資料2

第5世代の電子計算機に関する調査研究報告書

—— 知識ベース・システム ——

昭和55年3月

JIPDEC

財団法人 日本情報処理開発協会

この報告書は，日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて昭和 54 年度に実施した「第 5 世代の電子計算機に関する調査研究」の成果をとりまとめたものであります。

知識ベース・システム執筆者

（順不同）

氏　　名	所　　属
古川康一	電子技術総合研究所ソフトウエア部 情報システム研究室主任研究官
白井良明	電子技術総合研究所パターン情報部 視覚情報研究室室長
山崎正人	電子技術総合研究所制御部 情報制御研究室主任研究官
田村浩一郎	電子技術総合研究所制御部 論理システム研究室室長
辻井潤一	京都大学工学部電気工学第2教室 助教授

目　　次

1．はじめに

1. はじめに

知識ベースは，人工知能の応用システムを開発していく過程で生れた概念で，データベースと同様に情報をためておく基地であるがその情報の質は，単なる個別的事実の集合にとどまらず，一般的な知識（概念）や，知識の利用に関する知識などを含み，人間の有する知識に近いものである。

情報システムを，人間と計算機の２つの成分から成るシステムと考えるとき，計算機サブシステムの機能をより高度化することが，情報科学の１つの大きな目的となっている。知識ベースを有する計算機サブシステムを考えると，それは，実世界のモデルの表現が可能で，実世界のモデルを含まない従来の計算機サブシステムからの大きな飛躍が期待される。実世界のモデルを有する計算機サブシステムの特長を列挙すると，つぎのようである。

1) 処理プログラムと人間サブシステム間のインタフェースとして，実世界のモデルが利用でき，人間－機械インタフェースが改善される。

2) 柔軟な情報システムが実現される。データベースにおけるデータ独立性，プログラミングにおける機械独立性，移殖性などが容易に実現できる。

3) モデルに照らした処理プログラムの検証，プログラムの改良，合成などが可能となる。

4) 仕様のわずかな変更に伴う処理プログラムの変更が容易になる。

5) プログラムの蓄積性がよくなり，大規模なソフトウェア・システムの漸進的な開発が可能となる。

このような計算機サブシステムを作成する際に最も大きな問題となるのは，個々の問題の処理プログラムの記述ではなく，ある問題領域での知識，すなわち実世界のモデルの記述である。これまでに，各種の知識表現システムが開発されてきたが，それらは，その能力に一長一短があり，それらの利用法も確立されているわけではない。

本サーベイでは，まず，これまでに開発された各種の知識表現システムについて概観する。つぎに，システムの漸進的な開発にとって必須となる新しい知識の獲得と，その既存体系への吸収，すなわち学習の問題について述べる。このような閉じていない知識ベースは，不完全な知識の扱いを含む。とくに，そこでは，単調でない推論が問題となる。それは，常識による推論のもつ1つの側面であるが，人間が通常行っているように，不完全な知識でも，システムは，常識的な答えを出すことができる場合はよくある。たとえば，通常，「鳥は飛べる」と考えていて大きな間違いはないが，だ鳥や，ペンギンはとべないので，「鳥は飛べる」というのは不完全な知識である。ここで大切なのは，常識による推論が誤りであると気がついた時，その誤りを適確に訂正できる機能である。最近のこの分野での大きな成果であるこのような機能を有するTruth Maintenance Systemについての概説を行う。このシステムは高度な知識ベースを実現する際に最も重要な機能の1つである仮説の設定とそれに基づく推論のメカニズムを容易に実現できる点が注目される。

知識ベースを含む計算機サブシステムは，知識工学と呼ばれる分野で，多くの実験システムの開発がなされ，そこで開発の方法論も合わせて研究が進められている。それらのシステムの特徴は高度な専門家の知識を計算機サブシステムの中で知識ベースとして実現している点である。現在，医療，科学・工学などの分野で，各種の応用システムの開発が試みられている。それらについての詳細は，各論にゆずるが，ここでは，計算機科学の分野での応用が，これから一層大切になることを指摘したい。そのうち，データベースの高度化については，本報告の第3章でのべる。その他に，プログラミングの分野での，そのような傾向は，ソフトウェア工学の報告書を参照されたい。また，今後は，超LSIの利用技術の開発が，これから大きな研究課題となることが予想されるが，そこでのCADシステムも知識ベース・システムとして実現されるであろう。さらに機械翻訳システムも同様である。要するに，今後開発が期待される大規模システムのほとんどのものは，それが高度な機能を必要とすればするほど，

知識ベース・システムとして実現される可能性が高い。

知識ベースとデータベースは，全く異なる分野で研究され，それぞれ発展してきたが，それらの情報システムにおける役割は類似しており，同一の理論の上で論じられることが必要である。関係データベースは，述語論理の枠組で形式化されており，それを基にして統合が可能である。データベースの研究の成果は，大量データの組織化法と，そこからのデータの検索法で，これを知識ベースに取り込むことが必要となってくる。第3章のデータベースの高度化は，このような視点からも論ずる。

最後に，知識ベースあるいは高度なデータベースのためのハードウェアについて考察を行う。知識ベースにおける主要な処理は演繹操作であるが，その基本は，記号のパターンの照合を基にした記号処理である。また，そこでは，問題解決能力が必要とされるが，そのために非決定手続の処理も必要となる。このような機能を有するハードウェアとしてLISPマシン，PROLOGマシン，ACTOR マシン，AMORDマシンなどが考えられる。これらのサーベイは，マシンの理論のサーベイにゆずる。本報告書では，データベース・マシン，とくに関係代数演算を直接実行するマシンについてのサーベイを行う。

2．知識の扱い

2. 知識の扱い

2.1 知識の表現

2.1.1 概説——歴史的背景

人工知能研究は，その名前が示すように，人間の知能を，すなわち人間の知的な行動を支配するメカニズムを解明することを目的としてきた。しかしながら「知能」が人間の知的な行動を支えるメカニズムを指すとすれば，そのメカニズムによって操作される実体，すなわち「知識」も人間の知的行動を支える大きな要素である。人工知能の研究がゲームのように極度に限定された世界で研究されていた間は，そこで必要になる知識の量も少なく，知識の内容も一様であった。そこでの研究目的は知識の内容よりも，知識を操作するメカニズムに重点が置かれていた。

しかしながら，人工知能の研究がこういった局限された世界から離れて，ロボットの行動計画〔Fikes 1971, Falhman 1974〕・自然言語の理解〔Winograd 1972, Schank 1975, Bobrow 1977b, Wilks 1978〕・人間の視覚といった，より現実的な対象を研究しはじめるにしたがって，人間の知的行動を支えるもう1つの要素，すなわち知識の研究の重要性が強く認識されるようになってきた。

知能と知識の関係を，通常の計算機処理とのアナロジーで考えると，知識はデータ，知能はプログラムに対応する。しかしながら，人間の情報処理機構においては，この両者の関係は通常の計算機処理におけるプログラムとデータほど明らかには区別できない。人間の情報処理機構においては，処理される対象と処理する機構とが重層的な階層性をもっており，1つの「知識」をとりあげてみても，それはデータ的な性質とプログラム的な性質の両方を兼ね備えていることが多い。知識の問題は，計算機科学において，プログラムとデータとい

う２つの概念をもう一度吟味しなおす機会を与えたことになった〔 Bobrow 1975, Winograd 1975〕。

このような，人間の知的活動に焦点をあてた研究で，より現実的な応用システムを目指すものに，Feigenbaum 等の主唱する知識工学の流れがある〔 Feigenbaum 1977 〕。知識工学においては，ある特定の分野の専門知識（医学，有機化学，プログラミング）を，いかにして工学的なシステムにまとめあげるかの方法論が研究されてきたが，ここでも問題は知識表現へと収束してゆくことになった。特に，専門家の知識は，必ずしもシステム設計時に完全な形で把握されているわけではなく，部分的に整理された知識が膨大に存在する。これを逐次追加・修正しながら，完全なシステムにまとめあげてゆくことになる。すなわち，この見方からすると，専門家の知識は，システムによって蓄積・管理される対象（データ）としての性質をもっている。一方，専門家の知識は，例えば医療診断システムを考えると，患者の検査データ等を処理し，診断を下すという働きを行うものであり，患者の検査データを処理するプログラムでもある。ここでも，専門家の知識はプログラムであると同時に，システムによって管理されるデータとしての性質を持っていることになる。

以上２つの流れは，人工知能研究から出発して，知識の問題へと至った流れであるが，これに対して，より計算機科学的な方向から，知識の問題を把えつつある流れがある。すなわち，高機能なデータベース・システムの研究と，プログラミング方法論からのアプローチである。データベース・システムは，当初従業員の人事ファイル等の比較的よく整理された膨大なデータを統合して管理することから出発したが，その出発の当初から，利用者から見たデータの把握の仕方，user's view をどのように表現しておくべきかについて活発な議論があった。データをあるがままに見れば，単なる数値であり，文字列にすぎないが，利用者はこれに対して自分なりの意味づけを与えている。この利用者によるデータの見方を，どのように表現しておき管理するかは，知識表現の問題ときわめて類似している。従来のデータベースに貯えられたデータが純粋な意

味でのデータであるとすると，利用者の見方はそれよりも一段レベルの高いものであり，データについての知識といってもよいものである。実際，従来のデータベースの上に，論理的な演繹機能等のより高度な機能をつけ加えていこうという試みが活発に行われ始めている〔Gallaire 1977, Ohsuga 1975〕。データベース中に貯えられた汎用性の高いデータと，利用者の個別プログラムという両端のギャップをうめるものとしての「知識」の取扱いが，今後のこの分野の中心的課題となってきている。

一方，プログラミングの方法論，あるいはプログラミング言語の研究においても，従来のコントロールの流れを表面に出したものから，より記述的・論理的な手法への傾向が強まっている。人工知能用言語PLANNERの設計者であるC. Hewitt は，さらにその方法論を押し進め，データとプログラムをActorと呼ぶ統一的な概念で把握しようとし〔Hewitt 1973〕，また，プログラム自体を論理的演繹過程として把えるプログラミング言語PROLOG の開発とそのデータベース・システムへの適用など，人工知能とプログラミング言語の設計という分野の相異を越えて，研究交流が活発化してきている。知識表現用言語として開発されたKRL〔Bobrow 1977a, 1979〕と，Abstract Data Type を直接言語仕様に組込んだＣＬＵ等のプログラミング言語〔Liskov 1977〕が，非常によく似た機能を持っていることは，単なる偶然ではないであろう。米国では，人工知能とプログラミング言語の研究者間の交流も積極的に行われており，SIGPLAN (special Interest Group on Programming Language) とSIGART(Special Interest Group on Artificial Intelligence)の合同会議等も開催されている〔SIGART 1977〕。

2.1.2. 「知識の表現」の枠組――データとプログラム

知識表現の問題を考える場合，まず「知識」とは何かを明らかにしておく必要がある。従来の計算機処理においては，人間の持つ知識はデータかプログラムかのいずれかの形で表現され，与えられていた。したがって，人間の持つ様

々な知識も，このいずれかの形式で表現できれば，ことさら「知識」という抽象的な用語をもち出すまでもなく，データの表現，プログラミングの方法論だけを考えれば良いことになる。

知識をデータで表現するのが良いか，プログラムで表現するのが良いか，というのが知識表現における「データとプログラム」の問題である。この問題の設定自体は，以下の議論からわかるように，ある表現形をデータとみるかプログラムとみるかは相対的なものであり，あまり意味のある問題設定ではない。しかしながら，この問題設定は知識とは何かを考えるうえで，有力な手掛を与えてくれる。

人間の知識の中には，「元素の原子量」のようにきわめてデータ的な性質の強いものから，「与えられたデータの平均値を求める方法」のようにプログラム的な性質の強いものまで，さまざまなレベルがある。したがって，各知識の性質に応じて，データとして表現したり，プログラムとして表現したりすればよいことになる。しかしながら，人間の知識の中には必ずしもこのようにデータかプログラムかを判然とは区別できないものが実は非常にたくさんある。この間の事情を少し考えてみよう。

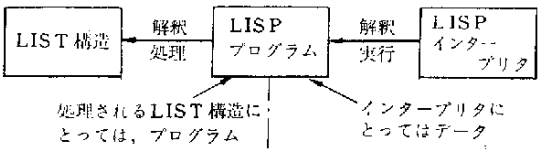


図 2.1　LISPシステムの構成

図 2.1 に，人工知能の分野でよく使われるプログラム言語LISPのシステムを模式的に示す。LISPの文法に従って書かれたプログラムは，その解釈し実行するLISPインタープリタにとってはデータである，と同時にそれは入力のリスト構造を解釈し処理するプログラムでもある，という二重性をもっている。

このように，ある表現をデータとみるか，プログラムとみるかは，それを解釈する側からその表現をとらえるか，その表現によって解釈される側からとらえるかによって変化する。

この階層性が，人間の知識の場合には複雑で入り組んだ構造になっている。すなわち，人間の知識体系の中には，「単なる事実」・「事実を操作するための知識」・「その知識を組合せるstrategy に関する知識」等々というように，さまざまなレベルの知識が混在して存在し，この階層性が無限の連鎖を形成している。したがって，同じ「知識」でも見方のレベルを変えると，データともプログラムとも考えることができることになる。

推論・演繹の問題から簡単な例をあげてこのことを考えてみよう。つぎの(A)・(B)２つの事実が与えられている。

「人間は，不死ではない」……(A)

「ソクラテスは人間である」……(B)

この(A)・(B)から，「ソクラテスは，不死ではない」を推論することを考えてみよう。

〔I〕 述語論理式

述語論理による知識の定式化は，(A)・(B)を同じレベルの知識であると考える立場である。述語論理では，この(A)，(B)２つの知識をつぎのような論理式で表現する。

$(\forall x)[HUMAN(x) \rightarrow MORTAL(x)]$ ……(A)'

$HUMAN(Socrates)$ ……(B)'

この(A)'，(B)'に対して，一階述語論理の推論手続を適用すれば，

$MORTAL(Socrates)$

が演繹できる。推論手続というのは，一階述語論理の推論規則を手続化したものが，(A)'・(B)'のような論理式の集合を与えられたときに，これをどのように操作できるかを記述したメタレベルの知識である。したがって，この推論手

続をプログラムとして実現すれば，論理式集合をデータとする推論システムができあがる。実際，この推論手続を計算機プログラムとして実現するための有効な手法がJ.A. Robinsonによって提案されて以来，論理式集合と定理証明プログラム（Theorem Prover）とを基本にしたシステムが多く作られている。SRIのロボットShakey の行動計画作成システムも，知識記述の基本部分にこの方式を使っている。

この種のシステムでは，(A)．(B)２つの知識をまったく同じレベルの知識とみており，論理式を操作するメカニズムは，定理証明プログラムだけに埋め込まれている。図2.2にこの種のシステムの概念図を示す。

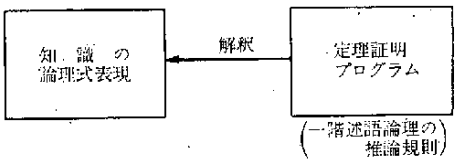


図 2.2　述語論理式と定理証明プログラム

〔Ⅱ〕 PLANNER　〔Hewitt 1972〕

(A)．(B)２つの知識の間にレベルの差を認めようとする立場がある。この立場の代表的なシステムとして，C. Hewitt によって提案された人工知能用の言語PLANNERがある。この立場では，(A)の「人間は不死ではない」という知識を，『「あるxが，不死ではない」ことを証明したければ，「そのxが人間である」というデータが存在しているかどうかを調べよ』という形で表現する。ここでは，(A)のような知識を(B)のような「単なる事実」とは区別して，「単なる事実」を操作するための，一段レベルの高い知識として把えている。「単なる事実」の側からみると，(A)のレベルの知識はプログラムの役割を果すことになる。PLANNERでは，知識を貯えるデータベースとして，２つのレベルをもっており，この２層の構造の上に，全体の実行を管理する

PLANNER－インタープリータが存在する。模式図を図2.3に示す。

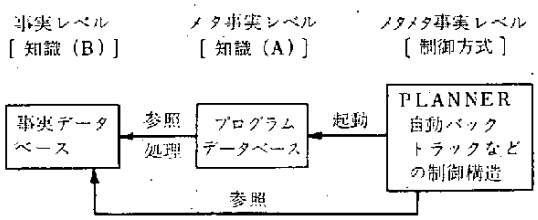


図2.3　PLANNERシステムの構成

PLANNERの特徴は，(A)のような知識がプログラムとしての性質とデータとしての性質の二重性をもつことを意識したことである。(A)のような知識のデータとしての特質は，PLANNER－インタープリータによって保障されている。すなわち，(A)のような知識を推論や問題解決の過程で適切に呼び出したり（pattern-directed function call）適用が失敗したときにうまく後戻りする（automatic backtracking）などの機能をPLANNER－インタープリータが分担しているために，すなわち，推論や問題解決や制御機構に関する知識がPLANNER－インタープリータに埋め込まれているために，(A)のレベルの知識はそれによって操作されるデータの性質をもつことになる。このPLANNERの考え方を，ほぼそのまま高機能データベースに適用した試みにChangのDEDUCE，DEDUCE Ⅱがある〔Chang 1976,1977〕。

〔Ⅲ〕 CONNIVER〔Sussman 1972〕

PLANNERは，人間の知識を計算機内に表現することへの手口を与えたという意味で大きな貢献をした。しかし，「単なる事実」・「プログラム的知識」・「PLANNERに埋め込まれた制御的知識」という3層の構造で十分かどうかについては議論がある。

特に，制御機構がすべてPLANNER－インタープリータに埋め込まれていることについての不満が多い。問題解決や推論のための制御機構に関する知

識もやはり対象分野に依存し，したがってシステム設計者によって記述されるべきだという主張から，PLANNERに代ってCONNIVERなどのコンテキスト機能を持った人工知能用言語が開発されている。

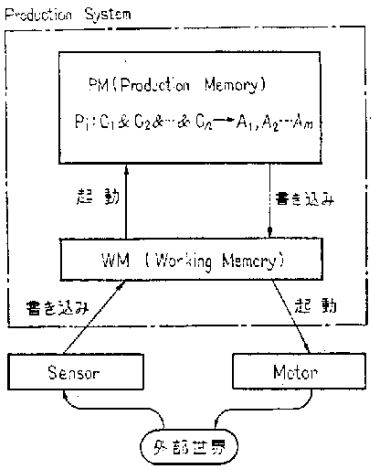
これらの新言語は，PLANNERの3層構造に代って，制御機構の記述を含めた4層の構造を主張しているといってもよい。CONNIVERの興味ある応用例に，ＭＩＴの「積木の世界」での行動プランニング・プログラムBUILDがある〔Fahlman，1974〕。

〔Ⅳ〕 プロダクション・システム

〔Davis 1975，1978，Rychener 1976,1977〕

データかプログラムかという問題に対して，（狭義の，Feigenbaum 流の）知識工学においては，Newell等が人間の記憶モデルとして提唱したプロダクション・システム〔Newell ,1973〕（以下，ＰＳと略す）を採用している。ＰＳでは記憶を，ルールの集合が貯えられている長期記憶（ＬＴＭ）と，処理の中間結果が置かれる短期記憶（ＳＴＭ）に分けて管理している。長期記憶中のルールがPLANNERの場合の定理型の知識に対応しており，システムによって管理されるデータとしての性質を持っている。図2.4にＰＳの基本構成，図2.5にＰＳを用いた医療診断システムMYCIN の構成図を示す〔Shortliffe 1976〕。

PLANNERとＰＳの大きな相異点は，前者がautomatic backtracking の機能を含めて，システム側（PLANNERーインタープリンタ）が，問題解決に関する制御構造のすべてを持っているのに対して，後者は，システム側に予め用意されている制御構造としては，簡単なconflict resolution の機構〔Rychener 1977，McDermott 1978〕だけで，制御構造に関する記述も，知識を記述する人間が，プロダクション・ルールの中に埋め込んでゆくことができることである。このことによって，ＰＳにおいては，PLANNERよりも柔軟な制御構造をとることが可能になっている。PLANNERのpattern



