

TM-0724

要素技術から見た知識
システム・アーキテクチャ

石塚 満(東大), 中川裕志(横浜国大),
飛鳥井正道(キャノン), 松山隆司(東北大),
石田 亨(NTT), 元田 浩(日立),
榎本哲夫(京大), 国藤 進(富士通)

May, 1989

©1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

I C O T T e c h n i c a l M e m o r a n d u m

要素技術からみた知識システムアーキテクチャ

昭和63年度

知識システム・ソフトウェアアーキテクチャ

ワーキンググループ（全体会議）報告書

K S A ワーキンググループ

石塚 満〔東京大学〕 主査

編集責任者 堀内 英一〔I C O T 第5研究室〕

I C O T

1 9 8 9 年 4 月

序

新しい知識処理技術を取り入れた次世代知識ベース・ソフトウェア構築への枠組、アプローチを明らかにすべく活動した。取り上げた重要な要素技術は次の通りである。

ロジックの拡張（特に類推と非単調論理）

仮説推論とその高速化手法

高次推論と計算量

制約問題解決

学習（特にExplanation-based Learning）

知識コンパイル

並列と協調分散処理

トップダウン・システム・アーキテクチャ（特に設計問題）

各委員には個々のテーマを担当してもらうと共に、個々の技術と他の技術の関係を明らかにし、新しい統合的な枠組みを目指す議論を行った。

また関連技術について最新の海外研究についての調査も行った。

各委員が描く次世代知識ベースのイメージは、それぞれの専門分野を背景にして焦点の置く位置が必ずしも同一ではない。ワーキンググループとしても単一の知識ベース・アーキテクチャを提示した訳ではないが、要素技術の相互関係を明らかにし、幾つかの次世代知識ベースの枠組を示している。これはI C O Tの知識ベース基本ソフトウェア構築に向けて参考になるものと思われる。

本報告書の出版にあたり、本研究の機会を与えて下さった淵一博I C O T所長、原稿執筆にご協力をいただいた委員およびオブザーバの方々、ならびにWGの活動・運営にご協力および助言をいただいたI C O T第5研究室の藤井室長、牛駒室長、および研究員の方々に感謝する。

平成元年 4月

K S A - W G 主査 石塚 満
責任編集者 堀内 英一

目次

昭和63年度K S A - W G 活動報告書	1
石塚 満 主査 (東京大学 生産技術研究所)	
論理のモデル理論は知識システム・アーキテクチャに貢献しうるか	17
中川 裕志 (横浜国立大学 工学部 電子情報工学科)	
仮説推論と他要素技術との相互関連性について	26
飛鳥井 正道 (キャノン(株) 情報システム研究所)	
仮説推論・協調的問題解決に基づく画像理解システム S I G M A	30
松山 隆司 (東北大学 工学部 情報工学科)	
問題解決の高速化・並列・分散処理	42
石田 亨 (N T T 情報通信処理研究所)	
知識コンパイル	47
元田 浩 (日立製作所 基礎研究所)	
説明学習に基づく統合化学習アーキテクチャ	65
榎木 哲夫 (京都大学 工学部 精密工学教室)	
高次推論と計算量	77
國藤 進 (富士通(株) 国際情報社会科学研究所)	

活動報告書

KSA:知識ベース・ソフトウェア・アーキテクチャ

石塚 満（東京大学 生産技術研究所）

1. はじめに

KSAワーキンググループは、新しい知識処理技術を取り入れた次世代知識ベース構築への枠組、アプローチを明らかにすべく活動を行った。前年度は仮説推論、問題解決モデルと知識表現、分散協調問題解決、知識獲得、設計型エキスパートシステムと5つの重要な要素技術に分割してサブワーキンググループを形成し活動したが、今年度は個々の技術の相互関連を強く意識し、統合的な視点、枠組みを打ち出すことを意図して、KSAワーキングに集まって議論を進めた。分散協調問題解決（DIC）と、設計問題知識獲得（KAD）に関しては、重要課題であり未整理で残された課題も多いことから、引き続きサブワーキンググループとして議論を進めることにした。

2. 活動方針

次世代知識ベースの枠組みについて議論し、その構築に向けてのアプローチを明らかにすることを目的にして活動した。重要な要素技術については、各委員の専門に応じて担当をお願いした。担当は以下のように決めた。

長澤委員：トップダウン・システムアーキテクチャ

松山委員：仮説推論、分散協調

小林委員：学習（特にEBL）

原口委員：ロジック+ α

國藤委員：高次推論と計算量

石田委員：並列と分散協調の関係、高速化手法

元田委員：知識コンパイル

中川委員：ロジック+ α （デフォルト推論、サーカムスクリプション）

溝口委員：トップダウン・システムアーキテクチャ

小川委員：分散協調

榎本委員：学習（特にEBL）

石塚主査：仮説推論、制約指向

個々の要素技術の中だけでの議論でなく、相互関連、統合的枠組の探求を意識して検討を進めることとした。

具体的活動内容はおおまかに次のようにして行った。

- 1) 関連分野の研究者（外部の研究者を含む）の講演に基づく討議
- 2) 文献調査による最新の研究動向の把握
- 3) 委員からの意見（レポート提出2回を伴う）に基づく討議

3. 活動内容

上記の1) 2) 3) 項の分類に従って、以下に順に記す。

3. 1) 関連分野の研究者の講演とそれに基づく討議

- a) 画像理解における推論方式 — 松山隆司委員（東北大） [6月15日]

画像理解システムSIGMAで用いられたボトムアップ、トップダウン解析の統合による推論法について説明があった。画像処理の低レベル処理部を除外すれば、仮説推論の枠組で画像理解を考えることができる。すなわち、正しい画像特徴の割合が最大になるようなsubsetの仮説を探ることが目標になる。

- b) 高次推論と計算量に関する問題提起 — 國藤進委員（富士通国際研） [6月15日]

高次推論（帰納、類推、仮説推論）をシンボル処理で実現する場合の計算量の壁についての話があった。知識表現のクラスと計算量の概念との関連の研究の重要性が指摘された。特に、高次推論機能の指數的組合せ爆発の問題を、多項式時間推論が可能なレベルにするという目標が示された。

並列計算による効率化の期待が大きい、並列計算が有効である問題のクラスは狭いことが示されている。また次のような学習スキーマの提案があった。

$$\begin{aligned} \text{学習システム} &= \text{学習プロトコル} \times \text{学習アルゴリズム} \\ &= (\text{例の提示法} \times \text{オラフル}) \times \\ &\quad (\text{演繹的アルゴリズム} + \text{非演繹的アルゴリズム}) \end{aligned}$$

- c) 形式言語の帰納推論と計算量 — 榊原康文氏（富士通国際研） [6月15日]

例からの文脈自由言語の文法の学習において、正負の例文のみでなく、例文の構造的記述（学習プロトコル）を与えることにより、文法の帰納的学習が多項式時間内に効率化可能である事を証明し、Angulinの方法との比較を行った。

○計算量の比較

正提示	正則のクラス	多項式時間
完全提示	正則言語	NP困難
極小最適教師	正則言語	多項式時間
〃	人脈自由言語	?

○ Angulin & Shapiro の方法との比較

Angulin & Shapiro の方法

最少最適教師 + 非終端記号質問

機原の方法

極小最適教師 + 構造的記述 (内部ノードのラベルを持たない)

c) 局所的制約解消機構を用いたスケジューリング

— 沼尾雅之氏 (IBM東京基礎研究所) [7月14日]

論理型言語、プログラムの視覚化、制約プログラミングに関連する研究として、PROEDIT/PROEDIT2、Update Propagation Network、ProBool、Constraint Prolog、Scheplanの説明があった。その後、特にProBoolによる仮説推論の方法と制約を簡単に記述できるようにしたConstranit Prologについて説明があった。

ProBoolはPrologにおける述語の値の領域を真、偽の2値から自由ブール代数に拡張したものである。不完全な知識や、ある仮説を前提として成立するような述語は、その証明に用いた仮説をそのまま述語の値とすることにより表現できる。述語の値としては任意の命題論理式がとれるので、仮説間の関係も論理的に記述できる。仮説推論はPrologの導出原理を用いて実行され、得られた仮説の無矛盾性は保証される。

Constraint PrologはProBoolの制約表現をホーン節定義の部分とは独立な制約ルールとして記述し、制約ルールによるパターンの監視により、実行の制御をしよう改良したものである。

c) 類似性の推論 — 原口誠委員 (東工大) [7月14日]

類推研究の歴史 (過去、現在、未来) の流れを概観した説明があった。

類推とは類似性に基づく推論であり、1) 極大類比の検出、2) ルールの変換と類比の対応、3) 既存の類似性を前提とした新たな類似性の帰結へと研究を進めてきた。類似性の検出とそれに基づくルールの変換による類推はPrologの演繹推論の枠組みで実行可能である。しかし、ルール変換後にinstantiateされないものが残ってしまうことがあり、このときには検出された類似性からその拡張を推論する必要がある。すなわち、

検出された類似性 A	質問 Q
質問 Q に答えられるような A の拡張	

これはルール変換の仕方をルールで与えられることに対応しており、今までのルールの変換もimplicitにガイドルールが与えられていたと考えることができる。ガイドルールをexplicitに与えない場合は概念/述語の関係を与え、関係の近いものを優先させることで拡張を一義的に決められる。

ガイドルールや関係の決定の仕方は今後の課題である。

e) Explanation-based Learningとその周辺 — 小林重信(東工大) [8月24日]

知識システムにおける知識獲得支援のためのシーズとして、(A) determination による類推が紹介され、次に(B) EBLとSBLの融合による学習が議論された。

(A) determination による類推の紹介

類推は図式的に述べれば

『ソース領域SにおいてPとQが成立し、ターゲット領域TでSと同じくPが成立すれば、TでQが成立する』

なる推論である。このような類推は強い(含意的な)領域理論が実際に与えることができない領域で有効であるが、推論の正当性を保証する理論(類推の正当性を与える背景知識)が欲しい。またTでの事実Q(T)の推論において元のソースSが持つ情報が用いられること(ソースの非冗長性)が必要である。ラッセルらは、この類推の正当性とソースの非冗長性を保証する背景知識として

『任意のxに対してP(x)ならばQ(x)であるか、あるいは
任意のxに対してP(x)ならば $\sim$ Q(x)である』

なる形式の知識を考えこれをdeterminationと呼んだ。このdeterminationを組み入れた演繹システムから類推の前提P(S)、Q(S)、P(T)からQ(T)が論理的に導かれる。

このようなdeterminationは

(イ) 論理プログラミングで容易に実現可能、

(ロ) 強い領域理論が(実際に)構築不可能な多くの問題領域や物理的プロセス、合理的にagentによって決定されるプロセスにおいて成立することから知識システムにとって有用であると考えられる。

Discussion: 確かにdeterminationによって類推は演繹推論の一部となり概念的にも整備されたが、そのdeterminationをどうやって獲得するのが今後の大きな問題である。特に強い領域理論が望めない分野でのdeterminationの獲得のしやすさ等を調べる必要があるだろう。

(B) SBLとEBLについて

SBL(similarity based learning)とEBL(explanation based learning)の長所と短所を比較検討し、両者の長所を取り込む形で知識ベースの知識を洗練化する学習方法が論じられた。アイデアをごく大まかに述べれば、EBLにおいてマクロを作るときの基本操作である事例に対する説明構造(証明構造)の節の裁断と項の変数化を、複数の事例に対する説明構造のそれに拡張してやることである。そのためにプロトキンの語のleast generalizationの概念を用いて、複数の説明構造の共通点を抜きだした説明構造のマクロ(最少EBG)を考える。これによって、継続的な事例提示に対処できる。また通常のマクロ生成方法では事例の数が増えたとき、マクロの数も増え、したがったマクロ検索などの操作性が

劣化するのに対し、マクロの数を抑えることができる等の利点がある。

最少EBGの問題点は、事例に対する説明構造が本質的に異なるとき、その最少EBGはもとの事例を再現できないことが生じうることである。そのために、

『後戻りなしに具体例が再現でき、かつマクロの数が最少』

という操作性規範を導入した最少EBGによる学習を論じた。このように、実行システムが効率よく具体例を再現できるという操作的な概念の学習と言うこともできる。

また例としてハノイの塔の操作的概念の学習に関する実行列が示された。

Discussion: 上記の操作性規範を満足するためには、事例の集合を分割し、その分割要素に対する最少EBGを考えなければならないが、その分割をどうするか？

f) 連想の仕方の学習 — 元田浩委員（日立）〔8月24日〕

幾何の問題解決、特に補助線問題を題材にして、連想用知識をいかにして獲得・学習するかが論じられた。そのために以下のような問題解決に対する考え方を採用している。

『人は問題解決が終了した時点で、問題の全体構造ではなく、ある意味のある部分構造を（後で利用するために）記憶する。その意味のある部分とは、幾何の補助線問題の場合は図形パターン戦略と呼ばれる連想用の知識である。

