

TR-197

知 識 表 現

— 論理的アプローチに焦点を当てて —

松 本 裕 治

August, 1986

©1986, ICOT

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

知 識 表 現

—論理的アプローチに焦点を当てて—†



松 本 裕 治**

1. はじめに

言語学においては、文の意味の表現法として、論理は非常に馴染みが深い。述語論理は数学の言葉として生み出されたものであるが、様相論理や内包論理など、自然言語の文の意味を表現するための論理に至っては、自然言語文の意味表現として、これら以上に記述力のある意味表現法はほかにはない。計算機科学における知識表現の問題でも論理は重要な役割を演じる。

知識表現について語るときに問題としなければならない事柄には、さまざまなものがある。しかし、それらの問題を最も簡単な言葉に集約して言うなら、何を表現するかということと、どのように表現するかということである。もちろん、この中には、それをどのように扱うかという意図も含まれていなければならない。

文の意味表現や、文章を理解するための知識には、どのような性質のものが必要であろうか。計算機科学において提案されてきた知識表現法は、次のように大きく分類することかできる。

- (1) 述語論理等の形式論理
- (2) プロダクションシステム等の手続き型言語
- (3) 意味ネットワーク等のグラフ的表現
- (4) フレーム、スクリプト等

ただし、本解説では、これらのそれぞれについての意義を述べ直すことはしない。これらについては、本学会誌に既に詳しい解説があるし¹⁾、そのほか、人工知能関係の入門書に、詳しい説明が見受けられる²⁾。

これらの知識表現法のうち、プロダクションシステムのような手続き型言語は、自然言語理解のような広い対象を扱うための知識表現法ではなく、ここでの議論の対象からは除外する。意味ネットワークとフレーム表現は、このように別に論じられることが多い。し

かし、例えば、is-a 階層などを特別視して、その他の概念の関係をひとまとめにしてモジュール化してみれば、意味ネットワークとフレームの違いはほとんど明確ではなくなる。Schank らの概念依存形式 (Conceptual Dependency)³⁾、スクリプト⁴⁾、MOPs (Memory Organizing Packets)⁵⁾ 等も、本質的には、意味ネットワークやフレームの一種である。これらをグラフ知識表現とでも呼んで統一的に扱うことにしよう。そうすると、知識表現法は、論理表現かグラフ表現かということになる。

本解説では、論理とは主に第一階述語論理を指すと限定する。なかには、一階述語論理の範囲を越える論理を対象にすることもあるが、一階述語論理は十分に強力な記述形式であり、さまざまな高階もしくは様相概念をその中に取り込むことが可能なのである⁶⁾⁻⁹⁾。また、計算機上に実現するという前提を持つ以上、一階述語論理を越える論理を扱うことは好ましくない。

本解説では、論理に関連した知識表現システムの例をいくつか紹介し、また、知識表現のための論理の拡張について解説することにより、知識表現における論理の重要性を浮き彫りにしたいと思う。

2. 知識表現における論理の立場

論理とグラフによる知識表現の違いおよび特徴は何であろうか。論理の最大の特徴は、その構文論および意味論がはっきりと定義されているということである。また、与えられた知識 (公理) の集合から推論によって導き出すことのできる事柄が、論理的帰結として明確に規定できることである。また、論理演算子や関係子のような豊かな記述子を持っていることである。

後者は、時にはまったく非常に堅苦しい言語のような印象を持たれることが多いが、実は、論理の特徴の一

† Knowledge Representation—In Favour of Logical Approaches
by Yuji MATSUMOTO (Institute for New Generation
Computer Technology).

** (財)新世代コンピュータ技術開発機構第一研究室

ただし、この点、様相論理のように推論規則が構文的に定めておらず、意味論に存在が依存されているものもあるし、モナディックユーニバース量詞¹⁰⁾のような、全量詞が存在して、形式的な推論規則が定めきれない論理もある。

つは、不完全な状況に対する記述力なのである。二つの事柄のいずれか一方が成り立つという選言的な表現は、論理和 (disjunction) によって簡単に表現できる。存在限量子 (existential quantifier) は、具体的な個体を示すことなく、ある性質を持つ個体の存在を暗示することができる。論理の否定は、完全否定や部分否定などのさまざまな否定を正確に記述し分けることができる。これらは、意味ネットワークやフレームなどによって容易に記述し得ないものである。

一方、論理を知識表現言語として用いることに対する最大の反論は、論理による表現があまりに平板であり、その推論に要する計算が非常に効率が悪いのではないかということである。その理由は、定理証明に関して提案されたさまざまな戦略が実用的に効果的ななかったからかも知れない。しかし、この問題は慎重に考える必要がある。論理における定理証明手続きが、時間的に効率よくなかったのは、その戦略が問題分野または問題の特徴に依存しない一般的な証明法を目標としていたからである。グラフ的な知識表現における推論が、グラフ上の探索によって、簡単に効率よく行えるのは、グラフ表現の特徴としてよく取り上げられるが、それは、実は、グラフのある種の枝 (例えば、is-a リンク) を特殊扱いしているからである。