WM: Short Term Memory (STM)
ともいう
PM: Long Term Memory (LTM)
ともいう

図 2.4　Production Systemの基本構成

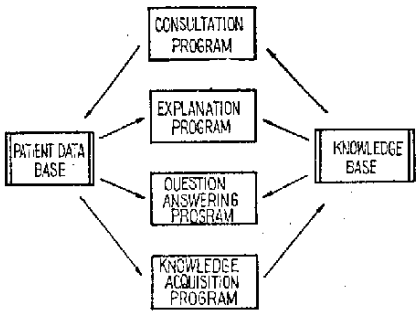


図 2.5　MYCINの構成

directed invocation の機能だけを実現したのがＰＳだと考えれば良い。ただし，ＰＳではプロセス間の依存関係等はほとんどシステム側では管理されておらず，co-routine 等の複雑にいり込んだ制御構造をとることはほとんど不可能である。Petri-Net等の考えを導入して，この欠点を解消しようと試みもある〔Zisman ,1978〕が，ＰＳの簡潔さを保持しながら，制御構造に関する記述をどのように導入してゆくかが今後の課題となろう。この課題は，2.2.2で述べるメタ知識の活用とも密接に関連しており，Davisらは，ＰＳにMeta-Ruleを導入することを提案している〔Davis　1977〕。また，PS的なアーキテクチャがこれまで成功してきたのは，広義の意味でのパターン認識的な問題であり，ロボットの行動プラン作成のような問題解決型のシステム，あるいは自然言語の構造解析のような問題に対して，ＰＳ的なアーキテクチャがどのように適用できるかについても今後の検討課題となろう。

2.1.3. 知識の呼び出し

〔1〕 プロダクション・システム

「知識の呼び出し」方の問題は，見方を変えれば知識相互の関係を，どの時点でとるかの問題である。もし知識相互間の関係が，プログラムを書く時点でわかっていれば，プログラムの中の必要な箇所でその知識を呼び出してやればよい（データとして表現された知識であれば，そのデータを参照する。プログラムとして表現された知識であれば，そのプログラムを呼び出す）。しかしながら，すべての知識の相互関係を事前に決定しておくことが可能でない場合も多い。

一般に，医療診断システムに代表されるKnowledge oriented なシステムにおいては，まず膨大な，あらかじめプログラムの形式では整理不可能な知識を対象としなければならない。例えば，MYCIN〔Shortliffe 1976〕は，患者についてのさまざまな検査結果と症状とから総合的に判断して，各患者に適合した抗生物質を決定するシステムであるが，この診断の際に必要となる典型的な知識は，次のような形式をしている。

「血液中の細菌が球状菌ならば，抗生物質A，B，C，Dが有効である」

「もし細菌が好気性があって，しかも血液中の白血球の数がある一定値以下であれば……」

このように，個々の知識は因果関係の連鎖として比較的整理されているが，これらの知識の量は膨大で，事前にすべての知識の相互関連をとらえることができない（ベテランの医師でも，自分の診断のプロセスをアルゴリズミックに人に説明することができないとか，数人の医師が同じ診断をしても，医師によって診断の根拠が違うといった報告がある。このことは，医師の診断のプロセスが事前にアルゴリズミックな決定木の形では表現できないことを示している）。また，知識個々には矛盾がなくても，全体としてみると，別々の知識が異なった診断に導くこともある。検査の精度や病状の認定も100％確実なものではない。

このような医療診断の世界を対象にして，MYCINのとった態度は，個々の知識は，あるまとまりとして記述し，個々の知識間の相互関連は診断が進むに従って，また診断の確かさに応じて適宜ダイナミックにとるという態度である。ある知識（病状と病因の因果関係etc）は，それが現在の診断に関連があると判断されると，システムによって自動的に呼び出され，ある（仮説的な）診断を下す。その診断は，さらにそれと関連がある（その診断を補強したり，否定したりする）と思われる知識の起動を促し，仮説的な診断をより確かな診断にしたり，仮説を却下したりする。このループを繰返すことによって，総合的にみて最も妥当な診断が下されることになる。

ある情報処理を行う場合に，従来のプログラミング手法では，まずその処理に含まれるコントロールの流れに注目する。この方向からのアプローチは，処理に本質的に含まれるコントロールの流れをそのままプログラム構造にひき移し，本質的ではなく単にプログラム作成の便宜のために導入される無意味なコントロールの移動（GO TO)は極力排けるべきだという考え方に発展し，構造化プログラミングの手法として定式化されてきている。しかしながらこの手法は，MYCINの場合のように，全体のコントロールの構造が不明確な分野，あるいはそれが事前にはとても決定できないような複雑な分野には，全く適用できない。ＰＳは，このような分野に対するプログラミングの手法である。すなわち，コントロールの流れを前面に出してプログラムを考える方法に対して，ＰＳはコントロールの流れを表面に出さずに，知識の単位に対応するルールの集合で処理の内容を記述してゆこうとする立場である。すなわち，通常のプログラミング手法においては，あらかじめ人間が知識相互間の関連を把握し，全体の処理の流れの中で，個々の知識の起動される場所を規定しておくのに対して，ＰＳでは起動されるべき時期がシステム（PSインタープリタ）によって実行時に決定されることになる。プログラミング手法になっているＰＳでの処理は，このため適用可能な知識単位（プロダクション・ルール）の選定を行うrecognizeフェーズと，選定された知識単位

の実行を行うactionフェーズを交互に繰返しながら進んでゆく。

Knowledge oriented なシステムになればなるほど，処理に関連する知識の量が大きくなり，プログラミングの時点ではとてもその全体をとらえきれないこと，また，次々に新しい知識がつけ加わってゆくことを考えると，実行時に知識相互間の関係を把握するという考え方は重要であろう。

〔Ⅱ〕 デモン

ＰＳの，実行時に知識相互の関連をとるという考え方，あるいは，処理の流れをプログラミングの時点ではすべて決定しておかないという考え方は，人工知能研究においてこれまでにもいろいろな形で提案されている。Charniackの提唱したデモンも，この流れの中で把えることができる〔Charniak 1972〕。デモンは，一種のソフトウェアによる割り込みであり，処理途中である条件が生じると自動的に呼び出され，実行されるソフトウェア・モジュールである。また，デモンは予め用意されているものだけでなく，他のソフトウェア・モジュールによって実行時に作り出されることもある。Charniak のデモン自身は，規定が曖昧で，デモンの「起動される条件」をどのように記述するか等に任意性がある。この点で，Charniak のデモンは，彼の自然言語理解システムに強く依存した形式で実現されているように思える。

PLANNERにおけるAntecedent 型の定理やErase型の定理は，事実データベースの操作を監視しているデモンであると考えることができる。また，次項で述べる知識記述用言語ＫＲＬ，ＦＲＬにみられる，スロットに対する操作によって起動されるプログラム群もまた，デモンの一種である。

〔Rieger 1978〕は，この種のデモン機構に代表されるプログラムの呼び出しを，従来のサブルーティンの呼び出し（Demand Computation）と対比して，Spontaneous Computationと総称し，人工知能における典型的な呼び出し機構であるとしている。

2.1.4. データ構造と知識表現

2.1.1 項では，データとプログラムの関係について考えたが，本項ではデータ構造ということについて考える。

前項でも述べたように，取扱う知識の量が膨大になると，関連のある知識を選択する方法が大きな問題になる。蓄積された知識を一応すべて適用してみる方式は，ほとんど不可能である。知識を入力する時点で，「どういう状況でその知識が有効となるか」を示す情報も含めて入力する必要がある。この方式をとったのが 2.1.1 項で述べたＰＳやPLANNER, CONNIVERであった。

しかしながら，この方式も膨大な知識を対象にしたり，より緊密な知識間の関連を表現するといった場合には問題がある。人間の場合には，ある知識の単位が想起されると，それ以外の知識の中で，現在想起されている知識と深く関連している知識・やや関連している知識・まったく関連のない知識などが区別されて，関連の深い知識単位はつぎのステップでは他の知識単位に比べると，はるかに容易に取出すことができるといったメカニズムをもっている。これによって潜在的には膨大な知識を記憶していながら，ある時点で想起可能な状態になっている知識は，そのごく一部であるということになる。この連想的な能力が，人間の情報処理能力の基盤になっている。

計算機における知識の記述においても，このような知識単位のある種のまとまりを記述する必要がある。R.Quillian の semantic network〔Quillian 1968，1969〕は人間の連想的な能力を，「関連する概念間にリンクをつける」という方式で実現したものである。データ構造によって人間の連想的能力を模擬するこの方式は，その後も名前をかえてひきつがれており，たとえば D。Bobrow，T.Winograd などのＫＲＬ〔Bobrow 1977a〕，I.Goldstein のＦＲＬ〔Goldsteim 1977〕なども連想的なデータ構造を中心にして，これにプログラムやデータを含めたさまざまな知識を埋め込んでゆくという方式をとっている。

〔I〕 ＦＲＬ，ＫＲＬ

ＦＲＬの記述例を，図2.6に示す。

この図からわかるように，ＦＲＬにおける記述は，次の2つの機構に支えられている。

(1) property inheritance

(2) embedded procedures in slots.

```
(FASSERT     MIT-AI
   (STREET   ($VALUE ( /545 Technology Square/ )))
   (CITY     ($VALUE ( /Cambridge/ )))
   (STATE    ($VALUE ( /MA/ )))
   (ZIP      ($VALUE ( /02139/)))     )

(FASSERT     MINSKY
   (AKO      ($VALUE ( MIT-AI )))
   (OFFICE   ($VALUE (  821   )))     )
```

図 2.6　ＦＲＬの記述例

(1)は，概念間の階層構造をもとにした操作である。すなわち「Minsky はＭＩＴの A.I. Lab. の研究者である」という事実をデータ構造で保持し，「ＭＩＴのA.I. Lab. の研究者」に関して定義された事実・プログラムは，すべてMinsky にも適用できることをＦＲＬのシステム側が管理してくれる。もちろん，同種の情報がMinsky に対しても直接定義されておれば，その情報が優先する。

(2)の機構は，ＫＲＬにもある。ＫＲＬでの模式的な記述例を図2.7に示す。

```
DAY  Frame
     YEAR
     MONTH
     DAY-NUMBER        WHEN-FILLED
                       (CHECK-RELATION  day-number month)
     DAY-OF-WEEK       TO-FILL
                       (APPLY  calendar-lookup
                          TO  year month day-number)
                       (APPLY  anchor-date-Method
                          TO  year month day-number)
     ASCII-FORM        WHEN-FILLED
                       (FILL  year month day-number)
```

図 2.7　ＫＲＬの記述例

ここでは，DAYというデータ・タイプ(KRLの用語ではUNIT)をKRLで定義したものである。ここで，DAY−NUMBER のスロットに具体値(instance 例えば31日)を代入すると，このスロットにWHEN−FILLED procedure として定義されている，日数と月の関係をチェックするprocedure(1つのslot に複数のprocedure が定義されていてもよい)が自動的に呼び出されて実行される。

FRL,KRLの記述は基本的には，object oriented あるいは concept orientedであり，個々のobjectについて知識を記述してゆくという方式であり，(1)・(2)の機構によってobject間の静的な関係を把えてゆこうというものである。

これに対して，PS, PLANNER,CONNIVER PROLOG 等は，event-orientedな記述を基本にしており，この種の object 間の関係をデータ構造として表現する機構はない。

〔Ⅱ〕 述語論理

述語論理式も，object間の関係を示す表現を述語に選び，この述語を基本にして論理式を組み立てることを基本としているため，広い意味でevent-orientedな知識記述である。これに対して，大須賀は一変数述語(HUMAN，MALE等)が基本的にはobjectについての記述であることに注目し，述語論理におけるこの部分の記述だけをFRL等にみられる階層構造に整理し，解答導出の過程で関連する論理式を検索する際のINDEXとして使用することを提案している〔Ohsuga 1975〕。大須賀は，この体系を高階のmany sorted logicで基礎づけることを考えており，FRL・KRL 等の人工知能用言語の論理的基礎づけを与える興味深い試みである。many sorted logic を導入しようとする試みは，〔Sandewall 1972〕にも見られる。

〔Ⅲ〕 Semantic Network

人間の連想機能をデータ構造によって実現しようとするQuillian の semantic network は，その後さまざまな角度から検討されている。とくに，〔Schubert 1976〕はλ一記法を含む高階の論理体系によって，また〔Hendrix 1975〕は，一階の述語論理によって，それぞれ semantic network の論理的意味づけを行っている。Schubert の試みは，非常に興味深いものであるが，そのネットワーク図式による表現を論理式へ変形する操作を定義したにとどまり，ネットワーク図式を用いた演繹操作等は全く未定義の状態である。同様な欠陥は，Hendrix の Partitioned Networkにもみられ，論理式の代りに semantic network を採用する利点をどこに求めるかを今後明らかにしてゆく必要があろう。

論理式を出発点として，それをネットワーク図式で表現しようとするこれらの試みとは別に，従来のネットワーク図式を出発点として，その論理的な意味を考えようとする試みが〔Woods 1975〕，〔Brachman 1976〕〔Nagao 1979〕によって行われている。

2.1.5. 対象に依存した処理の取扱い

これまでは，主として対象とする分野やそこでの処理には依存しない，知識記述の枠組について議論してきた。しかしながら，有機化合物の構造推定を行うDENDRALにおいては，有機化合物の構造を何らかの形式で表現し，それに対して，例えば，２つの構造式を結合する，あるいは，構造式中の一部の化学結合を切断する，等の処理をほどこさなければならない。一般的な枠組の中で，こういった分野に依存した処理をどのように組込んでゆくかが問題となる。

〔Ⅰ〕 プロダクション・システム

DENDRALでのプロダクション・ルールの例を図 2.8 に示す〔Feigenbaum 1971〕。

```
ESTROGEN      (BREAK   (14  15)(13  17))
              (HTRANS   +1  +2)
```

図 2.8　DENDRALのルール

図中，ESTROGENは，特定の化学構造式（ボンド・グラフという一種のネットワークで表現されている）を指しており，現在処理中の構造式中に，このESTROGENを示す構造が埋め込まれていれば，PSのrecognizeフェーズで認識され，次のactionフェーズでこのルールの右辺が実行される。右辺のaction記述も，BREAK，HTRANS 等のボンド・グラフに対する操作によって表現されている。このように，分野依存的な操作や処理を直接PSの仕様の中に組み込んでゆく手法は，MYCINによっても採用されている。

上述のDENDRAL・MYCIN の方法は，分野依存的な処理や操作が一般的な枠組の中に直接埋め込まれてしまっており，全体的なシステムとしては汎用的なものとはなっていない（MYCINの一般的な部分だけを抜き出して，E-MYCIN Essential MYCINが作られている）。PSにおいて，より汎用的な枠組を目指す方向としては，PSを人工知能用プログラミング言語とみなすRychener等のPsnlystの試みがある〔Rychener 1976〕。Psnlystにおいては，recognizeフェーズではパターン・マッチングを一般的手法として採用している。明らかに単純なパターン・マッチングだけでrecognizeフェーズにおけるすべての分野依存的な処理を記述するのは不可能な場合があるし，可能であっても不経済な処理となる。このため，Psnlystにおいては，パターン記述中に評価可能なpredicateを記述することを許している。actionに関しても，同様である。

〔Ⅱ〕 論理式によるシステム

対象に依存した処理の取扱いに関しては，partitioned network〔Hendrix 1975〕，大須賀のシステム〔Ohsuga 1975〕，PROLOG

〔Kowalski 1974〕等は，論理式を一般的な枠組として採用していることから，ほぼ同じように考えることができる。

大須賀のシステムでは，基本的にはデータベースへの知的アクセスを目指しているために，推論・演繹を行う一般的なcomponentと外部データベースへのアクセスを行うcomponentとのインターフェースを行う必要があり，大須賀は外部データベースによってその述語を満足する具体値が保持されている述語を，評価可能な述語とみなして推論する方式をとっている。

〔Ⅲ〕 PLANNER，CONNIVER，FRL，KRL等のプログラミング言語による方法

以上2つの方式と比べると，プログラミング言語を設計し，それによって知識記述を行うシステムにおいては，対象に依存する処理を組み込むことははるかに容易である。逆にこの種のシステムでは，プログラムを組むことによってかなりad-hocな，対象依存的な処理を組み込むことができるため，システム全体としてどのような論理的基盤を持っているのか，どの程度の問題を解くことができるのか等が明確にとらえられないという欠陥がある。極言すると，これらのプログラミング言語を使って書かれた対象依存的な処理の中から，一般的に使える手法を抽象し，これをプログラミング言語の仕様に反映することがこのアプローチでの課題である。前二者が，一般的な枠組をまず設定し，これに対象依存的な処理を埋め込もうとしているのとは逆の方向である。

PLANNER，CONNIVER，FRL，KRLと並べてみると，後ろにゆくほど，それまでの対象依存的なシステム作成の結果得られた，あれば便利であると考えられる機能が，一般化された形式でプログラミング言語の仕様にとり込まれている。しかしながら，現時点ではあまりに性急に多くの機能を一般化し，言語仕様の中にとり込みすぎているという批判がＫＲＬにはあること〔Lehnert 1979〕またＫＲＬのインタプリータ自体が巨大になりすぎてか

えって使いにくい等，問題はかなりありそうである。ある意味で，ＫＲＬはＰＬ／Ｉと同じで，便利と思われる機能をそのまま言語仕様にとり込みすぎており，もう少し理論的な整備を行ってから，必要な機能だけをうまく言語仕様に反映するという努力が必要なのかもしれない。

2.2. 知識の活用

2.2.1 論理式と演繹

知識の表現と，その活用方式の関係が最も直接的な形であらわれるのが，論理式と，それを用いて演繹を行う場合である。古典的には，一階の述語論理式と，これをもとに演繹を行う導出原理〔Robinson 1965〕がある。その後，この導出原理は，効率上の問題から種々の改良が加えられ，線型導出，semantic resolution，unit resolution等の改良型の戦略が考案されてきた〔Chang 1973〕。

しかしながら，この種の改良型戦略は結局取扱う論理式の形（例えば，unit resolutionの場合には，1つのリテラルからなるクローズを導出の対象にするetc）だけに注目して行うもので，人間が演繹を行う際の，「現在の，この定理の証明には，この公理とこの公理が関係する」といったような，より高次の知識を使うことは，大きくかけはなれていた。この種の，論理式で表現された知識よりも一段上位の知識の使用が必要であることは，HewittによるPLANNERの設計の際にも強調され，PLANNERでは証明に関連する公理を，recommendation listとして予め人間が与える方式がとられた。recommendation listは，人間が定理型の知識をPLANNERの基本関数であるTHGOAL を使って，プログラムの形で記述してゆくときに，同時に埋め込んでゆく方式がとられた（図2.9参照）。ただし，一階述語論理の導出原理にくらべて，PLANNERの効率が良いのは，このrecommendation list による関連知識の選択という機能にあるのではなく，もっと別の原因にあったことが，最近の研究では明らかになってきている

〔Reiter 1978〕。

```
(THGOAL (WIN POKER HAND)
        (THTBF BLUFF DRAWAFEW CHEAT))
```

THGOAL：PLANNERの基本関数
THTBF：recommendation list

図2.9　recommendation listの例

PLANNERは，推論・演繹の機能というものをcomputationの過程として把えようという試みであったが，これと対局的な立場に立つのが，PROLOGである。PROLOGにおいては，computationというものが演繹の立場から把えられる。PROLOG自体は，PLANNERのパターン・マッチングの機能を，導出原理におけるunificationに対応させ，すべてのcomputationの規則を述語論理式によって記述する。

PROLOGの式（プログラム）で記述されたものが，知識であるとすると，前述のPLANNERの場合にrecommendation listとして表現されたmeta一知識をどのように表現するかが問題になる。PROLOG自体は，論理式の形式をHorn clauseに限定しているために一般の導出原理に比べて効率が良いことは確かであるが，このままでは，導出原理の場合と同様に，meta知識の導入は不可能であり，その分効率が低下することは避けられない。