また問題解決がうまくいかないというフラストレーションがその図形パターン戦略を学習するときの引き金になる』

フラストレーションには、前向きなもの（証明がそれ以上進まない）と、後向きのもの（不確定サブゴール）の両者がある。図形パターン戦略とはそのようなフラストレーションが観測された時に、図形条件をチェックし、新たな点や線分（補助線）を生成し、作業領域に補助線を引くことにより得られる新条件をassertする手続きである。このような図形パターンを学習するために、

1. 補助線を必要とする問題を与えてフラストレーションとその解消願望を発生させる。
2. 過去に得た図形パターン戦略を用いて、または教師による教示によって補助線情報とそれにとまなう新条件を得て、問題を解決する。
3. 完成した証明木を分析し、図形パターン戦略を生成するために必要な部分を抽出する。
4. 抽出された部分木を、EBGと同じような手法で一般化して図形パターン戦略を得る。

Discussion: EBGでは概念学習、つまり、『ある事実を示すことができれば、それはある概念である』といった概念の性質ない定義にかかわるルールを学習することが目的だが、一方フラストレーションに基づく図形パターン戦略の学習にお

いては、問題解決において困ったときのとき方を指示する手続きの知識となっている。

フラストレーションの形態とそれを解消する連想用知識を、個別的な問題領域で調べれば、図形パターン戦略のような連想用知識でうまく問題解決と学習が可能な分野があるのではないか？フラストレーションに基づく学習は我々の直感とよく一致している。

g) 設計の諸相と知的CAD — 長澤勲委員（九工大）

建築設計支援システム — 手越義昭氏（広島工大）、

渚氏（和田建築技術研究所）[11月 9日]

設計の部門間分業、設計作業の流れ、設計作業のシステム化段階等、設計の諸相に関する講演があった。特にAIと設計について以下の指摘があった。

— AIの役割は現在稼動している設計システム内の膨大な知識を設計者に返すところにあるだろう。

— 設計者の要求を設計システムに速く反映するにのAIは役立つだろう。（これには適切な知識表現を選択することが必要）

— 設計データベースはAI研究とデータベース研究の接点にもなっている。（例えばObject指向データベース）

同時に実用的なレベルにまで開発が進められた建築設計支援システムのデモンストラクションが行われた。

3. 2) 文献調査による最近の研究動向

1988年8月に開催されたAAAI-88での発表論文を中心に各委員が分担して調査を行った。以下にその概要を記す。

■中川裕志委員担当分

• Kolaitis, P. G. and Papadimitriou, C. H.

"Some Computational Aspects of Circumscription", AAAI-88

• Gelfond, M. and Lifschitz, V.

"Compiling Circumscriptive Theories into Logic Programs Preliminary Report", AAAI-88

極小限定(circumscription)は、述語論理に基づく堅固なモデル論により、基礎理論としての重要性が認識されている。しかしながら、高階論理で記述されているために一般的な計算法がなく、実用化のための課題となっている。そこで、極小限定が一階述語論理で表現でき、機械的に計算可能となるような述語あるいは公理系のクラスを求めることが重要となっている。このようなクラスとしては、Lifschitzによるseparableというクラスが知られている。

Kolaitisはseparable よりも一般的なクラスで極小限定が一階述語論理で表現できるものとして、存在限量化一階論理式を示している。しかし、論理プログラムなどで表現される全称限量子については、論理プログラムの極小限定が一階述語論理式であることの必要十分条件は、その論理プログラムがbounded であることであり、これを調べる手続きは決定不能であることを示している。

一方、Gelfond は極小限定と論理プログラムとの直接的な関係を利用して、ある論理式の極小限定と同じ結果を計算できる論理プログラムを、その論理式から導こうとしている。ただし、この論理プログラムへの変換は関数を含まない節形式の論理式でしかも層状のものに限定されている。

(討論)

・関数記号を含みドメインが無限となる場合には、極小限定を一階述語論理で行うことは不可能なので、有限のドメインでの実用的な問題を探すことも重要ではないか。

■赤埴氏 (NTT、オブザーバ) 担当分

・Ginsberg, M. L.

"A Circumscriptive Theorem Prover", AAAI-88

ある論理式が与えられた公理系の極小限定かどうかを検査する定理証明器として、de KleerのATMSを拡張したものを応用している。

$\neg p$ の形の論理式(ATMSの仮説)の集合D、それ以外の論理式の集合Tにおいて、論理式 q が r によるconfirm されるとは、

1. $T \cup \{r\}$ が充足可能
2. $T \cup \{r\} \models q$
3. r がDの要素の選言標準形

であると定義する。このとき、 q が極小限定の論理的帰結かどうかを検査するためには、 q をconfirm する r (ATMSの環境)が求められ、 $\neg q$ がunconfirm であることを示せば良い。

ATMSの拡張としては、論理式 q の環境を $\langle e_1, e_2 \rangle$ のように、 q が成立する最も一般的な(ATMSの極小な)環境 e_1 と、 $\neg q$ が成立する最も一般的な環境 e_2 の対で表現する。また、述語 $\forall x p(x)$ と、その基礎例 $p(a)$ との順序を定義することにより、一階述語論理に拡張している。

(討論)

・ $\neg q$ のunconfirm という条件がなければATMSと同じである。 $\neg q$ のunconfirm という条件は、複数のモデルが存在した場合に極小なモデルを選択するときに利用される。

・拡張されたATMSは後向きであるため、バックトラックが不可避である。

■元田浩委員担当分

・Valiant, L. G.

"Functionality in Neural Nets", AAAI-88

ひとつひとつの動作が遅い、多数の素子からなるネットワークにおける学習について論じている。ここでは、

- ・複数の種類の学習が可能
- ・新しく学習することは前に学習した結果を乱さない (cumulative)
- ・前に学習した概念を使って新しい概念を階層的に学習することが可能 (hierachical)
- ・高々数ステップで学習完了

である学習を考えている。

(討議)

・ニューラルネットにおいて例えば連言を学習する場合、 $A \wedge B$ と $B \wedge A$ の学習は個別にしか行えず、連言を本当に学習させるためには交換法則をさらに学習させる必要があるが、この学習法も同様の欠点を持つのではないか。

・ネットワークのそれぞれのノードをひとつの事例に対応させると事例推論に対しても同様なアプローチが利用可能ではないか。

■小田利彦委員 (リコー) 担当分

・de Kleer, J.,

"A General Labeling Algorithm for Assumption-based Truth Maintenance",
AAAI-88

de KleerのATMSをpositive clauses (仮説の選言) を取り扱えるように拡張する場合、ラベル計算の無矛盾性と完全性が失われるために、従来はhyperresolution を用いることにより無矛盾性と完全性を維持していた。本論文では、仮説の否定を導入することにより、hyperresolution を用いずに、より効率的にラベル計算の無矛盾性と完全性を維持する方法を提案している。ただし、この方法はラベルの無矛盾性は維持されるが、完全性はinterpretationの環境 (Default Logic の extension) でしか維持されることが示されている。

(討論)

・標題に記述されたgeneral というより、仮説の否定を導入したことにより positive clausesを効率的に扱えるようになったという論文である。

■飛鳥井正道委員 (キャノン) 担当分

・Koff, C. N., Flann, N. S. and Dietterich, T. G.

"An Efficient ATMS for Equivalence Relations", AAAI-88

- Forbus, Kenneth D. and de Kleer, Johan

"Focusing the ATMS", AAAI-88

- Dixon, Michael and de Kleer, Johan,

"Massively Parallel Assumption-based Truth Maintenance", AAAI-88

Koffの論文は、ATMSにおいて $x = y$ などの同値関係を効率的に扱えるようなATMSを提案している。de KleerのATMSのように同値関係をConsumer Architectureで管理するかわりに、同値データベースで管理することにより、同値類内の項数の3乗の計算量を2乗の計算量に効率化している。

Forbusの論文は、ATMSを用いた問題解決における探索の制御をよりきめ細かに行おうというものである。従来のATMSのConsumer Architectureと異なり、探索の焦点となる環境focus environmentの定義を任意の仮説の集合に拡張し、focus environmentから含意される（従来のATMSでは無矛盾な）前件を持つルールを実行することにより探索の制御を行う。

Dixonの論文では、並列処理によりATMSを高速化した並列ATMSのコネクション・マシンによるアルゴリズムを提案している。並列ATMSでは、それぞれのプロセッサと無矛盾な環境に対応させることにより、従来のATMSのラベル計算におけるラベルの環境の極小性、無矛盾性などの処理が不要となっている。実験結果により、並列ATMSが飛躍的に処理速度を向上させ、インプリメンテーションを容易にしたことを示している。

（討論）

• Dixonのコネクション・マシンによる実現が目新しい他は、ATMSのminorな改善と言う感じである。

■ 榎本哲夫委員担当分

- Barletta, R. and Mark, W.

"Explanation-Based Indexing of Cases", AAAI-88

事例推論のための事例記憶を組織化する上で、事例検索を容易にするためのインデックスの方法として、EBLを事例推論に応用したEBI (Explanation-Based Indexing) を提案している。

従来の事例推論における事例に対するインデックスの方法では、事例の類似性に着目した帰納的な方法が用いられていたために、数多くの事例が必要であるだけでなく、本質からはずれたような「偶然の一致」による特徴をインデックスにするとという欠点があった。EBIではこのような欠点を克服するために、事例の類似性ではなく、事例の背後にある因果則や原理原則を用いた事例理解によりインデックスを行っている。

(討論)

・事例からのE B Lによる説明知識をどのように領域知識に取り込み、不完全な領域知識を完全にしていくかが重要ではないか。

■石田亨 (N T T) 担当分 — Partial Global Plan 関係

・E. H. Durfee and V. R. Lesser(Univ. of Massachusetts)

Incremental Planning to Control a Blackboard-Based Problem Solver,
AAAI-86

Using Partial Global Plans to Coordinate Distributed Problem Solvers,
IJCAI-87

Predictability Versus Responsiveness: Coordinating Problem Solvers in
Dynamic Domains, AAAI-88

統合型の協調プランニング手法Partial Global Plan (P G P) について、関連する三編の論文(1986-1988)から、その研究状況が報告された。まず、ブラックボードモデル上でのdecision controlとして、従来の純粋にdata-directed な手法に加え、問題解決の状態空間の抽象化とその上での観測データのincremental clusteringを行うことにより、近未来についてたてられたpartial planをlong-term view に基づいてincremental に修正していくという、goal-directed な手法を提案、vehicle monitoring Problem Solver (経時的snapshotの位置センタデータから乗り物の種類・航跡を同定する問題) について例証した。さらにさまざまな協調の方法の統合化モデルとして、P G P、node plan、local planの三種のプランからなるネットワークモデルを提案し、各プラン内でのactivityの順序管理を行うActivity Map、部分解の求解や解の合成を管理するSolution-Construction Graph の生成を通して、満足解を導出する(山登り法による)手法について述べられた。Multi-Agent の協調型プランニングでは、他者agent の行為や将来の相互干渉についての予測(predictability)が必要になるとともに、予測のズレに対する素早い対処(responsiveness)が要求される。そこで、time-cushionというプランニング時の時間余裕をP G Pにもたせることにより、P G Pを再構成することなくロバストなプランニングを実現する手法を提案し、実験結果から効率向上への貢献が示された。

■矢沢利弘氏(電力中研 — 寺野委員代理) 担当分

・S. Minton(CMU)

"Quantitative Results Concerning the Utility of Explanation-Based
Learning." AAAI-88

E B Lによる学習の目的は、生成された知識が真にその後の問題解決実行時の効率改善に寄与することにある。しかしながら、事例から知識をやみくもに創り出す

だけの従来の多くのEBLのアプローチでは、逆に知識が増大したが故に、効率をかえって悪化してしまうことも考えられる。特に問題解決の制御知識獲得を目指す著者らのPROGIDYのようなシステムでは、その可能性が大きく、そのルールが作られたことによりどれほどの資源（推論ステップ数etc.）の節約が達成されるのか、ルールを適用する際のコストパフォーマンス（マッチングコストetc.）はどれくらいか、などを考慮したutility 尺度の導入が必要になる。本論文では、説明に基づいて生成された制御ルール記述の単純化（compression）によるutility の向上を目指すとともに、生成の度にそのルールのutility を算出することによって、その有用性を評価し生成ルールを取捨選択していくシステムを提案している。

■石塚満委員担当分

・A. Giusberg(ATT Bell Lab.)

