

フレーム問題に関する考察

松原 仁 山本 和彦

電子技術総合研究所パターン情報部

要約

本稿では、いわゆるフレーム問題について、人工知能の立場からいくつかの考察を行なう。われわれの主張は以下に示す通りである。1) フレーム問題は、一般的には決して解決できない。2) したがって、解決を目指すことに意味がない。3) フレーム問題は、知識表現の枠組だけではなく、表現されている知識の内容にも関係する。4) 人間もフレーム問題を決して解決してはいない。5) 人間があたかもフレーム問題を解決しているかのように見えるのはなぜかを説明することこそが、未解決の興味深い問題である。

S o m e C o n s i d e r a t i o n s o n t h e F r a m e P r o b l e m

Hitoshi MATSUBARA , Kazuhiko YAMAMOTO

Information Sciences Division, Electrotechnical Laboratory

1-1-4 Umezono, Sakura-mura, Niihari-gun, Ibaraki, 305 JAPAN

ABSTRACT

This paper develops some considerations on the frame problem from the standpoint of artificial intelligence. The following are our assertions.

- 1) The frame problem can never be solved generally by computers.
- 2) So, the frame problem is not to be solved.
- 3) The frame problem has relations not only to the framework of knowledge representation, but also to the content of represented knowledge.
- 4) Human beings can never solve the frame problem.
- 5) A very interesting open question is why human beings seem to solve the frame problem.

1. はじめに

状態の記述方法および状態に作用する作用素の記述方法があらかじめ定義されているものと仮定する。ある状態Aにある作用素を適用して他の状態Bに遷移した際に、AとBの間で何が変わり何が変わらないかをどのように記述するかという問題のことを、一般にフレーム問題 [1, 2] と呼ぶ。作用素の作用が及ぶ状態の範囲をどのように限定するかの問題であると言い換えることもできる。知識の表現と利用にまつわる重要な概念として人工知能における議論の対象にしばしば取り上げられている。

本稿では、このフレーム問題について、いくつかの考察を行なう。われわれの主張は以下の通りである。

- 1) フレーム問題は、一般的には決して解決できない。
- 2) したがって、解決を目指すことに意味がない。
- 3) フレーム問題は、知識表現の枠組だけではなく、表現されている知識の内容にも関係する。
- 4) 人間もフレーム問題を決して解決してはいない。
- 5) 人間があたかもフレーム問題を解決しているかのように見えるのはなぜかを解明することこそが、未解決の興味深い問題である。

以下ではまずフレーム問題に関するこれまでの議論を簡単に振り返ってから、上記のわれわれの主張を展開することにする。

2. フレーム問題の発端

ここでは行動計画の自動生成の領域でフレーム問題が取り上げられた発端について説明する。人工知能には初期の頃から定理の自動証明という領域があり、行動計画の自動生成の領域はその応用として研究が始められた。行動を計画することを公理から定理を証明する過程としてとらえたのである。定理の自動証明は数学を対象としているためにそのための知識表現の枠組として形式論理が使われることが多かった。そのような経緯でフレーム問題は、最初に形式論理という枠組について議論されたのである。

積み木Aが積み木Bの上に乗っているという命題を、 $(on\ A\ B)$ と表現することにする。古典的な形式論理には状態の変化の概念がない。 $(on\ A\ B)$ は公理ないしは定理として扱われるので、ひとたび成立すると永久に成立していることになる。行動計画の自動生成は現在の状態から目標の状態に変化させるための手順を求めることが目的なので、状態が変化しなくては意味をなさない。そこで、状態の概念を陽に引数に持ち込んで、状態SでAがBの上に乗っていることを $(on\ A\ B\ S)$ として表現することがまず考えられた。しかしこの方法では、各状態・各作用素に対して、ある状態にある作用素を適用するとどうなるのかを、変化するしないにかかわらず全ての場合について記述することになる。状態数 m ・作用素数 n に対し $m \cdot n$ 個の規則を記述しなければならない。