〔GALLAIRE 1979〕は，この種のmeta一知識をPROLOGにおいて取扱う機構として，

1. 2階述語を導入（システムとして予め用意されたevaluableな2階述語を入れる ── 2階述語についての公理を入れることはしない）し，導出過程中の任意のクローズや項の参照を許す。
2. 演繹過程中で，指定された公理の追加，削除を動的に行う。
3. backtrackingの禁止を指定できる。
4. PROLOG表現中に，resolveすべきクローズや，その中でのリテラル

を指定することを許す。

等の機構を用意している。

このような枠組は，もちろん ad hoc なものにならざるをえず，PROLOGの本来もっていた論理性を失う危険性があるが，〔GALLAIRE 1979〕は，それが実際のシステム作成においては非常に有効であったこと，また，論理式表現の任意の個所に，この種のmeta一知識の記述ができ，PLANNERのrecommendation list といったような付加的な枠組を必要としないことの利点を強調している。すなわち，PLANNERのrecommendation list方式は，定理型知識及び事実データベース中の事実を諮意的に人間が知識記述時に分類し，この分類にもとづいて関連知識を指定する方式であったのに対して，このPLOLOGによる方式においては，論理式の形式を直接参照し，ある条件を満足するものが実行時に選択される方式になっており，一歩進んだ方式となっている〔次節のReference by name と Reference by description の項参照〕。

この他，論理式上での演繹機能をより効率よく行う方式としては，与えられた論理式集合に対して，予めresolve可能な節対を連結グラフとして表現しておき，この上でのグラフ探索を行って演繹のプランを作り，unificationはこのプランにもとづいて複数の節対に対して同時に行うといった〔Chang 1979〕の方式が提案されている。

2.2.2 Meta一知識の活用

前項では，PLANNER・PROLOGを例として，論理式で表現された知識のレベルと，その知識をいかに活用して，効率よく処理や推論を行うかについてのもう一段レベルの高い知識とがあることを指摘した。ここでは，〔Davis 1977〕に従って，この『「知識の活用の仕方」についての知識』をどのように取扱うかについて考えてみることにする。

Davisは，2.1.2で述べた「知識の呼び出し」について，従来の方式を整理し，次の2つの枠組があることを指摘した。すなわち，

(1) 外部 handle による呼び出し (reference by name)

(2) 知識の内容による呼び出し (reference by description)

通常のプログラミングにおいてもっともよく見られるサブルーティンの名前による呼び出しは，典型的な(1)の方式になっている。すなわち，サブルーティンの名前は，そのサブルーティンの実行する仕事とは全く無関係に，プログラマーが諮意的に選ぶことができ，他のプログラム中から，その名前を指定することによって呼び出すことになる。したがって，サブルーティン名というのは，そのサブルーティンにつけられた外部 handleとして機能している。

これに対して，ＳＲＩのロボット行動計画作成システムSTRIPS は，ロボットの基本行動を規定する各作用素について，図２.１０のように，その作用素を適用した後に，状態記述中に新らたにつけ加わる論理式と，消去されるべき論理式とをそれぞれADD−LIST，DELETE−LIST の部分に記述している。STRIPSは，このADD−LIST，DELETE−LIST を参照し，与えられた目標を達成するために必要となる論理式をADD−LIST 中に含む作用素を次に適用することになる。作用素の選定プロセスを知識呼び出しのプロセスであると考えると，STRIPS の方式は，呼び出しの際に知識の内容を参照して行っていることになり，典型的な(2)の方式になっている。

サブルーティン名による呼び出しは，人工知能研究の分野においてはその後 pattern directed procedure invocation の形で一般化される。すなわち，これまで名前という表現力のとぼしい外部 handleによって行っていたのが，pattern という，より

```
PUSH (ob, x, y)
Precondition
  (∃r) [INROOM (ROBOT, r)∧INROOM (ob, r)
    ∧LOCINROOM (x, y, r)]∧PUSHABLE (ob)
delete-list
  AT (ROBOT, $1, $2)
  NEXTTO (ROBOT, $1)
  AT (ob, $1, $2)
  NEXTTO (ob, $1)
  NEXTTO ($1, ob)
add-list
  AT(ob, x, y)
  NEXTTO (ROBOT, ob)
```

図 2.10　STRIPS における作用素の表現

表現力の高い外部handleによる呼び出しに置き換えることになる。

pattern directed procedure invocation は，また(2)の reference by description の方式にも使われる。例えば，ＰＳにおいては，各プロダクション規則の左辺のうち，その左辺（一般に変数を含むパターンによって記述されている）が，必要な規則を選択する際の invocation pattern の役割を果している。この場合には，「左辺→右辺」という知識記述の一部である左辺が同時に invocation のキーにもなっている。したがって，パターンによる関連知識の呼び出しは，外部handleとして使われる場合と，reference by desription として使われる２つの場合があることになる。

PLANNERの記述は，一方において reference by descriptionの方式であるパターン・マッチングによる goal-oriented な procedure invocation を持ち，他方において，recommendation list を使った外部handleによる関連知識の指定という両面を備えている。では，果して，この２つの記述方式だけで十分なのだろうか。Davis は，関連知識の呼び出しを記述する枠組の妥当性を判断する基準として，次の２つの評価基準を設定している。すなわち，

(1) validity

(2) expressiveness

validity というのは，外部handleあるいは reference by description において使われる description が，それによって呼び出される知識内容を忠実に反映しているかどうかという評価基準である。この評価基準からすると，後者の方式は，知識の内容の一部を呼び出しの手掛りとしているのであるから，問題はない。一方，前者による関連知識の呼び出しにおいては，知識内容とは無関係に諮意的につけられた外部handle（名前またパターン）を手掛りとしているため，この validity の保障は，知識記述者に完全に依存することになる。

一方，expressiveness は，どこまで豊富な内容を reference 機構に組み込むことができるかという評価基準である。一階述語論理式と，それを用いた導出原理においては，論理式で表現された知識は，各々その論理式が含むリテラ

ルによってインデックスづけされており，演繹過程において必要な論理式は，このインデックスを参照することによって選択される。この意味では，各論理式の呼び出しは完全にreference by descriptionの方式に従っており，validityという評価基準は満足している。しかしながら，この導出原理にもとづき，論理式の形（すなわち，知識記述の内容）にだけ注目して効率を上げようとする定理証明機構の改良戦略は，ほとんどすべて抜本的な効率の向上にはつながらなかったという意味で失敗しており，前項のPROLOG の項で述べたように，「知識の活用」に関する知識，すなわちmeta-知識の取扱いが問題となっている。言い換えると，通常の論理式で表現される知識をobject レベルの知識と呼ぶことにすると，object レベルでの知識内容だけでは，expressivenessの評価基準を満足できないことになる。

外部handleとして使用されるパターンは，従来のサブルーティン名というとぼしい表現力しか持たなかった枠組を，一段進めたものである。この方向の試みは，知識内容とは一応独立した，meta-知識の記述（いつそのobject レベルの知識が役に立つかという記述）を外部handleとして与えようという試みであったと総括できる。

meta-知識としての外部handleの記述（すなわち，expressivenessを豊かにしようとする記述）と，object レベルでの知識の記述を呼び出しの手掛りとしようとする試み（すなわち，validityの保障された知識の呼び出し）の両者の利点を合わせたものとして，Davisは彼のＰＳにもとづくKnowledge Acquifitionシステム（ＴIERESIAS)において，通常のプロダクション規則をデータとして，それに対して働くMeta-規則という通常のプロダクション規則よりも一段レベルが上の記述を与えることを試みている〔Davis　1978〕。

meta-規則の例を図2.11に示す。この例からわかるように，meta-規則の左辺においては，通常のＰＳにおけるＳＴＭに入れられるデータに対する条件（図2.11の条件1と2）の他に，通常のプロダクション規則の内容を調べる条件（条件3）も記述することができる。すなわち，通常のプロダクション規

```
METARULE001
If   1) you are attempting to determine the best stock to invest
        in ,
     2) the client's tax status is non-profit ,
     3) there are rules which mention in their premise the
        income-tax bracket of the client ,
then it is very likely (0.9) that each of these rules is not
  going to be useful.
PREMISE
 ($AND (SAME OBJCT CURGOAL STOCK-NAME)
        (SAME OBJCT STATUS NON-PROFIT)
        (THEREARE OLRULES ($AND
          (MENTIONS FREEUAR PREMISE BRACKET))
                                              SET1))
ACTION (CONCLUDE SET1 UTILITY NO 0.9)
```

図 2.11　Meta－規則の例

則を，その内容によって呼び出す機能（ reference by description ）と，それ以外に問題解決過程の状況を参照して適切な規則を呼び出す機能の両者が，meta−規則の形式で表現されることになる。

現在，このmeta−規則が有効に働くほど，プロダクション規則の数が膨大になるシステムがなく，その有効性は確かめられていない。しかしながら，PROLOGに対するmeta−知識の記述と同様に『「知識の活用」方法に関する知識』をシステムに対して与えようとする興味深い試みであろう。今後，高機能のデータベース等，大量のデータを対象とする実用システムが開発されるにつれ，このようなmeta−知識の記述は，ますますその重要性が認識されてゆくことになろう〔 Bundy 1979 〕。

2.2.3. 不完全な知識の活用

数学等の整備された体系における知識は別にして，普通人間の持つ知識は，対象が現実的なものになればなるほど不完全なものとなってくる。したがって，knowledge oriented なシステムにおいては，不完全な知識を有効に活用する処理や演繹の取扱いが常に大きな問題となる。数学的な意味での論理的演繹のように，論理式（知識）が不十分な場合においては結果が出てこないとか，知識間に矛盾があった場合には，あらゆる定理が真となるといった機構では，現実的な応用システムに知識を導入することはできない。このような不完全な知識をどのように活用してゆくかに関しては，〔Moore 1975〕，〔Doyle 1979〕等が論じている。

不完全な知識を対象とする場合には，知識ベースや事実ベースに明示的には宣言されていない事柄について知る必要があった場合，標準的な仮定（default assumption）を行って，以下の処理を進めてゆくのが普通である（例えば，旅行プラン作成の会話システム GUS〔Bobrow 1977b〕においては，対話の相手である人間が明示的に出発地を指定しない場合，dafault assumption として出発地を Palo Alto とし，以下の旅行プランを作成してゆく）。この default assumption の機能は，不完全な知識を取扱う際の基本的なものになっているが，この時問題となるのは「default assumption を行って，それにもとづいて処理をすすめた場合，もし将来の時点で矛盾が生じた場合に，どのようにしてその矛盾を生じる原因となった default assumption を検出し，それを解除するか」ということである。

これに対するもっとも標準的な解法は，backtracking である。すなわち，それまでに行われた default assumption を１つ１つ，矛盾がなくなるまで解除してゆく方法である。これまでの backtracking による方法では，時間的に近くで行われた assumption から順に解除してゆくのが一般的であるが，この場合には，矛盾を生じせしめた原因とは全く無関係な assumption までも，たまたまそれが時間的に後に行われたという理由だけから解除されてゆくことに

なり，そのassumptionから導かれた有用な結論までも，次々に失われてゆくことになる。これに対して，〔Doyle 1979〕のTruth Maintenance System（TMS）では，'dependency directed backtracking' を提唱している。すなわち，時間的な順序には無関係に，矛盾を導くことになった結論，その結論を導くことになった前提というように，論理的な関係を逆方向にたどってゆき，矛盾の原因となったdefault assumptionを検出する。検出した後は，そのassumptionにもとづいて行われた推論結果だけをすべて解除すれば良いことになる。推論の各過程において，導びかれた結論とそれを導びくもととなった前提とを，システムが常に保持しており，これを参照することによってbacktracking を行うことになる（PLANNERにおいてdecision pointを時間的順序にスタックしてゆき，backtracking する際には，このスタックの上から順に解除してゆくのとは対照的である）。

Doyleは，以上のような機能を，問題解決システムを推論部と推論根処管理部に分離し，常にいかなる推論をどのような根拠により行ってきたかを管理することにより実現した。'dependency directed backtracking'は，問題の解法について，システムが不完全な知識しか持っていない場合の効率の良い処理方式になっている。その意味で，扱う対象の知識が不完全な場合の処理に対する基礎理論であるnon monotonic logic〔McDermott 1979〕と並んで，この新しいbacktracking 技法は，今後種々の知識表現の枠組について適用されてゆくべきものであろう。

参 考 文 献

[Bobrow 1975] Bobrow, D.G., 'Dimension of Representation,' in Representation and Understanding (Eds. Bobrow and Collins), Academic Press

[Bobrow 1977a] ——, and Winograd, T., 'An Overview of KRL,' in Cognitive Science, Vol. 1

[Bobrow 1977b] —— et al., 'GUS, A Frame-Driven Dialog System,' Artificial Intelligence, Vol. 8

[Bobrow 1979] , and Winograd, 'KRL: Another Perspective,' Cognitive Science, 3

[Brachman 1976] Brachman, R.J., 'What's in a Concept,' in Proc. of COLING 76, Ottawa

[Bundy 1979] Bundy, A., et al., 'Solving Mechanics Problems Using Meta-level Inference,' in Proc. of 6th IJCAI, Tokyo

[Chang 1973] Chang, C.L. and Lee, R.C., 'Symbolic Logic and Mechanical Theorem Proving,' Academic Press

[Chang 1976] Chang, C.L., 'DEDUCE -- A Deductive Query for Relational Data Base,' in Pattern Recognition and Artificial Intelligence (Ed. Chen), Academic Press

[Chang 1977] , 'Further Investigations of Deduction in Relational Data Bases,' in Logic and Data Bases (Eds. Gallaire and Minker), Plenum Press

[Chang 1979] , 'Resolution Plans in Theorem Proving,' in Proc. of 6th IJCAI, Tokyo

[Charniak 1972] Charniak, E., 'Towards a Model of Children's Story Comprehension,' Ph. D. Thesis, MIT

[Davis 1975] Davis, R., 'An Overview of Production System,' Stanford AI memo, AIM-271

[Davis 1977] , 'Generalized Procedure Calling and Content-Directed Invocation,' in Proc. of Sym. on Artificial Intelligence and Programming Language, SIGART Newsletter, No. 64

[Davis 1978] , 'Knowledge Acquisition in Rule-Based Systems,' in Pattern-Directed Inference System (Eds. Waterman and Hayes-Roth), Academic Press

[Doyle 1979] Doyle, J., 'A Glimpse of Truth Maintenance,' in Proc. of 6th IJCAI, Tokyo

[Fahlman 1974] Fahlman, S.E., 'A Planning System for Robot Construction Tasks,' Artificial Intelligence, Vol. 5

[Feigenbaum 1971] Feigenbaum, E.A. et al. 'On Generality and Problem Solving: A Case Study Using DENDRAL Program,' Machine Intelligence, Vol. 6, Edinburg Univ. Press

[Feigenbaum 1977] Feigenbaum, E.A., 'The Art of Artificial Intelligence: I Themes and Case Studies of Knowledge Engineering,' in Proc. of 5th IJCAI, Cambridge

[Fikes 1971] Fikes, R.E. and Nilsson, N.J., 'STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving,' Artificial Intelligence, Vol. 2

[Gallaire 1977] Gallaire, H. and Minker, J., 'Logic and Data Bases,' Plenum Press

[Gallaire 1979] Gallaire, H., 'Issues in Controlling a Deduction Process in a Declarative Mode,' in Proc. of ETL Seminar, Tokyo

[Goldstein 1977] Goldstein, I., 'The FRL Manual,' MIT AI Memo 409, MIT

[Hendrix 1975] Hendrix, G.G., 'Partitioned Network for the Mathematical Modeling of Natural Language Semantics,' Ph. D. Thesis, Univ. of Texas

[Hewitt 1972] Hewitt, C., 'Description and Theoretical Analysis of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot,' MIT AI Memo, 251, MIT

[Hewitt 1973] Hewitt, C., 'A Universal Modular ACTOR Formalism,' in Proc. of 3rd IJCAI, Stanford

[Kowalski 1974] Kowalski, R., 'Predicate Logic as Programming Language,' Proc. IFlP 74

[Lehnert 1977] Lehnert, W. and Wilks, Y., 'A Critical Perspective on KRL,' Cognitive Science, Vol. 3

[Liskov 1977] Liskov, B., 'Abstraction Mechanism in CLV,' C. ACM, Vol. 20, No. 8

[Minsky 1975] Minsky, M., 'A Framework for Representing Knowledge,' in Psychology of Computer Vision, McGraw Hill

[McDermott 1978] McDermott, J. and Forgy, C.,' Production System Conflict Resolution Strategy,' in Pattern-Directed Inference Systems (Eds. Waterman and Hayes-Roth), Academic Press

[McDermott 1979] McDermott, D. and Doyle, J., 'An Introduction to Non-Monotonic Logic,' in Proc. of 6th IJCAI, Tokyo

[Moore 1975] Moore, R.C., 'Reasoning from Incomplete Knowledge in a Procedural Deduction System,' MIT AI memo, 347

[Nagao 1979] Nagao, M. and Tsujii, J., 'S-Net: A Formalism for Knowledge Representation Languages,' Proc. of 6th IJCAI, Tokyo

[Newell 1973] Newell, A., 'Production Systems: Models of Control Structures,' in Visual Information Processing (Ed. Chase), Academic Press

[Newell 1977] Newell, A., 'Harpy, Production Systems and Human Cognition,' Department of Computer Science, CMU

[Ohsuga 1975] Ohsuga, S., 'A Knowledge Structure and a Deductive Inference Rule Based on It,' in Proc. of 2nd UJCC

[Ohsuga 1979]_____, 'Toward Intelligence Interactive Systems,' in Proc. of IFIP WG 5.2, Workshop Seillac II on Methodology of Interaction, North-Holland

[Quillian 1968] Quillian, M.R., 'Semantic Memory,' in Semantic Information Processing (Ed. Minsky), MIT Press

[Quillian 1969]_____, 'The Teachable Language Comprehender,' C. ACM, Vol. 12

[Rieger 1978] Rieger, C., 'Spontaneous Computation and its Role in AI Modeling,' in Pattern-Directed Inference Systems (Eds. Waterman and Hayes-Roth), Academic Press

[Reiter 1978] Reiter, R., 'On Closed World Data Bases,' in Logic and Data Bases (Eds. Gallaire and Minker), Plenum Press

[Robinson 1965] Robinson, J.A., 'A Machine Oriented Logic Based on the Resolution Principle,' J. ACM, Vol. 12

[Rychener 1976] Rychener, M., 'Production Systems as a Programming Language for Artificial Intelligence Applications,' Ph. D. Thesis, CMU

[Rychener 1977]_____, 'Control Requirements for the Design of Production System Architectures,' SIGART, No. 64

[Sandewall 1972] Sandewall, E., 'An Approach to The Frame Problem and its Implementation,' Machine Intelligence 7, John Wiley and Sons

[Shortliffe 1976] Shortliffe, E.H., 'Computer Based Medical Consultation: MYCIN,' American Elsevier

[SIGART 1977] SIGART Newsletter, 64, Proc. of 'Symposium on Artificial Intelligence and Programming Language'

[Schank 1975] Schank, R.C., 'Conceptual Information Processing,' North-Holland

[Schubert 1976] Schubert, L.K., "Extending the Expressive Power of Semantic Networks,' Artificial Intelligence, Vol. 7

[Sussman 1972] Sussman, G.J. and McDermott, D., 'From PLANNER to CONNIVER — A Genetic Approach,' FJCC 41

[Sussman 1977] Sussman, G.J. and Stallman, R.M., 'Forward Reasoning and Dependency Directed Backtracking in a System for Computer Aided Circuit Analysis,' Artificial Intelligence, Vol. 9

[Winograd 1972] Winograd, T., 'Understanding Natural Language,' Academic Press

[Winograd 1975]_____, 'Frame Representations and Declarative - Procedural Controversy,' in Representation and Understanding (Eds. Bobrow and Collins), Academic Press

[Wilks 1978] Wilks, Y., 'Making Preference More Active,' Artificial Intelligence, Vol. 11

[Woods 1975] Woods, W.A., 'What's in a Link,' in Representation and Understanding (Eds. Bobrow and Collins), Academic Press

