

解説



エキスパートシステムにおける知識表現と推論†

新田 克己††

1. はじめに

エキスパートシステムは「専門家の知識を組み込み、専門家の代行あるいは補助を行うソフトウェア・システム」である。対象分野の専門家の知識が組み込まれていないシステムはいかに複雑な機能が組み込まれていても、エキスパートシステムとは呼ぶことはできない。エキスパートシステムの構築は、(i) 専門家から知識を抽出し、(ii) 計算機上にその知識を表現し、(iii) 表現された知識を利用して問題を解決する、という三つの段階によって行われる。(i)の知識抽出は、専門家へのインタビューや専門書による専門知識の修得、計算機による自動的な学習を含んでいる。エキスパートシステムの能力を決定する最も困難かつ最も重要な段階である。(ii)は「知識表現」と呼ばれ、フレーム、ネットワーク、論理式、手続き、ルール、オブジェクトなどによる知識表現の枠組みが研究されてきている。(iii)は「推論方式」が最も重要な技術的課題である。

知識表現と推論方式はエキスパートシステムの構築にあたり最初に決定すべき重要な選択肢であり、その良否によって以後の開発に大きな影響を与えるものである。以下にエキスパートシステムに用いられている知識表現と推論方式を概観する。

2. 知識の種類

ここでは知識の種類、および、エキスパートシステムを構築する上での問題点について述べる。

2.1 知識の種類

(1) オブジェクト知識とメタ知識

専門家の知識は以下のような種類に分類することができる¹⁾。

① 事実・関係についての知識

② 手続きについての知識

③ 判断についての知識

①は「～は～である」という型で表される知識であり、②は「～をするには～をせよ」という型で表される知識である。(3)は判断をするための規則や手続きであり、長年の経験によって得られた発見的知識もその一つである。

①②は専門分野について物事をどれだけ知っているかという知識量に関する。③の知識の量と質は専門家と素人を区別するものである。③の判断を効率的に行うためには、「知識の構造に関する知識」や「知識の使い方に関する知識」が必要である。たとえば囲碁において定石をいくらか知っていても、どの場面でもどの定石を使うのかを知らなければ役に立たないであろう。このような知識を「メタ知識」、ほかの知識を「オブジェクト知識」ということがある。メタ知識は問題解決において、推論の制御のために用いられる。

(2) 深い知識と浅い知識

知識は一樣なものではなく、問題解決のための知識のレベルが存在する。たとえば、「夕焼けだから明日は晴れる」という知識は経験的に得られたものである。これを裏付けるには、偏西風の影響などで天気は西から東へ移動するという気象学の知識が必要である。後者の知識(気象学の知識)は原理・原則に関する知識であって、厳格であるが、問題解決をこの知識だけで行うのはかえって煩わしいことになる。専門家は前者の経験的知識を豊富に持っている、その知識によって問題を解決すると考えられる。

問題解決のため直接用いる経験的知識を「浅い知識」と言い、物事の原理や構造に関する知識を「深い知識」と言う。浅い知識が不十分で問題解決ができないときには深い知識を用いて解決できることがある。

2.2 専門知識の表現の問題点

エキスパートシステムによる問題解決を行うには、専門家の知識を抽出し、計算機にのる形に形式化しな

† Knowledge Representation and Inference Mechanism of Expert Systems by Katsumi NITTA (Electrotechnical Laboratory), Computer Science Division

†† 電子技術総合研究所ソフトウェア部

ければならない。形式化とは、あらかじめ与えられた知識表現および推論方式の枠組みの中で記号処理として知識を整理することである。通常の問題を解決するには、解くべき方法がアルゴリズムとして定式化されているので、それをプログラミングするだけでよい。しかし、エキスパートシステム構築においては、専門家の持つ知識は自分自身でもその内容が良く理解できていないことが多い。したがって、システムの開発には当初から完全な専門知識が得られることはなく、専門家による評価とそのフィードバックが必要である。そのため、専門知識の引き出しと整理のための手助けが必要である。

3. 問題解決と学習

3.1 問題解決と推論

「推論」とはすでに所有している知識から新しい知識を導き出すことをいう。人間の問題解決は発想推論、演えき推論、帰納推論からなるといわれる⁸⁾。「発想」とは、新しい事実が生じたときに仮説を生成する過程であり、「演えき」とは、仮説から帰結を導く過程であり、「帰納」とは、仮説を生成し、帰結を検証する過程である。エキスパートシステムにおける問題解決は主に演えき推論でなされている。帰納推論や発想推論は、知識を自動的に形成する学習の方法として研究されている。

