

仮説的な推論—TMS, ATMSとその応用

寺野 隆雄, 篠原 靖志

1. はじめに

“明日の休みは、天気だったらハイキングで、雨だったら映画にしよう。そして、……”

たとえば、デートの前日には、このようにさまざまな状況を想定して予定を立てることがある。こんな処理をコンピュータに行なわせるには、Aの状況のときにはBを、A'の状況のときはB'を、…というように、A、A'と仮説を立てながら推論を進めるのが普通である。このような仮説的な推論の手法は、人間の問題解決の方法に近く、複雑な事象について限られた時間内に判断を下すのに有用である。

しかし、こうして得られた結論はいつも正しいとは限らない。結論が推論中に採用した仮説に依存するので、仮説が間違っていた場合の結論の正しさは保証されないのである。そのため、コンピュータ上で仮説的な推論を行なうには、仮定した事象が事実とならない場合、どの仮定が誤りであったのかを判定し、正しい推論の筋道を効率よく作り直すことが必要となる。それに対し、仮説を伴わない通常の推論では、処理中で一度判明した事象は事実として記録され、決して変更されないことに注目されたい。

人工知能技術にもとづくシステム、特にエキスパートシステムでは、仮説的な推論を行なう必要

がしばしば生ずる。これを扱う枠組みとして、TMS (Truth Maintenance System)—真実の管理システム—(何というおおげさな名前だろう！)あるいは、ATMS (Assumption-Based Truth Maintenance System)—仮説にもとづく真実の管理システム—の手法がある。

本稿では、この2つの手法を中心に仮説的な推論について、状態空間の探索手法[1]という立場から解説する。まず、推論処理における状態空間の概念について[2]にしたがって述べ、次にそれをを用いてTMS[3],[4],[5]、ATMS[6],[7],[8]の基本的な原理について解説する。そして、これらの手法の応用例と今後の課題について述べる。

なお、TMSでいう truth (真実) は1次的な仮説の真偽という意味合いが強いので、本来の人工知能の概念からはこれには belief (信念) という用語を用いるのが正しい。信念はひるがえしても怒られない(?)が、真実とわかった事象は絶対ひるがえるはずはないのであるから。そのため、TMSに対しては、“信念の改定機構 (Belief Revision System)” という用語を使うことも多い。(これでもおおげさな名前という気がする読者もいると思うが…)

そこで、本稿では人工知能研究者の慣例にしたがって、もっぱら、TMS、ATMS という略語 (これなら安心できる) のみを使って解説を進めることとする。

てらの たかお, しのはら やすし (勸電力中央研究所 〒100 千代田区大手町1-6-1 大手町ビル

2. 推論処理：状態空間の探索

人工知能システムによる問題解決は、コンピュータ内部に適当なデータ構造を作っておき、それを変更することによってなされる。各時点におけるこのようなデータ構造（のスナップショット）を状態という。ひとつの状態が別の状態に変化する時は必ず何らかのオペレータが作用する。

たとえば、よく知られたプロダクション・システムでは、推論処理の過程で生ずる状態をワーキングメモリに保持しておき、これに、ルールを作用させている。信念の改定を扱う場合、ひとつの状態は事実・信念とそれから導かれる結論の全体となる。

したがって、状態をノードに、その変換ステップをアークに対応させれば、これらの処理は（たとえ陽には定義されないとしても）木あるいはグラフを探索する処理と考えることができる。すると、これを効率化するには、OR分野ではよく知られた基本的なグラフ操作のアルゴリズムを適用すればよいことが容易に想像できることと思う。ただし、状態空間をどのように表現するかは非常に重要な問題であり、それによって問題解決の効率は大きく変化する。TMS, ATMSではこの点に大きな工夫がなされている。

以下では、図1のようなスイッチと電球からなる単純な回路を用いてTMS, ATMSの解説を行なう。まず、次のR1～R5のルールが成立することは容易にわかる。

〔一般ルール〕

- R1: IF Switch 1(X)
THEN Light A(X)
- R2: IF Switch 1(on) and Switch 2(X)
THEN Light B(X)
- R3: IF Switch 1(X) and Switch 2(on)
THEN Light B(X)
- R4: IF Switch 1(off) and Switch 2(X)
THEN Light B(off)