[Zisman 1978] Zisman, M.D., 'Use of Production Systems for Modeling Asynchronous, Concurrent Process,' in Pattern-Directed Inference Systems (Eds. Waterman and Hayes-Roth), Academic Press

2.3 知識の取得・学習

知識ベースシステムにおいては，知識の取得（ acquisition ），組織化（ organization ），適用（ application）が基本の３本柱になっている。このうち取得機能は，知識の入力をあつかう部分であり，外部（人間あるいは外界）から知識を単純に取込む場合と，内部の既存の知識を用いて（場合によっては教師の手助けを借りて）新たに知識を生み出す場合とがある。前者を単純取得，後者を学習と呼ぶことにしよう。

単純取得の場合，カードリーダや端末装置から単純にコードによって入力され，ほとんどの処理を経ずにこれまた単純に格納されるような場合と，広義のパターン認識（文字，図形，物体，音声，自然言語などの認識）機能を用いる場合とがある。コンピュータでは，知識はプログラムおよびデータであるから，現在のほとんどのコンピュータでは，前者のやり方をしており，後者は特殊な場合になっている。

知識の取得機構自体がそれ自身知識ベースであるから，取得に関する知識の取得，組織化，適用が考えられる。この再帰を進めると際限ないが，行きつく所は，記憶レジスタに還元される。レジスタは入力と出力を持つプロセスである。Milne & Milner（１９７９）では，プロセスの数学的モデルが見事に与えられている。つまり，プロセスの集合をＰ，入力の集合をＩＮ，出力の集合をＯＵＴとすれば，

$$P = OUT \times (IN \rightarrow P) \qquad (1)$$

である。（＝は同形の意）。この集合方程式の解がプロセス集合である。つまり，プロセスとは，出力と，入力からプロセスへの関数の対で与えられる。さらにプロセスは何種類かの入出力ポートを持ち，各集合が分類（ソート）されているものとする。レジスタももちろんこのモデルでうまく表現でき，ある入力Ｚがはいれば，自分自身が別なプロセスに変わり，それ以降，出力に対して

はZを出すプロセスになる。知識ベースシステムはこのレジスタモデルが複雑になったものとみなせば良い。つまり入力（知識の取得）があれば自分自身のふるまい（プロセス）がそれに応じて変化する。このように，知識の取得は，単純取得，学習に拘わらず，記憶およびそれによるシステムの変化に密接に関係している。

よりマクロ的な（心理学的な）見方をすると，ピアジェの用語を借りれば，知識の取得には，同化（assimilation）と消化（accomodation）がある。同化は，システム内にすでに存在する知識のスキーマに合わせて知識（データ）を取込む。パターン認識がこれに相当する。消化は，新しい知識を，システム内にすでに組織化された知識群に追加する機能であり，システム内の記憶の構造を変える。これらを(1)式のモデルで解釈すれば次のようになる。同化というのは，関数 $f \in [IN \rightarrow P]$ がIN の部分集合に対して同一であることを示す。この部分集合はパターン認識でのクラスと呼ばれるものに対応する。一方消化の方は，入力 $Z \in IN$ によって，$IN \rightarrow P$ におけるP内のプロセスPが以前のPと変わることに該当する。このように見ると，これまであまり明確でなかった同化と消化の概念をより明確な形で定式化できる可能性がある。

同化自体は単純取得であり，学習的な機能はないが，同化の方法を学習することがある。これはパターン認識において，サンプル集合によって認識のパラメータを調節するものに相当する。これも単純ながら学習であり，現にパターン認識関係者はこれを「学習」と称している。これに比して，消化は単純取得でありながら学習的な色彩が強い。これには，プログラムをプログラムライブラリーに単純に追加する単純なものから，既存の知識ベースと一貫性を保ちながら，その構造を変えて新知識を消化するようなより知的なものまで，様々なレベルがある。後者のような知的なものの研究はまだ少ない。これを行なうためには，簡明でしかも意味規則が明確な知識表現形式が必須であろう。

同化や消化は外部から知識が与えられる単純取得であるのに対して，学習は内部から新しい知識を生み出す。消化は内部の知識を修正，更新するので，こ

の意味で学習に近い。

学習には，パースの用語を借りれば，帰納（ induction ）と推測（ abduction） とがある。帰納は，三段論法における小前提（事実）と結論とから大前提（規則）を見出す推論である。前述した同化学習もこれに近いが，しかし現在のパターン認識技術における学習ではパラメータ学習が主体であり，規則のような構造的なものは生み出さない。人間の場合，パターンを学習するのは，恐らく規則を見出すのが主で，パラメータ学習は，あるとしてもわずかであろう。こういったパターンの構造自体の学習はWiston が試みている。しかしまだ本格的に規則を学習する試みは少ない。その中で，メタDENDRALによって，質量分析に関する規則を見出し，その規則自体，化学界で貴重な発見となった例が成功したもののひとつである。また，Lenatは整数論に関して，「興味深い」定理を生み出すためのシステムを作成した。この場合は単なる帰納の域を超えている。

一方，推測（ abduction ）は，大前提（規則）と結論から，小前提（事実）を見出すものであり，物理学におけるモデル作りの作業に似ている。パースのいう推測は単純であるが，モデル作りの本質である。しかし実際に意義深いモデルを作るためには推測だけではなく様々な知的機能を必要とする。こういう学習はＡＩ研究のひとつの夢であろう。

こういった学習も(1)式のプロセスモデルで解釈可能である。ただし〔ＩＮ→Ｐ〕の関数集合が担当複雑な構成になる。

この他，いわゆる「学習」と呼ばれるものの中で，刺激（入力）と反応（出力）の対による強化学習がある。これは心理学ではスキナーの理論，生理学では，細胞間の結合の重みの変化の理論を土台として，工学的にはパーセプトロンなどが考えられた。この種の学習は１９５０年代から６０年代にかけて盛んに研究されたが，構造自体の変化という視点が欠けていたために研究の行詰まりを招いた。人間の学習（ＣＡＩ）およびコンピュータ自体の学習に関しても，最近はピアジェの考え方が注目されている。

この他，Samuel のチェッカーの学習プログラムがあるが，これも基本的にはパラメータ学習である。これはそれなりに成功し，プログラム作成者自身よりも強いプログラムになったが，本格的により強いものになるためには，やはり構造を変革する機構，つまり，規則を自分自身で改良したり，さらには新たに発見したりするような機構が必要であろう。この場合もやはり簡明でかつ表現能力の高い知識表現形式が最大のキイポイントになる。このように，消化や学習の立場から本当の意味でのプログラム自動合成が達成されることになる。

本稿の用語でいう単純取得と学習との大きな差異のひとつに，行動変容の基準の有無がある。学習の場合，ある基準を最適化するように行動変容するのに対して，単純取得は，単に外界から知識を取込むだけで，特に基準が指定されない。学習の場合でも，教師付きの場合には基準は教師側だけが持っていて，学習者側では持たないこともある。いずれにせよ，(1)式のプロセスモデルで学習モデルを作るためには，基準の最適化をはかるプロセスと学習者の行動プロセスの相互作用を考える必要がある。教師付きの場合には，これにさらに教師プロセスが加えられる。

迷路を最短距離で脱出するようなネズミの“学習”のような場合はもちろんのこと，帰納や推測にも基準がある。フィードバックによって，適用範囲がより広くかつ簡潔な規則や事実が求められるからである。

以上をまとめると，知識の取得は次のようになる。

- 取得
  - 単純取得
    - 同化
    - 消化
  - 学習
    - パラメータ学習
    - 帰納
    - 推測

このように見てくると，知識の組織化や適用に比較して，取得に関する研究

は立遅れている。この3者は，知識表現を軸にして相互に密接に関連し合っており，逆に，これまでのように，組織化と適用だけに注目して知識の表現形式を考るのではなく，取得までも視野に入れて，より秀れた表現形式を追求することが重要である。

2.4. 知　識　工　学

2.4.1　知識工学の誕生

計算機を用いた情報処理は社会の各部に侵透してきたが，その処理の内容はあまり大きな変化がない。つまり人間が行ってきた単純な作業を計算機に代行させることである。例えば給与計算，銀行のオンライン処理などは，いくつかの簡単な法則に基いて計算を行っているにすぎない。そこに用いられている法則を知ることも容易で，とくに熟練者でなくても短期間にマスターできる。

そのほかデータベースを用いて情報を検索するような場合を考えてみる。通常はデータベース内の情報を索引によって取出したり，キーワードによって関連する情報を抽出する。その場合でも計算機は簡単な処理をしているだけで，ユーザーが望みの情報を得ることができるか否かは，ユーザーの与えるキーワードの組合せによって決まる，このように計算機は大量のデータを扱ったり，高速に処理するという目的に使われ，作業内容は簡単な処理のくり返しである。

いっぽう人間の知能に近い機能を実現しようとする試みもある。その一つに人工知能がある。人工知能の研究には2つの立場がある。つまり知能を人工的に作る立場と，人間の知能を分析してそのモデルを作ることである。大部分は第一の立場をとっているが，ここでも同じ立場をとる。その中にもいくつものアプローチがある。定理を自動的に証明したり，計算機のプログラムを自動作成するための基礎理論を追及する研究と，一定の分野を限ってその分野の問題を解決するシステムを作る応用的な研究は対象的な2つの立場である。知識工学が実用化したのは当然後者の研究である。

知識工学実現には，多くの知識を持つシステムが必要であり，力学におけるニュートンの法則のように，少数の法則で多くを説明できるものは存在しないであろう。たとえば病気の診断という分野を取上げてみる。医者と同等な知識を持つシステムは，多数の個別な知識を持っていなければならない。その知識は簡単な式で記述できないものが多い。また医者に個性があるように，知識の範囲も明確ではない。このような分野の知識をどのようにして表現して計算機に入れるかという研究が知識工学の中心課題である。たとえ長期間かかって知識を蓄えたとしても，それは人間のように消滅することなく，新しい知識を加えていくことによって進化させることができる。このようにして限られた分野ではあるが，名医に近い知識を持つシステムを作ることもできよう。

なお「知識工学」は，第５回人工知能国際会議（ＩＪＣＡＩ－７７）の招待講演でスタンフォード大学のファイゲンバウムが提唱した。[1)]

2.4.2. 知識工学の代表例　DENDRAL

知識工学が実用化された例はまだ少ない。その中でDENDRALとよばれるシステムが代表的な例である。[1),2)]このシステムを作るプロジェクトは，１９６５年にスタンフォード大学の計算機科学科のファイゲンバウム，ブキャナン等のグループと，同大学の質量分析研究のグループによって共同で着手された。目的は質量分析器と核磁気共鳴器の出力データから，有機物質の構造式を決定することである。図2.1.2と図2.1.3にデータの一部を例示する。

DENDRALは３つのモジュールで構成される。

第一のモジュールは構造式の生成を行う。例えば，$C_nH_{(2n+2)}O$ で表わされる物質で，ｎ＝４の場合には７種類の可能な構造がある。ｎが大きくなると，可能な構造の種類は急増し，ｎ＝10では約１０００種類になる。構造の生成には化学の知識だけで十分であり，この部分はすでに化学の分野では定式化されている。

第二のモジュールは与えられた構造から，質量分析スペクトルデータを予測

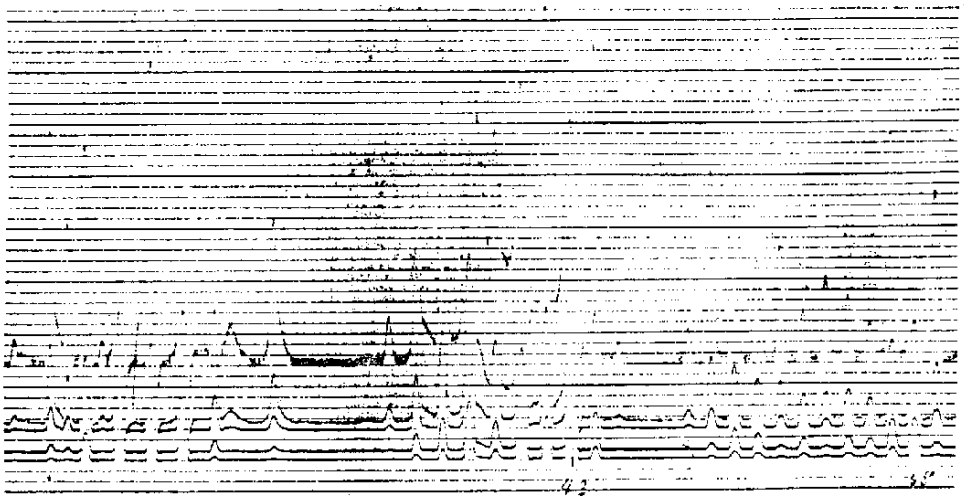


図 2.1 2　質量分析スペクトルデータの例
（ n－ブタンの一部 ）

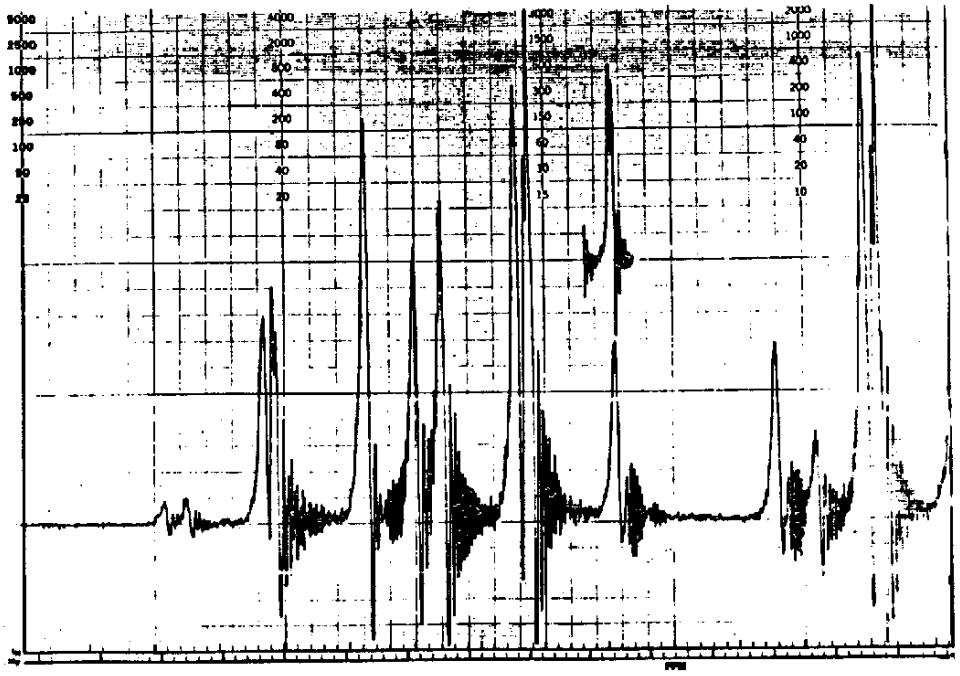


図 2.1 3　核磁気共鳴スペクトルデータの例
（ ジクロベンゼンの一部 ）

し，実際に得られたデータと比較する。もしうまく合致すれば，その構造式を持つ物質であることが結論される。もし効率を考えなければこの２つのモジュールで十分である。つまり，はじめに質量分析スペクトルから物質の化学式を推定し，その化学式を満足する可能な構造をすべて生成し，予測されるスペクトルとデータを比較して最も合致するものを選べばよい。ところが$C_6H_{13}NO_2$は可能な構造が約１万もあり，この方式では能率が悪い。

そこで第三のモジュールが必要になる。それは可能な構造の生成の計画を立てる。この部分が人間の熟練者の知識である。つまり質量分析スペクトルを眺めて，物質の核となる構造（基）の一部を推定するのである。例えばどことどこにスペクトルのピークがあり，その大きさがある関係を持てば物質はケトン類であるという推定を行う。もしケトン類であることがわかれば，可能な構造の種類は少なくなる。その少ない候補の中から，安定な構造から順にチエックしていけば，試行錯誤の回数は大幅に削減される。

基の存在を予測するためには質量分析スペクトルだけでは十分でないため，核磁気共鳴スペクトルも併用し，かなり広範囲な物質まで効率よく推定できるようになった。DENDRALはスペクトルデータから構造式の推定という分野では人間の熟練者とほぼ同じ能力を持つといわれている。

DENDRALは現在もスタンフォード大学の質量分析研究所で化学の専門家によって使用されている。他の研究者も計算機ネットワークを通して使うこともできる。英国ではこのシステムをエジンバラ大学に移転するプロジェクトが始まっている。

2.4.3 応　用　分　野

ここでは，現在実用化されているシステム，実用化をはかっているもの，あるいは研究が進められている分野などを述べる。

(1) 数式の記号処理

数式が与えられたとき，そこに含まれる変数に数値を与えて数値計算によ

って答を出すことは計算機の得意な分野である。数式を記号のまま処理する試みは，1961年MITのSlagelが行った。それは大学の教養で習う程度の積分を行うプログラムで，「Symbolic Automatic INTegration」を約してSAINTとよばれた。[3] たとえば，

$$\int \frac{x^4}{(1-x^2)^{5/2}}\,dx$$

を入力として与えれば

$$\arcsin x+\frac{1}{3}\tan^3 \arcsin x-\tan\arcsin x$$

という答を約11分の後に得ることができた。

その後実用になる数式処理システムの作成が各所で研究された。1969年からMITのプロジェクトMACで着手したシステムは，「MAC's SYmbolic MAnipulator」を約してMACSYMAと呼ばれた。[4] そこには数学的知識が取込まれて現在まで発展し続け，数式の積分，微分，簡単化，因数分解，極限などを求めることができる。その能力は大学の学生以上で現在，数学者，物理学者によって使用されている。

数式処理では記号処理に適した言語LISPを用いることが多い。通常の計算機では，リスト処理やデータ型の検査に時間がかかる。また，桁数の多い整数を扱わなければならない。そこで数式処理に適した専用プロセッサを作る試みもある。[5]

(2) 医療のコンサルタント

スタンフォード大学の計算機学科と医学部が協力して，血液感染と脳膜炎に関する診断システムMYCIN[6]を作った。これは，問診によって得られる情報や，血液の中の細菌の培養によって得られる情報を入力として，その患者の病名と，治療に用いる薬を出力する。

システムは，医学知識をつぎのような法則として持っている。

もし「Aであり，Bであり，……，X」であるならば，「Yである確からしさはPである」

これはプロダクション・ルール[7]とよばれている。図2.14に法則の例を示す。

```
RULE037
    IF: 1) THE IDENTITY OF THE ORGANISM IS NOT KNOWN
           WITH CERTAINTY, AND
        2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
        3) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
        4) THE AEROBICITY OF THE ORGANISM IS AEROBIC
  THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE(.8)
        THAT THE CLASS OF THE ORGANISM IS
        ENTEROBACTERIACEAE
RULE060
    IF: THE IDENTITY OF THE ORGANISM IS BACTEROIDES
  THEN: I RECOMMEND THERAPY CHOSEN FROM AMONG THE
        FOLLOWING DRUGS:
        1-CLINDAMYCIN          (.99)
        2-CHLORAMPHENICOL      (.99)
        3-ERYTHROMYCIN         (.57)
        4-TETRACYCLINE         (.28)
        5-CARBENICILLIN        (.27)
```

図2.14　MYCINのプロダクション・ルールの例

前者は培養した菌に関する法則で，後者は治療のために与える薬に関する法則でもある。診断のための推論は，いわゆる逆向きである。すなわち，「Yとなるためには，AとBと……，Xを満足しなければならない；Aであるためには，A′，B′，……，でなければならない」というように目標から前提条件に向かって推論を進めていく。

すべての場合を探索するのは効率が悪いため，確からしさを導入し，ある事象の確からしさが一定より下がればそれは否定されたものとみなして探索を打切っている。しかし，いくつもの法則を組合せて導いた事象の確からし

さをいかにして決定したらよいかは問題である。

MYCIN 以外にも，Rutger 大学の緑内障の診断と治療用のCASNET[8]，ＭＩＴの腎疾患用システムPIP[9]，日本では心不全用システムMECS-AI[10]などが作られている。なお知識工学の医療への応用に関しては文献１１が詳しい。