問題解決を行うには、その問題が独立な部分問題に分割できるならば、おのおのの中間的な問題を解決し、その結果を統合することにより最終的な解が求められる。問題解決における中間段階（中間状況）を節点で表すとき、問題解決の全体はこれらの節点で作られる「AND/OR 木」の探索と考えることができる（図-1）。木のルートは結論（仮説）、木の末端は観測データ

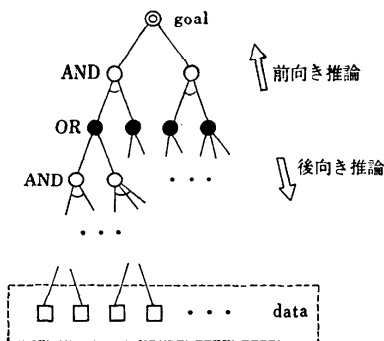


図-1 AND/OR 木

ータに対応する。節点が「AND 節点」のときにはその節点の子の節点の部分問題がすべて解決したときに解決され、「OR 節点」のときには子の節点のいずれかが解決したときに解決されたと判断される。

問題解決は、観測データが与えられ、そのデータを基に推論を行って、結論を求めることで行われるのである。このとき、AND/OR 木を末端（観測データ）からルート（結論）へと探索していく方法を「前向き推論（データ駆動）」といい、逆に、仮説をあらかじめ設定し、AND/OR 木をルートから下へたどる探索方法を「後向き推論（ゴール駆動、モデル駆動）」という。さらに、前向きと後向きの両方を組み合わせて行う推論を「双方向推論」という。

一つの節点から探索できる枝が複数あるときには、探索順序を決定するための戦略が必要である。盲目的探索としては、最も深い節点を優先的に探索する方法（縦型探索）、同じ深さの節点を均等に探索する方法（横型探索）があり、発見的知識を用いる探索としては、評価値の高い節点を優先的に探索する方法（最良優先探索）、などがある。

3.2 代表的な知識表現と推論

代表的な知識表現と推論方式を以下に紹介する²⁾⁻⁶⁾。

(1) プロダクション・ルール

プロダクション・ルールによる知識表現では、知識は

IF <条件> THEN <結論/手続き>

の形のルールの集合として表される。作業メモリ（4.1節参照）には事実データが格納されている。OPS 5 における事実データの例を以下にあげる。先頭の person はクラス名、^ の付いたもの (name と mother) は属性、^ のないもの (taro と hanako) は値である。

(person ^ name taro ^ mother hanako)

プロダクション・システムにおいては、前向き推論は、作業メモリの内容とルールの条件部を比較し、条件部を満足するルールの結論部を実行することに相当する。後向き推論は、作業メモリと結論部を比較し、結論部を満足するルールの条件部を実行することに相当する。適用可能なルールが複数存在するときには、そのうちの一つを選択する戦略（これを「競合解消戦略」という）が必要である。戦略には、条件部が最も詳細なルールを選ぶもの、作業メモリに最も新しく追加されたデータにマッチしたルールを選ぶもの（これを「イベント駆動型」という）、条件部やルールに付

加されている重要度が最も高いルールを選ぶもの、最初にマッチしたルールを選ぶもの、手続きで選択されるルールを決定できるものなどがある。

ルール表現の利点は、知識が断片的で独立性が高いので、知識の変更や追加が容易であること、「もし～ならば～をする」形式の表現は人間の断片的な知識の表現と近いので表現が自然であること、表現が一様で管理しやすいこと、などである。

欠点は、知識の全体構造が把握しにくく、矛盾検出が困難であること、デバッグがやりにくいこと、推論に柔軟性がないこと、などである。

(2) フレーム

フレームは静的な知識の典型的な枠組みを表現するデータ構造である。代表的なフレーム型知識表現言語 FRL では、一つのフレームは、フレーム名、スロット (slot)、ファセット (facet)、値、などからなる。「フレーム名」はそのフレームが表現する概念の名前である。「スロット」はその概念の持つ項目名を表す。「ファセット」はデータの型 (値、暗黙値、値の拘束条件、付加手続きなど) を表す。付加手続きは特定のスロットを参照したときに起動される手続きである。「値」はデータや手続きである。フレームは is_a リンクによってほかのフレームと結合され、上位概念/下位概念による階層構造を構成する。下位フレームは上位フレームの情報を継承する。