一般にある作用素の適用によって変化するのは状態の記述の中のごく一部である。

たとえば電灯の付いている部屋の中でコップを持ち上げても電灯は一般に付いたままである。上記の記述方法を用いると、「コップを持ち上げても電灯は付いたままである」、「コップを置いても電灯は付いたままである」、「コップから水を飲んでも電灯は付いたままである」、などという規則をいちいち記述しておかねばならない。もしも状態の記述の量を減らしたければ、命題はそれが初めて成立する状態でのみ記述することにして、その状態以降はその命題が陽に否定されるまではずっと成立しているとみなす、という方法もある。この方法を用いれば確かに状態の記述の量は減らすことはできるが、ある状態である命題が成立しているかどうかを知るために、その命題に関する最も近い記述を求めて以前の状態に溯って推論する必要が生じる。

フレーム問題は、このように記述の量と共に推論の量にも関連する概念である。別の言い方をすれば、フレーム問題は、知識を表現してそれを利用する際に必要な空間の量と時間の量に関係がある。また、フレーム問題は単にコンピュータ上の効率だけではなく、人間がどのように対処しているかという点で哲学にも心理学にも関連する [3, 4]、認知科学に共通の問題なのである。

3. フレーム問題の解決の試み

フレーム問題の解決策としてまず考えられたのが、形式論理に対する副作用の導入である。STRIPS [5] という行動計画の自動生成システムで採用された。ひとつひとつの作用素に対し、その作用素を適用するための条件 (precondition)、その作用素の適用によって状態に新たに加わる命題リスト (add-list)、逆に状態から取り除かれる命題リスト (delete-list)、を定義しておく。作用素を適用する度にこの定義にしたがって公理ないしは定理を実際には書き換えることにすれば、過去に溯った推論を行なう必要はない。したがって推論の量の点では一応の改善になっている。

しかし STRIPS においては、作用素をその作用対象と切り離して表現しているので、作用の副作用が作用対象によって異なる場合には機能しないという限界がある。たとえばカップが受け皿の上に乗っている状態を考えてみよう [6]。下の受け皿を移動すると上のカップも付いて移動するが、上のカップを移動しても下の受け皿は付いて移動しない。この差は「移動」という作用素だけでは表現できない。この限界を解消すること自体は簡単である。個々の物体に対しその物体に適用できる動作をもれなく記述しておくという方法がある。また、動作記述子の引数に物体名も陽に書いて、個々の物体ごとに異なる副作用を記述するという方法もある。どちらの方法でもカップと受け皿の例を記述することができるようになる。ただども基本的には物体ごとの記述を必要とするので、元のフレーム問題の発端に話に戻ってしまうのである。

高階の述語を導入することによってカップと受け皿の例を記述できるようになる。Sandewall は UNLESS という高階の述語を導入している [2]。" IS(object, property, situation) " は object という物体が situation という状態で property という性質を持つことを表わす述語であるとする。同様に、" ENDS(object, property, situation) " は object という物体が situation という状態で property という性質をもはや持たないことを表わす述語であるとする。さらに、" succ(situation, action) " は situation

という状態でactionという動作を行なった後の状態を値とする関数であるとする。

IS(object, property, situation)
UNLESS(ENDS(object, property, succ(situation, action)))
IS(object, property, succ(situation, action))

という推論規則は、"IS(o, p, s)"が定理で、"ENDS(o, p, succ(s, a))"が定理でない(証明できない)ならば、"IS(o, p, succ(s, a))"は定理と見なす、という意味を持っている。objectの種類によってENDSの記述を変更することにすれば、カップと受け皿の例は記述可能である。

UNLESSオペレータを用いれば、すべての場合をもれなく記述する必要はなく、例外に相当する場合だけを記述すればよい。しかし、ある状態である定理が成り立っているかどうかを調べるには、過去の状態に溯って推論を行なう必要がある。推論の量の点でフレーム問題が解決できていないことになる。記述の量の問題を推論の量の問題に転嫁しているに過ぎない。