"Knowledge-Based reduction: A New Approach to Checking Knowledge Base for Inconsistency and Redundancy," AAAI-88

ネットワークを介して複数人が協同で知識ベース開発を行う際の知識ベース管理機構として、前向きプロダクションシステムに対する無矛盾性と非冗長性をチェックするメカニズムを提唱している。そのために、各ルールに対し、その帰結部が導出される根拠になっている症候（finding）のリテラルをラベル、ルールがその依存関係（レベル）の順に一つずつ処理されるときに生成される不完全なラベルを部分ラベルが記録され更新されていき、これらの（部分）ラベル群を管理することにより、知識ベースの冗長性・矛盾性をチェックするというもの。

■榎本哲大委員担当分

・P. Struss(Siemens Corp.)

"Global Filters for Qualitative Behaviors," AAAI-88

物理システムの挙動予測を導出する従来の定性推論では、現実の解としては存在しえない挙動をも導出してしまうことがある。この理由としては、local criteria のみに基づくことから、本来は識別すべき挙動におけるある時点での状態が、定性的に同一視されてしまうことにより、挙動の全体像、即ち完全な履歴を参照することなく、無意味な状態遷移の分岐を生じることにある。本論文では、対象システムの挙動をトポロジカルに解析するための手法である位相面解析の考え方を導入し、位相面上での空間的な制約（ex. 同一システムの挙動軌跡が枝分かれしたり、交差したりすることはないetc.）を定性的に表して大域的な制約として用いる。これにより、局所制約に基づいて生成される不必要な状態分岐が除外できることになる。失点-ばねの二次系システムの挙動予測に対して適用を試みている。

■原口誠委員担当分

A. J. Bonner(Rutgers Univ.)

"A Logic for Hypothetical Reasoning," AAAI-88

「現在のルールベースRに対し、将来別の情報を仮定して、どんな事実Cの成立を結論付けられるか」といったqueryをルールベースRにindependentに扱うことができる直感主義的Horn Logicの拡張論理を論じている。そこではimplicationやorは、古典論理のそれに対し、より構成的な意味をもち、かつmaximal model存在等、Logic Programと極めて類似した性質をもつ。故に、直観的な解釈とcomputationalな強みをもったものになっている。

■小林重信委員担当分

Explanation-Based Learning (EBL)に関する下記に関する合計39編の論文の紹介があった。

レビュー	1件
一般化の方式	7件
操作性規範	3件
不完全制	3件
制御不可能制	4件
EBLとSBLの統合化	7件
問題解決	4件
応用	10件

EBLに関するこれまでのほぼすべての論文をカバーしていることになる。

3. 3) 委員からの意見に基づく討議

各委員に各々の担当課題に関する中間的な報告と、他の課題との関係・統合的な枠組に関する意見と、計2回の報告をお願いし、次世代知識ベースの姿について議論した。以下にまず第1回日の議論の概要を記す。

■國藤進委員 — 学習と計算量

学習モデルと計算量克服が重要である。富士通国際研榊原氏出席の会議COLTでは理論的な学習研究者が多く、AIでは理論的に見て無理な学習をしようとしているという批判があったそうである。計算量が少ない学習で解けるクラスは現在のところ驚くほど小さい。準最適な解を求める方向に進むのではないか。

溝口：NP完全でも解けないと決め付けるのはまずい。現実には有限な問題の大きさとの関係を考えるべき。

小林：理論的学習の人と機械的学習の人がかい離しているようだ。

榎木：C A Iなどにうまくつかえるような学習理論でなければならない。

■中川裕志委員 — ロジック + α

仮説推論を高速化するには領域知識をうまく利用することが重要。

石塚：問題をうまく切りだせば仮説推論が活用できる場が多い。

長澤：本当に役に立つ道具が小道具あるいは発見的知識が固有の領域ごとにゴロゴロしている。これらを統一的に扱う努力が必要。

溝口：A T M Sにもcontext スイッチなどで余分な知識を書けば高速化できる。
(この知識も一種の領域知識である。)

飛鳥井：篠原氏によるA T M SにおけるNoGoodを実行中に学習する方法がある。

■小林重信委員 — 問題解決の枠組

エキスパートシステムは合成的問題への適用が中心になる。

基礎的な知識の集積→類型化→探索問題+推定問題

國藤：浅い知識によるエキスパートシステムしかニューラルネットやファジーには置き代わらない。

榎木：個性の強いA I的手法と一般的なシステム的手法の組合せが重要。

■石塚満委員 — 仮説推論

数理計画法をホーン節の推論へ応用する展開が期待される。

中川：deductive databaseなどへの応用を考えてどうか。

■溝口理一郎委員 — トップダウン・システム・アーキテクチャ

知識レーベル学習と記号レベル学習の対比。インタビューをうまく組み込む方法を考える。領域固有の知識を使わないとうまいインタビューができない。M I Sでは最初に与える述語をどう決めるかが重要。

■元田浩委員 — 知識コンパイル

深い知識を用いた知識コンパイルの枠組みについて話があった。

中川：似たような問題に対してどの程度よくコンパイルされたレベルから使い始めたら良いかむずかしい問題であろう。

■原口誠委員 — ロジック + α

シンボル間の弱い結合を利用した類推が重要。つまり意味的に近いものを利用するのでその近さを定義するようなシソーラスを作ることが重要であろう。

■ 榎本哲夫委員

既存の知識から説明できない状況（インパス）がトリガーとなって学習する方法、インパスの解消法が学習の結果である。

溝口：インパスにおいて誤った知識の適用はC A Iなどにおいては問題。

原口：過去のインパス解消の事例を自動的に組織化することは困難である。

■ 石田亨委員

推論高速化、並列A I、分散A Iについて

－並列化できるところとHeuristicsで効果を上げるところと、どう切り分けるかが課題

－並列アルゴリズムのためのHeuristicsというのものもあるのではないか

■ 飛鳥井正道委員

知識ベースシステム構築上の問題点、仮説推論について

■ 新田克己委員

法律に関するエキスパートシステムにおける記号、その効用と限界について

－法律においては、述語さえバインドされていない

－法的推論では優先度を用いた推論が必要

2回目の討議では、各要素技術間の相互関係を明らかにして、次世代知識ベースの統合的枠組、アプローチについて議論した。以下にその概要を記す。

■ 石塚満委員

次世代知識ベースの課題として、「知識の幅の拡大」と「知識獲得のボトルネック」をあげ、演繹的推論機構を骨格とし仮説推論など高次人工知能技法を付加したアプローチの必要性を指摘、Rasmussen によるオペレータの情報処理モデルとの対応から、ヒューリスティックな経験知識を基幹に置きながらも、これをバックアップするため、高速化された推論機構が働くようなアーキテクチャが示された。

■ 中川裕志委員

「論理のモデル理論は知識システムアーキテクチャに貢献しうるか」

領域知識の中でも、特に推論制御のための知識の重要性が指摘され、E B L や C B R による方策への期待が述べられた。極小限定や一般論理プログラミングの意味論などのモデル論的アプローチのもつ長所と局所的知識コンパイルへの可能性が示された。またこれに付随する問題として、計算量的な問題と、局所的モデル大

域的モデル間での知識整合性の維持、文脈依存性を扱うための組織的体系の整備が課題として挙げられた。

■小川均委員

分散協調問題解決サブWGの議論をまとめた報告があった。分散協調問題解決の定義と分類の明確化が図られた。要素技術として、エージェント間での処理に関する事項として、①問題・知識分割と割り当て、②協調、③通信、④問題解決プロトコルに整理が図られた。エージェント自身の処理に関する事項は、①他のエージェントに関する推論、②不完全で矛盾する情報の取り扱い、③問題解決、に整理された。

■石出亨委員

並列と分散の関係、区別についての話があった。そして、並列・分散問題と並列・分散環境とは区別して考えるべきである。逆に並列環境の並列性をフルに利用するには分散問題に取り組むべきである等の考えが示された。

■小林重信委員

問題解決の統合化および学習の統合化と題して考えが示された。問題解決の基本的な方法として1) Rule-based Reasoning (RBR)、2) Case-based Reasoning (CBR)、3) Search-based Reasoning (SBR)を挙げ、これらは相補的な関係があり、統合化が期待される。学習方式についてはSBL (Similarity-based Learning)、EBL (Explanation-based Learning)、類推を挙げ、適用領域を示した。

■榎本哲夫委員

学習システムの一つの統合的アーキテクチャを提示し、受動的学習から能動的学習への方向が示された。またEBLの応用についても話があった。

4. まとめ

次世代の知識ベース・ソフトウェア・アーキテクチャについて、新しい知識処理技術を取り入れた枠組について議論した。取り上げた重要な要素技術は次の通りである。

ロジックの拡張（特に類推、非単調論理）

仮説推論とその高速化手法

高次推論と計算量

制約問題解決

学習（特にEBL）

知識コンパイル

並列と協調分散処理

トップダウン・システム・アーキテクチャ（特に設計問題）

各委員の研究に基づく意見と共に、最新の海外の研究についての調査も行った。

本KSAワーキンググループの各委員の意見は、ICOTテクニカル・メモとしてまとめる予定である。ICOTならびに他機関における今後の知識システム・ソフトウェアに役立つことを期待します。

論理のモデル理論は 知識システム・アーキテクチャに貢献しうるか

中川 裕志

(横浜国立大学 工学部 電子情報工学科)

1. 効率化へ向けて

・領域向きの知識システム

実用性のある知識システムを目指す流れは、(1)汎用の知識システムにおいてヒューマン・インタフェースを向上することによって対象領域の専門家が汎用システムを使いこなせるようにする流れ、と(2)領域向きのシェル、に分かれる。いずれにしても、計算量を適度に抑えるためには対象領域固有の知識を利用せざるを得ない。

・領域知識の分類

領域知識は少なくとも、(1)事実に知識と(2)推論制御のための知識、に分けることができる。このうち形式化すなわち計算機への移植がむずかしいのは(2)の推論制御のための知識である。つまり、対象領域の専門家自身も明確に意識していない、ないしは定式化していない知識である。この種の知識を獲得するひとつの方策として Explanation Based Learning や Case Based Reasoning が期待される。将来的に見れば、コネクショニストモデルによる学習のアプローチは有力だと考えられる。しかし、この方向はシステムと外界との境界面、すなわちパターン認識など非記号的な外部入出力と記号表現との間をつなぐ処理が現状では主力である。したがって、すぐにこの方向から従来の記号処理を完全に(あるいは大きな部分を)置き換える可能性には疑問が残る。逆に言えば、記号処理の仕事でも部分的には置き換えると効率化が計れる可能性を探ることがよい戦略である。

2. 論理からのアプローチ

上記の効率化の視点に立つと、人工知能の主流のひとつである論理を基礎にする研究は、このような状況に何等かの貢献をなし得るのかどうかは、見極めておくべき論点である。そのために、まず人工知能における論理の役割について概観しておく必要があるだろう。

・モデル理論の長所短所

論理型言語以外のところで論理が役だっているのは、論理のもつ数学的性質の美しさを利用したモデル理論からのアプローチであろう。極小限定や層状な一般論理プログラムの意味論はこの路線での研究である。このようなアプローチの長所

は、例外として特別な処理を必要とすることが（原理的には）ないことであり、与えられた知識から導出可能な解（あるいは新たな知識）に対して組織的な意味付けができる点である。なお、一般論理プログラムとは、前提に否定を含んでもよい論理プログラムである。層状とは、否定が必ず他の述語によって定義されることを保証するような構造のプログラムである。そのモデル論的意味論は論理の長所を知識処理システムに活用するさいに重要となる基礎を与える。付録A.に、最近提案された有力な意味論である完全モデル意味論を示した。

一方、論理を計算化した場合の処理量は知識の規模が大きくなると膨大となり、多くの場合、組合せ爆発を起こして現在の計算機の処理能力を越えてしまう。このような論理およびそのモデル理論の性質を考えると、大規模な知識ベースをもつ知識システムを全面的に論理あるいはモデル理論の枠で捉えることは計算量的な困難が付きまとうことが分かる。