フレーム自体は静的なデータ構造であり、推論を行うには外部に推論機構を必要とする。フレームの推論は付加手続きの起動によって行われるが、そのほかに、最近では、ルールとフレームを組み合わせた知識表現を採用し、推論は主にルールの起動によって行う方法がある。

フレームによる表現の利点は、知識を階層的に整理することができること、宣言的な表現だけでなく付加手続きを埋め込むことができること、ルールと合わせた知識表現が可能なこと、などである。

欠点は推論の制御がユーザにまかされているので、その記述の負担が大きいことである。

(3) オブジェクト

オブジェクトとは、それ自体の内部状態とその内部状態に対する手続きからなる計算の単位である。フレームと類似の構成をとることが多く、性質も似ている。しかし、推論がオブジェクト間のメッセージの交換で行われることと、オブジェクト内部の情報は必要とき以外は外部に公開されないことなどが異なる。

利点と欠点はフレームとほぼ同じであるが、制御情報がオブジェクト間に分散されているので、概念間の独立性が高く、シミュレーションに向いている反面、全体の整合をとるのはより困難である。

(4) 述語論理

論理学は人間の思考を形式化する手段として長期にわたって研究されてきた。論理にはいろいろな種類があるが、その中で述語論理が最も重要である。述語論理では、たとえば

Human is fallible.

Socrates is a human.

は、それぞれ

$$\forall x \text{ human}(x) \rightarrow \text{fallible}(x)$$

human (socrates)

と表現される。述語論理の推論はルールと同様に前向き推論と後向き推論がある。前向き推論は一種の三段論法で行われ、後ろ向き推論は一種の背理法を用いて行われる。Prolog は一階述語論理 (の一部) に理論的基礎を置く言語である。

論理式の利点は、人間の思考過程に近い自然な表現ができること、前提が正しければ結論が正しいことが保証されること (厳密な議論ができる)、宣言的知識と手続き的知識の区別がない (同一の形式をしている) ことなどである。

欠点は、定理証明についての制御的知識を表現できないことである。

(5) ネットワーク

ネットワークによる知識表現では、個々の概念は節点で表し、概念間の関係はリンクで表現される。代表的なリンクには instance_of, is_a, part_of などがある。is_a リンクで結合された上位概念の情報は下位概念に継承される。

推論はネットワークの照合で行われる。質問がなされたときには、それに対応したネットワークが生成され、照合によって解が検索される。照合において付加手続きが起動できるものもある。

ネットワークによる知識表現の利点は、記述が静的であり、知識を自然に表現できることにある。

欠点は、推論の結果が正しい結論であることの保証がないこと、ネットワークが大規模になったときの照合戦略が単純でないこと、などである。

(6) 手続き

手続きとしての知識表現においては、知識は事実を操作するプログラムとして表される。推論制御の知識

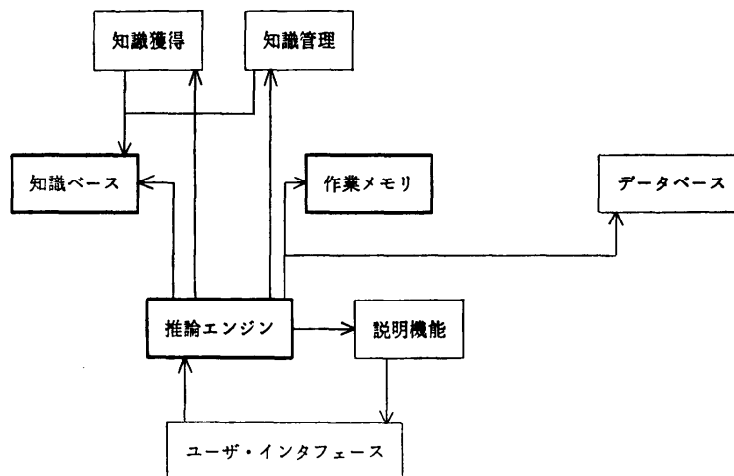


図-2 知識ベースシステム

もプログラム中に埋め込まれるので、発見的な知識を表現することができる。

利点は、実行効率が良いこと、非単調な事実を表現するのが容易なことである。

欠点は、知識の局所性がなく、改定が容易でない点である。

4. エキスパートシステムの知識表現と推論

前章で知識表現と推論方式について述べたが、それぞれに特徴があり、エキスパートシステム構築に無条件に採用することはできない。以下にエキスパートシステムにおける知識表現と推論の現状について述べる。知識表現と推論は密接に関連しているので、4.2と4.3の区分は説明の便宜上振り分けたものであって厳密な区別はしていない。