また、グラフによる階層構造がいろいろな意味を持っているのは^{12), 13)}、グラフ表現の表現力が高いことを示しているのではなくて、いわば、グラフ表現に確立した意味論がないという欠点なのである。実際、これらは、明確な意味さえ与えられれば、一階述語論理に還元することができる性格のものである^{14), 15)}。

一方、論理的記述の平板さについては、知識の表現だけでなく推論自体を論理を用いて記述することができる^{16), 17)}ことを指摘しておきたい。これについては、後述する。

すべてを論理で表現するのは効率の低下がある。グラフ表現には意味の定数と表現力に問題がある。では、知識表現はどのような方向に進むべきであろうか。次章で、この問題に関連するいくつかのアプローチを眺めてみようと思う。

また、本章の内容に関連して、大森 19¹⁸⁾ ~ 20¹⁹⁾ などが参考になると思われる。

3. 論理表現とグラフ表現

本章では、論理とグラフ表現の融合に関するいろいろな手法を検討する。本章を以下のように二つの部

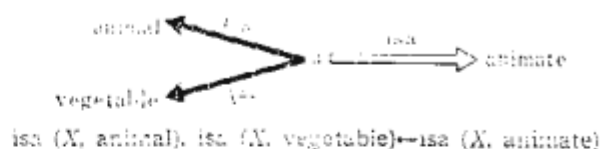


図-1 拡張意味ネットワークによる節表現の記述

に分けて説明するが、これは形式上の分類であって、両者の差は、連続的なものであることを注意しておく。

3.1 グラフ表現による論理の拡張

例えば、Deliyanni らによる拡張意味ネットワーク (Extended Semantic Networks)²⁰⁾を考えてみよう。彼らは、すべての述語を二引数述語で表現し、任意の節表現 (clausal form) による論理式を意味ネットワーク状に表現することを提案している。グラフの枝は述語名に対応し、節点は述語の引数となる項に対応する。この枠組みでは、意味ネットワークの意味は元の論理式によって正確に定義できる。また、節表現の右辺と左辺にある述語それぞれに対応する枝を区別して (色分けして) 表現することにより、融合原理 (resolution principle) による推論過程をグラフの糖選として表現できる。

図 1 に示すのは、拡張意味ネットワークによる、節表現の例である。節の左辺にあるリテラルを黒い枝で、右辺のリテラルを白抜きの枝で表現している。融合演算 (resolution) は、同じ名前を持つ枝で、その両端に現れる項がそれぞれ単一化可能なものを糖選させることに対応している。このアプローチにおける意味ネットワークの使用は、融合原理による証明過程の制御のための戦略の一種と考えることができ、その意味では、Kowalski²¹⁾ や Siekel²²⁾ によるグラフ表現を利用した証明戦略と同様の効果を持ったものとみることができる。

また、McSkimin の意味ネットワーク²³⁾は、概念間の包含関係 (階層構造) を意味ネットワーク (彼らは意味グラフと呼んでいる) によって表現し、意味ニューフィケーションという拡張された単一化を推論に利用している。彼らが扱っているのは、拡張 Π 節 (extended Π -clause) という、多ソート一階述語論理 (many sorted first order logic) の一表現である。拡張 Π 節は、節の雛形 (clause template) と変数への代入 (substitution) を示す集合 (の集合) のことによって表現される。clause template は、構文上は普通の節表現とほとんど同じであるが、述語名までが変数として扱われる。この変数は、その他の変数とともに、substitution set によって与えられる領域 (具体的な項および意味

グラフで表現される概念を用いた集合演算によって定義されている（ただし、述語名に対応する変数だけは有限集合でなければならないという制限が付く）。拡張 II 節は、構文的に相似の節をひとまとめに表現した節であるといえる。例えば、次に示すのが、拡張 II 節の例である。

$((\alpha, x, y), \{[LIKE]/\alpha, (little \cap boys)/x, [SANTA], ice-cream/y\})$

この節は、“All little boys like Santa and all types of ice cream” という文の意味を表現している。ここで、大文字で示されているのは個体名、小文字で示されているのは意味グラフで定義される概念の名前である。意味ユニフィケーションとは、拡張 II 節に対する単一化操作である。グラフ表現された階層概念を利用することによって、かなり複雑な単一化を高速に行うことができる。

3.2 論理によるグラフ表現の拡張

意味ネットワークの論理的な記述力の弱さを補うために、意味ネットワークの枠組みに論理演算子や限量子による記述を直接取り入れたものとして、Schubert の拡張された意味ネットワーク²⁵⁾や Hendrix による Partitioned Networks²⁶⁾ などがある。しかし、これらのアプローチは、論理と意味ネットワークとの結合としては、正しい方向とは言えない。