3. モジュール化

知識を計算量的困難を引き起こさない程度の大きさに分割できることは、重要な性質である。このようなモジュール化は領域知識の性質により可能性があることをエキスパートシステムの研究成果は示唆している。しかし、効率よく問題を解決するためには、さらに、問題を単純な探索問題の部分と深い推論を必要とする部分に分割できることが望ましい。そうすれば、探索部分の高速化はコネクションマシンなどを利用して、大きな効率化が狙えるのではないか。ただし、そのような分割ができない問題は局所的に並列化できないため全体の並列化を行うしかないが、それを無理にやっても、どの程度、高速化に自身寄与するかは未知数（あるいは疑問）。すると、やるべきことは与えられた知識がこのような分割可能かどうかを機械的に調べる方法の開拓である。一般的には難しい問題だが、例えば関数のない一般論理プログラム（いわゆる universal formula）などはこのような分割がやりやすい。つまり、大量な事実の探索は単位節とのマッチングに絞られるし、その上での論理操作も比較的単純（連言や選言）だからである。

・知識コンパイル

上記のような分割が可能であるとすると、極小限定のもつ知識コンパイルの側面を利用することができる可能性がある。例えば、例外を現す abnormal 述語の定義を極小限定で計算しておくことなどが、応用できる。また、EBLやCBRも活用しやすくなるのではないか。

・大域的モデルから局所的モデルへ

知識コンパイルなどの処理は知識獲得段階で行っておくことになるから、かなり大規模な知識ベースに直接適用することも不可能ではない。しかし、現実の応用では、知識の局所的性質のみで解決できる場合が多いと予想される。もちろん、

実行時の効率を考えれば、知識を局所化することは重要である。このような点を考えると、モデル理論に基づく知識コンパイルのような処理も局所的に行っておく方法は魅力的である。ただし、局所的モデルと大域的モデルの間での知識の整合性を維持するという問題は残る。

・文脈依存的解釈の必要性

局所的モデルは大域的モデルという文脈の中で利用される。このような文脈依存性のある問題は従来の方法論の不得手とするところである。思い付くものといえば高々isa関係などによる知識階層化程度である。(Gelfond, Lifschitz の提案した与えられた公理系からその極小限定と等価な論理プログラムを変換して導出する方法でもisa階層までしか扱っていない。付録B、参照) 文脈依存性をより組織的に扱える体系が望まれる。問題を記述するという点においては、状況理論ないし状況意味論が、文脈依存性を直接に表現する意味論として有力である。ただし、状況理論では、問題の記述が中心で具体的な推論に関しては研究が進んでいない。状況理論に基づく推論システムの試みのとしてETL中島氏の提案しているPrositがある。その具体的メカニズムはまだ明かではないが、文脈依存性を扱う文脈の切り替えなどには、前記の推論制御の知識などが有用ではないかと考えられている。

このような状況において知識処理システムのアーキテクチャを考慮するために検討すべき点は、まとめると以下になる。

- ・論理に基づく整合性のある意味論の確立
- ・知識のモジュール化
- ・モジュール内での論理的整合性
- ・モジュール間での整合性
- ・外部環境すなわち文脈依存性を考慮した推論

付録

A-1. 層状プログラムの意味論

1. 層状プログラムと反復不動点意味論

極小限定と関係の深い論理プログラムの意味論の研究は、最小不動点意味論[Apt82]からSLDNFの完全性[Jaf][Llo]の研究へと大きな発展を見せた。しかし、プログラム中に否定を含む一般論理プログラムの意味論の構築には、従来の単純な延長線上では不満足が残った。そこで、有力な方向として、否定述語が階層化されて現れるという制限を満たす層状プログラム(Stratified Program)が研究され、多くの重要な結果が示された[Apt87][Geld87][Prz87]。層状プログラムの意味論の研究において極小限定との重要な接点が存在する。この節では、この点について説明する。まず、層状プログラムの定義から始める。

【定義】 層状プログラム[Apt87]

一般論理プログラムPが層状プログラムであるとは、Pが条件(1)(2)を満たす共通部分を持たない節の集合 $P_1, \dots, P_n$ で表される場合である。

(1) P_i の節の頭部の述語および体部に現れる正の述語は、その定義節が、 $\bigcup_{j \leq i} P_j$ に含まれる。

(2) P_i の節の体部に現れる負の述語は、その定義節が $\bigcup_{j < i} P_j$ に含まれなければならない。 □

この定義における節の集合 P_i を層(stratum)と呼ぶ。また、条件(1)(2)を満たすPの分割 $\{P_1, \dots, P_n\}$ を層化(stratification)と呼ぶ。条件(1)(2)を満たせば、負述語は自分自身を呼び出すことがない。つまり、負述語はそれを定義する述語が全て別の述語であることが保証できる。このことは、次に示す反復不動点意味論と呼ばれる層状プログラムの意味論にも反映されている。

【定義】 反復不動点意味論[Apt87]

$\{P_1, \dots, P_n\}$ なる層化を持つ層状プログラムPを想定する。基礎例の集合 $M(i-1)$ に層 P_i に属する節を作用させて新しい基礎例の集合 $M(i)$ を得る操作を T_i とする。このとき、次式により $M(i)$ ($i=1, \dots, n$)を計算するとしよう。

$$M(0) = \emptyset, \quad M(i) = \bigcup_{j \geq 0} T_i^j(M(i-1)) \quad (i=1, \dots, n)$$

(ただし、 T_i^j は T_i をj回作用させること)

このようにして計算したときに $M(n)=M(n-1)$ となったならば、これをPの反復不動点という。□

ところで、与えられたプログラムが層状プログラムであったとき、層化は一意に決まるとは限らない。しかし、 $M(n)$ は層化によらず一意であることが証明されている。[Apt87]

さて、極小限定との関係に話を戻そう。ここで重要なのは次の定理である。

【定理】 点別極小限定と反復不動点の一致[Lif87b]

層化 $\{P_1, \dots, P_n\}$ を持つ層状プログラムPでは、次の性質が成り立つ。

Pを(直接的ないしは間接的に)定義するのに用いられる述語の集合をQとする

と、

$\wedge \text{Circum-p}[P;p;Q]$: プログラムPにおけるQをパラメータとしたpの
 $p \models P$ 極小限定の連言

を満たす唯一のHerbrandモデルを持ち、これは反復不動点M(n)に一致する。□

この定理はひとつの見方によれば、点別極小限定を計算する方法として層状プログラムが与えられるとみれるので、極小限定の計算化という点から興味深い。逆の見方では、層状プログラムのモデル論的意味論を点別極小限定が与えるともいえる。

2. 完全モデル意味論

論理プログラムあるいは演えきデータベースと、極小限定を含む非単調論理との関係を議論する方法としてPrzymusińskiは完全モデル意味論という意味論を提案している[Prz87][Prz88]。完全モデル意味論を考えた動機から始めよう。次のプログラム[Prz87]について考えてみる。

```
avoid-math(x)
  ← businessman(x) ∧ ¬good-mathematician(x),
businessman(Iacocca)
```

このプログラムはふたつの極小モデルM1, M2を持つ。M1では avoid-math(Iacocca) がtrueとなり、期待通りのモデルである。しかし、M2では good-mathematician(Iacocca)がtrueとなり期待に反するモデルであるが、やはり極小である。この例では、容易にわかるように、本体部の負述語を優先的に計算することにより、M1を選ぶことができる。完全モデル意味論はこの直感を定式化したものである。ここでは、次に示すような結論部が述語の選言であってもよい一般的な場合を考える。

$C1 \vee \dots \vee Cp \leftarrow A1 \wedge \dots \wedge Am \wedge \neg B1 \wedge \dots \wedge \neg Bn$

この論理式に関して次に述べる優先順位を考える。

【優先順位】[Prz87]

- (1) 前提部の否定述語B1, ..., Bnは結論部より高い優先順位を持つ
- (2) 前提部の肯定述語A1, ..., Amは結論部と等しいかより高い優先順位を持つ
- (3) 結論部の述語C1, ..., Cpは同じ優先順位を持つ □

さて、このような優先順位において、高い優先順位の述語をできる限り極小化しようとしたモデルとして完全モデルが与えられる。形式的には次の定義による。

【定義】 完全モデル[Prz87]

あるモデルMからある基礎例を削除し別の基礎例を追加して同じプログラムを満たす新しいモデルNを作ったとしよう。このとき、NがMよりpreferableであるとは、Nに追加した基礎例の述語の優先順位が、Mから削除した基礎例の述語の優先順位より低いことである。Mよりpreferableなモデルが存在しないとき、Mを完全モデルという。□

完全モデルは必ずしも一意ではないが、次の定理が知られている。

【定理】 完全モデルの極小性[Prz87]

全ての完全モデルは極小モデルである。また、もしプログラムが前提部に否定を含まなければ、逆も成り立つ、すなわち全ての極小モデルは完全モデルである。

□

【例】 完全モデルでない場合 [Prz87]

プログラム $p \leftarrow \neg q$, $q \leftarrow \neg p$ はふたつの極小モデル $M1=\{p\}$, $M2=\{q\}$ を持つ。しかし、優先順位の定義により、 p は q より順位が高く、 q も p より順位が高いため、 $M1$ は $M2$ より preferable かつ $M2$ も $M1$ より preferable なため、完全モデルではない。

□

また、プログラムが層状の場合には、必ず1個以上の完全モデルが存在することが知られている。さらに、結論部が述語1個からなる層状の論理プログラムの場合にはHerbrandモデルに限定すれば、完全モデルが一意的に決まることも重要である。

さて、いよいよ完全モデルと極小限定との関係について述べる段階に来たが、その準備として、完全モデル意味論を定義しよう。

【定義】 完全モデル意味論 [Prz87]

(必ずしも層状とは限らない)一般の論理プログラム P の全ての完全モデルの集合を $\text{perf}(P)$ と書き、これによって完全モデル意味論を定義する。すなわち、 P からの完全モデル意味論による定理 F とは、 $\text{perf}(P) \models F$ を満たすものである。□

ここで極小限定との関係として、次の定理が重要である。

【定理】 完全モデル意味論と極小限定の等価性 [Prz87]

P を $\{P1, \dots, Pn\}$ という層化を持つ層状の論理プログラムとする。このとき、自由変数を持たない任意の論理式 F に対して、 $\text{perf}(P) \models F$ は $\text{Circum}[P; P1 \dots Pn] \vdash F$ の必要十分条件である。□

この定理により、優先順位付き極小限定から導ける論理式は完全モデル意味論に対する推論エンジンで計算でき、ひいては論理プログラムにおけるSLDNF resolutionが優先順位付き極小限定の健全な推論メカニズムとして利用できる可能性のあることが示唆される。

B. 論理プログラムへの変換

極小限定は一般には、計算可能な一階述語論理式にはならないので、実用化に困難が付きまとう。そこで、極小限定と論理プログラムとの直接的な関係を利用して、ある論理式の極小限定と同じ結果を計算できる論理プログラムをその論理式から導こうという研究が最近発表された [Gelf88]。この方法は、今のところ、関数を含まない節形式の論理式でしかも層状なものにしか適用できない。関数を含まない節形式とは、全称限量子の内側に存在限量子のない論理式、すなわち全称論理式である。

さて、上記の条件を満たす節の連言で表される公理系を A とする。このとき、 $P1, \dots, Pk, Z$ の各々は、 A の中の共通部分を持たない述語の集合とする。また、 U は名前の一意性を表す公理であるとする。さらに、 A 中の節の各々には Z の要素である

述語は高々1個しか現れないとする--《假定》。このとき、 $\text{Circum}[A \wedge U; P_1 > \dots > P_k; Z]$ を Circ と略記することにする。次に、 $P_1 > \dots > P_k$ という順位の層化を持つ層状の論理プログラム Π のインタープリタ Ans の動作を以下のように定義する。

```

      | yes,
      if  $U \wedge \text{Circum}[\Pi; P_1 > \dots > P_k] \models W$ 
 $\text{Ans}(\Pi, W) =$  | no,
      if  $U \wedge \text{Circum}[\Pi; P_1 > \dots > P_k] \models \neg W$ 
      | 不定 上記以外の場合

```

ただし、 U は Unique Name Assumption を表す公理である。最後に $\text{Replace}(A)$ と $\text{Resolve}(A)$ というふたつの変換手続きを導入する。

$\text{Resolve}(A)$: A の節を用いて、 Z の各要素に関して resolvent を求め、 A に追加する。《假定》により resolvent は Z の要素を含まない。