4.1 知識ベースシステム

エキスパートシステムは「知識ベースシステム」として構築されることが多い。一般的な知識ベースシステムは、知識を格納する「知識ベース」と問題解決を行う「推論エンジン」と事実を格納する「作業用メモリ」と周辺の支援モジュールからなっている(図-2)。知識ベースシステムとしての構成をとると、知識表現の枠組みと推論機構を標準化することができる。したがって、知識を追加・削除・変更しながら開発されるエキスパートシステムには都合がよい。最近では、知識ベースシステムの推論エンジンと支援モジュールを「エキスパートシステム構築用ツール」として商用化されるに至っている。

エキスパートシステム構築ツールは、もともとは特定のエキスパートシステムのために開発された知識表現と推論の枠組みをほかのシステム開発に用いられるように汎用化したものが多い。たとえば HEARSAY から AGE が、MYCIN から EMYCIN が、CASNET から EXPERT が生まれている。ツールを使うことによって、開発のコストは LISP や Prolog などの人工知能用言語で初めから開発するのに比べて 100 分の 1 に減少するとも言われている⁹⁾。

EMYCIN, OPS5, EXPERT などは、知識表現と推論は一つの方式を提供するものである。EMYCIN はルールを後向き推論するものであり、OPS5 はルールを前向き推論するものである。EXPERT はルールを FF (事実から事実へ)、FH (事実から仮説へ)、HH (仮説から仮説へ) に分類し、前向き推論で FF, FH, HH と起動していくものである。AGE は黒板モデルによるツールであり、イベント駆動とゴール駆動のいずれの推論方式も可能である。

第2世代のツールは、KEE, ART, LOOPS, EX-CORE, ESHELL などのように、複数の知識表現と複数の推論方式(ハイブリッド方式という)を提供している。特に、表現方式はルールとフレームを融合し、前向き/後向き推論が可能なものが多い。

EUREKA や PICON では、高速な推論を行うことによって実時間制御を行う目的で用いられるものである。

また、最近では Shell-KABA, apes のようにパーソナル・コンピュータ上で動作するものも開発されて

きている。

4.2 知識表現に要求されるもの

エキスパートシステムに用いられる知識表現としては、

- (1) 強力な表現力を持つこと、
- (2) わかりやすい(自然な)表現であること、
- (3) 追加・変更が容易であること、

などが評価の対象となる。

(1)の「強力な表現力」とは、いろいろな種類の知識を表現できることを意味する。これは前述のオブジェクト知識、メタ知識を表現できることであり、具体的には、因果関係、あいまいな知識、時間の変化、常識的知識、パターンの知識などを表現できることである。

知識ベースを構築するにあたり、知識が厳密に記述できない(不確実)、あるいは知識が信頼できない(非単調)、という場合がある³⁾。

「不確実な知識」の表現には、確信度(CF: Certainty Factor)、主観的確率、Dempster-Shaferの理論、Fuzzy論理などがある。たとえば、MYCINにおいては、ルールに確信度を導入して、結論の信頼度を表している。確信度は-1から1までの値をとる。-1は確実に否定的であることを表し、+1は確実に肯定的であることを表す。たとえば、

PREMISE: (\$AND X Y)

ACTION: (A CF_r)

なるルール(CF_rは結論Aの確信度)において、Xの確信度をCF_xとし、Yの確信度をCF_yとすると、(\$AND X Y)の確信度CF_{xy}はmin(CF_x, CF_y)で表され、結論Aの確信度はCF_{xy}とCF_rの積になる。

正しいと信じられていた知識が、別の知識の追加によって正しくなくなることがある。時間とともに変化する知識や常識としての知識(たとえば「TOMは鳥である」から「TOMは飛べる」と思っていたところ、「TOMはペンギンである」ことが判明したので「TOMが飛べる」は成立しなくなる)がその例である。このような知識は、論理による知識表現においては「非単調論理」として形式化され、手続き型の知識表現においては作業メモリの書き換えとして表現される。

知識の整合性を管理し、非単調な知識処理を実現する機構としてTruth Maintenance System (TMS)²²⁾がある。これは作業メモリ中のデータの追加・変更が

