- isa(mammal,P).

```

図5 簡略化されたDCKRによる知識表現例

ただし図5は、説明を簡単にするために簡略化したDCKR表現になっている。図5に示した知識は、図4から読み取れる知識を直接書き下したものであることが分かる。ここで図5で使われている述語isaの定義を以下に示す。

```
isa(Up,P) :- P=isa:Up ; sem(Up,P).
```

以上の準備をしてから、もし「clyde#1のbloodtempは何か」を知りたいければ、

```
?-sem(clyde#1,bloodtemp:X).
```

を実行すれば良い。Prologインタプリタは図5のプログラムを用いて、

```
X = warm
```

と答えて来る。

次に「何がmammalですか」ということを知りたければ、

```
?-sem(X,isa:mammal).
```

を実行すれば、Prologインタプリタは、

```
X = clyde#1;
```

```
X = elephant;
```

```
X = human;
```

と次々に答えて来る。

更に「clyde#1にはどのような性質があるか」という質問に対して、

```
?-sem(clyde#1,P).
```

を実行すれば、

```
P = age:5;
```

```

P = isa:elephant;
P = color:gray;
P = isa:mammal;
P = bloodtemp:warm

```

と答えて来る。

セマンティック・ネットワークでは表現しにくかった全称限量表現 Everyone who lives at Maple Street 43 is a programmer という知識もDCKRを用いれば、