$\text{Replace}(A)$: Z の各要素 Z_i に対して、 Z_i と同じ数の変数を持つ新しい述語 $*Z_i$ からなる新述語の集合 $*Z$ を導入する。そして、 A 中の $\neg Z_i$ を $*Z_i$ で置き換える。

手続き Replace により、与えられた公理系を論理プログラムと見なした場合の節の頭部に現れる負のリテラル $\neg Z_i$ は新しく導入された正のリテラル $*Z_i$ に変換されるので、この手続きによって実行可能な論理プログラムが得られている。また手続き Resolve は $*Z_i$ がもともと $\neg Z_i$ であったときの意味を保てるように Z_i との関連を付けるものである。このとき次の定理が成り立つ

【定理】 極小限定と論理プログラムの等価性

節の集合として与えられた公理系 A から、まず $\text{Resolve}(A)$ つきに $\text{Replace}(A)$ を施して得られた論理プログラム Π が、 $P_1, \dots, P_k, Z, *Z$ に関して $P_1 > \dots > P_k > Z > *Z$ という順位の層化となるような層状プログラムとしよう。このとき、 A 中の述語の任意の基礎例 W に対し Circ との間に次の関係が成り立つ。

- (1) $\text{Circ} \models W$ の必要十分条件は $\text{Ans}(\Pi, W) = \text{yes}$
- (2) W が $P_1, \dots, P_k$ に属する述語の基礎例の場合は、 $\text{Circ} \models \neg W$ の必要十分条件は $\text{Ans}(\Pi, W) = \text{no}$
- (3) W が Z に属する述語の基礎例の場合は、 $\text{Circ} \models \neg W$ の必要十分条件は

$\text{Ans}(\Pi, \text{Replace}(\neg W)) = \text{yes}$ $\square$

《例》 論理プログラムへの変換

次の公理系について考える。

```

Tweety  $\neq$  Opus , Tweety  $\neq$  Joe , Opus  $\neq$  Joe
bird(x)  $\wedge$   $\neg$  ab(x)  $\rightarrow$  flies(x),
bird(Tweety), penguin(Opus),
penguin(x)  $\rightarrow$  bird(x)
penguin(x)  $\rightarrow$   $\neg$  flies(x)
 $\neg$  flies(Joe)

```

これを上記の方法で変換した論理プログラムは次のようになる。

```

flies(x) ← bird(x) ∧ ¬ab(x).
bird(Tweety).
penguin(Opus).
bird(x) ← penguin(x).
*flies(x) ← penguin(x).
*flies(Joe).
ab(x) ← bird(x) ∧ penguin(x).
ab(Joe) ← bird(Joe).

```

最後の2個の節は手続きResolveによって得られたものである。等価性の定理によりこの論理プログラムのインタープリタAnsでの実行結果が、もとの公理系の極小限定と一致することが保証されている。もちろん、全ての公理系をこの変換手続きによって層状プログラムに変換できるわけではない。

参考文献

- [Cla] Clark, K.L.: Negation as Failure, in Logic and Database Plenum Press, pp.293-324 (1978)
- [McC80] McCarthy, J. : A form of non-monotonic reasoning, Artificial Intelligence Vol.13, pp. 27-39 (1980).
- [Rei] Reiter, R : circumscription implies predicate completion (sometimes). Proc. AAAI-82, pp418-420, (1982)
- [Apt82] Apt, K.R., Emden, M.H.V.: Contributions to the Theory of Logic Programming, JACM 29(3), pp.841-862 (1982)
- [Jaf] Jaffar, J., Lassez, J.-L., Lloyd, J.: Completeness of the negation as failure rules, Proc. IJCAI-83, pp500-506, (1983)
- [Lif84] Lifschitz, V.: Some results on circumscription, AAAI Workshop on Non-Monotonic Reasoning, pp. 151-164 (1984).
- [Llo] Lloyd, J.W.: Foundations of Logic Programming, Springer-Verlag (1984) 邦訳 論理プログラミングの基礎、産業図書 (1987)
- [She] Shepherdson, J. C. : Negation as failure, the Journal of Logic Programming, Vol. 1, No.1, pp. 51-79 (1984).
- [Lif85a] Lifschitz, V.: Computing circumscription, Proc. of 9th IJCAI, p. 121-127 (1985).
- [Lif85b] Lifschitz, V.: Closed-World Databases and Circumscription, Artificial Intelligence Vol.27, pp229-235, (1985)
- [Han] Hanks, S, and McDermott, D.: Temporal Reasoning and Default Logics, Yale/csd/rr#430 (1985)
- [McC86] McCarthy, J.: Applications of Circumscription to Formalizing Commonsense Knowledge, Artificial Intelligence Vol.28, pp.89-116 (1986)
- [Lif86a] Lifschitz, V.: On The Satisfiability of Circumscription,

- Artificial Intelligence Vol.28, pp17-27, (1986)
- [Lif86b] Lifschitz, V.: Pointwise Circumscription: Preliminary Report, Proc. of AAAI-86, pp.406-411 (1986)
- [Gelf86] Gelfond, M., Przymusinska, H.: Negation as Failure: Careful Closure Procedure, Artificial Intelligence 30, pp.273-287 (1986)
- [deK] deKleer, J.: An assumption-based truth maintenance system, Artificial Intelligence 28, pp127-162 (1986)
- [Lif87a] Lifschitz, V.: Circumscriptive Theories: A Logic-based Framework for Knowledge Representation, Proc. of AAAI-87, pp.364-368 (1987)
- [Lif87b] Lifschitz, V.: On the Declarative Semantics of Logic Programs with Negation, in Foundations of Deductive Databases and Logic Programming, Morgan Kaufmann, pp.177-192 (1987)
- [Prz87] Przymusinski, T.C.: On the Declarative Semantics of Deductive Databases and Logic Programs, *ibid*, pp.193-216 (1987)
- [Apt87] Apt, K.R., Blair, H.A., Walker, A.: Towards a Theory of Declarative Knowledge, *ibid*, pp.89-148 (1987)
- [Geld] Gelder, V.A.: Negation as Failure using Tight Derivations for General Logic Programs, *ibid*, pp.149-176 (1987)
- [Prz88] Przymusinski, T.C.: On the Relationship between Logic Programming and Nonmonotonic Reasoning, Proc. of AAAI-88, pp.444-448 (1988)
- [Gelf88] Gelfond, M., Lifshitz, V.: Compiling Circumscriptive Theories into Logic Programs Preliminary Report, Proc. of AAAI-88, pp.455-459 (1988)
- [Kol] Kolaitis, P.G., Papadimitriou, C.H.: Some Computational Aspects of Circumscription, Proc. of AAAI-88, pp.465-469 (1988)
- [Gin] Ginsberg, M.L.: A Circumscriptive Theorem Prover: Preliminary Report, Proc. of AAAI-88, pp.470-474 (1988)

仮説推論と他要素技術との相互関連性について

飛鳥井 正道

(キャノン(株) 情報システム研究所)

問題解決の過程では、対象についてすべてを知っていることはまれであり、通常は限られた知識に基づいて推論しなければならないことが多い。不完全な知識に基づいて結論を導き出すためのひとつの方法は、不完全な知識を補うためになんらかの仮説を立てて推論を行なうことであり、このような推論方法を仮説推論という。仮説推論は不完全な知識を扱うような問題を容易に記述できる枠組みを与えてはいるが、推論速度が遅いことなどいくつかの問題点を抱えている。ここでは de Kleer により提案された ATMS を用いた仮説推論を中心に、このような問題点を解決するための方法を、他の技術要素との関係を基にして述べてみたい。

1 高速化・効率化

仮説に基づいて推論を行なうときに、知識ベースの管理を行なうための機構として de Kleer の提案した ATMS [1] があり、仮説推論を行なうための効率的な枠組みを与えている。ATMS を用いた仮説推論システムは、問題解決機構と ATMS とから構成される。問題解決機構は、推論の実行ごとに推論の結果を理由付け (justification) として ATMS に与える。ATMS は与えられた理由付けを記録するとともに、これらの理由付けからデータがどのコンテキストで信じられているかという信念 (belief) を計算 (ラベル計算) して問題解決機構に返す。

ATMS を用いた仮説推論を高速化するための手法としては、並列処理により ATMS を高速化した並列 ATMS をコネクション・マシン上に実現した Dixon ら [4] の方法がある。

また、類似した前件を持つ理由付けのラベルを計算するときに、ATMS では前件の共通部分のラベルを重複して計算してしまうという欠点がある。ルールベースシステムを問題解決機構として用いる場合の解決方法として、ルールベースシステムにおけるパターンマッチングの高速化アルゴリズムである Forgy の Rete アルゴリズムにラベル計算を組み込むことにより高速化をはかるという Inference 社の ART [6] の方法など [7, 10, 13] がある。

2 探索

de Kleer の ATMS はすべての仮説を等しく扱うために、多くの仮説を扱うような問題では仮説の組合せ的爆発の問題が不可避である。このような大規模な探索空間から解を効率的に探索するための従来の手法は、下記の 2 つのアプローチに大別できる。

1. 探索の効率化
2. 探索空間の縮小

探索の効率化としては、基本的には問題解決機構が探索の焦点となる環境(focus environment)を制御することにより、効率的な探索を行なう de Kleer らによる方法 [3] が考えられる。さらには、問題解決機構と ATMS との間にインターフェイスを設けることにより探索の制御を行なう井上 [12] の方法など [15, 16] や、仮説に不確実性の情報を付加することにより探索の制御などを行なおうとする董ら [17] や劉ら [19] の方法など [20, 21, 22] がある。

探索空間の縮小としては、de Kleer により拡張された ATMS [2] を利用して仮説間に制約関係を記述する飛鳥井らの方法 [10] がある。しかしながら、仮説の論理結合などの簡単な制約関係しか記述できないのが問題である。董ら [18] は仮説に状況という概念を導入し、これを後述する時間関係の推論に応用しているが、同様に飛鳥井は仮説が導入された状況を考慮することにより、仮説の組合せを制限するための柔軟な枠組み [11] を与えようとしている。また、E B L により矛盾の学習を行なうことにより探索空間を縮小させようという矢沢らの方法 [23] もある。

3 基礎・拡張

3.1 論理

ATMS を論理的に基礎付け、拡張したものとしては Reiter ら [5] の CMS (Clause Management System) がある。CMS は、問題解決機構から与えられた命題論理式の集合 Σ と命題論理式 C から、極小支持節 S の集合を求める。ATMS は CMS において、 Σ をホーン節 (ATMS の理由付け) の集合、 C を命題 (ATMS のノード)、 $\neg S$ を仮説である命題の論理積 (ATMS の環境) に特殊化したものとなる。CMS は確かに ATMS を一般化してはいるが、極小支持節の計算は高価であり、逆に一般化したことにより ATMS の効率を犠牲にしている。同様に ATMS を含めた Truth Maintenance Systems のモデル論的な基礎付けを行ない、発想推論や Reiter のデフォルト論理などとの関連を考察したものとしては、井上 [14] による研究がある。また、最近自然言語処理において、様々な状況 (文脈) における意味の多義性などを扱うために状況意味論が提案されており、これを支える理論として状況理論がある。状況理論を論理的に解釈すると、命題の真偽が状況に依存して決定されるという点において、命題の真偽が仮説である命題の論理積のもとで決定されるという仮説推論と類似している。また、状況理論の支持という概念は、モデル理論の論理的帰結という概念に対応させることができるが、状況と命題の支持関係自体も命題として扱えることが、状況理論の特色であるように思う。いずれにせよ、状況理論と仮説推論などとの論理的な関係についての研究が必要であると思われる。

3.2 時間

時間を扱う推論としては、Allen の時間論理や、様相論理を用いた時相論理など論理によるアプローチの他に、古くは CONNIVER から Inference 社の ART や IntelliCorp 社の KEE などのように複数のコンテキストを扱うことによるアプローチがある。後者は、それぞれのコンテキストをある時点に対応付け、コンテキスト間に順序を定義することにより、時間的な変化を表現しようとするものである。ATMS も複数のコンテキストを扱い、それらのコンテキスト間に順序関係が存在する点では、これらのシステムと同様である。しかしながら、ATMS の順序関係は時間を表現しているのではなく、コンテキストをユニークに特徴付けている仮説の集合間の包含関係を表現している。従って、ATMS は組合せ問題に適しているが、組合せの時間

(順序) 関係に依存するような問題には適さないと言われている [8]。この問題を解決するために、ATMS の仮説に状況と言う概念を導入して、仮説が導入された時間的な順序関係などを扱うとする董ら [18] のアプローチがあり、仮説推論と時間を扱う推論との融合の試みとして興味深い。