あり、新しい事実の発生や従来は正しいと信じられてきた事項の否定があったときに、その波及効果を計算し、知識を無矛盾に保つものである。

因果関係の知識においては、原因と結果の間に時間的な遅れが生じることがある。時間を表現するには、時間を離散的なものとするか、連続的なものとするか、また、原因となる行為をある瞬間に行われたと考えるか、一定の幅をもつ区間内に行われたかなどの選択により表現方法が異なる。論理による知識表現においては、sentenceの真偽値が時間によって異なるとする時相論理がある。また、多重世界を表現する機能を持つ知識表現においては多重世界による時間の表現が可能である。

知識の表現に関して、かつては、知識を「宣言的」に表現するか、「手続き的」に表現するか論争があった。宣言的表現では、「～は～である」の形の知識を静的データとして表現するものであり、知識の改定が容易である利点があった。手続き的表現では、「～をするには～をせよ」の形の知識をプログラムとして表現するものであり、高速の推論が可能である。現在では、ルールとフレームの融合(たとえば、KEE)や、論理式とフレームの融合(たとえば、Krypton)、論理式とオブジェクトの融合(たとえば、ESP)、フレームへの付加手続きの採用、他言語の呼び出し機能、などにより、両方の表現が可能なものが増えてきている。

(2)の「わかりやすい表現」とは、人間の思考に近い表現であることを必要とする。知識表現は、専門知識の引き出しと整理のための思考方法を提供するからである。フレームは心理学の研究から生まれたものである。また、ルールや論理式による知識表現は人間の思考に近いとされている。厳密な表現が必ずしもわかりやすい表現とは限らない。たとえば、微分方程式で知識を記述し、その厳密解を求める代わりに、定性的な知識表現と解で十分なこともあり、メンタルモデルとの関連でも研究されている。

(3)の「追加・変更しやすい」とは、知識の部分的改定がしやすいことを意味する。知識を一部修正したために全体の整合がとれなくなることがないように、知識はなるべく局所的であることが望ましい。また、知識の改定により、知識が矛盾を生じることのないように、知識管理が容易であることが必要である。知識の修正がしやすいためには知識の表現形式が一様であるほうが望ましい。

4.3 推論方式に要求されるもの

エキスパートシステムにおける推論方式においては

- (1) 高速な推論であること、
- (2) 推論過程について説明ができること、
- (3) 厳密であること、
- (4) 他の可能性を示せること、
- (5) 協調問題解決ができること、
- (6) いろいろな推論方式が利用できること、
- (7) 他言語の呼び出しができること、

などが評価の対象となる。

(1)知識量や事実が増加したり知識が複雑化したりすると、問題解決に時間がかかるようになる。推論が妥当な時間に行えないと、エキスパートシステムとしては役に立たない。高速な推論を行うには、「AND/OR木の探索戦略の利用」と「知識のコンパイル」の二つの方法がある。前者に属するものとして「メタ知識の利用」と「知識の構造化」、後者に属するものとして「ルールのコンパイル」や「推論機構の部分計算」の技法がある。

メタ知識は、知識の使い方についての知識であり、不要な検索を防ぐのに用いる。メタ知識を用いた推論を「メタ推論」という。ルール型の知識ベースにおいては、プロダクション・ルールはあらかじめグループに分けることができる。メタ知識はルールとして与えられ、検索するルール・グループの絞り込みに用いられる。また、ルール・グループを木構造に整理して検索の制御に用いる「文脈木」も同じ目的のための手法である。フレーム型においては、付加手続きを起動することでメタ知識が表現される。論理型においては、オブジェクト知識とメタ知識の融合 (amalgamation) の研究が行われている。これは証明機構 (推論エンジン) 自身を論理式で記述することにより、論理式で制御情報も記述しようとするものである。証明機構を論理式で表現することは、推論の速度を低下させる。しかし、利用する論理式 (知識ベース中の論理式) が既知ならば、部分計算の技法により証明機構をその論理式向きに最適化することで速度向上が図れる。

コンパイルには、ルールの条件部のコンパイル、結論部のコンパイル、などがある。条件部のコンパイルの一種としては、パターン照合の重複を避けることによって高速化を図る RETE アルゴリズム²⁶⁾が知られている。

(2)の説明機能は推論の妥当性を示す上で重要である。説明機能において代表的なものに、MYCINで用

いられた how と why のコマンドがある。how コマンドは「なぜそのような結論が得られたか」を質問するものであり、AND/OR木においてルートから現在の節点へ至るまでの推論の経過を表示させるものである。why コマンドは「どうしてそのような質問をするのか」を質問するものであり、AND/OR木において現在の節点から下の部分木の経路を表示させるものである。