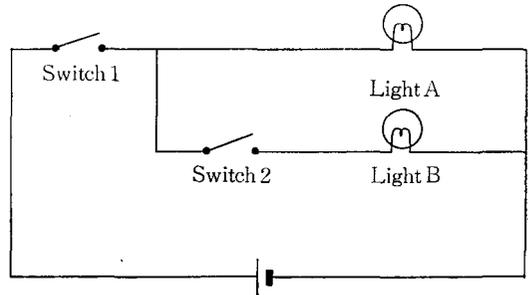


図1 スイッチとランプからなる回路

R5: IF Switch 1(X) and Switch 2(off)
THEN Light B(off)

Switch 1(X), Switch 2(X), Light A(X), Light B(X) はそれぞれスイッチと電球の状態を表わす述語である。たとえば、ルールR1は電球Aの状態がスイッチ1の状態と一致することを示す。

今、図1の回路で、Light A, Light Bのオン/オフについては、実際に検査できるがSwitch 1, Switch 2については、直接検査できないとしよう。このような系の状態を想定するために、R1～R5の一般的なルールの他に、仮説にもとづくルールを使うこととすると、推論を進めるにはさらに以下のようなルール群が必要となる。

〔仮説ルール〕

- A1: IF Maybe Switch 1(off) THEN
Switch 1(off)
- A2: IF Maybe Switch 2(off) THEN
Switch 2(off)
- A3: IF Maybe Switch 1(on) THEN
Switch 1(on)
- A4: IF Maybe Switch 2(on) THEN
Switch 2(on)

ここで、Maybe Pは、not Pがいえないかぎり、Pは正しいと信ずることを意味する。

〔矛盾についての宣言〕

- C1: IF Switch 1(on) and Switch 1(off)
THEN contradiction
- C2: IF Switch 2(on) and Switch 2(off)

THEN contradiction
 C3: IF Light A(on) and Light A(off)
 THEN contradiction
 C4: IF Light B(on) and Light B(off)
 THEN contradiction

上のルールは、オンとオフとが両立しない概念であることをコンピュータに知らせるのに必要である。

3. TMS : 状態空間の縦型探索

TMSでは、ひとつの状態は各データの真偽とそれらの論理的導出関係のネットワークであらわされる。TMSは、データの真偽が新しくわかった場合、矛盾を生じていない状態をひとつ探索する。状態を表わすネットワークは、ノードの集まりとして表現する。各ノードは、in, out のどちらかの値をとる。in, out はそれぞれ論理学の真、偽に対応し、ノードが信じられること、信じられないことを表わす。各ノードは次のような3つ組で表現される。

識別子 データ サポートリスト
 (n1 Light B(off) (SL (n2 n3) (n4)))

識別子はノードを区別する名札であり、データはこのノードの示す情報を表わす。サポートリストは、TMSに特有の概念であり、このノードが正しいと信じられる根拠 (justification) をまとめたものである。この例では、(n2 n3) を in サポート、(n4) を out サポートといい、n2, n3 がともに in であり、かつ n4 が out であるときに

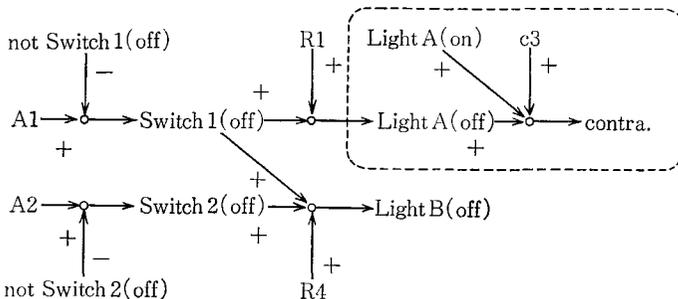


図3 新事実 Light A(on)が観測された時の矛盾の発生

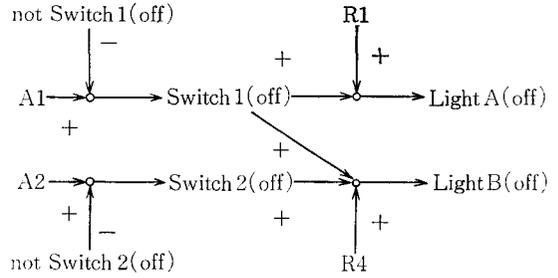


図2 事実 Light B(off)が観測された時の状況

限って、n1の値はinになる。

今、Switch1 (off) という事実が観測されたとき、矛盾の生じないひとつの状況として、推論推論システムが次のような状況を見つけたとする。

Switch 1(off) かつ Switch 2(off) かつ
 Light A(off) かつ Light B(off)