3.3 分散・協調

ATMS を分散協調問題に応用した例 [25, 26] や、協調問題に応用した例 [24, 9] はあるが、いずれもエージェントが持っている信念の基礎集合を ATMS の仮説の集合に対応させ、エージェントの信念をこれらの仮説の集合から導出される命題の集合として表現しようとしている。しかしながら、ATMS では命題がどのような仮説の集合のもとで成立するかを、ラベルとして implicit に表現しているために、知識と信念の論理におけるような nested belief ($B_A(B_B(p))$) が表現できないことが難点であり、エージェントの信念を扱うための今後の課題であると思われる。

参考文献

- [1] de Kleer, J., An assumption-based TMS, *Artificial Intelligence* 28, pp.127-162 (1986)
- [2] de Kleer, J., Extending the ATMS, *Artificial Intelligence* 28, pp.163-196 (1986)
- [3] de Kleer, J. and Williams, B.C., Back to backtracking: Controlling the ATMS, *Proc. AAAI-86*, pp.910-917 (1986)
- [4] Dixon, M. and de Kleer, J., Massively Parallel Assumption-based Truth Maintenance, *Proc. AAAI-88*, pp.199-204 (1988)
- [5] Reiter, R. and de Kleer, J., Foundations of Assumption-based Truth Maintenance Systems : Preliminary Report, *Proc. AAAI-87*, pp.183-188 (1987)
- [6] ART Reference Manual, Inference Corporation (1987)
- [7] 飯島、成田、吉田、泉、仮説ネットワークを用いた仮説推論器、人工知能学会研究会資料 SIG-FAI-8701-2、pp.7-14 (1987)
- [8] 飛鳥井、ATMS について、ICOT 仮説推論SWG 資料HYR-2-2 (1987)
- [9] 飛鳥井、森沢、村田、浅野、仮説推論と協調推論の融合の一実現法、人工知能学会第2回全国大会、pp.137-140 (1988)
- [10] 飛鳥井、村田、森沢、浅野、エキスパートシステム構築ツールCHORUS(2)の仮説推論機能一、情報処理学会第37回全国大会、pp.1218-1219 (1988)
- [11] 飛鳥井、Situating ATMS、人工知能学会第3回全国大会発表予定
- [12] 井上、導出を用いた仮説探索、情報処理学会第35回全国大会、pp.1567-1568 (1987)

- [13] 井上、太田、仮説推論システム A P R I C O T / 0 による知識コンパイル、人工知能学会研究会資料 SIG-KBS-8805-6、pp.51-60 (1988)
- [14] Inoue, K., A Model Theory for Hypothetical Reasoning, *ICOT Technical Report TR-336* (1988)
- [15] Forbus, K.D. and de Kleer, J., Focusing the ATMS, *Proc. AAAI-88*, pp.193-198 (1988)
- [16] 西岡、堀、池田、溝口、角所、言語処理システムにおける ATMS の利用法について、情報処理学会第 37 回全国大会、pp.1117-1118 (1988)
- [17] 董、中川、不確実な知識における ATMS、情報処理学会知識工学と人工知能研究会資料 52-1、pp.1-8 (1987)
- [18] 董、中川、assumption と situation、日本ソフトウェア科学会第 4 回大会論文集、pp.119-122 (1987)
- [19] 劉、西田、堂下、不確実な情報による ATMS の制御、日本ソフトウェア科学会第 4 回大会論文集、pp.127-130 (1987)
- [20] 青山、石塚、仮説の重み利用による高速化機構を備えた仮説推論システム、人工知能学会第 2 回全国大会、pp.57-60 (1988)
- [21] D'Ambrosio, B., Truth Maintenance with Numeric Certainty Estimates, *Proc. 3rd Conference on Artificial Intelligence Applications*, pp.244-249 (1987)
- [22] Laskey, K.B. and Lehner, P.E., Belief Maintenance: An Integrated Approach to Uncertainty Management, *Proc. AAAI-88*, pp.210-214 (1988)
- [23] 矢沢、篠原、仮説に基づく推論における機械学習方式について、電子情報通信学会人工知能研究会資料 88-28、pp.1-8 (1988)
- [24] Martins, Joao P., Reasoning in Multiple Belief Spaces, Technical Report 203, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY (1983)
- [25] 横尾、石田、分散協調問題解決における ATMS の利用、人工知能学会第 2 回全国大会、pp.141-144 (1988)
- [26] 河本、川上、猪股、西村、分散協調型問題解決システム Discovery - オブジェクトの並列処理 -、情報処理学会第 36 回全国大会、pp.1501-1502 (1988)

仮説推論・協調的問題解決に基づく 画像理解システム S I G M A

松山 隆司

(東北大学 工学部 情報工学科)

1. はじめに

我々は画像理解における推論方式に関する研究を行っており、文献[1]においてその推論モデルを提案するとともに航空写真理解システム S I G M A を設計・試作した。本年度はモデルの持つ論理的意味について詳細に検討を加え、提案したモデルが仮説推論の拡張として定式化できることを明らかにした。このことから、仮説推論がエキスパートシステムや記号処理に基づく問題解決システムだけでなく、画像理解といった認識システムの論理的枠組みとしても有効であることが明らかになった。

一方、S I G M A の推論モデルは、オブジェクト指向計算に基づいた分散知識表現、協調的問題解決方式を基本に設計されている。すなわち、画像から認識された個々の対象物のインスタンス（たとえば家や道路）は、それぞれ独自の知識に基づいて独立に推論を行う能力を持った a g e n t とみなされ、周囲の環境や他の対象物との間の空間的關係についての推論を行う。本年度はこうした観点からの検討も行い、インスタンス間の協調的問題解決のための同期メカニズムについて考察した。

以下では、2 において S I G M A の推論モデルを概観し、3 では仮説推論に基

づく定式化、4では分散問題解決のための同期メカニズムについて述べるが、それらの詳細については文献[1][2]を参照していただきたい。

2. SIGMAにおける推論モデル

SIGMAは次の3つのエキスパートから構成されている(図1)。

(1)Geometric Reasoning Expert(GRE):このエキスパートは、システムにおける中心的な役割を果たすエキスパートで、対象物の幾何学的構造および対象物間の空間的關係に基づいた推論を行う。

(2)Model Selection Expert(MSE):このエキスパートは、GREからの要求によって起動され、今注目している対象物の具体的なモデルとその画像上での見え方を推論する。

(3)Low Level Vision Expert(LLVE):このエキスパートではMSEで決められた画像上での対象物の見え方に基づき、実際に画像を処理し、指定された(線や領域などの)画像特徴を抽出する。

SIGMAにおける空間的推論はGREおよび推論機能を持ったagentである対象物のインスタンスによって行われ、その方式は以下の考え方に基づいている。一般に、2つの対象物のクラス01、02の間の関係 $REL(01,02)$ は、 $01=f(02)$ 、 $02=g(01)$ のように2つの関数を用いて表現することができる。02のクラスのインスタンスを s とすると、関数 f は s をそれと関係 REL を満たすクラス01のインスタンス $f(s)$ に写像する。同様のことが関数 g についても成り立つ。SIGMAでは、ある対象物のインスタンスが見つかり、それとある幾何学的関係を満たす対象物に対する**仮定**(上の $f(s)$ に当る)が生成される。仮定は、

- (1) 目的とする対象物が存在すべき位置を表す領域
 - (2) 目的とする対象物の特徴に関する制約条件の集合
- として表現される(図2)。

GRE内の全ての部分情報(evidence)-対象物のインスタンスおよびそれらによって生成された仮定-は、共通のデータベース(図1のIconic Database)に蓄えられ、それらの間の**整合性**が調べられる。このデータベースには、対象世界における対象物の位置情報を表すための幾何学的空間(具体的には2次元配列)が用意

されており、全ての部分情報はこの配列上の領域として表現される。GREは、この配列上での領域間の位置関係（重なり）を利用し、関係のある部分情報を検索する。言い換えると、システム内の全ての部分情報が空間的位置情報に基づいて指標付けられていることになる。データベースにはこうした位置情報のほかに対象物のインスタンスや仮定に関するシンボリックな情報（属性や関係、制約条件）も蓄えられる。

今、クラス02のインスタンス s が関係 REL に基づいてクラス01の対象物に対する仮定 $f(s)$ を生成したとし、Iconic Database 中で $f(s)$ とクラス01のインスタンス t とが互いに重なったとする（図3(a)）。このことは、 $f(s)$ に課せられた位置的制約条件が t によって満たされたことを意味する。このとき GRE は、 $f(s)$ に付けられた他の制約条件と t の持つ属性とを比較し、それらの間の整合性を調べる。もし $f(s)$ と t との間に整合性があると、それらは1つの situation を形成すると言う。すなわち、situation とは互いに整合性のある部分情報の集合として定義される。図3(a)の場合、GRE は整合性のある2つの部分情報 $f(s)$ と t を統合（同一視）することによって、その situation の解を求める。この統合の結果、対象物のインスタンス s と t の間にシンボリックな関係 REL が設定される。このプロセスが SIGMA におけるボトムアップ解析による対象物の関係付けである。

一方、図3(b)に示すように、同じクラスの対象物に関する2つの仮定 $f(s)$ と $h(u)$ とが、互いに相手の制約条件を満足し1つの situation を形成している場合、GRE は次のような推論を行う。 $f(s)$ と $h(u)$ との間に整合性があるので、前のプロセスと同じく両者を統合する。しかし、今の場合 situation を構成している全ての部分情報が仮定であるので、前のようにインスタンスの間に関係を設定することによってその解を求めることができない。そこで GRE は、統合された仮定（各仮定に付けられた制約条件の積）を満たす対象物のインスタンスの検出を MSE に依頼する。この要求は MSE からさらに LLVE へと引き継がれ、実際の画像の中から目的とする対象物に対応する画像特徴が探索される。要求された対象物のインスタンスが検出されると、その結果が GRE に返される。GRE は、MSE より返されたインスタンスを Iconic Database に登録し、その新しいインスタンスと、仮定を生成していた元のインスタンス（図3(b)では s と u ）との間にそれぞれ対応する関係を設定する。こうした解析プロセスが SIGMA におけるトップダウン解析である。

以上述べたようにSIGMAでは、部分情報の蓄積、統合を通じてボトムアップ、トップダウンの2つの解析が1つの推論過程として統合され、状況に応じて-situationの中にインスタンスが含まれるかどうかによって-いずれかの解析プロセスが起動される。また、仮定の生成や記号的関係の設定は個々のインスタンスが独自の知識に基づいて行い、GREは部分情報の整合性の検証を通じてインスタンス間の協調、同期を実現している。

3. 推論モデルの論理的枠組み

画像理解における推論方式を考える場合、次の2つの問題点が重要となる。

(1) 情報の不足: シーンの記述を作成するのに必要な情報量に比べ、画像に含まれる情報が少ないうえ、セグメンテーションの不完全性によって意味のある特徴が抽出できないことがある。

(2) 誤った情報: セグメンテーションの不完全性や雑音によって誤った特徴が抽出されることがある。

よく知られているように、導出原理などによる通常の演繹的推論は、予め知識として与えられた論理式の集合によって規定された世界の情報を導出(検索)するだけで、新たな情報を生成するという機能は持たない。すなわち、通常の演繹的推論では、結論を得るために必要な完全な情報が予め与えられていると仮定しており、画像理解にそのまま適用できない。

一方、仮説推論は、与えられた知識だけでは観測事実が説明(証明)できないときに、システムが仮説を生成し、説明が可能となるようにしようとするもので、不完全な知識に基づく推論が可能となる。すなわち、仮説推論を用いると、シーンについて得られた部分情報と対象に関する知識に基づいて仮説を生成し、その仮説の妥当性を画像データを調べることによって検証するという推論が実現される。この方式では、生成された仮説と入力データとが矛盾しないかぎり、その仮説が正しいと考えることになり、情報が不足しているような問題に対しても何らかの結論を得ることができる。

SIGMAにおける推論方式は正しくこうした考え方に基づいており、仮説推論を用いてSIGMAの推論モデルの論理的枠組みを表現できる。以下では、S

I G M A の推論方式を述語論理を用いて説明する。

【例 1】認識対象 01 が 02 と関係 REL を満たし、01、02 それぞれのインスタンス a、b が認識されているとする。この情報を論理式で表すと次のようになる。

公理 1: $\forall x [01(x) \supset \exists y (02(y) \wedge \text{REL}(x, y))]$

事実 1: 01(a)