(3) 有機化合物の構造決定

豊橋技科大の佐々木らによる，CHEMICSおよびCHEMICS-F は専門的なシステムとして成果をあげている。[12,13]

CHEMICSは核磁気共鳴（NMR），質量分析（MS），紫外線，赤外線分光（UV．IR）データと分子式を入力とし，これから可能な分子構造を出力する。DENDRALがＭＳのデータを主に扱かっているのに対し，これはC[13]-NMR を中心に他のUV．IRやＭＳも積極的に使用した。より総合的なシステムと思われる。

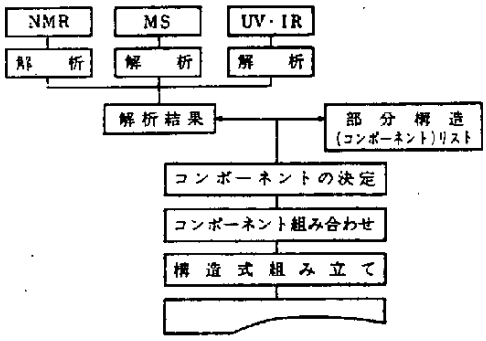


図 2.1 5　CHEMICS 概念図

CHEMICSの処理の流れを図 2.１５に示す。まず，これらスペクトルと部分構造に関する化学的な知識とから存在しそうな部分構造の候補を抽出する。約２００種のC１３-NMR のスペクトルと部分構造の対応法則は，このグループの長年の結果を蓄積したものであり，このシステムの特長である。UV．

ＩＲスペクトルからは，カルボニル基，水酸基，エーテル性酸素，炭素－炭素多重結合の存否，不飽和性の情報が得られる。ＭＳからは，有機質量分析の経験則にもとづき，存在すべき，またはあってはならない骨格構造の情報が得られる。

つぎに，得られた部分構造から，トポロジカルに可能なすべての構造を出力する。この部分はDENDRALとほとんど同じである。ただ，CHEMICSの方が入力情報が多いだけにユーザーのアドバイスに頼る度合いが少ないと思われる。

得られた候補の構造の数を絞るため，DENDRALでは各候補についてマススペクトルを合成し，入力スペクトルとよく一致するものだけを結果として出力している。一方，CHEMICSの場合には，CHEMICS－F になって主要な化合物についての特徴的なスペクトルをファイルに蓄積しておき，候補構造のスペクトルの特徴をファイルから取り出し，これと入力スペクトルを比較することにより候補の数を絞る方式がとられた。

CHEMICSには，現在約４５０万の化合物の構造を含むCA－RSF (Chemical Abstract Registry Structure File) と結合して更に強力なシステムにしようとする計画もある。[14)]

(4) その他の種々の研究

スタンフォード大学のヒューリスティック・プログラミング・プロジェクトでは，知識工学の研究を大規模に行っている。その主なものをつぎに示す。

PUFF [1)]: 人間の呼吸の状態をモニターし，肺機能障害の診断を下す。100の症例に基づいて検査データと診断の関係を表わす法則を抽出し，別の１５０例で性能テストを行い，人間と９０％の一致をみている。

MOLGEN [1)]: 分子遺伝学の実験計画を立てる。生物学，遺伝学，化学，トポロジーおよび装置に関する知識を持ち，与えられた目標を達成するための実験計画を立てたり，それを改良したりする。

CRYSALTS[15]：　白質のX線回析像から，蛋白質の3次元構造を指定する。回析像も3次元濃淡画像であるため，入力データの記述も課題である。

SACON[16]：構造解析用プログラムの使い方のコンサルタントを行う。構造物の形や材料，負荷から，各部の変形や応力を求めるためのプログラム・パッケージがあるが，その使い方は簡単でない。そこで素人でも使えるように，質問応答によって助言をする。

スタンフォード大学以外でも研究されている。それ等の中にはつぎのものがある。

PROSPECTOR[17]：ＳＲＩで開発されているシステムで，鉱物資源に関して地質学者に助言を与えることが目的。調査データと銅山の銅埋蔵量の関係を表わす法則を見つけ，専門家の判断と比較した。

Adviser[18]：ＭＩＴのコンサルタント・システム。数式処理用プログラムMACSYMAの使い方を教える。

ＩＣの設計等：ＭＩＴの人工知能研究所では，電気回路に関する知識工学の研究を行ってきた。まず，電気回路の解析を行うシステムを作り[19,20]それから故障を見つけて直す研究を行った。[21]現在はＬＳＩ回路の設計のコンサルタント・システムを開発中である。[22]

2.4.4. X線光電子分光（ESCA）による分析

電総研の情報制御研究室と高温電子材料研究室は協力して，ESCAによる組成分析システムを開発した。[23,24]

組成分析は化学分析の基本であり，

- 重量分析
- ガスクロマトグラフィー
- 発光分光分析
- 原子明光分析

◦ X線けい光分析

◦ 放射化分析

◦ 質量分析

など，それぞれ特長を持った多くの方法がある。ESCAの場合は，定量性は劣るが，全元素を対象とし，極微量（ppbオーダー）の分析ができるのが特色である。

ESCAのデータには，このほか，

① 化学シフト（ピーク位置が分子構造によってわずかに動く）を利用した分子構造の解析

② 半導体や固体物性の研究に対して重要な，

◦ フェルミレベル

◦ 電子の状態密度

③ 反磁性物質と内核ピークの形の関係

④ 固体中の電子の平均自由行程

など，多くの情報が含まれている。このシステムは②の分野への拡張が検討されている。

(1) ESCAの概要

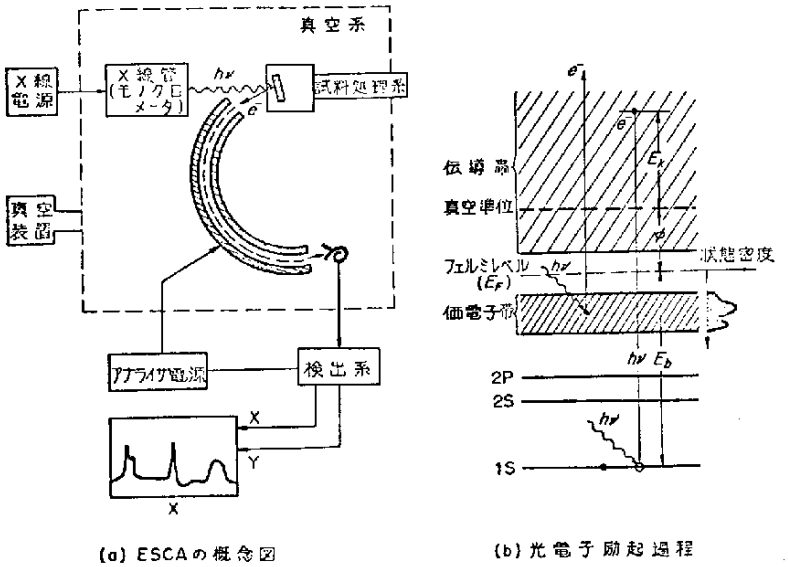
図2.16にESCAの原理を示す。

ESCAは高真空中におかれた試料に単色化されたX線を照射し，その結果光電効果によって飛び出してくる電子をエネルギー分析することによって，試料中の電子のエネルギー準位と密度を測定する。

飛び出してくる電子の運動エネルギーEk，その電子の結合エネルギーEb，X線のエネルギー$h\nu$，装置の仕事関数ϕとの間には，次の関係がある。

$$Eb = h\nu - Ek - \phi$$

図のアナライザにかかる電圧を走査することにより，検出器に到達する電子の運動エネルギー，したがって結合エネルギーが走査され，スペクトルが得

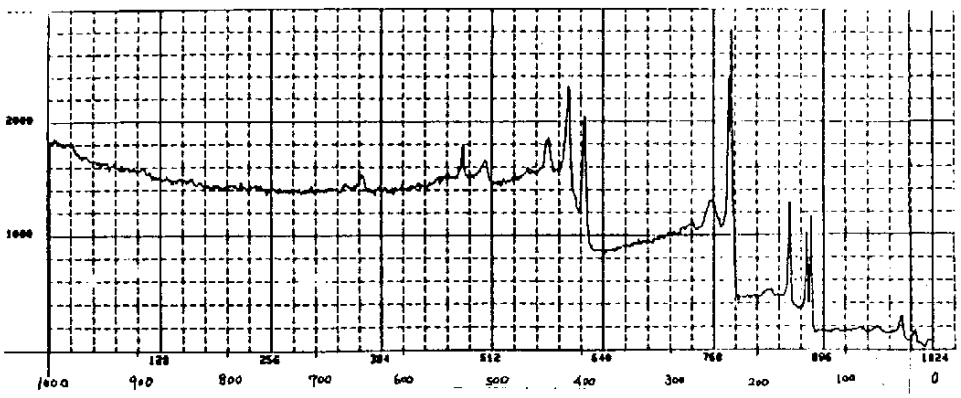


(a) ESCAの概念図　　(b) 光電子励起過程

図2.16　ESCAの原理

られる。

図2.17はESCAスペクトルの例である。鋭いピークの左右に生ずる大きな段差と，ピークの左側に生ずる非弾性散乱によるなだらかなピークは理論的解明も充分でなく，ESCAのスペクトルの解析を難しくしている。



総合エネルギーEb（ev）

図2.17　ESCAスペクトルの例

(2) 分析システム

組成分析システムは3つのステップよりなる。

第1のステップでは，スペクトルの全ピークに対して解釈の候補，すなわち，可能な元素名と電子軌道名を与える。ここでは，ピークの面積は問題にしないので，上にのべた問題をさけるため，まず簡単な低域カットの炉波を行なう。この結果は図2.18のように段差やなだらかなピークが除かれる。

図2.18　前処理後のスペクトル

つぎに，各ピークの位置と，システムが持っている元素と電子軌道に対応する結合エネルギー値の知識（図2.19参照）を比較することにより解釈の候

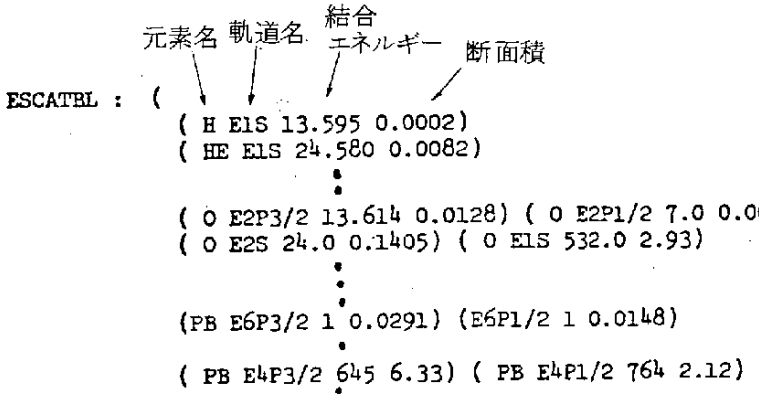


図2.19　知識の一部

補が求められる。ここでは，ノイズにうもれがちな小さなピークをのがさないためと照合の効率を上げるために知識主導形の方法がとられている。すなわち，ある元素が候補としてあげられると，その元素の全ピークの存在すべき位置が知識から求められるので，せまい範囲に限ってピークの探索を行なうことができる。照合の幅は，化学シフトを見込むためあまり小さくできない。このためこのステップで得られる解釈の候補は1ピーク当たり2～3個になる場合が多い。

第2のステップでは，解釈の候補を，次の法則により評価し確らしいものにしぼる。

① 同一元素内では，化学シフト値は電子軌道によらずほぼ一定である。

② 同一元素内では，ピークの面積は図2.19に示した断面積値にほぼ比例する。

解釈の候補と元素ごとにまとめ，2つの法則を反映する評価関数CFによる評価を行なって信頼すべき解釈に到達する。CFはシフトを考慮に入れたピーク位置にあるべきピークの存否を期待される強度でウエイト付けしたものを使用し，多くのデータにより充分な識別能力が確認されている。また，図2.20に示すような大きなピークの肩に乗った，いわゆるショルダーピークを指摘できることもこのシステムの特長である。

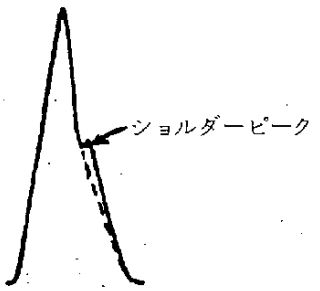


図2.20 ショルダーピーク

第3のステップでは，上で求められた元素について，ピーク面積を原スペクトルから計算する。これを断面積で割った値の比として組成比が求められる。

(3) 結　果

図2.17，図2.18に示したPb Mo_6S_8のスペクトルの各ピークに対する解釈は，図2.21のように微量の酸素を含めて正しく求められ組成式

```
CF = 0.93
((565 857 163.1) ( S E2p3/2 165 1.11))      - g
((565 857 163.1) ( S E2p1/2 165 0.567))     - g
CF = 0.86
((158 479 532.2) ( O E1s 532.0 2.93))       - h
CF = 0.93
((88 987 36.1) ( MO E4p3/2 35 0.86))        - f
((88 987 36.1) ( MO E4p1/2 35 0.445))       - f
((1003 789 229.5) ( MO E3d5/2 227 5.62))    - a
((491 786 232.4) ( MO E3d3/2 230 3.88))     - b
((302 620 394.5) ( MO E3p3/2 392 5.94))     - c
((162 602 412.1) ( MO E3p1/2 410 3.04))     - d
((53 505 506.8) ( MO E3s 505 2.34))         - e
CF = 0.675
((61 1004 19.5) ( PB E5d5/2 19 1.58))       - i
((32 1002 21.5) ( PB E5d3/2 22 1.11))       - j
((643 882 138.6) ( PB E4f7/2 138 12.73))    - k
((432 877 143.5) ( PB E4f5/2 143 10.01))    - l
((162 602 412.1) ( PB E4d5/2 413 13.02))    - d
((98 578 435.5) ( PB E4d3/2 435 8.87))      - m
((42 363 645.5) ( PB E4p3/2 645 6.33))      - n
```

図 2.2.1　解　釈　結　果

Pb：Mo：S＝1：7.4：9.5 が得られる。

システムは，非常に小さなピークを検出したり，ショルダーピークを指摘したり，また装置にごく微量付着した不純物を見出すなど，高い性能を発揮し，組成分析に関しては専門家のレベルにある。

2.4.5. 医用画像解釈

わが国の医療費は年々増加し，約１０兆円に達している。

これに伴って医療産業の生産額も急激に増加している。医学の技術が進歩するに従って高度な医療サービスを受ける人口も増え，医者や検査官の不足が問題になる。

医療において診断のための検査の占める比重は大きい。検査データは数値で

表わされたり，画像として表示されることが多い。検査データの解釈は医者が行っているが，一部は計算機を補助として用いている。その中でも画像データの解釈は計算機にとって扱い難く，自動化も遅れている。ここでは医用画像処理の現状を概観し，今後の課題として知識工学の適用を述べる。

(1) 医用画像処理の現状

医用画像処理の研究は情報処理の立場からは１０年以上も前から始められているが，その処理内容はつぎのような２種類に大別できる。

① 画像の生成：医者が見るための画像を作る。（たとえば，超音波像，X線断層像（CT），ＲＩ画像の生成。）

② 画像の解釈：画像から必要な情報を抽出する。（たとえば，白血球の計数，がん細胞診断，X線写真認識，眼底写真解釈）

このうち①はかなり実用化されている。さらに赤外線写真（サーモグラフィー）の応用も研究されている。

②はより高度な処理が要求されるため，比較的実用化が遅れていたが，最近一部の分野では処理装置が製品化されている。血球の計数のような簡単な処理は早くから実用化されているが，子宮がんの細胞診，白血球の種類の識別と異常白血球の検出，染色体細胞の分類などもほぼ実用化されている。つまり顕微鏡写真にしたとき，細胞が重なっていない場合にはうまく解釈できる。処理手順は，まず細胞を一つ分離してその細胞だけの解釈を行い，それが終了するとつぎの細胞について同じ処理を行う。画像内のすべての対象細胞についての解釈が終った後で総合的評価を行う。このような場合には，画像処理は一つの細胞だけを対象とするため，比較的単純な手法ですむ。

いっぽうまだ実用化されていない例として，胸部X線写真，胃のX線写真による集団検診の自動化，眼底写真の解釈などがある。これは多数の細胞を含んだ写真のように一部だけを分離して考えることができない。一部は全体として見てはじめて意味がある。とくにX線写真では対象としている肺や胃だけでなく，体の他の部分が重なって見える。医者の診断も複雑で，対象と

なる臓器の性質だけでなく，その周囲にある各種の部分の性質を知っていなければならない。現在もこのような写真の解釈は医者以外によっては行われていない。

図 2.2.2 に胸部X線写真の例を示すが，肺の部分には肋骨が横切り，中心付近には細い血管が重なって見える。また中央から右にかけて心臓があり，その影が向かって右の肺の左下の一部と重なっている。肺がんや結核の診断のためには，肺の中にある異常な影を見つけなければならない。一般には肺部は明るく，異常な影は暗い。そこで明るい領域にある暗い部分が異常であるとする方法が考えられる。ところが上述のように骨や血管の影も重なっているため，正常な影と異常な影を区別しなければならない。そのための研究としてこれまで肋骨境界の識別[25]，心臓りんかくの抽出[26]，血管影の識別などが10年間にわたって行われてきた。また限られた実例についての診断も試みられ，ある程度の診断はできるようになった。しかしまだ実用に耐えるようなシステムを作る段階には至っていない。

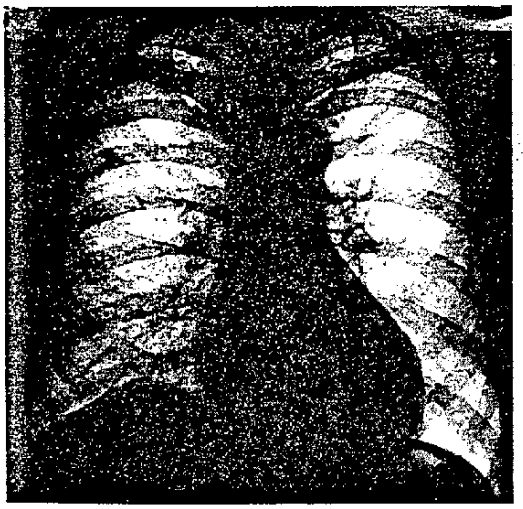


図 2.2.2　胸のX線像の例

(2)　知識工学の導入

医用画像処理を行って医者の診断の助けとするためには，各種の知識を用いなければならない。それは，

①　解剖学的知識

②　対象画像の作成と画像の性質に関する知識

③　画像処理の知識

④　診断に関する知識

⑤　対象とする患者に関する知識

である。たとえば胸部X線写真による診断では，どのような機器を用いてどのような撮影条件で画像を作成したか，また写真フィルム上の濃淡は対象物のどのような性質を表わしているかという②が必要である。③によって画像を解釈して特徴を抽出する。その場合には肺，心臓，骨，血管など位置や大きさの知識①を用いる。診断を行うためには，⑤を考慮し，疾患と画像との対応を調べながら異常な状態を見つけるという④が重要な役割を果す。

以上述べた知識ははっきり区別することがむづかしい。解剖学的知識に基づいて画像を解釈する場合に，①，②，③が密接に結びついている。しかし少なくともはっきりと分類できる知識もある。①では肺の位置や大きさ，②では対象物のX線吸収係数とフィルム上の濃淡，③では明るさ一様な領域の抽出法，④では病気によってひき起される肺の変化である。⑤においては問診によっても集められる患者の年令，病歴などは，とくに知識と名づけなくてもよい簡単な事実である。このように独立した知識はデータベースに入れておいて利用すればよい。

いっぽう医学とか物理学とはっきり分類できない知識もある。医者が経験によって獲得した診断術がこれに当る。そのようなノウハウはある程度教科書に述べてあるが，はっきり定式化し難いものが多い。この種の知識を何とかして計算機内に表現して蓄積してゆくことが望ましい。知識の有効性は絶えず実験によって確かめて，もし不都合が生じたらそれを修正する。このようにして長い間にはかなり医者に近い水準の知識を持たせることができよう。現在困難な医用画像処理も，多様な知識を有効に用いることによって解決されると思われる。