McDermottらの非単調論理(non-monotonic logic) [7] は、高階の様相述語(Mで表現する)を用いることによって定理の数の「非単調」な変化に対応を図っている。非単調という言葉の意味をまず説明しよう。古典的な形式論理では公理ないしは定理が新たに加わることによって定理の数は単調に増加する。直感的に言えば、知識は決して減少することがない。その意味において単調論理である。それに対し、高階の様相述語を導入したとすると、公理ないしは定理が新たに加わることによって必ずしも定理の数が増加するとは限らない。減少することもある。ゆえに非単調論理と呼ぶのである。SandewallのUNLESSオペレータの一般化と見ることもできよう。

"M(電灯が付いている)"は、その状態における他の記述と「電灯は付いている」の命題が矛盾しない限り、すなわち「電灯は付いている」の命題が陽に否定されていない限り、「電灯は付いている」と見なしてよい、という様相命題である。この記述が存在すると、もしも「電灯は付いている」ことが偽であると判明していなければ、「電灯は付いている」が真と仮定して推論を進める。後で「電灯は付いている」ことが偽と判明したならば、この命題を真と仮定したことによって得られた定理はすべて消去することになる(これが非単調の所以である)。人間の常識推論を表現するために考え出された枠組である。

非単調論理におけるフレーム問題の取り扱い、UNLESSオペレータにおける取り扱いと非常によく似ている。例外だけを記述すればよいので記述の量の点については改善されているが、ある命題がある状態で真か偽かを決定するためには過去に溯って推論を行なう必要があるため、推論の量の点では解決になっていない。非単調論理と同様に形式論理の範囲の中で常識推論の実現を目指す枠組として、McCarthyのcircum-scription [8] やReiterのdefault logic [9] などが知られているが、どれもがフレーム問題に関しては(非単調論理と同じ意味で)解決されていない。表面上フレーム問題を取り繕ったように見えてはいても、問題のある場所から他の場所へ移動しているだけに過ぎないのである [10]。

ここまでは形式論理の枠内におけるフレーム問題の取り扱いについて述べてきた。形式論理の枠外でのフレーム問題の取り扱いにも触れておくことにする。

Minskyの提案したフレーム理論 [11] (この概念はフレーム問題とは無関係であることに注意されたい) は、階層性・性質の継承・デフォルトなどの特徴を有した人間の常識推論のモデルである。「鳥」のフレームにはデフォルトとして「飛べる」と記述してある。その下位フレームである「つばめ」には飛べる飛べないに関する記述はない。この場合は上位フレームの性質が継承されるので「つばめ」は「飛べる」ことがわかる。同じく「鳥」の下位フレームである「ペンギン」には陽に「飛べない」と記述してある。フレーム間の制御の移動が推論に対応している。

フレーム理論も、形式論理における高階述語の導入のように、フレーム問題の解決にはなっていない。例外だけを記述すればよいという意味で記述の量の点では改善されているが、(例外でない)知識を取り出す際には上位のフレームを順々にたどっていく必要がある。推論の量の点では解決になっていない。このことは特に階層が深い場合や多重の性質継承が存在する場合などに表面化しやすい。

手続き的に知識を表現する [12] と、宣言的に表現する場合とは異なり、知識や推論がプログラムの中に完全に埋め込まれるので、フレーム問題があたかも出現しないかのように見える。しかし、これは表面化していないだけであって、決して解決されていないのである。フレーム問題が全体に分散した状態なので局所的に取り出すことが難しいに過ぎない。

中島のProlog/KR [13] やわれわれのRAPS [14] においては、フレーム問題の「解決」という言葉を、「システムが知識の書き換えを(バックトラックなどを含めて)自動的に行なうので、ユーザはフレーム問題を考慮する必要がない。」という意味で使用しているので、ここでの議論からは除外する。

4. 考察

(1)

第3章でフレーム問題に対する解決の試みを振り返ってみたが、どれも本質的な解決にはなっていなかった。フレーム問題を解決するには記述の量と推論の量の両方を解決しなければならないが、解決策はみなどちらか一方を犠牲にしてもう一方の解決を図ったに過ぎなかった(特に推論の量を犠牲にして記述の量を減らそうとした試みが多かった)。フレーム問題は、そもそもの問題の設定条件において、一般的には決して解決できないと思う。すなわち、状態の記述方法と状態に作用する作用素の記述方法が問題の設定条件になっている限り、コンピュータは決してフレーム問題を解決できない。より宣言的に知識を表現しようとするほどフレーム問題が表面化しやすいので、形式論理においてフレーム問題が最も顕著なのは当然である。しかし、形式論理を知識表現の枠組と見なした場合の短所にフレーム問題を取り上げるのは大きな誤りである。セマンティック・ネットワーク、プロダクション・システム、フレーム理論などのいかなる枠組を持ってきたとしても、フレーム問題は決して解決されない。たとえ手続き的に知識を表現したとしても解決にはならない。解決の可能性の有無を