事実 2: 02(b)

このとき

ゴール: $\exists z (02(z) \wedge \text{REL}(a, z))$

を通常の導出反駁法で証明すると、 $z \mid f(a)$ という束縛が得られ、ゴールが証明される。ここで f は公理 1 を節形式に変換する際に導入されたスコールム関数である。

一方、S I G M A では、公理 1 と事実 1 から $f(a)$ という仮定が生成され、 $f(a)$ と整合性のある他の仮定やインスタンスが探される。今、 $f(a)$ が b と整合性があるとする、システムはそれらが同一の対象を表すと考え、 $f(a)=b$ という仮説を新たに生成する。この仮説を利用すると、 $\text{REL}(a, b)$ が新たな事実として分り、 $z \mid b$ という束縛が得られる。これが S I G M A におけるボトムアップ解析での推論方式である。

【例 2】次のような公理、事実が与えられたとする。

公理 1: $\forall x [01(x) \supset \exists r (03(r) \wedge \text{REL1}(x, r))]$

公理 2: $\forall y [02(y) \supset \exists t (03(t) \wedge \text{REL2}(y, t))]$

事実 1: 01(a)

事実 2: 02(b)

ゴール: $\exists z (03(z) \wedge \text{REL1}(a, z) \wedge \text{REL2}(b, z))$

このとき通常の推論方式では、与えられた情報からはゴールは証明できない。S I G M A では、公理 1、2、事実 1、2 から仮定 $f(a)$ と $g(b)$ を生成し、それらの間の整合性を調べる。整合性が確認されるとシステムは $f(a)=g(b)$ という仮説を生成する。その結果ゴールが証明され、 $z \mid f(a)$ という束縛が得られる。さらにシステムは、 $f(a)$ という仮定を満たす画像特徴の検出を起動し、 z に束縛されるインスタンスを探す。これが S I G M A におけるトップダウン解析における推論方

式である。

図4は、上の推論方式を仮説推論の枠組みとして記述したもので、SIGMAにおける仮説がどのように生成されるのかがよくわかる。すなわち、通常の仮説推論では、仮説として考えられる論理式が予め（固定）決められているのに対し、SIGMAでは $\text{ground term} = \text{ground term}$ といった形式の等式が動的に生成される。言い換えると、同一の対象を指している ground term の同定というのが、SIGMAにおける認識過程の中心的機能であるということになる。

通常の仮説推論では観測された情報は常に正しいとして推論が行われる。しかし、先に述べたように画像理解では、セグメンテーションの不完全性によって誤った情報が抽出される。このため、画像理解では、純粋な論理に基づいた仮説推論のほかに誤った観測事実の検出機能が要求される。文献[2]では $\text{maximum explainable observations}$ （知識と生成された無矛盾な仮説集合によって説明可能な観測事実の集合を考え、その大きさが最大となるような仮説集合を選択する）という概念を導入することによって仮説推論を拡張することを提案したが、その計算アルゴリズムはまだ明確ではない。

4. 協調的問題解決のための同期機構

これまでは大局的な観点から推論方式を述べたが、SIGMAでは個々の対象物のインスタンスが独自の知識に基づいて推論を行うという局所的推論のレベルがある。このレベルでは、多数の agent が協調しながら推論を実行する分散型問題解決が行われる。以下では、各 agent における知識表現と推論の同期機構について述べる。

SIGMAでは各認識対象のモデルはオブジェクト指向計算における1つのクラスとして表現され、実際に認識された対象物はそうしたクラスのインスタンスとなる。各クラスには認識対象の属性や isa 、 part-of 関係を表すためのスロットのほかに、そのクラスのインスタンスが推論を行うために用いる知識がルールとして蓄えられている。

ルールは、インスタンスが他のインスタンスと協調的推論を行うことができるようにするため、通常のプロダクション規則を拡張した形式として表現される。

すなわち、各ルールは図5のようにcondition部とaction部に加えてhypothesis部を持っている。

各インスタンスは、それが属すクラスに蓄えられているルールのcondition部を調べ、それが満たされるとhypothesis部を評価する。これによってそのインスタンスと関係がある他の対象物に対する仮定（上の $f(a)$ や $g(b)$ ）が生成される。仮定が生成されると、そのルールの評価は一時停止される。生成された仮定と整合性のある他のインスタンスがGREによって求められると、そのインスタンスが元の仮定を生成したインスタンスに渡される。これによって停止されていたルールが再び活性化され、action部が評価される。action部では仮定を満たす（GREから渡された）インスタンスとの間の記号的な関係をスロットに書き込むとともに、相手のインスタンスにそのことを通知する。このことによって、2つのインスタンスの間に整合性のある双方向の関係が設定される。新たな関係が設定されると、別のルールのcondition部が満たされ、そのルールを用いた推論が始められる。

今あるsituationが $\{f(a), g(b), c\}$ という部分情報から構成されている（これらの間の整合性がGREによって確かめられた）とする。GREは、インスタンス c をこのsituationの解と見なし、仮定 $f(a), g(b)$ を生成した元のインスタンス a, b に c を渡す。このことによって a, b 内で評価が一時停止されていたルールが同時に活性化され、 a, b, c の3つのインスタンス間での関係の設定動作が協調して行われる。言い換えると、 a, b 内で独立に評価されていたルール間の同期が、それらによって生成された仮定を通じてとられることになる。

実際のSIGMAでは、こうした空間的関係設定のためのルールのほか、複数のルールの評価を制御するメタルールなど、いくつかのタイプのルールが利用でき、各インスタンスによるより柔軟な推論が可能となっている。（詳細は文献[2]を参照）

5. おわりに

ここでは画像理解システムSIGMAにおける推論方式について検討を行い、

(1) 推論モデルの論理的枠組みが仮説推論の拡張として形式化できること

(2) 分散型問題解決における a g e n t 間の協調動作が仮説生成機能を持った
ルールによって実現できること

を示した。今後の問題としては、誤った情報に対する処理・推論アルゴリズムの
確立、処理・推論の並列化がある。

〔参考文献〕

〔1〕 松山, ハング: 画像理解システム S I G M A, 情報処理学会論文誌, 26
巻, 5号, pp.877-889, 1985

〔2〕 Matsuyama, T. and Hwang, V.: SIGMA: Knowledge-Based Aerial Image
Understanding System, PLENUM 1989 (to be published)

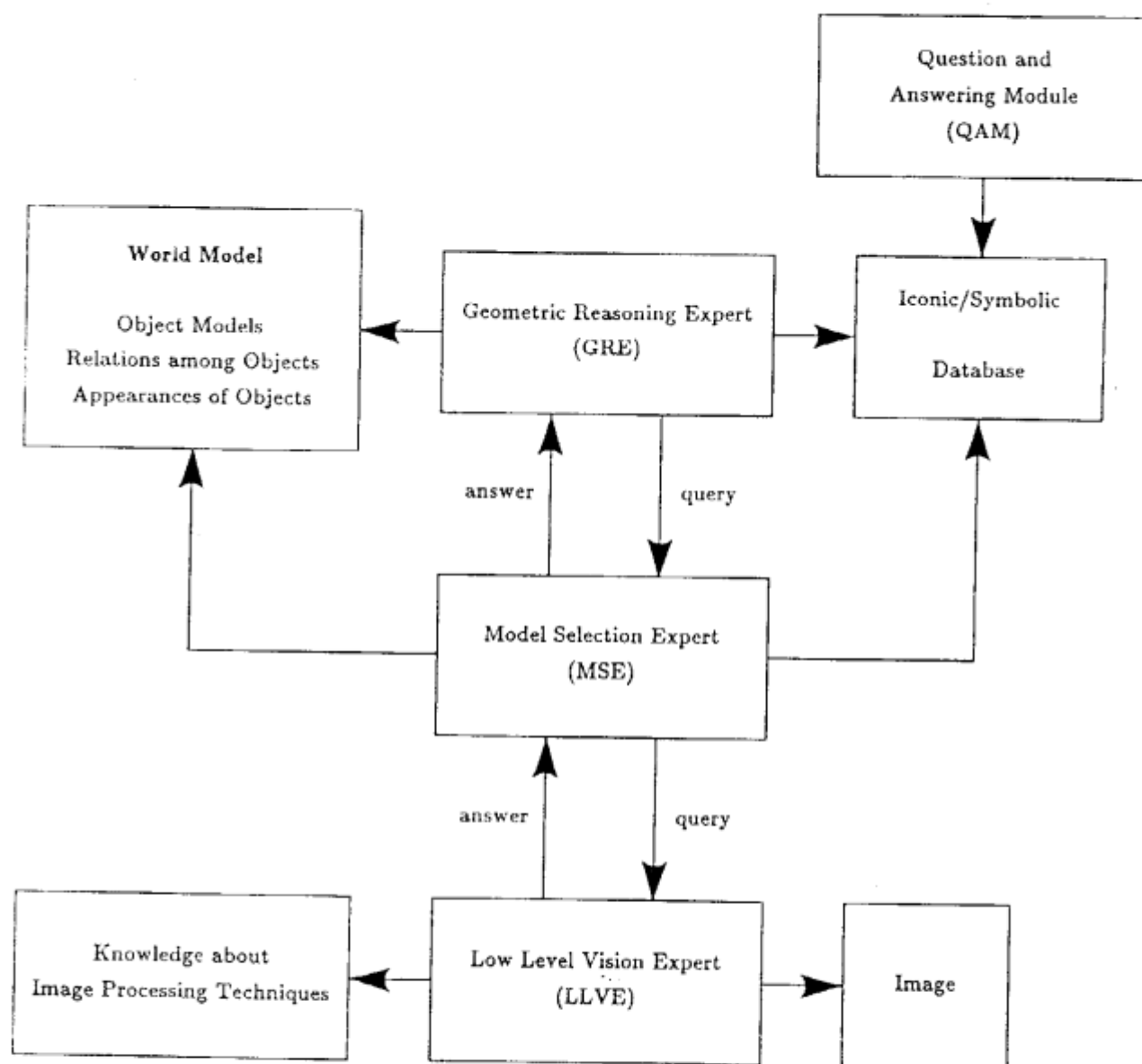
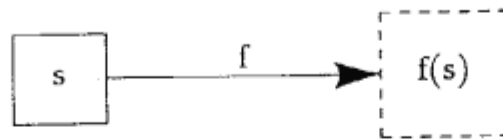
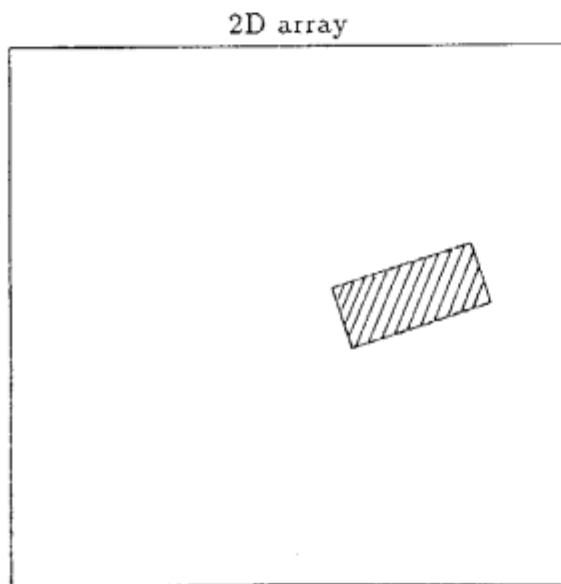


図1 システムの構成



(a)Hypothesis generation



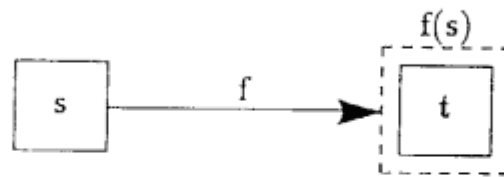
Iconic Description

Constraints:
 (AND (EQUAL OBJECT-TYPE ROAD)
 (AND (LESSP TOTAL-LENGTH 100)
 (GREATERP TOTAL-LENGTH 50))
 (AND (LESSP AVERAGE-WIDTH 15)
 (GREATERP AVERAGE-WIDTH 10))
 (AND (LESSP AVERAGE-DIRECTION 50)
 (GREATERP AVERAGE-DIRECTION 30))))