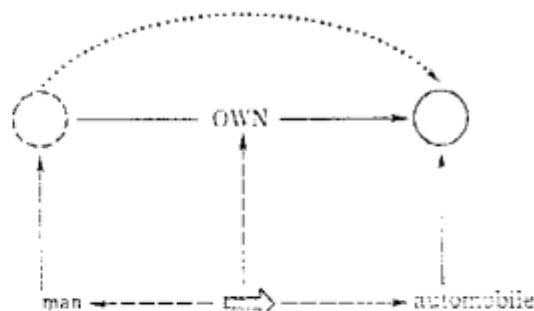
図-2 の (a) と (b) は、“Every man owns a car” という文に対する Schubert の拡張された意味ネットワークと Hendrix による Partitioned Networks による記述例を示している。これらは、意味的には、次の論理式と等価である。

$$(\forall x)(\exists y)(man(x) \rightarrow (car(y) \wedge own(x, y)))$$

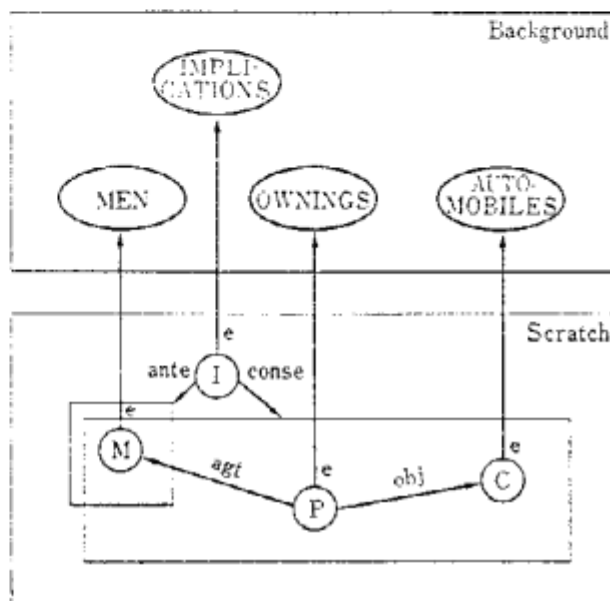
両者の記述とも、論理演算子を意味ネットワークの節点として記述できるようになっている。図-2(a) の Schubert の例では、限量子を節点の性質として記述している。图中、破線で示された節点は全称限量されていると理解し、実線の節点は存在限量されていると理解される。限量子間に依存関係があるときには、図のように節点間にそれを示す枝が張られる。

一方、Partitioned Networks では、節点はすべて存在限量されていると理解される。Partitioned Networks で扱う論理は、多ソート一階述語論理であるので、全称限定は常に含意記号を伴って表現できる。例えば、次の式の左辺の論理式は、右辺のように理解される。

$$(\forall x \in Q, p(x) \Leftrightarrow (\forall x)(Q(x) \rightarrow P(x)))$$



(a) Schubert の拡張された意味ネットワーク



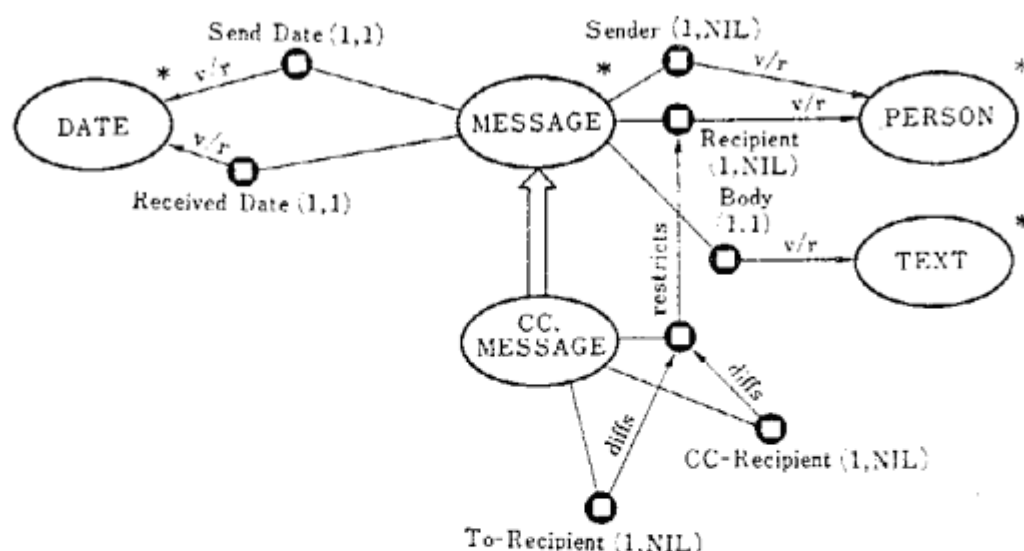
(b) Hendrix の Partitioned Networks

図-2 “Every man owns a car.” の記述

Partitioned Networks は、図-2 (b) のように、長方形によって区切られた意味ネットワークで、各長方形は表現のレベルを表しているが、論理的には、変数のスコープに対応している。含意記号は、図のように、特別の節点によって表現されるが、結論部 (conse に指された長方形) が、条件部 (ante に指された長方形) を含んでいるとき、これらに共通に参照される概念が全称限定されていると解釈される。

彼らの仕事は、意味ネットワークの拡張というよりも、ネットワークによる論理式の表現法というべきものである。ところが、表現そのものの議論に終始してしまっており、ネットワーク上での推論は、グラフの探索やグラフ同士のパターンマッチにより行われることが示唆されているものの、効率に及ばず影響および推論の完全性などについては深い考察が行われていない。

グラフ表現と論理の共存については、Rich による考察²⁷⁾が一つの参考になる。彼の提案は、グラフ的な表現とそれと等価な述語論理表現を同時に持つハイブ



"A CC-MESSAGE is a MESSAGE, at least one of whose Recipients is a To-Recipient and at least one of whose Recipients is a CC-Recipient."