枠組の性質に帰すべき問題ではないのである。

フレーム問題を例外の取り扱いに即して説明するならば、記述の量も推論の量もふやすことなく例外を取り扱うにはどうすればよいか、という問題に帰着される。例外を取り扱うためには、あらかじめ例外だけをもれなく記述しておくか、知識が必要になった時点で例外かどうかを判定するための推論を行なうか、記述と推論とを適当に組み合わせて併用するか、いずれにしる結局はどれが例外であってどれが例外でないかの情報を保持しておかなければならない。したがって、記述の量も推論の量もふやさずに例外を取り扱うことは原理的に不可能なはずである。

(2)

フレーム問題は一般的には解決できないのであるから、当然ながら一般的な解決を目指すことには意味がない。新しい知識表現の枠組の設計は、フレーム問題の解決を目指してではなく、常識推論の実現を目指して行なわれるべきである。しかし、特定の知識表現の枠組における記述の量と推論の量との間のトレードオフについて考慮することは非常に重要である。記述の量を犠牲にして推論の量を減らそうとするのか、逆に推論の量を犠牲にして記述の量を減らそうとするのか、の決定は知識表現の本質的な分岐点になるからである。

記述の量は空間的な概念なのに対し推論の量は時間的な概念なので、両者をそのまま比較することは不可能である。もしもなんらかの共通の尺度を準備して両者の「加算」を定義することができれば、両者間のトレードオフを定量的に評価できるようになるはずである。個々の知識表現を対象として記述の量と推論の量の「和」の最小化を目指すことには十分な意味がある。この問題のことをフレーム問題と呼ぶのはもはや適当ではないと考える。

(3)

従来フレーム問題に関する議論では、もっぱら知識表現の枠組だけが考慮の対象となっていた。しかし、実際にどのような知識が表現されていて、その知識がどのように利用されるのかも、フレーム問題と決して無関係ではあり得ないのである。前述したようにフレーム問題は知識の表現と利用における空間的・時間的な効率に関係している。具体的な知識の内容と無関係に効率が決定されるはずはない。よく使う知識とあまり使わない知識とを同等に扱うのは非効率的である。

ある例題をある枠組で効率よく表現できたとしても、他の例題もその枠組で効率よく表現できるとは必ずしも限らない。もしある例題をある枠組で表現して記述の量の問題を解決できたとするならば、それはその例題について解決したと考えるべきであって、その枠組について解決したと考えてはならないのである。記述の単位および推論の単位にどれだけの情報が含まれているかによって、必要となる記述の量と推論の量が増減する。フレーム問題を論じる際にその影響を無視してはならない。

(4)

視点を変えて人間はどのようにフレーム問題を取り扱っているかについて考えてみ

よう。「人間はフレーム問題を解決できるのにコンピュータはできない。この違いがコンピュータは人間並みの知能を持ってない理由なのである。」という主張も存在する。はたして、人間は本当にフレーム問題を解決しているのであろうか。われわれは人間も決してフレーム問題を解決してはいないと考える。状態の記述方法と状態に作用する作用素の記述方法を問題にしている限り、コンピュータに解決できないのと同様に人間にも解決できない。記述の量と推論の量の両方を減らすことは人間にとっても不可能なはずである。