このときの論理的導出の様子を表わすと図2のようになる。線についた+, -のラベルはそれぞれその始点の述語が終点の述語のin, outサポートであることを示す。ここで、図2は、状態空間を表現したひとつのノードであることを強調しておく。

図2の状況では、Switch 1(off), Switch 2(off) という事象を仮定しているが、この時、さらに Light A (off) という事実が観測されるとこのままでは矛盾が生じる(図3)。その場合には、Switch 1(off), Switch 2(off) という仮説が誤っていたことがわかり、この仮説を訂正して矛盾のない新しい状況を作り出す必要がある。

そこで、システムは、図3に示されるネットワーク (論理的導出関係) をたどることで、仮説 Switch 1(off) の誤りを見出し、それを棄却して推論をやりなおす。

その結果、新たに Switch 1 (on), Switch 2(off) という仮説を採用することとなり、観測事実と矛盾しない状況の解釈を保つことができる(図4)。これを論理的導出関係に沿ったバックトラック (Dependency Directed Backtracking) という。TMSは、信念の変化に応じて矛盾のない状態をひ

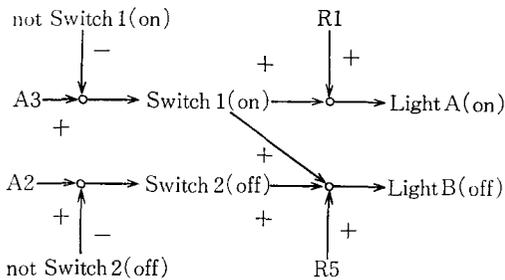


図 4 矛盾を解消した状況

とつずつ作り出すという意味で、状態空間を縦型に探索していることとなる。

4. ATMS : 状態空間の横型探索

前節のTMSでは、矛盾を生じない状態をひとつだけ保持していた。これに対して、ATMSは、事実と矛盾しない状態を同時にすべて保持する。そして、新たに観測された事実と矛盾する状態を順次消去していく。これは、状態空間を横型に探索する方法といえる。このような探索を効率よく行なうため、ATMSでは仮説 (assumption) という考え方を明示的に用いている。

仮説とは、観測事実と矛盾しないかぎりには自由にその真偽を選択できるデータである。上記の例では、仮説ルール中に maybe とともにでてくる Switch 1 (off) や Switch 1 (on) などがこれにあたる。ATMSでは状態は各々仮説の組合せに対応させることができる。これを表現するために各ノードにTMSにおけるデータと (in サポートのみからなる) サポートリストに加えてそのデータが真となるための前提となる仮説の組 (環境: environment) のリスト (これを label という) も保持している。そしてあるデータが新たな観測事実と矛盾を生じた場合、そのデータの環境が許されない仮説の組合せであったと判断して、その組合せを棄却する。

すなわち、その組合せを含む環境を

すべてのノードのラベルから消去する。このような矛盾を生じる仮説の組合せを **nogoods** と呼ぶリストで管理して、無駄な推論を防いでいる。この操作によって、各データを真とするために必要な環境を維持できる。厳密には、各ノードは次のような構造をもつ。

識別子 データ

(n1 Light B(off))

ラベル

{{Switch 1(off)} {Switch 2(off)}}

サポートリスト

{{(n2 n3) (n6)}}

たとえば、前の例で、事実 Light B(off) が観測された時点における状況を、仮説の組ごとに表示すると図 5 のようになる。図 5 に示された仮説の組の箱を View Point または World という。