図-3 KL-ONE による記述例

リッドな知識表現法である。システムに要求されるのは、一方の表現に対する追加変更が他方の表現に正しく反映されるような機構を備えていなければならないということである。この要求がどのように実現されるのかは大きな問題であるが、彼の議論は、非常に一般的なものであり、具体的な示唆があるわけではない。両方の表現は本質的に同じものを表しているが、必要な推論の種類によって、両者の特質を使い分けられなければならないというのが彼の提案である。

意味ネットワークと論理表現の共存として、Richの方法とは異なる方法が Krypton^{29,30)}で提案されている。Krypton は、語彙レベルの記述を扱う T Box と呼ばれる部分と、論理的な言明を扱う A Box と呼ばれる二つの部分より構成される。名詞的な概念の定義とそれらの間の関係（同値、包含、排他など）は、T Box によって意味ネットワークとして記述される。A Box は、それらの概念を用いて表現される「文」に相当する言明を論理式によって記述する。T Box の記述には、彼らが設計した KL-ONE^{30,31)}という意味ネットワークが用いられている。

KL-ONE は、Brachman らによって、長年にわたって設計された知識表現言語で、彼らは、意味ネットワークと呼んでいるが、基本的な構成要素は、Concept (フレーム名に相当) と Role (フレームのスロットに相当) であり、フレームと呼んでも差し支えない。このタイプの知識表現法としては、KL-ONE は思想的にも統一され、非常に完成度の高い言語である。例を一つ示すことによって、KL-ONE の概要を説明して

みよう。図-3 は、KL-ONE の記述例である。楕円で囲まれたのが Concept で、太い矢印は、Concept 間の継承関係を表す。細い矢印は、Role や、Role に対する制限などを記述するのに用いられる。Role は、円で囲まれた小さい正方形で表されている。Role は、名称以外に、数についての制限を持つ。たとえば、Sender という Role の下に書かれた (1, NIL) は、その Role に含まれる要素の数が 1 以上いくつでもよいという意味で、Number Restriction と呼ばれている。このように、Role は一般に集合を表すと仮定されている。また、Role から v/r という矢印が出ているが、これは、Value Restriction の略で、その Role に当てはまる概念を指し示す。したがって、MESSAGE の意味は、「一つの通信文と、一人以上の送り手と受け手を持ち、それらは人であって、送信日と受信日は、いずれも唯一の正付を持つ」と定義されていると理解することができる。また、diffs という矢印は、Role Set differentiation と呼ばれ、集合を表している Role の細分化を行うために使用される。この図では、CC-MESSAGE は MESSAGE の一種であるが、受け手が、二種類に細分化されていることを意味している。そのほかにも、KL-ONE は、いろいろな特徴を持つが、一つ注意しておきたいのは、Role は決して default 値としては扱われないし、上位概念から継承される値も決してキャンセルされることがないということである。これは、他のフレームシステムとは大きく異なる点である。しかし、このために、KL-ONE では、概念の包含関係が正確に定義できることが保証されるし、

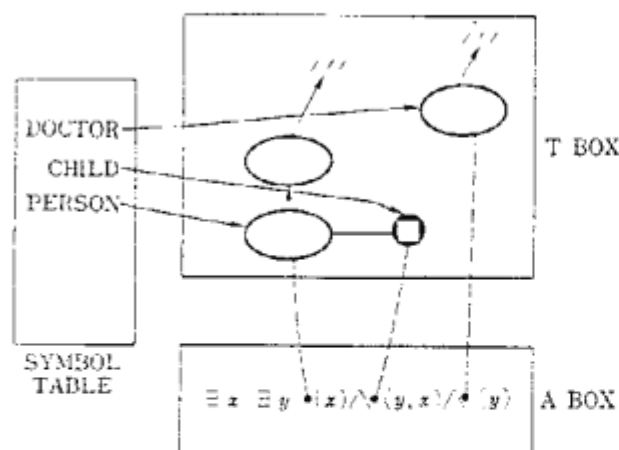


図-4 Krypton の概念図

新しい概念が定義されたときに、それを意味ネットワーク内の正しい位置に置く処理 (classification と呼ばれている) の自動化に役立っている。

A Box は一階述語論理によって記述される論理式よりなる。ただし、述語名、関数名、定数などはすべて T Box に記述された記号が用いられる。図-4 に、Krypton の構成の概念図を示す。Krypton では、T Box は概念間の包含関係や独立性を調べるために使われ、A Box は論理的な言明の等価性や含意性を調べるために使われる。

A Box に対して使用者からの質問が与えられた場合、Krypton は、T Box を参照しなければならないが、Rich¹⁷⁾のように T Box の記述を一階述語論理に翻訳して使うようなことはしない。融合原理の推論過程に必要なのは一つの矛盾する記述の発見であるが、そのために T Box を利用することができる。この部分はまだ完成されたわけではないが、Stickel 流の定理証明法³²⁾を用いることが示唆されている。