人間もフレーム問題に直面していると感じることもある。未知の対象の問題解決を行なう際に、既知の対象との類推がうまくとれないとフレーム問題が生じる。類推がとれない未知のゲームをマスターする場合を考える。この場合にはそのゲームにおける状態の記述方法と状態に対する作用素の記述方法を最初から記憶しなければならない。ゲームを完全にマスターするまでは、ある状態である命題が成立しているかどうかを判定するためかなりの記述の量もしくは推論の量を必要とすることが実感される。たとえば詰め将棋を解く際に実際に駒を盤の上で動かす人（ちなみに本来は頭の中だけで解かなくてはいけない）は、詰め将棋におけるフレーム問題に直面して解決ができなくなっているのである。

(5)

人間においては多くの場合フレーム問題が「問題」にすらならない。人間もフレーム問題を解決していないとすれば、人間があたかもフレーム問題を解決しているかのように見えるのはなぜであろうか。この問いに対する答えはまだ出ていない。この謎の解明を目指すことこそが人工知能の興味深い問題であると思う。

詰め将棋を完全にマスターすると、頭の中だけで局面を自由に操作することが可能になる。数十手先の局面も直感的にほぼ誤りなく思い浮かぶようになる。記述の量も推論の量も問題にならなくなるのである。フレーム問題は解決できないとすれば何が起こったのだろうか。フレーム問題に直面しなくなったと考えざるを得ない。初心者はフレーム問題に直面して解決ができないのに対し、熟練者はフレーム問題に直面しない。熟練者は、もはや状態と状態に対する作用素という形式化をしていないのであろう。人間がフレーム問題に直面しなくなる過程において学習が大きな役割を担っていることは確かである。

5. おわりに

本稿ではフレーム問題に対するわれわれの基本的な考え方について述べた。フレーム問題は、人間にもコンピュータにも一般的には決して解決できない、というのがその中心になっている。われわれは決してフレーム問題を考えること自体に意味がないと主張しているのではない。ある知識をある枠組で表現しようと試みる際に、フレーム問題がどのような場所にどのような形で出現するのかを明らかにしておくのは重要である。フレーム問題をどのように取り扱っているのかは、その知識表現の最も特徴的な指標として機能するからである。

謝辞

東京大学工学部機械工学科井上 博允教授および井上・稲葉研究室の皆様，電子技術総合研究所中島 隆之パターン情報部長および画像処理研究室の皆様，AIUEOならびに寺小屋のメンバーの皆様に感謝します。

参考文献

- [1] McCarthy, J. and Hayes, P. J. : "Some philosophical problems from the standpoint of artificial intelligence", Machine Intelligence 4, pp.463-502 (1969)
- [2] Sandewall, E. : "An approach to the frame problem and its implementation" Machine Intelligence 7, pp.195-204 (1972)
- [3] Fodor, J. A. : "The modularity of mind", MIT Press (1983)
- [4] 土屋 俊 : "心の科学は可能か", 東京大学出版会(1986)
- [5] Fikes, R. E. and Nilsson, N. J. : "STRIPS: a new approach to the application of theorem proving to problem solving", Artificial Intelligence 2, pp.189-208 (1971)
- [6] Hayes, P. J. : "A logic of actions", Machine Intelligence 6, pp.495-520, (1971)
- [7] McDermott, D. and Doyle, J. : "Non-monotonic logic I", Artificial Intelligence 13, pp.41-72 (1980)
- [8] McCarthy, J. : "Circumscription- a form of non-monotonic reasoning", Artificial Intelligence 13, pp.27-39(1980)
- [9] Reiter, R. : "A logic for default reasoning", Artificial Intelligence 13, pp.81-132(1980)
- [10] Hanks, S. and McDermott, D. : "Default reasoning, nonmonotonic logic, and the frame problem", AAAI-86, pp.328-333(1986)
- [11] Minsky, M. : "A framework for representing knowledge", in Winston, P. (ed.) "The psychology of computer vision", McGraw-Hill (1975)
- [12] Winograd, T. : "Frame representations and the declarative/procedural controversy", in Bobrow, D. G. and Collins, A. (eds.) "Representation and understanding", Academic Press (1975)
- [13] 中島 秀之 : "知識表現とProlog/KR", 産業図書(1985)
- [14] 松原 仁, 井上 博允 : "プロダクション・システムによるロボットの行動計画の自動生成", 情報処理学会知識工学と人工知能研究会, 41-8 (1985)