(3)　胃X線診断

官庁では35才以上の公務員に胃のX線写真による集団検診を行っている。X線撮影装置を積んだ車の中で，X線写真を何枚も撮り，それを持ち帰って

医者に診断してもらう。通常は撮影を行う組織には専属の医者はいなくて，診断のために臨時に医者を依頼している。この診断をある程度自動化して，明らかに健康な人をあらかじめスクーリングできれば，医者の負担は軽減される。

胃自体はX線をあまり吸収しないため，バリウムを飲ませ，バリウムの像を見ることによって胃の疾患を診断する。胃の各部分の状態を見るために，バリウムの量，人間の姿勢を変えて6枚程の写真を撮る。1人の患者につき6枚を観察し，それぞれ顕著な疾患を検出し，あいまいな場合にはいくつもの写真を比較して診断を下し，もし疑わしい疾患が検出された場合には直接撮影で再検査にまわす。胸部X線写真と較べるとつぎのような特徴がある。

① バリウムを飲ませたため，胃の形がよくわかる。

② 複数の写真を用いて信頼性のある判定ができる。

③ 胃の形，位置が人によってまた姿勢，時間によって大きく変化する。したがって他の臓器との重なり具合も変わる。

④ 診断のため多数の写真を見なければならない。

この中で③は自動化にとって好ましくないが，①，②は処理を簡単化し，④は自動化の効果を増大する。

胃X線診断の概要を見ながらその自動化のための問題点や課題を調べてみる。

(a) 立位充満像

立った姿勢でバリウムをかなり多量に飲んで撮影した像で，胃の大部分はバリウムに満たされ，その形がよく表われている。図2.23に例を示す。図2.24には胃各部の名称を示す。通常は図2.24のハッチングのようにバリウムが満たされ，その部分はX線が吸収されてフイルム上では白くなる。これが焼付けられれば，図2.23のように黒いシルエット状になる。正常な胃はなめらかな曲線のりんかく像を形成するが，胃壁に異常があれば，その部分が硬くなって直線化したり，異常を反映して不連続になった

図 2.23　立位充満像

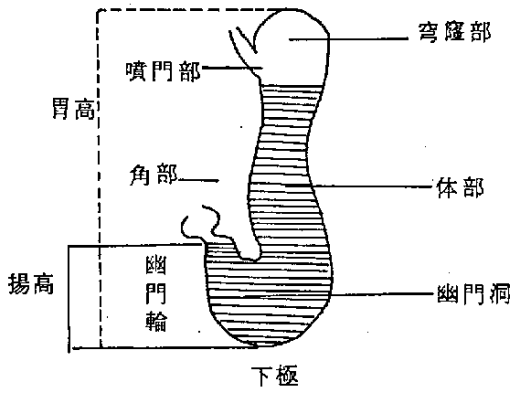


図 2.24　胃各部の名称

りする。異常を見つけるためには胃壁の局部だけを見るのではなく，全体の一部として見なければならない。少なくとも局部を見ている場合にも，それが胃のどの部分に相当するのかは知っていなくてはならない。

これまで充満像の画像処理の研究は少し行われている。胃の陰影やりんかくの抽出[28,29]が早く着手されたが，まだ種々の胃に対して適用できる矛軟性のある方法は確立されていない。胃の典型的な型である長胃の充満像から主要な部分を識別する研究もあるが[30]，全体の正答率はまだ５０％程度である。

(b)　背臥位二重造影像

バリウムと発泡剤を飲み，バリウムが胃壁に付着した後に背を下にして横になれば，大部分のバリウムは穹窿部（図 2.2 3 ）に溜まり，他の部分は胃壁に付着したバリウムだけが見える。図 2.2 5 に示すように胃壁の部

図 2.25　背臥位二重造影像

分は白っぽくなり，それは胃の表側と裏側の壁の重なった像となっている。胃壁の凹部にはバリウムがよく付着するため，胃壁のひだを見ることができる。胃壁に異常があれば，ひだがそこに集中したり直線化したりする。また局部的な隆起部や陥凹部も検出できる。[31)]

しかし二重造影像は充満像のりんかくのように明確な特徴はなく，正常な像でも微細な変化がある。また壁が二重に映っているだけでなく，他の臓器の陰の影響も受け易い。従って計算機を用いて自動化するのは容易ではない。これまでこの種の研究は行われていないと推定される。

(c)　研究の進め方

電総研視覚情報研究室では胃X線診断の検討を行っている。従来の研究，医学書，医者の話を参考にし，これまで研究室で行ってきた物体認識の研究の経験からつぎのようなアプローチをとることが考えられている。

①　立位充満像から胃の領域を近似的に求める。

② 胃の各部のだいたいの位置を決める。

③ ②を参考にして胃の領域をさらに正確に求める。

④ 胃の型を決め，胃の領域外を調べる。

⑤ ④を参考にして胃の領域をさらに正確に求める。

⑥ 胃のりんかくの胃常を調べる。

⑦ 上を参考にして他の充満像を同時に処理する。

⑧ 穹窿部を見つけて異常を調べる。

⑨ 二重造影像で穹窿部を見つけ，それを手がかりにして他の領域を近似的に求める。

⑩ 他の臓器を調べ，重なっている部分を調べる。

⑪ 明るさの局所的異常を調べる。

⑫ 明るさの変化している部分を検出し，その方向性を知り，ひだの集中を検査する。

以上のうち⑨以下は二重造影像の処理を含むため超未踏といってよい。しかし⑧までは従来の画像処理の技術プラス専門的な知識によって達成できると思われる[32]。⑧までを行う間に知識工学のノウハウを知り，それより先の困難な問題に立ち向うことができればよいと考えている。

2.4.6. メタシステム

前項までに述べた種々の知識システムに対し，これらに用いられた知識工学的手法を整理，一般化し，知識システムを構成するためのシステムを作る研究も行なわれている。このようなシステムをメタシステムと呼ぶことにする。

Rutgers 大学の EXPERT，スタンフォード大学の EMYCIN やＡＧＥはその例であり，これらメタシステムの上で多くのコンサルテーションシステムの開発が行なわれている。一方，これらとは少し目的が異なるメタシステムとしてスタンフォード大学の Meta—DENDRAL がある。これは知識システムではなく，知識システムの重要な要素である専門知識を作り出すための

メタシステムである。ここでは，これら２種のメタシステムについてのべる。

(1) EMYCIN [33)]

機能的には 2.4.2 にのべたMYCIN の専門知識部分を取り除いたものである。MYCIN の特徴である。

① プロダクションルールによる法則の記述

② 逆向きの推論

③ 信頼係数による探索の制御と結果の評価

④ 英語によるシステムとユーザとの質問応答

⑤ 結果や途中の推論，システムからの質問に対してユーザが質問できる機能

はそのまま保存しつつ，医療以外を含めた多くの分野に応用できるよう一般化されている。

法則の入力，更新やユーザとの質問応答部分は分野を限定しないために新たな工夫がなされた。法則の入力はMYCIN の英語に対してAlgol 風の ' terse ' と呼ばれる内部のLISP形式に近いものに改められた。図 2.2 6 は，ある法則(RULE　６８)を ' terse ' および内部のLISP形式で表わしている。また，システムは文法上の誤りやミススペルのチェックを行なう。QAモジュールはユーザーが入力した法則の中からキーワードを抽出して辞書を作り，文型のマッチングとキーワード検索による英語による質問応答を可能にしている。

EMYCIN 上で作られたシステムには，医学分野でのMYCIN，PUFF，HEADMED（精神病）のほか有限要素法の数値計算プログラムの使用法のコンサルテーションシステムSACONなどがある。

```
Rule 68

If:  1) The material composing the sub-structure
        is one of the metals,
     2) The analysis error (in percent) that is
        tolerable is less than 5,
     3) The non-dimensional stress of the sub-
        structure is greater than .5, and
     4) The number of cycles the loading is to be
        applied is greater than 10000
Then: It is definite (1.0) that fatigue is one of
 the stress behaviour phenomena in the sub-structure
```

(a) 法　　則

```
If  Composition = (LISTOF METALS) and
    Error < 5 and
    Nd-stress > .5 and
    Cycles > 10000
Then   Ss-stress = fatigue
```

(b) 'terse' 表現

```
PREMISE: ($AND
           (SAME CNTXT COMPOSITION (LISTOF METALS))
           (LESSP* (VAL1 CNTXT ERROR) 5)
           (GREATERP* (VAL1 CNTXT ND-STRESS) .5)
           (GREATERP* (VAL1 CNTXT CYCLES) 10000))
ACTION: (CONCLUDE CNTXT SS-STRESS FATIGUE TALLY 1.0)
```

(c) LISP表現

図 2.2 6　法則の 'terse' 表現とLISP表現

(2) EXPERT [34)]

Rutgers 大学ではすでに CASNET による緑内障の診断治療システムを開発しているが，EXPERT はこれを他の分野にも応用可能なように一般化したものである。このシステムは，FORTRANで作られているため高速で移殖性が高く，ミニコンでも使用できるのが特徴であろう。

法則の大部分はプロダクションルールの形で記述されるが一部にCASNETも使用できる。EMYCIN との大きな違いは推論法則の適用方式である。EMYCIN ではゴールから出発する逆向き推論であったが，EXPERTは前向きである。また法則は３つのレベルに分けられ実行の順序はこのレベルによって決められる。

診断は，次の３つの要素によってなされる。

① 問診，検査，または推論によって得られた事実（ findings：Fで表わす）。

② 病気の分類，病気のメカニズに関する仮説（ hyposeses ：Hで表わす）

③ これらを論理的に結びつける法則，これはさらに次の3種に分けられる。

- F－F法則：FからFを導く法則
- F－H 〃 ：F 〃 H 〃
- H－H 〃 ：H 〃 H 〃

法則の適用順序は，F－F→F－H→H－Hと定められている。EMYCINの場合，もっぱら法則の並べ方によっていた実行順序を法則群の間に設定し，かつ全体としてボトムアップは推論方式をとっているのが大きな違いである。扱かう問題の性質と制御方式との関係は今後，大いに研究されるべきであろう。

EXPERTで実現されたシステムは医療コンサルテーションシステムに限られている。ミズーリ大学と共同でのリューマチの病理と治療のほか緑内障，甲状腺疾患の診断，治療に関するシステムが作られている。また日本では，日米協力科学研究プロジェクトにより，緑内障の問診システムG3/EXPERT[11]が作られている。

(3) AGE （Attempt to GEneralize）[35]

知識工学の専門家でないユーザーが高度な知識ベースシステムを作れることを最終目標にして開発が進められている。知識工学上の種々の技術をパッケージ化してあり，またユーザーとインタラクティブな形でシステムを構成するための質問応答機能をもっている。

ＡＧＥでは，ユーザーが問題をいくつかのレベルに階層化し，推論過程の制御方式を工夫できるよう，カーネギメロン大学のHEARSAY－Ⅱ で用いられたのと同じ，黒板モデルがとり入れられている。ＡＧＥの構成要素は次の3つである。

① 分割された小問題を解くためのＫＳ（Knowledge Sources）と呼ばれる小領域

② 小問題間の階層構造を記述し，ＫＳ間の情報交換を行なうための黒板

③ 黒板上に書かれたどの事項に焦点を当て，どのＫＳを次に起動するかを決定するメカニズム

図2.27はＫＳと黒板との関係を示している。ユーザーは，まず，問題を木構造で表わされるいくつかの問題に分解する。図2.28は蛋白質の構造解析システムCRYSALISの問題の構造である。ＫＳはこの各ノードに対応して作られる。ＫＳには，その小問題を解くための法則と推論の制御メカニズム（前向きか逆向きか等）が導入される。ＫＳ間をつなぐのは図2.28

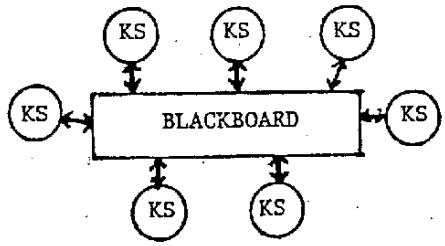


図2.27　黒板とＫＳ

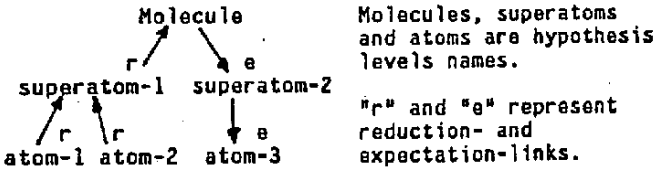


図2.28　CRYSARISの問題構造

の矢印に相当する。EVENT，推論の結果予想されるEXPECTATION，または逆向き推論において新たに生成されるゴールである。黒板上に書かれたこれらの情報により，次にどのＫＳを起動すべきかが決められる。基本的な制御方式はEVEN−DRIVEN，EXPECTATION−DRIVEN，GOAL−DRIVENの３種であり，ユーザーはさらに細かい指定まで含めて多くの制御方式を選択することができる。

システムは，ユーザーに対して相当詳しいガイドを与える。図2.29はそ

```
Do you want to work on Hypothesis.structure?: y
Aquiring HYPOTHESIS.STRUCTURE
Name of LEVEL1 ?
  A level name can be any literal atom not already used for the
  name of another level. Type Done to quit.
: sentence
Attribute?
  Attributes must be unique within a level.
: words
Attribute?
     [etc.]
Name of LEVEL2 ?
: wurd  [misspelled name--to be corrected later]
Attributes?
: letters
[The remainder of hypothesis acquisition and editing deleted.]
: print  [print the hypothesis structure]
LEVEL1 - SENTENCE
         WORDS - NIL
         ENGLISHWORDS - NIL
         LETTERS - NIL
LEVEL2 - WURD  [mispelled level name])
         LETTERS - NIL
         ENGLISHWORD - NIL
         ORDREINSENTENCE - NIL  [misspelled]
         POSSIBLEWORDS - NIL
LEVEL3 - LETTER
         CRYPTLETTER - NIL
         ORDERINWORD - NIL
         RELFREQ? - NIL
```

図2.29　システムガイドの例

の一例である。

現在，AGE上ではCRYPTO と呼ばれる，文字対文字対応による暗号文を解読するシステムのほか，ブリッジのビッドを行なうシステムが作られ，また前出PUFFについてはEVENT−DRIVEN とEXPECTATION−DRIVEN による制御方式の異なる2種のシステムが作られ，システム構成の容易さと，制御方式選択の自由度は実証されている。

(4) Meta−DENDRAL[36,37,38]

Meta−DENDRALのプロジェクトは，①知識システムを作る場合に，専門家から専門知識を得るプロセスがボトルネックになるのでこれを解決したい，および，②計算機により未知の新法則を発見したいという2つの動機により始め

られ，現在までに，いくつかの新法則を含む貴重な成果が得られている。

Meta−DENDRALは質量分析スペクトルと試料の構造を結びつける法則を生成する。試料に電子を当てると，どこかで化学結合が切れ，イオン化されたいくつかの部分構造ができる。出来たイオンの質量を求めたのが質量分析スペクトルであり，どの場所でどの位の確率で化学結合が切れるかの法則（フラグメンテーションの法則）がわかればスペクトルと試料の構造を結びつけることができる。

ユーザーは構造既知の物質の構造式と，スペクトルデータを入力し，さらに法則生成の過程で必要なさまざまの制限条件や助言をシステムに与える。

法則生成の過程は次の３つのステップよりなる。

第１のINTSUM は，スペクトルの全ピークに対して，考えられるすべての部分構造による解釈を試みる。そして分子構造と生ずる部分構造との関係を統計的に調べ，結果をDENDRAL SUBGRAPH の形にまとめる。

第２のRULEGENでは，上で得られた事実にそって，可能なフラグメンテーションの法則を生成する。法則は，一般的なものから出発し，事実に合わせて階層を追って特殊化される。特殊化されすぎて適用の場がほとんどなくなったり，あるいは逆に，一般的すぎてほとんど意味をなさなくなることがないように，その法則を支持する例，および反例にもとずく評価を行っている。

第３のRULEMODは，RULEGENで生成された法則の候補について，詳しい評価を行ない，法則の調整を行なう。冗長な法則をとり除き，同じ事実に支持された法則を併合し，反例がでた法則は特殊化して反例を除き，また反例が出ない範囲で一般化できるものは一般化する。

図２.３０は，アミン類に関するINTSUM の働きを示している。物質１，２におけるA１，A２，A３ の切断はＢ１の形にまとめられ，Ａ４だけは別にＢ２となる。RULEGENは任意の原子Ｒと別のＲの間で結合が切れるという一般的な法則から出発し，図２.３１のようにユーザーが与えた制限にしたが

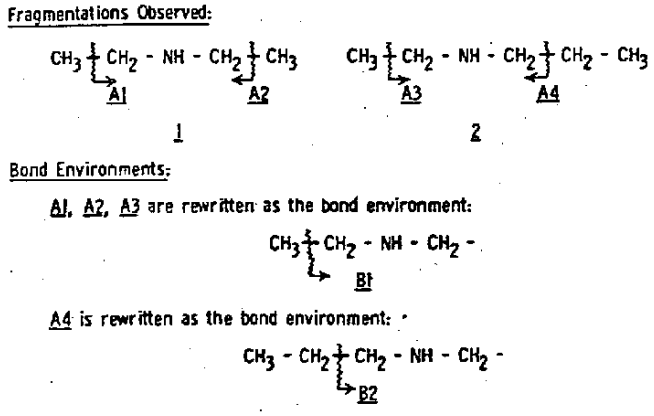


図 2.3 0　INTSUM の例

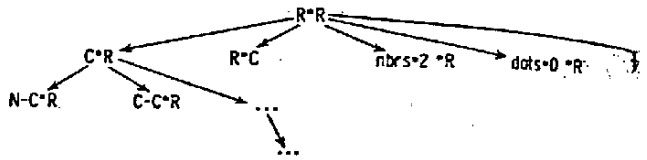


図 2.3 1　法則生成の過程

って，木構造を展開し，法則を特殊化していく。例では，左辺のRがまず炭素Cに置きかえられ，さらにN－Cとなる。この例では，これ以上の特殊化によっては評価が上がらないため，この枝に関する展開は，ここで打ち切られる。このような探索が木全体について行なわれ多くの法則の候補が作られる。

図 2.3 2は，RULEMODの例である。ここではまずRULE−1とRULE−2

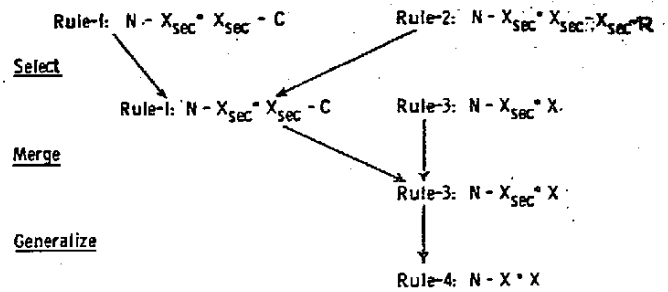


図 2.3 2　法則の調整

が比較され冗長なRULE−2 が除かれる。つぎに，同じ事実を証明するRULE−1とRULE−3が併合されて，この場合はRULE−3 となる。

さらに，これは反例が出ない範囲で一般化されRULE−4 となり，これが結果として出力される。これらの過程で，ユーザーが与える制限条件や助言は図2.3.3に示すように，専門的で細かいものである。

```
specify fragmentation process constraints? Y/N. : [Y]

prompt for all constraint values? Y/N. : [Y]
forbid cleavage of more than one bond
     to the same atom? Y/N. : [Y]
forbid cleavage of aromatic ring bonds? Y/N. : [N]
allow default definition of aromatic rings? Y/N. : [Y]
forbid cleavage of double and triple bonds? Y/N. : [Y]
minimum number of carbons in a fragment? : [2]
allowed hydrogen transfers? : [2 1 0 −1 −2]
maximum number of bonds allowed to cleave
     in a single step process? : [2]
maximum number of steps in a fragmentation process? : [2]
maximum number of bonds allowed to cleave
     in a multiple-step process? : [2]
maximum number of rings allowed to fragment
     in a multiple-step process? : [1]
allowed neutral transfers (other than H)?
     : [CO −1]
any other constraints? Y/N. : [N]
```