4. 知識表現のための論理の拡張

論理によって知識の表現を行うときに、例えば一階述語の論理式の集合をそのまま用いるのでは、心もとないことはすでに述べた。論理による知識表現については、その記述力と計算可能性という二つの大きな問題がある。したがって、一階述語論理以上の記述力を持つ論理を扱いたい反面、むやみに高階の論理などを使うわけにはいかない。記述された論理式についての処理が計算機の上で効率よく動かなければ、実用システムとしては問題だからである。本章では、知識表現のための論理の拡張およびそれに関連する問題について考えてみたい。

4.1 メタレベル記述

前にも述べたように、一階述語の記述力は、実は、非常に強力である。例えば、人が自分の知識や信念などに対して行う推論等を記述しようとする、それはもはや一階述語の枠組みを越えてしまっているように感じられる。しかし、知識や信念に関する記述を一階述語論理の対象言語として、項によって記述してしまい、それらに対する意味の定義や推論をすべて一階述語の範囲で記述することがある程度可能なのである。

McCarthy²¹⁾は、個体概念と言明についての知識の記述を一階述語の対象言語として、一階述語論理の枠組みの中で記述してみせている。これによって、参照的に不透明な文 (referentially opaque sentences)^{*}などの記述と意味付けを行っている。また、Moore^{6), 8)}は、これもやはり一階述語論理の範囲で、知識やそれを用いた行動に関する記述を行っている。彼は、人の持つ知識に関して様相 S4 論理風の可能世界意味論を与え、その記述と可能世界を対象言語として一階述語論理によって記述した。彼は、ゲンツェン流の自然演繹法を証明過程に用いている。これらの議論は、一見、高階もしくは様相概念が必要と思われる記述を、一階述語論理によって行うことを示した点で重要であるが、実現については、問題を残している。

より実用性を意識した論理の拡張として、Bowen 等によるメタプログラミングによる Prolog の拡張^{16), 17)}がある。彼らの議論は、Prolog 上に実現することを前提としており、実現する段階で、対象言語の論理変数とメタ言語の変数を同一レベルで扱っている点など、McCarthy や Moore の議論と同じ視点で論じるわけにはいかない。しかし、Prolog のような現実の言語に、推論の制御の記述やデータベースに対する事実の追加削除に伴う無矛盾性の判定^{**}を埋め込むことの容易性と柔軟性を示した点で評価すべきであろう。

このようにメタレベル記述は、柔軟性と高い記述力を備えた強力な表現法を提供してくれるが、その反面、推論の処理速度に対する危険がある。しかし、いったん記述が完了してしまえば、部分計算によって、同じ効果を持つより高速実行に適した記述に変換することにより実用上の問題をある程度回避できる可能性

* たとえば、'John knows Mike's telephone number.' のような文では、たとえ、Mike の電話番号が Mary の電話番号と等しいからといって、'John knows Mary's telephone number.' とは結論できない。このように、語句が示す内容について「透明」でない文は、referentially opaque sentence または oblique sentence と呼ばれる。

** これについては、文献 33) も参照されたい。

がある。竹内らによる Prolog メタプログラミングの部分計算による高効率化の研究³⁹⁾は、知識および知識に関する処理の記述とその効率的な実現への示唆を与えてくれる。

4.2 default 推論

論理が不完全な事実の記述に優れた表現法であることは既に述べた。しかし、一般に、論理が許している推論は、論理によっていったん記述されてしまった事実は完全であるという仮定のもとに行われる。一度与えられた事実は、未来永劫成り立つと仮定されている。つまり、論理は、宣言文や存在限量文のような不完全な言明を記述し得るが、述べられた事実自体は絶対に正しいことを主張するのである。しかし、人間が持つ知識の中に 100% 信じてよい知識というのはなかなかありそうにない。正しいと結論付けられた事実が実は不確かなものであったり、後で覆されたりすることがあるものである。

このような意味での不完全な知識を扱う試みは、フレームなどでは、default 値やデモンなどによって実現されている。しかし、それではまったく不十分で、このような現象に対する統一かつ形式的な記述を行うには、やはり、論理に頼るしかないのである。

Reiter の default logic³⁶⁾ や McDermott & Doyle による非単調論理³⁷⁾などは、不完全な知識のもとでの推論を論理的に定式化した代表例である。両者とも、ある事実が証明できないということを様相記号で表現して、一階述語論理を拡張している。両者の最も大きな違いは、default logic ではその様相記号を含む式が推論規則の記述にしか用いられず、表面的には一階述語の論理式しか現れないのに対し、非単調論理では、様相記号を論理式自体の構文要素として認めてしまっている点である。McDermott³⁷⁾は、その後、非単調論理に対して様相論理的な定式化を行おうとしたが、うまくいかなかった^{*}。その原因の追究および意味論に基づく非単調論理の改良が Moore³⁸⁾ によって行われている。このような一連のアプローチは、不完全な知識に対する論理的な形式化としては、非常に面白い反面、実現についての問題は、まったく解決されていない。実際、彼らの枠組みを計算機上にそのまま修正なしに実現するのは不可能である。ただし、Doyle の Truth Maintenance System³⁹⁾は、その実現について