(3)厳密であることとは、推論の結果として得られた結論が、論理的に正しいこと (正当性) をいう。論理型の場合、推論過程が論理式の定理証明に対応するので、正当であることが保証されている。

(4)故障診断などにおいては、解が一意に決定できないことがある。このようなとき、推論過程を後戻り (バックトラック) して別の解を探索したり、複数の推論過程を平行に実行したりする機能が必要である。これは推論の履歴の保存や作業メモリの内容を異なる推論過程ごとに分割することで実現することができる。ARTで採用されている「ビューポイント (view point)」は後者の例である。ビューポイントは、次に紹介する「黒板」をさらに発展させたもので、作業メモリ内で事実の集合を格納する単位である。時間の経過による状況の変化はビューポイントの連鎖で表現され、状況の変化の可能性が複数あるときにはビューポイントの分岐で表現される。ビューポイントにより複数解の探索を平行に行うことができる。

(5)知識は一カ所にまとまって存在しているとは限らず、複数の専門家の協力で問題解決を行う場合がある。このようなときには、一つの知識ベースが複数の知識群 (知識源: Knowledge Source) からなっていると考え、これらの知識群の協調によって問題の解決を行う機能が必要である。音声認識システム HEARSAY-II では音響学、音声学、構文論、意味論などの専門知識を総合して、信号データから文を識別するものである。それぞれの専門家の知識をルールで表現し、その協調で問題解決を行うのであるが、「黒板 (Black Board)」とよばれる大局的メモリを用いて知識群の調整を行っている。黒板の中にいくつかのレベルがあり、あるレベルにデータが書かれるとそのレベルの関連の知識源が起動される。

(6)問題解決にはきめの細かい推論が必要である。たとえば、双方向推論が可能であれば多様な問題に対処できる。プロダクション・ルールにおいては、一つのルールが前向き/後向きのいずれにも使える方式

と、ルールが前向きルールと後向きルールの2種類からなる方式がある。以下のルールは後者の例である。前向き推論の実行中に、以下のような goal (...) を含むような前向きルールがあると、 X が成立することを証明するために後向きルールによる後向き推論が行われ、証明が終了すると前向き推論に戻るものである。

IF: (\$AND goal (X) Y)

THEN: ...

そのほかに、浅い知識だけで問題解決ができないときに、深い知識を用いて浅い知識を生成し、それを用いて問題を解決する技法が研究されている²¹⁾。

(7)他言語の呼び出しは特に設計システムなどにおいて重要な機能である。

4.4 応用分野による知識表現と推論

エキスパートシステムはいくつかのタイプに分類することができる。以下に、代表的なタイプの特徴と知識表現、推論方式について述べる。

(1) 分析型システム

分析型システムとは、「解釈システム」や「診断システム」などのように、結論の候補が有限であり、観測データからもっともらしい結論の一つを決定するタイプのシステムである。このタイプのシステムにおける問題解決は、結論の候補の数が少ないときには、ゴール主導型の後向き推論による方が効率が良いことが多い。問題解決のための重要な要素技術は、推論が妥当であることの説明機能、あいまいな知識の表現方法などであり、そのほかに、解釈システムにおいてはデータに混入する雑音の扱いが、診断システムにおいては (i) 複数の故障があるときの推論、(ii) 対象物についての完全な知識が得られないときの扱い、などが問題となる。

質量分析器のデータと分子式から構造式を決定する DENDRAL においては、問題解決は分子式から構造式の候補を生成し、その構造式から予想される質量分析器のデータと実際のデータを比較して確からしい一つの構造式を決定することによる。一つの分子式から生成される構造式 (構造式の生成は手続きの形で与えられている) は膨大なものになるため、専門知識を利用して生成される構造式の数进行を絞る必要がある。そのため、プロダクション・ルールにより専門知識の表現を行い、前向き推論によって、存在しえない構造式のクラスを排除している。

地中のセンサ (ディップメータ) からの観測データから地質構造を推定する Dipmeter Advisor では、

問題解決は雑音を含む観測データから4種類のディップパターンを検出し、パターンの変化から地層や断層を認識することになる。地層/断層の認識にはプロダクション・ルールで書かれた知識を前向きに推論することによって行われる。