図2.3.3　ユーザーからの制限，助言の例

Meta−DENDRALが出力する法則は，その分野に関して専門家の出すものと同じレベルにある。特に，Meta−DENDRALがKetoandrostaneに関して求めたフラグメンテーションの法則群は，専門家が作り出せなかったものであり，結果は高く評価されている。

2.4.7. 知識工学の研究開発

これまで，知識工学の応用の紹介と，やゝ詳しい適用例を述べたが，ここでは知識工学の研究の進め方を考える。

扱う知識は対象に依存するため，何を対象として選ぶかによってそのむづかしさが変わる。例えば前述のDENDRALはシステムとして一応でき上るのに約5年を要した。またMYCINは3年位でできたが，まだ実用にまで至っていない。前述のPUFFは実験システムを作るのに1年を要したにすぎない。PUFFのための実際に行った作業内容は，医学的な専門家から知識を吸収するのに50時間，プログラムを作るのに10週間であった。この差は対象の複雑さと，どの段階まで開発するかの違いによることはもちろんであるが，積み重ねた経験の効果も大きいと思われる。

対象とする分野の専門家の知識は，経験的，実験的であいまいな場合が多い。このような知識を計算機で記術するためには，情報処理の専門家が知識を分析して組織化することが必要になる。この仕事を行う知識工学者ともいうべき人は，はじめは対象分野に関しては素人であるが，専門家との接触によってある程度の知識水準に達する。しかし知識工学者の役割は，既知の知識を有効に利用する方法を見つけることであり，対象分野の新しい知識を見つけることではない。従って有用なシステムは，すぐれた専門家と知識工学者の協力によってはじめて可能になる。

知識工学によって作るシステムが備えるべき条件としてつぎのことを忘れてはならない。

① 対象分野の専門家が知識をつけ加えたり修正できる。

② ユーザーが自然にシステムと会話できる。

③ 必要に応じて，ユーザーに答を出した理由を説明できる。

以上の条件を備えれば，システムを信頼して使うことができる。

すでに述べたように知識工学の対象分野は多様である。たとえば，機械工業の設計・生産の分野では，機械の専門家と知識工学者との協力が必要である。胃X線診断では，医者，X線技術者，知識工学者が協力することになろう。

いずれにしても，知識工学者の仕事量が多くなる。前節で述べたようなメ

タシステムはこの仕事量を軽減するために開発された。しかし，どのメタシステムも多かれ少かれ対象分野を制限している。真に対象分野に依存しない汎用メタシステムはまだ存在しない。これは，知識の表現，獲得，推論機構，自然言語処理などの研究の進展を待たなければならないであろう。

参 考 文 献

1) E.A. Feigenbaum: The Art of Artificial Intelligence, Proc. of 5th International Joint Conference on Artificial Intelligence (1977).

2) E.A. Feigenbaum, G.B. Buchanan and J. Lederberg: On Generality and Problem Solving: a Case Study Using the DENDRAL Program, Machine Intelligence, Vol. 6, Edinbargh Vniv. Press (1971).

3) J.R. Slagle: 大学初年度微積分学の記号積分問題を解くヒューリステイック・プログラム, コンピューターと思考, 好学社(1969).

4) J. Moses: Selected Papers by Prof. Moses, Project PIPS, Electrotechnical Laboratory (1975).

5) 井田, 後藤: 数式処理プロセッサ, 情報処理, Vol.21, No.1, pp.19-27

6) E.H. Shortlife: Computer Based Medical Consultations, MYCIN, American Elsevier Publishing Co. (1976).

7) R. Davis, B. Buchanan and E. Shortlife: Production Rules as a Representation for Knowledge-Based Consultation Program, Artificial Intelligence, Vol. 8, No. 1, pp. 15-45 (1977).

8) S.M. Weiss, C.A. Kulikovsky and S. Amarel: A Model-Based Method for Computer Aided Medical Decision Making, Artificial Intelligence, Vol. II, No. 1, 2, pp. 145-172 (1978)

9) P. Seolovits and S.G. Pauker: Categorical and Probalistic Reasoning in Medical Diagonis, ibid., pp. 115-144.

10) 開原他： 人工知能の手法を用いた診断治療のコンサルテーション，医用電子と生体工学，Vol.17，pp 73-77(1979).

11) 開原他： 医療知識工学，情報処理，Vol.20，№ 11，pp.991-1000(1979).

12) S. Sasaki: Automated Chemical Structure Analysis Systems, Determination of Organic Structures by Physical Methods, Vol. 5, Academic Press, 1973.

13) 佐々木： 有機化合物構造決定の自動化に関する研究，分析化学，Vol.23，№ 9，1974.

14) 工藤，阿部，佐々木： ＣＡＳデータと自動構造解析ＣＨＥＭＩＣＳシステムの機能との有機的結合，化学と情報に関する討論会論文集，日本化学会，1979.

15) R. S. Engelmore and A. Terry: Structure and Function of the Crysalis System, Proc. IJCAI, pp. 250 - 256 (1979).

16) J. S. Bennette and R. S. Engelmore: SACON: A Knowledge - Based Consultant for Structural Analysis, Proc. 5th IJCAI, pp. 47 - 49 (1979)

17) J. Gashning: Preliminary Analysis of the Prospector Consultant System for Mineral Exploration, Proc. 5th IJCAI, pp. 308 - 310 (1979).

18) M. R. Genesereth: The Role of Plans in Automated Consultation, Proc. 5th IJCAI, pp. 311 - 319 (1979).

19) G. J. Sussman and R. Stallman: Heuristic Techniques, in Computer - Aided Circuit Analysis, IEEE Trans. Vol. CAS - 22, No. 11 (1975)

20) G. J. Sussman: Electrical Design; A Problem for Artificial Intelligence Research Proc. 5th IJCAI, pp. 894 - 900 (1977)

21) R. M. Stallman and G. J. Sussman: Forward Reasoning and Dependency - Directed Backtracking in a System for Computer - Aided Circuit Analysis, Artificial Intelligence, Vol. 9, No. 2, pp. 135 - 196 (1977).

22) G. J. Sussman, J. Holloway and T. F. Knight, Jr.: Computer Aided Evolutionary Design for Digital Integrated Systems, AI Memo No. 526, MIT (1979).
（第５世代コンピュータ調査研究のセミナーで講演，前刷集に含まれている。）

23) 山崎，伊原： トップダウン法によるスペクトルデータの自動解析，電総研彙報，Vol.40，№ 10，1976.

24) M. Yamazaki, H. Ihara: Knowledge-Driven Interpretation of ESCA Spectra, Proc. 6th IJCAI, pp. 995 - 997, 1979.

25) H. Wechsler and J. Sklansky: Automatic Detection of Rib Contours in Chest Radiographs, Proc. of 4th IJCAI, pp. 688 - 694 (1975).

26) 春田，鳥脇，福村： 直接撮影胸部X線写真の心輪部線情報による先天性疾患の自動分類，通信学会パターン認識と学習研究会資料，PRL76-67（1977）

27) 鳥脇，末永，福村： 間接撮影胸部X線写真における肺の異常陰影の自動識別，医用電子と生体工学，Vol. 12，№ 5（1974）

28) 末塚，高谷，磯部： 医用画像の記述と識別，第１４回日本ＭＥ学会大会論文集，P.591（1975）

29) 相馬，福島，宇都宮，宮下： 胃X線写真を処理，第１５回日本ＭＥ学会大会，P,107（1976）

30) 森，二木： 長男の立体充満像の自動スクリーニングの試み，第１５回日本ＭＥ学会大会，P.109(1976)

31) 常岡，内田： レントゲン診断学Ⅱ，木本誠二監修，現代外科学大系，中山書店(1974)。

32) Y. Shirai: On Application of 3-Dimensional Computer Vision, Bul. ETL, Vol. 43, No. 6, pp. 358-377 (1979).

33) W. van Melle: A Domain-independent Production-rule System for Consultation Programs, Proc. 6th IJCAI, pp. 923-925, 1979.

34) S.M. Weiss, C.A. Kulikowski: EXPERT: A System for Developing Consultation Models, Proc. 6th IJCAI, pp. 942-947, 1979.

35) H.P. Nii, N. Aiello: AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs, Proc. 6th IJCAI, pp. 645-655, 1979.

36) B.G. Buchanan, E.A. Feigenbaum, N.S. Sridharan: Heuristic Theory Formation: Pata Interpretation and Rule Formation, Machine Intelligence 7, 1972.

37) B.G. Buchanan: Issues of Representation in Conveying the Scope and Limitations of Intelligent Assistant Programs, Machine Intelligence 9, 1979.

38) B.G. Bachanan, D.H. Smith et al.: Applications of Artificial Intelligence for Chemical Inference 22. Automatic Rule Formation in Mass Spectrometry by Means of the Meta-DENDRAL Program, J. of the American Chemical Society, Vol. 98, No. 20, pp. 6168-6178, 1976

39) 岩田一明： 設計・生産システムの現状と問題点，日本機械学会誌，Vol.79，№692 (1976)

3．データベースの高度化

3. データベースの高度化

3.1. 知識情報処理とデータベース

データベースの研究は，ソフトウェア工学の研究と同様に，知識情報処理に対して，２つの側面を持つ，第１に，知識情報処理システムの構成要素として，データベースを考えることができる。この場合，データベースは，知識ベースと共に，推論システムを構成し，問題解決に役立つことが要求される。第２に，データベースの高度化自身が，知識情報処理システムの応用と考えることができる。

第１の側面，すなわち，推論システムの構成要素としてデータベースを考える場合，推論システムの他の構成要素である推論機構と知識ベースとの整合性が大切となる。強力な演繹的推論を行わせるためには，一階述語論理での証明法を活用することが必須である。質問応答処理をこのような方法で行うアプローチは，非評価方式（ Non-Evaluational Approach ）と呼ばれている。その方式では，データベースは，他の一般的な知識と共に，ある形式理論（ Formal Tbeory) の公理を成すと考える。すなわち，非評価法では，データベースを構成する個別的知識と知識ベースを構成する一般的な知識は，質問応答過程で全く同じ扱いを受ける。ゆえに，この場合，これら２つのベースを区別する理由はなくなる。このような方式の利点は，第１に，システムが簡潔になる点である。第２に，PROLOG の上で示されたように，「計算」と「検索」が同じ概念でとらえられる点である。それらは，共に定理の証明の副産物として得られる［ van Emdan 1977］，第３の利点は，非評価方式そのものから由来する，強力な演繹能力を有する点である。欠点は，データベースの有する高速集合演算機能が利用できない点である。

この欠点を補う方式として考え出されたのが，評価方式である。評価方式で

は，データベースと他の一般的事実は全く別のものとみなす。すなわち，一般的事実から成る知識ベースは，ある形式理論の公理を成すと考え，データベースは，その理論の１つの解釈，あるいはモデルであると考える。質問応答は，Yes－No質問では，その質問が与えられたモデルにおいて真となるか偽となるかを調べて答が得られる。また，ある条件を満足するものを求める質問では，その質問を真とするようなモデルの中の事実を探して答が得られる。このような推論は，そのモデルの下でのみ成り立つ推論であり，そのモデルを，人間の有する知識と考えれば，自分の知っているすべての知識に基づく推論となるので，一種の帰納的推論と考えられる。以上は，評価方式のモデル理論的解釈であるが，〔Reiter 1977,1978〕は，評価方式の証明論的解釈を与えている。それは，「閉じた世界」の仮定（Closed World Assumption CWA）に基づく，新たな推論規則の導入により行われる。その推論規則は，「証明できない事柄は，その否定が真である」とする規則である。この規則は，ＫＲＬなどの知識表現システムで，default reasoning（既製推論）として知られているものの一種であり，また，PLANNER の THNOT 命令の働きと同じである。

評価方式では，質問処理は，集合演算によって行われる。とくに，データベースを関係データベースとすれば，質問は関係代数式によって表わされる。

評価方式の利点は，データベース，とくに関係データベースとの整合性の良い点である。質問処理は関係代数式の実行によってなされる。関係代数は，集合演算であり，その実行の高速化は，データベース・マシンとして実現されつつある。

評価方式の最大の欠点は，演繹能力が不足している点である。それは，前述のＣＷＡが成り立つ場合しか扱えない。しかし，それらの扱いについての研究も進展しており，少くとも未知情報の扱いは，可能となる見通しが得られている〔Cold ，1978〕。

評価方式は，知識情報処理におけるデータベースの第２の側面にも深く関係

している。評価方式は，既存のデータベース技術の上で成り立っているので，直接データベースの高度化につながる。

次節では，評価方式での演繹や最適化など，データベースの高度化の基礎となる技術の詳細について論ずる。

3.2. 評価方式での演繹的推論

評価法の解釈に，モデル理論的解釈と証明論的解釈があることは，前節で述べた。この2つの解釈に対応して，演繹方式も2通り考えられる。以下にその2つの方式について述べる。

3.2.1 モデル理論での演繹

データベースを形式理論のモデルと考えるとき，質問処理は，モデル上での論理式の評価によりなされる。データベースは，その際，形式論理中のすべての述語に対する真理値を与えていないのが普通である。それは，基本的な述語の真理値のみを与えている。さらに，その真理値の与え方は，対象となる集合の要素のすべての組合せに対する真理値表ではなく，真となる組合せのみを表の形で表わしている。

述語の真理値がデータベースの形で与えられている述語を基本述語，対応する真値を与える組合せの集合を基本関係と呼ぶ。基本述語以外の述語を仮想述語，対応する関係を仮想関係と呼ぶ。仮想述語の真理値を定めるためには，仮想述語が基本述語によって定義されていることが必要となる。この定義は，実はデータベースがモデルとなっている形式理論の構成要素である。

質問が仮想述語によって表わされているとき，それを基本述語のみから成る表現に変換できれば，その質問は評価が可能である。この，仮想述語表現から基本述語表現への変換が演繹過程と考えられる。

この変換には，2種類ある。第1は，含意関係に基づく変換で，第2は等価

関係に基づく変換である。

(a) 含意に基づく変換

含意関係に基づく変換は，論理学での３段論法に対応し，変換された質問の答は，一般には，もとの質問に対する答の一部分となる。論理の証明では，１つの解を求めるのが基本操作であり，すべての解を求めるためには，すべての可能な証明を行うことが必要となる。〔 Chang , 1978〕は，そのために，connection graph を用いた，すべての証明プラン生成法〔 Sickel , 1977〕,〔Chang and Slagle , 1977〕を採用している。この方式の利点は，証明法との整合性が良い点である。さらに，個々の知識が含意命題として表わされ，その結果，断片的となる（すなわち，$P(x)=Q(x) \vee R(x) \vee S(x)$ とすると，$Q(x)\rightarrow P(x)$，$R(x)\rightarrow P(x)$，$S(x)\rightarrow P(x)$の形で表現され，$P(x)=Q(x) \vee S(x)$ をひと塊りの知識と考える必要がない）ので，知識ベースの拡張が容易である（もし，$P(x)=Q(x) \vee R(x) \vee S(x) \vee T(x)$となることが分ったなら，単に $T(x)\rightarrow P(x)$ を付け加えればよい）。ここで，等号性が問題となるが，それは，すべての証明を行うことにより，結果的に等号性を仮定していることになる。すなわち，データベースの立場で考えれば，「知っていることはすべて答えた」ということになる。このことを，それが正しいもののすべてであると解釈するのは，前述した既製推論の一種である。（既製推論を仮定しないと，$\{ y \mid (\forall x) P(x)\rightarrow U(x, y) \}$ のような質問の評価ができなくなって具合が悪い。すなわち，この場合には，すべての x に対して$\sim P(x)$が真かU（x，y）が真であるかを調べなければならないが，既製推論によらないで$P(x)$が真となるｘの集合を求められない）

(b) 等価関係に基づく変換

等価関係に基づく変換では，変換前と後で答の集合が同一である。等価関係は，論理学での等価命題か，あるいは，関係代数での等式によって表わされる。〔 古川，１９７９〕は，関係代数方程式に基づく変換によって，演繹を行う方式を導入した。代数による方法は，集合演算を基にしており，ある条

件を満たすすべての解を求めるのに適している。質問処理は，以下のようにして行われる。まず，論理式による内包表現として与えられた質問｛ｘ｜W(x)｝は，and，or ，imply ，(∀y),(∃y) などに関する真理値規則により，単純な述語に関する質問｛ｘi｜Pi(ｘi)｝と，それらの間の集合演算，すなわち関係代数式に変換される。つぎに，この関係代数式は，仮想関係を定義する関係代数方程式により，順次，基本関係より成る関係代数式に変換されて行く。この変換過程が演繹的推論に相当している。

ここで問題となるのは，仮想関係が再帰的に定義されている場合である。その場合には，上に述べた方法によって，質問を基本関係のみから成る関係代数式に変換することは不可能である。その解決方法は，通常再帰方程式を解くのに用いられる，最小不動点を求める方法によればよい。

3.2.2 証明論での演繹

〔Reiter，1978〕は，ＣＷＡ（Closed World Assumption）に基づく既製推論を用いて，前節の等価変換法での関係代数式への変換規則を導き出した。Reiter の導出では，｛ｘ｜(∃y) P(ｘ，y)｝の形の式から成る関係代数式が得られる。そして，各｛ｘ｜(∃y)P(ｘ，y)｝の形の質問に対して演繹操作が施される。演繹操作は，resolution 法に準じて行われる。resolution 法との相違は，演繹の各段で，演繹操作と並行してデータベースへの問いを発している点である。この方法によって，データベース中に置かれた具体的事実と同時に，知識ベースから演繹的に導かれる事実があれば，それも含めて，答を求めることが出来る。

Reiterの方法の優れている点は，この質問処理法が，一般のＣＷＡを仮定しない質問処理過程に完全に埋め込まれている点である。（しかしながら，この一般の質問処理過程も，「すべての解」については，前節の(a)含意に基づく演繹と同様，既製推論を仮定している）

3.3. 実用化技術

高度な機能を備えたデータベース・システムが従来のデータベース・システムに取って替わるためには，第1に表現力が損われていないこと，第2に機能と効率を合わせて考えたときのシステムの能力が，従来のシステムのそれを凌ぐことである。

3.3.1 表現力の拡張

関係データベースの欠点として挙げられるもののうち，表現力に関するものは，自然な構造情報を表現できない点と，未知情報の扱いが困難である点の2つが主なものである。

関係データベースは，いくつかの関係と呼ばれる一様な，構造のない表から構成されている。関係は，レコードの集まりと考えられるが，関係モデルでは，それを要素の間に順序等の構造をもたない単純な集合とみなしており，関係を処理する関係代数の諸演算も，この数学的な捉え方を出発点にしている。そのため，たとえば画像情報のように，データが2次元空間上の並びという構造を持つとき，その構造を自然に表現できない。便宜的に，構造自身を，たとえば座標値としてデータ化し，＜x座標，y座標，濃度＞の形をレコードの集合として関係を定義することは可能である。しかしながら，データの持つ構造は，他の特性値などと異なる種類の syntactic な情報であり，そのような構造をうまくとらえることによって，処理が単純化されるのが普通である。

この問題を解決する手掛りの1つは，述語論理での項（ term ）による構造の表現である。項は，関数形式 f (x)などが許され，この関数によって構造を表現すればよい。たとえば，2進木構造は t（x，u，y）のように表わせる。ここで，uは節の値で，x，yはそれぞれ左部分木，右部分木である。tは，構造組立関数であり，LISPの cons に当たるようなものである（ただし，この t

は，構造を実際に作る命令ではない）。構造を利用者が扱えるようにするには，そのレベルでの構造を操作するための言語が必要となる。その言語の第1の候補は，すでに述べた通り，述語論理自身である。システムの内部での論理操作は，関係代数が最も適しているが，この間のギャップは未だ埋められていない。利用者言語，内部表現共，他の可能性をも含めて研究すべきである。