の一つの方法を示している。

同様の問題への、これらとは異なるアプローチとして、McCarthy の Circumscription がある⁴⁰⁾。これは、特定の述語に対して最小の解釈を与えようとするもので、Reiter の Closed World Assumption (CWA)⁴¹⁾の一つの定式化になっている。より具体的には、McCarthy は、CWA を意味論的に捉え、ある述語の Circumscription (Predicate Circumscription) を、その述語に対して最小の解釈を与えるすべてのモデルで成り立つ式の集合と考えた。McCarthy はその後、最小解釈を与える対象を述語だけではなく任意の論理式にまで広げている⁴²⁾が、いずれの場合も定式化が二階述語論理の論理式で行われており、計算機による実現には問題がある。

それでは、不完全な知識に対するこのようなアプローチは、非現実的なのであろうか。これについては、次のような見方が考えられると思う。

Clark の Negation as Failure (NAF)⁴³⁾も CWA の一つの実現であるが、今まで説明したアプローチとは性質が異なる。彼は、否定が働に表現できない Prolog (Horn 節集合) に対して、否定の明確な意味を与えるために、Completion という概念を考えた。Horn 節は、一つのアトムだけを結論部として持つ条件式であり、結論部の述語についての一つの十分条件を与えるが、特定の述語についてのこのような十分条件をすべてまとめてその述語の必要条件でもあるとみなすのが Completion の考え方である。Clark は、このような必要条件を表す論理式の求め方を示し、それをその述語の Completion axiom と呼んでいる。Completion axiom によって定義し直された述語の論理的否定は、元の Horn 節の集合を完全な定理証明系で実行した場合に有限時間で失敗する要素の集合 (finite failure set) と正確に一致することがわかっている⁴⁴⁾。これは非常に重要な結果である。つまり、実際には対象とする Horn 節集合に何の変更も行わずに、その最小解釈と正しい否定の実現が可能であることを示しているのである。

節表現された論理式の集合の中で、ある述語が Horn である⁴⁵⁾限り、その述語の Circumscription は、その述語に関する Completion axiom を含意することがわかっている⁴⁶⁾。つまり、この条件のもとで、

* P の否定、 $\neg P$ が証明できないことを示す ΔP という表現に用いている Δ というオペレータを様相記号として非単調論理を拡張したときに一番もっともらしい様相 BS 論理が、実は一階述語論理に崩壊 (collapse) することがわかったのである。

^{*} データベースなどの世界において、与えられた事実のみが真であり、それ以外はすべて偽であるとする立場のこと。

^{**} 節表現された論理式の集合の中で、述語 P が Horn であるというのには、どの節にも P の正の出現が両方一つしかないことである。

Completion は Circumscription に含まれる概念ということになる。両者の差がどのようなものであるかは、よくわかっていないが、Completion は、Circumscription の重要な近似であると思われる。

Lloyd による Prolog の拡張⁴¹⁾は、NAF の健全な (sound) 性質を保ったまま Prolog で記述できる論理式の範囲を拡張している。彼は、Prolog の節の本体部に任意の一階述語論理式を許し、そのような節を、否定を含む一般の Prolog の節に変換する方法を示している。NAF の健全性を保っているというのは、ある論理式の証明が有限の時間内で失敗するならば、その式の否定が、Completion を施した後の論理式の集合の論理的帰結になっているということである。ただし、その反対の完全性は保証されていない。変換後のプログラムが、Prolog によって実行可能になるために、彼は、節の本体部の表現に、多ソート一階述語論理を仮定しているが、知識表現への応用のためには、これは特に影響ない制限であろう。

4.3 表現法と計算量

本節の主旨は、論理の拡張に関する問題ではなく、文章や状況の理解および推論を行うに際して、知識や問題の表現法が、記述量や計算量に大きく影響することがあるということを指摘することにある。その一例として、枠の問題 (frame problem)⁴²⁾とその解決法について考えてみよう。枠の問題とは、ある動作などによって、状況が変化したときに、実際に変化したものと不変なものとをどのように記述し分けるかという問題である。変化したものを記述するのはやむを得ないとして、変化しないものまでいちいち記述していたのでは膨大な処理が必要になってしまう。

McCarthy & Hayes は、situational calculus⁴³⁾において、動作によって変化する状況を、状況を表すパラメータを設けることによって記述し、ある動作によって不変なものを枠の公理 (frame axiom) として記述した。例えば、 x, y は箱、 z は位置、 s は状況、 c は色を表す変数として、

$$\forall x, y, z, s, c. (\text{Location}(x, y, s) \rightarrow \text{Location}(x, y, \text{paint}(z, c, s)))$$

は、ある箱に色を塗ったからといって、どの箱の位置も変化しないことを表す枠の公理である。

また、STRIPS⁴⁴⁾も問題の記述には、一階述語論理を用いているが、状況を論理式の集合と捉え、論理式の追加削除によって状態の変化を記述している。こうすると、確かに枠の問題は起こらないが、過去の状況