```

sem(X,profession:programmer) :-
    sem(X,isa:human),
    sem(X,location:maplestreet43).

```

として素直に記述できる。この知識はそのままPrologプログラムとして実行され、Maple Street 43に住んでいる人の職業を計算することができる。

以上で図2、図3の意味が比較的明確になったと思われる。自然言語理解システムを作成する上で、意味的な処理を行なうための知識が必要になる。この知識もDCKRを用いることで容易に記述することができる [田中88]。

3. 言語理解システムの構成

前節では、知識と推論の問題を論じ、セマンティック・ネットワークやフレームによる知識表現の問題点を指摘した。そして、ロジック・プログラミング言語の典型であるProlog上での知識表現形式DCKRを説明した。Prologに裏付けされたDCKRは、セマンティック・ネットワークやフレームの問題点を克服できる事を述べた。

次に言語理解システムの一般的な構成を示し、その中でDCKRの果す役割について説明する。図6に言語理解システムの一般的な構成図を示す。

4. 知識情報処理のための基本計算機構

述語論理による知識表現はこれまで哲学者や言語学者によってしばしば論じられて来た。しかしこの知識表現は心理学者によって、必ずしも支持されていない様である。述語論理で良く使われる演繹的な推論が人間の行う推論と必ずしも馴染まないとする見解が心理学者の間に少なからずあるからだろうか。ロジック・プログラミング言語の典型であるPrologは、この演繹的な推論を基本計算機構としている。しかしこの事はPrologにより他の推論機構がシュミレート不可能なことを意味しないことに注意すべきである。

Prologでむしろ重要なことは、推論方式うんぬんより、ユニフィケーションが基本計算機構になっていることである。ユニフィケーションはパターン照合の最も基本的でしかも強力な操作である。パターン照合が全ての認知活動の基本であると主張したとしても、そう大きな間違いではないだろう。だとすれば、ユニフィケーションを基本計算機構に取り入れたPrologのそのものに、哲学者や心理学者の関心がもっと寄せられても良いのではないか。

言語理解システムの作成の立場から、構文処理と意味処理とを融合させたモデルが必要であるということが、かつて主張された。これは心理学的にも妥当なモデルだと考えられている。この融合もPrologの枠内で、極めて素直に実現可能なことが知られている。更にそこでは、いわゆる構文解析用に特別なプログラムを作る必要はない。Prologに組み込みの機能で代用できるからである。

我々は頭脳のなかの限られた資源を用いて様々な知識情報処理を行っている。問題毎に異なる知識情報処理のためのプログラムが動作すると考える事もできよう。しかしその場合であっても、それら全てに共通する部分があって、それを中核として様々な知識情報

処理を行っていると考えerことは自然であろう。その中核の部分とPrologの持つ基本計算機構とを重ね合わせてみる事はできないだろうか。[Fordor 83]による水平的な処理を行うための基本計算機構としてである。前章までに述べて来たように、Prologの基本計算機構をそのまま用いて、従来個別に行われていた構文処理、意味処理、知識継承の問題、手続き付加の問題を統一的に扱うことができる。この多様な問題を統一的に扱えるという事実は筆者の予想を越えるものがあった。

以上のことから、[Fordor 83]の水平的な処理を行うための機構とPrologの基本計算機構とを重ね合わせてみるという問題に、哲学的、心理学的な検討を加える価値が全くないとは言えないように思われる。たとえ結論が否定的であったとしても、人間の持っているものと、どの点が異なるかを明らかにすることができれば、それだけでも意味があると思うのであるが。

跋言

本研究の契機は、ICOTの淵一博所長とのディスカッションにある。筆者の良き指導者であり助言者である同氏に感謝する。

参考文献

- [Barwise 83] Barwise, J. and Perry, J.: Situations and Attitudes, The MIT Press (1985).
- [Clocksin 81] Clocksin, W.F. et al.: Programming in Prolog, Springer-Verlag. (1981).
- [Colmerauer 78] Colmerauer, A.: Metamorphosis Grammar, in Boic (ed): Natural Language Communication with Computers, Springer-Verlag, 133-190(1978).

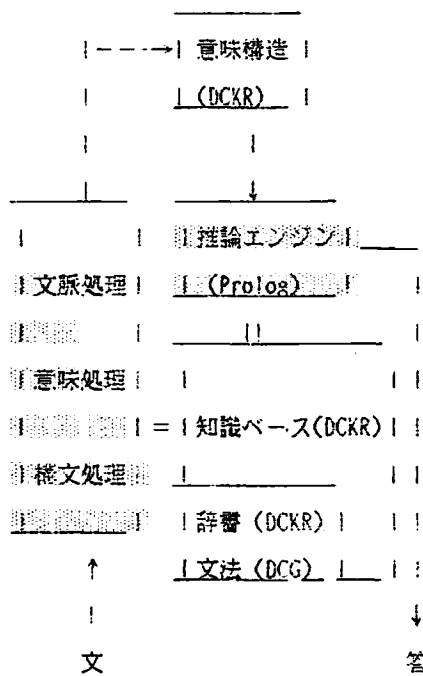


図6 DCKRと自然言語理解システム

図の左側には言語処理を行なう部分があり、図の真ん中に知識ベースがある。文が入力されると、知識ベースを用いて構文処理、意味処理、文脈処理が行なわれる。

[Coimeraire 78],[Pereira 82],[Matsumoto 83]の方法によれば、文法規則をDCGとよばれる形式で記述しておけば、それを最終的にPrologプログラムに変換することができる。そして、意味処理、文脈処理を行なうプログラムをDCGの文法規則の中に書込んでおくと、変換後のPrologプログラムに処理すべき文を与えて実行すれば、構文処理、意味処理、文脈処理を融合した言語処理を行なうことができる。

特に構文処理については、従来必要だったプログラムの全てをPrologに組み込みの機能で代用できる。意味処理についても、DCKRで記述された意味処理用の辞書を用いることにより、意味処理用のプログラムの殆ど全てをPrologに組み込みの機能で代用できる【田中 86】。図6の斜線を引いた部分はPrologに組み込みの機能で代用できることを示し

ている。

図6で、言語処理結果がDCKR形式になっていることに注意されたい。そうすることにより、例えば疑問文に対する応答は、(2章で述べた様に) DCKR形式の(疑問文に対する)意味処理結果をそのまま実行して答えを得ることができる。Prologに組み込みの機能をそのまま用いて、疑問文に対する推論問題解決を行なうことができる。図6の推論エンジンの部分は、Prologが代行してくれるので推論問題解決用のプログラムを作る必要もない。

明確に述べなかったが、DCKRに関するこれまでの説明から、Prologを用いると、プログラムとデータとを区別する必要がないことが分かる。従来多くのプログラミング言語では、プログラムとデータとをはっきり区別していた。そこではプログラムは動作する能動的なもので、受動的なデータを適宜使いながら問題解決を行うものであると考えられていた。ところが記号処理用言語の典型とされるLispやPrologでは、データもプログラムも同じ形式をしており区別されない。我々人間は果たして記憶の中で、プログラムとデータとを別形式として区別し保持しながら使っているのだろうか。

最後に文脈処理の問題について触れておきたい。文脈を理解するためには、文を処理した結果としての知識の形式をどの様にしたら良いかを深く考えなければならない。文脈理解の過程で様々な推論が行われることは良く知られている。したがって文脈理解についても、推論機構の裏付けのある知識を文から抽出しておくことは重要であろう。また比喩や隠喩の問題も、推論と深く関係している。この種の問題の解決には長期を要するだろうが、解決に向けての努力をすべき時期に来ていると思われる。それにより言語と知識との関連がより一層明確になるだろうし、知識表現の形式についてもよりよい考え方が生まれるように思われる。

[Fordor 83] Fordor, J.A.:
Modularity of Mind. The MIT Press(1983).

[小山 85] 小山晴生 (他) :
Definite Clause Knowledge Representation.
Proc.of LPC'85, 95-106(1985).

[Lloyd 84] Lloyd, J.W.:
Foundations of Logic Programming.
Springer-Verlag(1984).

[Matsumoto 83] Matsumoto, Y. et.al.:
BUP--A Bottom-up Parser Embedded in
Prolog. New Generation Computing, 1, 2,
145-158(1983).

[Mukai 85] Mukai, K.:
Unification over Complex Indeterminates
in Prolog. Proc. of LPC'85, 271-278(1985).

[Nilsson 80] Nilsson, N.J.:
Principles of Artificial Intelligence.
Tioga, (1980).

[Pereira 80] Pereira, F. et.al.:
Definite Clause Grammar for Language
Analysis --A Survey of the Formalism and
a Comparison with Augmented Transition
Networks. Artificial Intelligence, 13, 231
-278(1980).

[田中 86] 田中穂積 (他) :
知識表現形式DCKRとその応用、コンピュータ
ソフトウェア、日本ソフトウェア学会編、岩波書
店 (掲載予定) .