未知情報の扱いについては，〔Cold，1972〕が3値論理，すなわち，真，偽，未知を真理値とする論理によって関係代数の拡張を行った。通常の2値論理では，存在限定子による表現（∃x）P(x)の形で未知情報を表わし得るが，この種の情報は，CWAでは扱えないので，一階述語論理の完全な証明能力を必要とし，効率の面から実用にならない。論理表現と，拡張された関係代数との結合が，今後の課題である。

3.3.2. アクセスの効率化

関係データベースでは，アクセス経路を指定した検索手続きを記述できないので，利用者は，検索効率に関してはシステムの能力に委ねるしかない。そのため，システムによる最適化が重要な課題となる。

最適化の研究は，いくつかなされている。〔Astralan et.al，1975〕では，質問を解析して，システムが用意した索引を利用できる部分と利用できない部分に分類し，演算の順序を変えて，最適な手続きを作り出している。ある場合には，索引を一時的に作成するような手だてをも行っている。

〔Smitl et.al，1975〕および〔Hall，1976〕では，関係代数式の上で，結果に影響を与えない範囲で，すなわち，等価変換規則によって演算の順序を変更し，最適化を図っている。

〔古川，1979〕では，冗長表現および階層表現を関係代数方程式により表わし，それを変換規則として用いて，Smitl et al. あるいはHall と同様に，関係代数式上の操作により最適化を図っている。この最適化は，前二者と比べて，より大局的な最適化になっている。すなわち，データの構造を反映した処

理アルゴリズムに最適化されている。

関係代数式の変換による最適化は，演繹とともに知識情報処理技術としてとらえることができる。すなわち，変換の際に用いられる変換規則は，見方を変えれば，システムが保有しているデータ表現に関する知識と考えられる。この種の知識を蓄積することによって，最適化機能の増強を図ることが可能である。

表現力の拡張の項で述べた，構造情報の表現については，従来は種々のデータ構造を表現する物理的手段の研究が主になされてきた。しかし，関係代数は，あくまでも構造のない表に対して定義されているので，構造化されたデータ表現を，直接関係代数によって操作することはできない。上に述べた方法により，階層構造については，ある程度までの最適化が可能であるが，ポインタによるデータのアクセスまでは行えない。このギャップを埋めることがここでの課題であるが，それは不可能ではなく，自動プログラミングの考えを適用することによって可能となるであろう〔古川，米沢，1980〕。

3.4. 質問・応答の円滑化

3.4.1. 交信手段の多様化・高度化

データベース・システムを利用するとき，システムとの交信手段が豊富で，かつ人間にとって自然であることが望ましい。現在の技術水準で考えれば，Query By Example（QBE）に代表される非手続き的な図式言語が，工学的なセンスで最も使い易い言語であろう。言語は，使い易さと同時に，表現力が大切である。その点，QBEは，関係データベース用の言語であり，前節にも述べた通り，構造情報の扱いには機能が不足している。

自然言語は，勿論，最も使い易く，表現力が豊かな言語である。人間は，さらに，自然言語と共に，図を交信手段として用いる。「百聞は一見にしかず」であり，図を伴った説明は，通常，非常に分かり易い。自然言語と図の両者を同時に扱った研究が〔Waltz，1979〕などに見られる。それを交信の手段とす

る研究は，未だ見られないが，これから活発に研究されるようになるであろう。

自然言語による質問応答システムについては，自然言語処理ワーキング・グループの報告書を参照されたい。

3.4.2. 知的対応機能

自然言語あるいは図による交信によって質問応答の円滑化を達成するためには，まず第1に質問を理解する能力が必要なことは言うまでもない。この部分は，前節の自然言語あるいはイメージ理解の問題と考えることができる。会話による，質問者とシステム間での概念の同一化も，自然言語理解の一部分として捉えた方がよいであろう。それは，あいまい性の除去，意味の理解などの自然言語特有の問題だからである。しかしながら，Default Reasoningによる意味の推察と，それに基づく仮説的推論などを含めると，単なる自然言語理解の枠組から外れてきて，むしろ，知的対応機能と言った方がより適切であろう。

知的対応機能の目的は，一言で言えば，質問者の意図の推察に基づく，システムの知的な対応の実現である〔Joshi ,1977〕。

上に述べた質問の推察機能の他に，知的対応の重要な機能は2つある。それは，助言機能と，知的な応答機能である。

助言機能の単純な例に，答えが空となる場合の説明機能がある。すなわち，質問の評価の結果が，質問者の意図に反して該当する項目が無かったとき，システムがその原因を明らかにすることによって，質問者は，自分の思い違いに気がつき，正しい質問の作製に役立つはずである。この説明機能を実現するための研究が，〔Janas ,1979〕によってなされている。あるいは，もっと進んで，システムが，質問者が望むような質問を自動的に作り出すこともあり得る。

知的な応答機能の例としては，要約による応答がある。

3.5. データベース開発技術

データベースの開発には，入れ物，すなわちデータベース・システムの開発と，中身の開発とがある。入れ物の開発は，大規模ソフトウェア開発技術に依存し，中身の開発は，いわゆるデータベース設計論と，データ入力手法が問題となる。ソフトウェア開発技術については，ソフトウェア工学WGの報告書を参照されたい。

本節では，データベース設計問題を中心に論ずる。データベースの設計は，モデルが時間と共に変化しないスタティックな場合と，時間と共に変化するダイナミックな場合で，その問題が大きく異なる。

スタティックな場合は，データベースの初期設定時に，データ・モデルを完全に設計することになる。データ・モデルを記述するための言語には，データの意味を扱わない単純なものと，意味の表現を目視したものとがある。意味を扱わないものは，データベース定義言語として，実際の商用データベース・システムで用いられている。

関係モデルでの関数従属性あるいは多値従属性などは，意味的な制約条件を表わしている。たとえば，ある会社組織で，従業員が，各々唯一つの部門に属している場合には，従業員→部門の関数従属関係が成り立つ。この条件は，意味的制約条件として，データベースの更新時に入力の誤りの検出に利用できる。関数従属性あるいは多値従属性は，関係データベースの概念スキーマの設計に利用される。これらの従属性に基づくスキーマ設計法は，多くの研究がなされている。

従属性は，問題領域のかかえている意味特性の一部分しか反映していない。より広範な意味記述を扱うデータ・モデル記述言語に，関係モデルから発展したChenのEntity Relationslip Model，階層モデルから発展したBachmanのRole Model などが知られている。

意味記述は，人工知能の分野で研究が進んでおり，多くの知識表現方式と，

そのための言語および処理系が開発されている。そこで開発された知識表現技術は，データ・モデルの記述に応用することが可能であろう。

ダイナミックな世界の扱いは，より困難である。新たなデータが既存のモデルと矛盾をきたした場合，入力の誤りか，世界が変って，モデルが成り立たなくなったかの２つの場合があり得る。この問題は，帰納的に導かれた事実がモデルを構成している場合に起こり，そのときには，モデルの変更が必要となる。これは，Truth Maintenance System における標準的推論の扱いと同じである。

推論を簡単化し，知識表現をも単純化する新たなモデル，あるいは一般的事実のデータからの導出は，学習の問題として知られている。〔Ohsuga, 1979〕は，この問題を，関係代数の除算により行うことを提案している。この問題には，例題からプログラムを生成する文法推測法（Grammatical Inference）を用いた概念の構造化〔藤崎　他〕などの研究も関連している。

3.6. 非定形データの扱い

データベースの研究は，ほとんどが，一定の概念構造をもったデータの集まりを対象としたものに集中しているが，概念構造が定まらないデータの処理も大切である。たとえば，文章や，知識ベースなどがこれに当たる。

文章あるいはテキストの処理は，データベースの処理と異なる手法を要する。その基本的な操作は，与えられた文字列をテキスト中から探し出す，string matching の機能である。string matchingを主な機能とする言語にSNOBOLがある。さらに，テキスト処理のための専用ハードウェアの提案もある。

知識ベースの概念構造は，ＫＲＬなどで網羅的に研究されたが，必ずしも任意の知識をその構造に一意的に当てはめることが容易でなく，そのアプローチは成功していない。セマンティック・ネットワークでは，要素－集合関係，集合包含関係などの基本的な概念構造のみを扱って，ある程度成功している。そ

の場合，個々の知識は，タイプ分けされず，知識間の関係が，上記の概念構造を用いて関係づけられている。個々の知識の属性は，データと共にそこに置かれる。

非定形と定形の中間的な構造を有するものに，辞書のようなデータの集合がある。辞書は，ある程度構造化されているが，厳密でなく，かつ，形式が多様である。このような多様な形式を，オートマトン，あるいは，生成文法によって規定し，パージングに利用する研究がいくつか報告されている〔有川〕，〔辻井〕。

非定形データを定形化する研究も重要である。文献データベースを作成する場合，論文の自動抄録などの技術が，このために必要となる。

3.7. データベース・マシン

データベース・マシンは，大量データ処理を高速化するのがねらいであることは，言うまでもない。本報告では，集合演算，とくに関係代数演算を高速に行うデータベース・マシンについて述べる。関係代数演算で，その効率が問題となるのは，Projection，JoinおよびDivision である。このうち，Divisionは，Projection に伴う計数処理とJoin によって実現される〔古川，1979〕ので，ここでは，ProjectionとJoin のみを考える。

関係R1 の組数をm，R2の組数をnとする。R1でのProjection は，通常はO（m^2）の計算量を必要とし，R1とR2のJoinはO(m×n) の計算量を必要とする。この計算量のオーダを減少させるのが目的である。

データベース・マシンは，上記の演算の高速化の他に，いくつかの条件を満足しなければならない。そのうち，最も重要な条件は，柔軟性である。データベースの大きさ，レコード長などが自由に選らべて，かつそれらが変更されたときに，柔軟に追従できることが望ましい。さらに可変長レコードの処理も要求される。

さらに，ハードウェア化の難易，システムとしての整合性などが問題となる。

本調査では，以上の各点について，各種の方式の比較検討を行う。比較検討を行う方式としては，ＲＡＰに代表されるLogic per Track方式，Hashed Bit Array方式，STARAN などの本格的な連想メモリ方式，および並列分類方式を考える。

3.7.1 Logic per Track 方式

Logic per Trach方式は，ＲＡＰに代表される。すなわち，ディスクの各トラックに論理素子が付加されており，ディスクの１回転で，論理素子での判定結果に応じて，各レコードに付けられたマーク・ビットをON—OFF する。

ＲＡＰでのPROJECTION は，３つのマークビットを用いることにより，結果として得られる関係のサイズ，すなわち組数分のディスクの回転で，計算される。

ＲＡＰでのJoin には，結果として新しい関係を作るFull Joinと，Joinする一方の関係を出力関係と考え，それをマークするだけでよいImplicit Joinがあり，それらの処理アルゴリズムは異なる。Implicit Joinの場合は，ハードウェアで用意された２つの関係のクロス・マークを行う命令によってマーク付けが行われる。そのマーク付けは，制約条件を与える関係，すなわち，結果には現われない関係の組数に比例する。ＲＡＰでは，一度に複数の条件をテストできるので，実際の回転数は，（関係の組数）／定数である。

Full Joinでは，新たに作られる結果は，入力の両関係から組み立てられるので，その回転数は｜R１｜＝mあるいは｜R２｜＝nである。ｍとｎを事前に知れば，min(m,n)である。

Logic per Track方式は，物理的なディスクの構造に，システム構成が制約されるので，柔軟性に乏しい。さらに，多人数による同時利用を考えると，マーク付けの演算中は，そのデバイス全体をロックする必要があり，効率が落ちるなどの点も指摘されている。

3.7.2. Hashed Bit Array 方式

LEECH マシン〔McGregor, 1976〕やCAFS〔Babb ,1979〕などに代表される。PROJECTION およびJOIN ともに，ハッシングにより行う。ハッシングのためのビット・アレイを持つ。ハッシングは誤りを許す方式をとっている。PROJECTION では，ハッシングによる誤りを回復できない。たとえば，（a,a,b,b,c,c）をレコードとし，たまたまaとbがハッシングにより，ビット・アレイの同じビット位置（複数個かも知れない）に写像されたとすると，bはすでにビット・アレイ中に現われているとみなされ，結果としては（a，c）しか得られない。CAFSでは，誤り確率を十分に小さくすることができ，実際にはほとんど誤りを起こさないことを保証している。処理時間のオーダは，mに比例しているが，m項目のハッシングのみでよく，ディスクをm回転させる必要はない。

JOINは，ハッシングによって荒いふるいがかけられ，ホスト計算機で，絞られたレコードの集合の対が，実際にFull Join される。この場合には，ハッシングによる誤りは，この段階で検出されるので，問題はない。処理時間は，ホスト計算機での処理を除けば，m＋nである。LEECH マシンでは，R1をハッシュし，ついでR2を，R1のハッシュ表を検索表として用いてハッシュし，同時にR2のハッシュ表を作り，さらにもう一度，R1を，R2のハッシュ表を検索表としてハッシュしている。これにより，R1とR2の両方についてふるいをかけている。その手間は，2m＋nである。

この方式は，ハードウェア構成が複雑になり，汎用性への対応が困難である。アルゴリズム的にも，ハッシングのフェーズとホストでの処理の2段階を必要とし，すっきりしていない。多くの数による同時アクセスについては，RAPの場合と同様，ロックが広範囲にわたり，性能は著しく劣化する。Join がEqui-Joinのみしか実行できないのも難点である。

3.7.3 連想メモリ方式

STARAN に代表される方式で，演算時にデータベース中のデータを特別の連想メモリに持って来る(stage)ので，stage型ともよばれる。

STARAN でのPROJECTIONは，ＲＡＰと同様のビット操作で行えるが，比較すべきデータは，専用のレジスタに置かねばならず，staging以外に，m回の連想探索を必要とする。

Implicit Joinは，出力関係を連想メモリに持って来て，他の関係のレコードを１つづつ比較レジスタに置いて連想探索を行えばよい。したがって，その処理時間は，制約条件となる関係の組数に比例する。

stage型の欠点は，stageを高速に実行するための，大容量の高速チャンネルを必要とする点である。さらに，連想メモリの大きさが，stageすべき関係のの大きさより小さい場合には，処理を分割しなければならない。

また，Full Join を行う機能は，連想メモリにはないので，それは別に行う必要がある。このため，前項のHashed Bit Array方式と同様，システムの構造が，ハードウェア，ソフトウェアとも，複雑になる。

連想メモリの１セルの長さが固定長であることも，柔軟性の点に関して，満足できない。

高速大容量で，かつ安価な連想メモリの製造技術も，確立していない。それは，上に述べた柔軟性，あるいは汎用性のあるアーキテクチャが未だ考えられないことにもよる。すなわち，十分な需要があるとも思われないので，開発をためらっていることもあるであろう。

連想メモリは，考えは単純で，機能もすぐれており，多くの応用が期待できそうであるが，上に述べたような理由により，その開発の見通しは，あまり明るくないと予想される。

3.7.4. 並列分類方式

並列分類方式によるデータベース・マシンの提案は，ほとんど見当たらない。しかし，そのアイデアは古くからあり，電子ディスク上の並列分類アルゴリズムや，データフロー型の並列分類アルゴリズムがいくつか提案されている。それらについての性能を比較したものを図 3.1 に示す。

	アルゴリズム	ハードウェア比較器	計算量	記憶容量
Intelligent Disk	Gyro Sorting （ℓ行m列） $n=\ell m$ ↓ ホストによるソーティング （各行について）	ℓ 1	$O(\ell m^2)$ $\ell \times O(m\log m)$	n m
Rebound Sorter	Rebound Sorting	n－1	3n	2n
Merge Sorter	Pipelined 2way Merge Sorting	log n	2n+logn−1	2(n−1)
Heap Sorter	Pipelined Heap Sorting	log n	2n	n (core)

図 3.1

この図からも分かる通り，Merge Sorter と Heap Sorter が，$\log_2 n$ 台のプロセッサ（ハードウェア比較器）により，オーダ n で，n コの項目を分類する。ハードウェア比較器が非常に単純であれば，その台数は，オーダ n となっても構わないかも知れないが，実際には，オーダ $\log_2 n$ が現実的である。その点で，この 2 つの方式が注目に値する。Merge Sorter の利点は，磁気バルブなどの電子ディスクを利用できる点である。一方，〔 Tanaka , 1980 〕は，Heap Sorter と Searcher を組み合わせて，データフロー計算機の演算成分とすることにより，関係代数演算がオーダ n で実行できることを示した。

この技術は，柔軟性もあり，多くの点で，利点を有している。データフロー計算機での，データの転送問題が解決できれば，実用価値が高まるであろう。さらに，ソフトウェア的に見ても，関数型プログラミング技術と結びつき，将来の発展の可能性を秘めている。

参 考 文 献

[van Emden, 1977 M. H. van Emden, "Computation and Deductive Information Retrieval," Proc. IFIP Working Conference on Programming Concept, 1977.

[Reiter, 1977] R. Reiter, "An Approach to Deductive Question-Answering," BBN Report No. 3649, Bolt, Beranek and Newman, Inc., 1977.

[Reiter, 1978] R. Reiter, "On Closed World Data Bases," In Logic and Data Bases (Eds. H. Gallaire and J. Minker), 1978.

[Codd, 1978]

[Codd, 1979] E. F. Codd, "Extending The Database Relational Model to Capture More Meaning," 1979 ACM SIGMOD Conference, 1979.

[Chang, 1978] C. L. Chang, "DEDUCE 2: Further Investigations of Deduction in Relational Data Bases," In Logic and Data Bases, (Eds. H. Gallaire and J. Minker), Plunum Press, 1978.

[Sickel, 1977] S. Sickel, "Formal Grammars as Models of Logic Derivations," Proc. Fifth IJCAI, 1977.

[Chang, 1977] C. L. Chang and J. R. Slagle, "Using Rewriting Rules for Connection Graphs to Prove Theorems," IBM Research Report RJ 2117, 1977.

[古川 , 1979] 古川"データベースの知的アクセスに関する研究", 電総研研究報告 No.804, 1979.

[Astrahan, 1975] M. M. Astrahan and D. D. Chamberlin, "Implementation of a Structured English Query Language," CACM, 18, 10, 1975.

[Smith, 1975] J.M. Smith and P.T.Y. Chang, "Optimizing the Performance of a Relational Algebra Database Interface," CACM, 18, 10, 1975.

[Hall, 1976] P.A.V. Hall, "Optimization of a Single Expression in a Relational Data Base System," IBM Journal Research and Development, 20, 3, 1976.

[Waltz, 1979] D.L. Waltz and L. Baggess, "Visual Analog Representations for Natural Language Understanding," Proc. Sixth IJCAI, 1979, 926 - 934.

[Joshi, 1977] A.K. Joshi et al., "Approximate Responses from a Data Base Query System: An Application of Inferencing in Natural Language," Proc. Fifth IJCAI, 1977.

[Janas, 1979] J.M. Janas, "How to Not to Say "NIL" — Improving Answers to Failing Queries in Data Base System," Proc. Sixth IJCAI, 1979, 429 - 434.

[Ohsuga, 1979] S. Ohsuga, "Forward Intelligent Interactive Systems," Proc. The IFIP W.G. 5.2 Workshop Seillac II on Methodology of Interaction, North-Holland, 1979.

[Hollaar, 1979] L.A. Hollaar, "Text Retrieval Computers," IEEE Computer, March, 1979.

[McGregor, 1976] D.R. McGregor and W.N. Dawson, "High Performance for Database Systems," In Systems for Large Databases, North-Holland, 1976.

[Babb, 1979] E. Babb, "Implementing a Relational Database by Means of Specialized Hardware," ACM TODS, 4, 1, March 1979.

[Tanaka, 1980] Y. Tanaka, et al., "Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer," To be presented at IFIP

昭 和 55 年 3 月 発 行

発行所　財団法人　日本情報処理開発協会
東京都港区芝公園３－５－８
機 械 振 興 会 館 内
ＴＥＬ　(434) 8211(大代表)

印刷所　山　陽　株　式　会　社
東京都港区虎ノ門１－９－５
ＴＥＬ　(591) 0248