ここで、各箱の下側はその View Point で新し

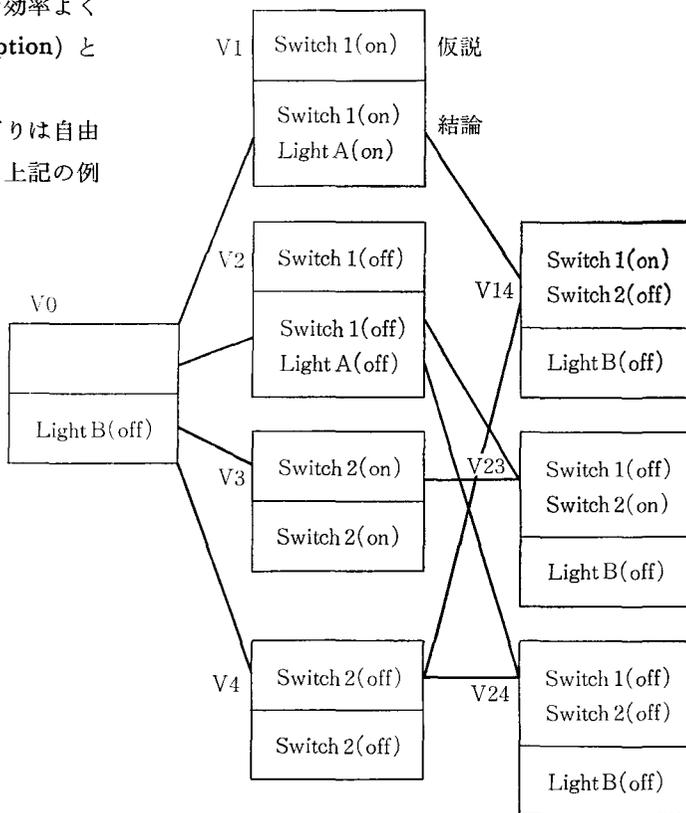


図 5 事実 Light B(off) における View Point の状況

く得られる結論を、上側はその前提として採用した仮説を示す。なお、各 View Point からは、それ以前に得られた結論（すなわち、図の左側にある箱の結論）はすべて参照できる。

この状況で、さらに、新事実 Light A (on) が観測されたとすると、V2 では矛盾が生じる。すると、V2 および、V2 に従属する V23, V24 は棄却され、同時に V3 も棄却されて V14 のみが残る。このようにして、新しいノードが与えられるつど、矛盾しない解釈の維持ができる。

ATMS の手法はループのチェックなどが不要で効率がよいことが知られている。この手法では特に nogood の表現を最適化しており、データごとにラベルの示す最小の環境を保持できるよう、集合演算算法に工夫がなされている。ただしオリジナルの ATMS は、複数の互いに矛盾する状態を並列に保持したままの推論を効率的に行なうことを目的に設計されているために、前節で述べたような1つの妥当な状態のみを導く場合の効率はあまりよくない。そこで状態空間を最良探索できるように ATMS の推論過程を制御する手法も提案されている。これを仮説にもとづく導出関係に沿ったバックトラック (Assumption-Based Dependency Directed Backtracking) という [9]。

5. 応用例と今後の課題

TMS は、もともと回路の状態を記号的に解析するシステムを実現するために開発されたものである [3] が、ここで採用されている Dependency Directed Backtracking の手法は、仮説的推論においてばかりでなく再推論を行なう場合にはすべて利用できる。TMS を用いた実用的なエキスパートシステムとしては、変電所のレイアウト設計支援システムがある [10]。

ここでは、システムがレイアウトを階層的に生成するとき、その依存関係を保持している。そして、システムが提示したレイアウトの一部を、利用者が変更した時に TMS の機能を用いて関連設

備の再配置を有効に行なうようになっている。

それに対し、ATMS はスケジューリング問題を扱うエキスパートシステムでしばしば用いられている。その代表的な例は [11] に紹介されている。また、ATMS の開発者は、これを回路の故障診断問題に適用している [12]。

エキスパートシステムで故障診断問題を扱う場合、通常、単一故障を仮定することが多いが、[12] で提案されたシステムでは複数原因による故障の候補をすべて導くのに、ATMS の機能を利用している。

また、TMS, ATMS は、ハイブリッド型と呼ばれる、最近の高機能のエキスパートシステム開発ツール [13] に実装されている [14], [15]。ただし、これらの機能を現実の問題に使おうとする場合、必要な知識を適切な形で記述することが非常に難しくなる。これらの手法を使いこなすには、適切な問題を選択するとともに、その表現方式についていっそうの研究開発が必要である。

6. おわりに

本稿では、仮説的な推論をコンピュータで行なう典型的な手法である TMS, ATMS について、状態空間の探索法という視点から解説した。これらの手法のコンピュータへの実現方法については TMS についてはたとえば [5] などの教科書が、ATMS については、De Kleer の論文 [6] が参考となる。また、仮説推論に関連する論理的な話題は本稿では触れなかったが、これについては、たとえば、[16]などを参照されたい。

参考文献

- [1] Nilson, N. J. (白井, 辻井, 佐藤(訳)): 人工知能の原理. 日本コンピュータ協会, 1983.
- [2] Hayes-Roth, F. et al. (eds.) (AIUEO(訳)): エキスパート・システム. 産業図書, 1985.
- [3] Stallman, R., and Sussman, G. J.: "Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis." *Artificial Intelligence*,

Vol. 9, No. 2 (1977), pp. 135-196.