がすべて失われてしまうので、時間に伴う状況の変化などについての推論は行えない。

Kowalski らの event calculus⁴⁵⁾は、時間に伴う状況の変化を一階述語論理 (特に Prolog での実行を考えて Horn 節論理) で表現しているが、NAF を用いることにより、枠の問題を避け、しかも、新しい事実の追加のみによって、以前には結論付けされてしまった事実を直す非単調な推論を実現している。

ここで述べたかったのは、同様の問題を論理によって扱う場合でも、問題の表現法によって、記述能力や問題解決に必要な計算量が大きく変わってしまうことがあるということである。

5. あとがき

知識表現の問題について、非常に狭い範囲であるが、論理的なアプローチを中心に紹介した。結論的なことは言えないが、少なくとも知識表現における論理の重要性和有用性は検証できたのではないだろうか。

また、論理によるアプローチが、決して処理速度の速さには繋がらないということも、ここで述べておきたかった点である。メタレベル記述による階層的な記述を部分計算などによって、効率的に実現するための基礎的な研究や、非単調論理のような人間の常識推論のための論理を Negation as Failure などを用いて、効率上の悪影響を与えずに実現する方法などが、論理に基づく知識の表現と利用に対する新しい展望を開いてくれるのではないだろうか。

参考文献

- 1) 文藝工学特集, 情報処理, Vol. 20, No. 12 (Dec. 1985).
- 2) Barr, A. and Feigenbaum, E. A. (eds.): Handbook of Artificial Intelligence, Pitman (1981). 田中幸吉, 編 一徳器訳: 人工知能ハンドブック, 共立出版.
- 3) Schank, R.C.: Conceptual Dependency: A Theory of Natural Language Understanding, Cognitive Psychology, Vol. 3, pp. 552-631 (1972).
- 4) Schank, R.C. and Abelson, R.P.: Scripts, Plans, Goals, and Understanding, Lawrence Erlbaum (1977).
- 5) Lytinen, S.L. and Schank, R.C.: Representation and Translation, Research Report No. 234, Department of Computer Science, Yale University (May 1982).
- 6) Moore, R.C.: Reasoning about Knowledge and Action, Proc. 5th IJCAI, pp. 223-227 (1977).