音声認識システム HEARSAY-II では、解釈の知識は、音声信号から音素の生成、音素から音節クラスの仮説の生成、音節クラスから単語の仮説の生成、... などの知識源 (プロダクション・ルールの集まり) から成り、7つのレベルを持つ黒板を用いる。知識源にはレベルの低いデータを合成して高いレベルの仮説データを求める (ボトムアップ) ものと、高いレベルの仮説を低いレベルのデータから検証する (ボトムアップ) ものがあり、真実度と優先順位を用いて推論制御を行う。

感染症の診断システム MYCIN においては、システムが患者に質問をし、その答えを入力することによって、病気の候補を決定し、適切な治療法を指摘するものである。知識を確信度付きのプロダクション・ルールで表現し、後向き推論を行っている。

緑内障の診断システム CASNET においては、知識は観察、病理生理学的状態、疾患カテゴリの三つのレベルからなる。第1レベルと第2レベルの節点は連想リンクで結ばれ、第2レベルと第3レベルの節点は分類リンクで結ばれ、第2レベルの節点同士は因果リンクで結ばれている。各リンクには信頼度を示す数値が付加されており、観測データが与えられると、原則として前向き推論で疾患カテゴリが決定される。

そのほかに ONCOCIN, MECS-AI はルールによる表現を、INTERNIST (CADUCEOUS) は意味ネットワークによる表現を、PIP ではフレーム表現を採用している。

一般的に、知識ベース作成が容易なプロダクション・ルールや、表現力の強力なフレームが分析型システムに向いている。

(2) 合成型システム

合成型システムとは、「計画システム」や「設計システム」のように、拘束条件を満たしながら最適解を求めるタイプのシステムである。このタイプのシステムは、解の候補が無限にあり、あらかじめ仮説をたてておくことが困難である (最適解があるか否かもわからないことが多い)。問題解決に重要な要素技術は、パターン情報の記述 (またはパターンを扱う言語とのインタフェース)、高速な推論、大容量の知識の扱い、

使いやすいユーザ・インタフェースなどである。また、計画システムにおいては、(i)問題の規模が大きく、行動予測をすべて行えないときの扱い、(ii)計画が複数の行為者によって実行されるとき調整、などが問題となり、設計システムにおいては、(i)拘束条件が多岐にわたり、それらを統合できないときの扱い、(ii)設計過程における選択が結果に与える影響がすぐには分からないときの扱い、(iii)膨大な拘束条件を示すパラメータの管理、(iv)シミュレーションや表現の変換(たとえば回路図からレイアウト図へ)や検証のように多種にわたる技法の開発、などが問題である。

分子遺伝学の実験の計画を行う MOLGEN では、遺伝子に関する知識によって抽象的な計画をたて、それを詳細化するものである。計画は、計画の戦略を決定する戦略空間、制約条件から計画設計を行う設計空間、実際の操作に具体化する計画空間からなる。詳細化はその計画が後で捨てられることがないと判断されるまで延期される「最小委託 (least commitment) 方式」を採用している。知識は LISP とフレーム・ベースのツールである UNITS により表現されている。

コンピュータ・システムの構成を設計する XCON では、構成要素の知識と制約条件(構成要素の連結、配置など)を用いて、注文にあう機器構成を作るものである。設計手順は OPS5 のプロダクション・ルールで書かれており、前向きに推論される。

設計型システムでも、プロダクション・ルールやフレームを用いるのが一般的であると思われる。

(3) 制御型システム

制御型システムとは、対象物のモニタから信号を受け取って状況を把握し、予測を行って、適切な操作を行うシステムである。

制御型システムにおいては、(i)リアルタイムに結論を出す必要があること、(ii)信号に雑音や誤りがあっても、安全な方向に操作する必要があること、などが問題である。したがって、最も重要な要素技術は推論の高速化である。

オペレータがオペレーティング・システムの監視と制御を行うのを援助する YES/MVS では、JES (job entry system) の行列空間の保持、コンピュータ間のネットワーク・コミュニケーション、ハードウェア・エラーへの対応、などの6つの仕事を行う。YES/MVS は実時間で動作し、MVS (multiple vertical storage) のメッセージを直接に解釈して、コマンドをオペレーティング・システムに送ったり、オペレータに

指示を送ったりする。OPS5 の上を実現されている。

NAVEX は、スペースシャトルの速度と位置をレーダのデータから推定し、エラーの検出と警告を行うシステムである。ART を用いて構築されており、知識はルールとフレームで表現されている。

FALCON は化学プラントにおいて、センサや警報装置からのデータから異常を検出するシステムである。プラントの構成要素の故障の影響や入力異常に対する出力の影響などの知識を用いてデータの解釈を行う。知識はルールとネットワークで表現され、前向き推論による解決を行う。