- [4] Doyle, J.: "A Truth Maintenance System." *Artificial Intelligence*, Vol. 12, No. 3 (1979), pp. 231-272.
- [5] Charniak, E., Riesbeck, C. K., and McDermott, D. V. (白井, 片桐 (訳)): 人工知能プログラミング. 日本コンピュータ協会, 1986.
- [6] De Kleer, J.: "An Assumption-Based Truth Maintenance System." *Artificial Intelligence*, Vol. 28, No. 2 (1986), pp. 127-162.
- [7] De Kleer, J.: "Extending the ATMS." *Artificial Intelligence*, Vol. 28, No. 2 (1986), pp. 163-196.
- [8] De Kleer, J.: "Problem Solving with the ATMS." *Artificial Intelligence*, Vol. 28, No. 2 (1986), pp. 197-224.
- [9] De Kleer, J.: "Back to Backtracking: Controlling the ATMS." *Proc. AAAI '86* (1986), pp. 910-917.
- [10] 吉田健一 (他): "知識工学の変電所レイアウト・システムCADへの応用." 日立評論, Vol. 67, No. 12 (1987), pp. 45-48.
- [11] Hackson, P. C., and Maletz, M. C.: "Critical Path Resource Allocation Using ART Viewpoints." *Proc. 6th International Workshop on Expert Systems and Their Applications* (1986), pp. 405-415.
- [12] De Kleer, J.: Williams, B. C.: "Diagnosing Multiple Faults." *Artificial Intelligence*, Vol. 32, No. 1 (1987), pp. 1-55.
- [13] 寺野隆雄: "知識システムの構築環境." 計測自動制御学会講習会資料—エキスパートシステム: 方法論と応用 (1987), pp. 83-96.
- [14] Clayton, B.: *ART Programming Primer*. Inference-Corp, 1985.
- [15] Morris, P. H., and Nado, R. A.: "Representing Actions with an Assumption-Based Truth Maintenance System." *Proc. AAAI '86* (1986), pp. 13-17.
- [16] 人工知能学会誌: "特集: 高次人工知能へ向けてのパラダイム." Vol. 2, No. 1 (1987).

1987年9月号

© 日本オペレーションズ・リサーチ学会. 無断複写・複製・転載を禁ず.

(33) 611

最新刊

パソコン・パッケージによる 例解線形計画法

平本 巖・木下昌男・栗原和夫共著 A5・1800円

ソフト別売 定価80,000円

入門者向けに、線形計画法におけるパソコン応用を解説。プログラム・パッケージを用いて、線形計画問題を解きすすむうちに理解を深めることができる。併せてプログラム・パッケージも販売。(ソフトウェア御希望の方は小社営業部まで。)

主要目次 線形計画法入門 (単体法 感度分析 2段階単体法他) 例題編 (生産計画問題 栄養問題 混合問題 多期間計画問題他) パーソナルコンピュータの活用 (手法理解のためのLPパッケージ 実務に利用するためのLPパッケージ 教育の場を利用するためのLPパッケージ他)

Computer Today 定価880円
好評発売中

9月号特集

オブジェクト指向言語の新展開

■別冊 プログラム移植 定価1380円

数 理 科 学

10月号特集 / 9月20日発売

定価930円

場

場とはなにか

内山龍雄

粒子と場

江沢 洋

並進不変性と場の量子論

中西 襄

弦理論における場

畑 浩之

場の理論と超流動との接点

中原幹夫

ベクトル場の極限——特異摂動理論

池上宜弘

場の数理とその存在の認識

大矢雅則

進化的に安定な場

松田博嗣・山村則男

動物発生における場

若原正己

行動科学と場

松原 望

場の認識論

坂本百大

<別冊>

宇宙 —なぜ始まったか 定価2000円

真空の相転移とインフレーション宇宙, ニュートリノとX線天文学等, 眩目の宇宙研究最前線.

サイエンス社

東京都千代田区神田須田町2-4 安部徳ビル

☎03(256)1091 振替 東京7-2387