- 7) McCarthy, J.: First Order Theories of Individual Concepts and Propositions, in *Machine Intelligence 9*, pp. 129-147, Ellis Horwood Ltd. (1979).
- 8) Moore, R.C.: A Formal Theory of Knowledge and Action, in Hobbs, J.R. and Moore, R.C. (eds.), *Formal Theories of the Commonsense World*, Ablex Publishing Corporation, pp. 319-358 (1985).
- 9) Hughes, G.E. and Cresswell, M.J.: *An Introduction to Modal Logic*, Methuen and Company Ltd. (1968). 三浦 聰他訳, 様相論理入門, 恒星社厚生閣 (1981).
- 10) Montague, R.: The Proper Treatment of Quantification in Ordinary English, in Thompson, R.H. (ed.), *Formal Philosophy, Selected Papers of Richard Montague*, Yale University Press (1974).
- 11) Dowty, D.R., Wall, R. and Peters, S.: *Introduction to Montague Semantics*, Reidel (1981).
- 12) Woods, W.A.: What's in a Link? Foundation for Semantic Networks, in Bobrow, D.G. and Collins, A. (eds.), *Representation and Understanding: Studies in Cognitive Science*, Academic Press, pp. 35-82 (1975). 堀一博監訳: 人工知能の基礎, 近代科学社 (1978).
- 13) Brachman, R.J.: What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks, *IEEE Comput.*, Vol. 16, No. 10, pp. 30-36 (Oct. 1983).
- 14) Simmons, R.F. and Bruce, B.C.: Some Relations between Predicate Calculus and Semantic Net Representations of Discourse, *Proc. of 2nd IJCAI*, pp. 524-530 (1971).
- 15) Hayes, P.J.: The Logic of Frames, in Metzger (ed.), *Frame Conceptions and Text Understanding*, de Gruyter, pp. 46-61 (1979).
- 16) Bowen, K.A. and Kowalski, R.A.: Amalgamating Language and Metalinguage in Logic Programming, in Clark, K.L. and Tarnlund, S.A. (eds.), *Logic Programming*, Academic Press, pp. 153-172 (1982).
- 17) Bowen, K.A.: Meta-Level Programming and Knowledge Representation, *New Generation Computing*, Vol. 3, No. 4, pp. 359-383 (1985).
- 18) Hayes, P.J.: In Defence of Logic, *Proc. 5th IJCAI*, pp. 559-565 (Aug. 1977).
- 19) Moore, R.C.: The Role of Logic in Knowledge Representation and Commonsense Reasoning, *Proc. AAAI-82*, pp. 428-433 (1982).
- 20) Israel, D.J.: The Role of Logic in Knowledge Representation, *IEEE Comput.*, Vol. 16, No. 10, pp. 37-42 (Oct. 1983).
- 21) Deliyanni, A. and Kowalski, R.A.: Logic and Semantic Networks, *C. ACM*, Vol. 22, No. 3, pp. 184-192 (March 1979).
- 22) Kowalski, R.A.: A Proof Procedure Using Connection Graph, *J. ACM*, Vol. 22, No. 4, pp. 573-595 (Oct. 1975).
- 23) Sickel, S.: A Search Technique for Clause Interconnectivity Graphs, *IEEE Trans. Comput.*, Vol. C-25, No. 8, pp. 823-835 (Aug. 1976).
- 24) McSkimin, J.R. and Minker, J.: A Predicate Calculus Based Semantic Network for Deductive Search, in Findler, N.V. (ed.), *Associative Networks*, Academic Press, pp. 205-238 (1979).
- 25) Schubert, L.K.: Extending the Expressive Power of Semantic Networks, *Artif. Intell.*, Vol. 7, pp. 163-198 (1976).
- 26) Hendrix, G.G.: Encoding Knowledge in Partitioned Networks, in Findler, N.V. (ed.), *Associative Networks*, Academic Press, pp. 51-92 (1979).
- 27) Rich, C.: Knowledge Representation Languages and Predicate Calculus: How to Have Your Cake and Eat It Too, *Proc. AAAI-82*, pp. 193-196 (1982).
- 28) Brachman, R.J. and Levesque, H.J.: Competence in Knowledge Representation, *Proc. AAAI-82*, pp. 189-192 (1982).
- 29) Brachman, R.J., Fikes, R.E. and Levesque, H.J.: Krypton: A Functional Approach to Knowledge Representation, *IEEE Comput.*, Vol. 16, No. 10, pp. 67-73 (Oct. 1983).
- 30) Schmolze, J.G. and Brachman, R.L.: Summary of the KL-ONE Language, *Proc. the 1981 KLONE Workshop*, Fairchild Technical Report, No. 618 (May 1982).
- 31) Brachman, R.J. and Schmolze, J.G.: An Overview of the KL-ONE Knowledge Representation System, *Cognitive Science*, Vol. 9, No. 2, pp. 171-216 (1985).
- 32) Stickel, M.E.: A Nonclausal Connection-Graph Resolution Theorem-Proving Program, *Proc. AAAI-82*, pp. 229-233 (1982).
- 33) Miyachi, T. et al.: A Knowledge Assimilation Method for Logic Databases, *Proc. 1984 International Symposium on Logic Programming*, pp. 118-125 (Feb. 1984).
- 34) Takeuchi, A. and Furukawa, K.: Partial Evaluation of Prolog Programs and its Application to Meta Programming, *Proc. IFIP-86*, (1986).
- 35) Reiter, R.: A Logic for Default Reasoning, *Artif. Intell.*, Vol. 13, No. 1, 2, pp. 81-132 (Apr. 1980).
- 36) McDermott, D. and Doyle, J.: Non-Monotonic Logic I, *Artif. Intell.*, Vol. 13, No. 1, 2, pp. 41-

- 72 (Apr. 1980).
- 37) McDermott, D.: Non-Monotonic Logic II, J. ACM, Vol. 29, No. 1, pp. 33-57 (1982).
- 38) Moore, R. C.: Semantical Considerations on Nonmonotonic Logic, Artif. Intell., Vol. 25, pp. 75-94 (1985).
- 39) Doyle, J.: Truth Maintenance Systems for Problem Solving, MIT AI-TR-419 (1978).
- 40) McCarthy, J.: Circumscription—A Form of Non-Monotonic Reasoning, Artif. Intell., Vol. 13, No. 1, 2, pp. 27-39 (1980).
- 41) McCarthy, J.: Applications of Circumscription to Formalizing Common-Sense Knowledge, Artif. Intell., Vol. 28, pp. 89-116 (1986).
- 42) Reiter, R.: On Closed World Data Bases, in Gallaire, H. and Minker, J. (eds.), Logic and Data Bases, Plenum Press, pp. 55-76 (1978).
- 43) Clark, K.: Negation as Failure, in Gallaire, H. and Minker, J. (eds.), Logic and Data Bases, Plenum Press, pp. 293-322 (1978).
- 44) Jaffar, J., Lassez, J.-L. and Lloyd, J.: Completeness of the Negation as Failure Rule, Proc. IJCAI-83, pp. 500-506 (1983).
- 45) Reiter, R.: Circumscription Implies Predicate Completion (Sometimes), Proc. AAAI-82, pp. 418-420 (1982).
- 46) Lloyd, J. W. and Topor, R. W.: Making Prolog More Expressive, J. Logic Programming, Vol. 1, No. 3, pp. 225-240 (1984).
- 47) McCarthy J. and Hayes, P. J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence, in Meltzer, B. and Michie, D. (eds.), Machine Intelligence 4, Edinburgh University Press, pp. 463-502 (1969).
- 48) Fikes, R. E. and Nilsson, N. J.: STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving, Artif. Intell., Vol. 2, pp. 189-208 (1971).
- 49) Kowalski, R. and Sergot, M.: A Logic-based Calculus of Events, New Generation Computing, Vol. 4, No. 1, pp. 67-95 (1986).

(昭和61年6月18日受付)