5. おわりに

エキスパートシステムの知識表現と推論方式について述べた。エキスパートシステムは知識工学と密接な関係にあり、その技法を多く採り入れている。しかし、知識表現の枠組みと現実の知識とのギャップはまだ大きい。しかもエキスパートシステムでは、現実の問題解決ができなければならないという要求があり、問題解決の速度やインタフェースまで考慮しなければならない。エキスパートシステム構築ツールも機能が強化されているが、汎用のツールがどこまで可能かどうかは興味ある問題である。

謝辞 原稿をまとめるにあたり、貴重な助言や資料をいただいた上野晴樹氏(電機大)、国藤進氏(富士通国際研)、平岡薫氏(富士通)、山口高平氏(大阪大)、中島秀之氏(電総研)に感謝いたします。

参考文献

- 1) Feigenbaum, E. A. et al. (ed.): 人工知能ハンドブック(訳) I~III, 共立出版(1983).
- 2) 中島秀之: 論理に基づく知識の表現, 情報処理, Vol. 26, No. 12, pp.1512-1519 (1985).
- 3) 石塚 満: あいまいな知識の利用と表現, 情報処理, Vol. 26, No. 12, pp.1481-1486 (1985).
- 4) 小林重信: プロダクションシステム, 情報処理, Vol. 26, No. 12, pp.1487-1496 (1985).
- 5) 小川 均: フレーム理論に基づく知識表現言語, 情報処理, Vol. 26, No. 12, pp.1497-1503 (1985).
- 6) 岡本敏雄: セマンティック・ネットワーク・システム, 情報処理, Vol. 26, No. 12, pp.1504-1511 (1985).
- 7) 伊藤昭治他: 医療診断エキスパートシステムにおける知識表現, 情報処理, Vol. 27, No. 7, pp.702-710 (1986).
- 8) 国藤 進他: 知識工学の基礎と応用, 計測と制御, Vol. 24, No. 6 (1985).

- 9) コンピュータ利用を変革する AI ツール, 日経コンピュータ (3.18. 1985).
- 10) 大須賀節雄他: 知識ベース入門, オーム社 (1986).
- 11) 上野晴樹: 知識工学入門, オーム社 (1985).
- 12) 中島秀之: 知識表現と Prolog/KR, 産業図書 (1985).
- 13) 昭和 60 年度新世代コンピュータに関する技術動向及び適用分野等の調査研究報告書, 日本機械工業連合会, 新世代コンピュータ技術開発機構 (1986).
- 14) Bobrow, D. et al. (ed.): 人工知能の基礎(訳), 近代科学社(1978).
- 15) エキスパートシステム (～その最新ツールと事例集～), 日本工業技術センター (1986).
- 16) 国分明男他: 意味ネットワークに基づく知識表現法とそのハードウェア化, 電総研調査報告 212 (1985).
- 17) 稲田 裕他: ESHELL を用いたスケジューリングシステムの開発, 情報処理学会, 第 32 回全国大会 (1986).
- 18) Begg, V.: エキスパート CAD システム(訳), 啓学出版 (1986).
- 19) エキスパートシステムに関する調査報告書, 日本電子工業振興協会 (1986).
- 20) 田中幸吉他: 知識工学, 朝倉書店 (1984).
- 21) 山口高平他: 設計と診断を融合したシェルの構成, 情報処理学会, 第 33 回全国大会 (1986).
- 22) Doyle, J.: A Truth Maintenance System, Artificial Intelligence, Vol. 12, No. 3, pp. 231-272 (1979).
- 23) Woods, W: What's Important About Knowledge Representation?, Computer, Vol. 16, No. 10, pp. 22-29 (1983).
- 24) Waterman, D. A.: A Guide to Expert Systems, Addison Wesley (1986).
- 25) Hayes-Roth, F. et al.: Building Expert Systems, Addison Wesley (1983).
- 26) Forgy, C. L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence, Vol. 19, No. 1, pp. 17-37 (1982).
- 27) Stefik, M. et al.: The Organization of Expert Systems, A Tutorial, Artificial Intelligence, Vol. 18, No. 2, pp. 135-173 (1982).
- 28) McSkimin, J. et al.: A Predicate Calculus Based Semantic Network for Deductive Searching, Associative Networks, Academic Press (1979).
- 29) Turner, R.: Logics for Artificial Intelligence. (昭和 61 年 10 月 8 日受付)