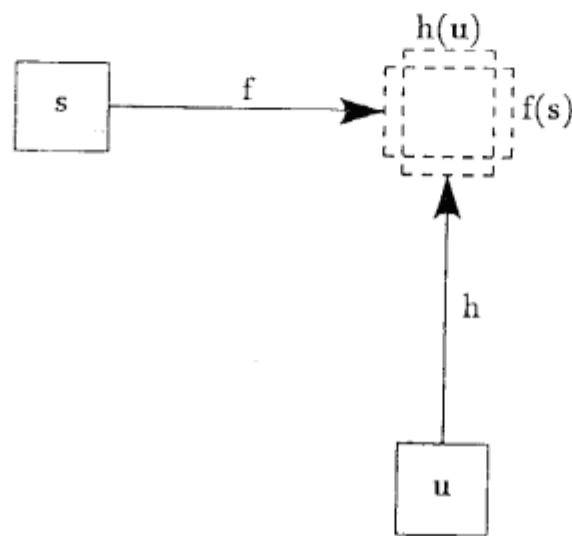
Symbolic Description

(b)Description of a hypothesis

図2 仮定の生成



(a) Situation which activates the bottom-up analysis



(b) Situation which activates the top-down analysis

図3 部分情報の蓄積

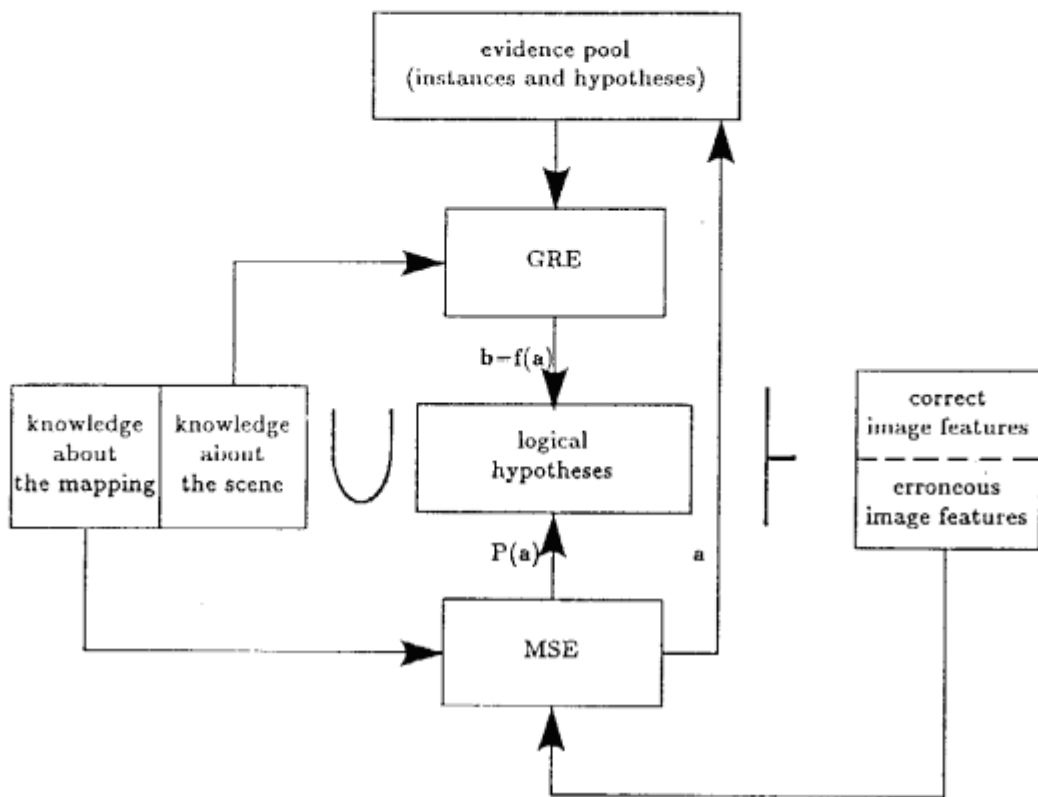


図4 SIGMAにおける推論の枠組

$R_{house-road-driveway}$
 <control-condition>: *along(self.front-road,self)*
 <hypothesis>: $P_{connect-house-road}$
 <action>: *self.connecting-driveway=solution.*

図5 空間的推論のためのルール

問題解決の 高速化・並列・分散処理

石田 亨
(N T T 情報通信処理研究所)

1. 問題解決の高速化

(1) 推論エンジンの高速化

プロダクションシステムの場合、Bliss版OPS5以降の高速化の実績は以下のとおりである。

- ・条件照合アルゴリズム ～2倍程度
- ・最適化 ～2倍程度
- ・処理系構築技術（記述言語、コンパイル方式） ～4倍程度

技術的には様々な手法が提案されているが、それほど大きな進歩は得られなくありつつある。今後は用途によって各種技術を使い分ける方向にあると思われる。

(2) 推論制御

推論制御知識を学習しようという試みが行われており、定量的な効果が示され始めている。

- ・PRODIGY EBLを利用 (Minton, IJCAI87, AAAI88).
- ・SOAR chunkingによる高速化.

しかし推論の高速化には様々な手法があるため、競争相手が多いため、推論制御知識の獲得が定着するかどうかはこれからが勝負と思われる。

(3) リアルタイム・エキスパートシステムにおける高速化

① 注視点制御

センサの入力フィルタを状況に応じてコントロール研究が行われている。

- ・ Adaptive Intelligent System (B. Hayes-Roth)

② 応答時間の保証

通常のリアルタイムシステムの手法 (Dynamic Stepの測定) は、以下の理由で AI システムでは使えない。

- ・ ルールの発火回数が予測できない。
- ・ 各ルールの発火に要する時間が予測できない。

推論実行中にタイマ割り込み (制限時間内で結論をだす) を行ったり、問題解決時間の概算する研究が行われている。

- ・ Interrupt Driven Architecture (B. Hayes-Roth)
 瀕死の患者の生命維持装置をモニタする研究。
- ・ Approximate Processing (Lesser, AI Magazine)

2. 並列処理

(1) 問題解決の並列性

① エキスパートシステム (Heuristic Programming) の並列性

プロダクションシステムの経験では、エキスパートシステムに含まれる並列性は以下のとおりである。

- ・並列条件照合： 5 ～ 10 倍
- ・並列発火： 2 ～ 9 倍

並列性は言語にではなく応用にある。Heuristic Programmingではほとんどサーチは行われない。おそらく、GHCで書いても並列PSで書いてもその並列性は似たようなものであろう。

② ATMS の並列性

ATMSは問題の本質が並列サーチである。de Kleerは以下の提案を行っている (AAA1-88)。

- ・各プロセサに1個のワールドを割り当てる。従って、ラベルはなし。
- ・各ワールド毎にデータ個数分の長さを持つビットベクトルを持つ。
- ・データが各ワールドで成立するかどうかは対応するビットのon/offで表す。

Connection Machine上での結果はSymbolics上の逐次処理系の70倍といわれるが、探索の制御が入ると並列化効果は落ちてくるだろう。

(2) 並列計算機の利用

- ① 並列マシンで得られるのは高々10～100倍の性能向上である。計算のオーダが変わるわけではない。コスト性能比をよくするのか、応用を限定してトップデータを出すのかという選択が必要であろう。

- ② 粒度と応用のマッチが必要である。

10台のPEを使う	-----	Heuristic Programming
1000台のPEを使う	-----	Search

3. 分散処理

分散プランニングの一連の研究の中に、並列処理とは異なる分散固有の問題をみることができる。分散プランニングはMulti-Agent PlanningとDistributed Planningに大別される。

① Multi-Agent Planning

Multi-Agent Planningには以下の2種の研究がある。応用としてはロボット（従って、例題はブロックワールドということになる）を頭において研究されてきた。

- ・複数のエージェントのプランを1箇所で構築。

自分のためだけのプランニング。

STRIPS(1971), NOAH(1977)

他のエージェントに処理を頼むことを含めてプランをたてる。

Konolige, Nilsson(1980)

複数のエージェントに協力して処理することを頼むためのプランニング。

Rosenschein(1986)

- ・独立に構築された複数のエージェントのプランを1箇所で調整。

排他的調整： 臨界領域を検出して同期信号を埋め込む。

Georgeff(1983), Stuart(1985)

② Distributed Planning

Distributed Planningは空間的に分散された問題（例題は複数のセンサによる軌跡の追跡(DVMT: Distributed Vehicle Monitoring Testbed)）を頭において研究されてきた。

- ・複数のエージェントが部分的なプランを交換しながら分散してプランニングを行う。

Durfee, Lesser(1987)

4. 並列と分散の関係

(1) 並列

問題を複数のプロセサで手分けして解く。

- ・ 並列プロセサの構成は既知。
- ・ 制御プロセサが存在。
- ・ プログラムを並列化するには、プログラム中の並列処理可能な部分を検出すればよい。

(2) 分散

分散された処理を統合して問題を解く。

- ・ 分散プロセサの構成は必ずしも既知ではない。
- ・ 制御プロセサは予め存在しない。
(どれかがその役割を果たすことはできる。)
- ・ 分散プログラムを統合するには、並列処理される複数のプログラムから同期すべき部分を検出すればよい。(Synchronizing Multi-Agent Plans)

(3) 並列・分散へのアプローチ

- ・ 並列マシンの上でも分散問題は解ける。(地理的分散についてもシミュレーションはできる。)逆に分散環境を利用して並列処理をすることもできる。(通信オーバーヘッドは大きくなるが。)
- ・ 並列・分散問題と並列・分散環境とは区別して考えるべきである。並列か分散かは問題の性質による、あるいは問題の解き方によるのであって、実行環境と混同してはならない。
- ・ 逆に並列環境の並列性をフルに利用するには、分散問題に取り組むことを考えるべきである。分散問題を集中環境で処理すれば、多くの計算機パワーを要し、大規模並列計算機が有効となるかもしれない。
- ・ 例えば、現在人間が分散して実行している処理をそのままプログラムし、中央の計算機で処理する(例えば、ネットワークオペレーション)ことを考えてはどうか。

知識コンパイル

元田 浩

(日立製作所 基礎研究所)

1. エキスパートの問題解決法と知識コンパイル

エキスパートシステムは、従来、エキスパートが持つ経験的知識（ヒューリスティックス）としての専門知識を主な対象として研究が進められてきた。経験則はタスクに直結した知識であり、問題の解決には極めて効率の高い性能を示す。しかしながら、経験則がうまく働くのはあくまでもその知識に適合した状況が現れたときに限られ、少し異なる状況には全く役に立たない。大規模なエキスパート・システムを構築する場合には、経験則という形であらゆる局面に対処できる知識を前もって収集することは不可能である。行き詰まった場合にエキスパートが熟考するプロセスを再現できなくてはならない。熟考できるためには対象を深く理解していなければならない。このようなアプローチの必要性はエキスパート・システムの脆弱性を解決する手段の一つとしても早くから指摘されている[Hart 82]。

深い理解に基づく熟考プロセスを深い推論と呼ぶことにする。このときに用いられる知識は、その分野における基礎的で原理的な知識や、考察の対象となるものの構造や部品の機能に関する知識であり、これらを総称して深い知識という。

我々が物事を理解したとき、頭（脳？、心？）の中に何かが記憶され、あるいはすでに記憶されている何かが再構成され記憶が強化される。例えば、我々は対象の構造に着目し階層的に把握し、かつ、注目している現象に注視の焦点を合わせることができる。しかも、個々の部品の機能を解釈する際に、その部品の果たす役割を考え、起こり得る挙動のうちの適切なもののみを抽出する。さらに部品の幾つかをまとめて一つの機能を持つグループと認識し、階層的に理解していく。この過程で、知識が構造化され、記憶が強化される。いったん理解すると、類似の構造に対しては、記憶されている解釈をそのまま適用する。このような現象は学習と深くかかわっている。既知の知識で理解できないとき、思い悩む。解法を教示されると、何故それで解けるのかと理解しようとする。理解できると記憶が強化される。このような能動的な学習により知識が構造化され、類似の問題に対しても適用可能な戦略知識が獲得される。

知識コンパイルという考えは、まさにエキスパートが、このような原理原則に戻って熟考するプロセスを通して、与えられたタスクに直接役に立つ強力な知識を生成する現象を計算機化しようという試みであり、人間の理解行為や学習行為の一面の機械化とも考えられる。

深い知識として何を用いるか、何を目的として知識を生成するか（どのような形の知識を生成するか）により、同じ演繹的な枠組みで診断、設計など幅広い領域の知識をコンパイルすることができる。この考えは、一見帰納的とも思える一般化（Generalization）や差別化（Discrimination）にも適用できる[Anderson 83-1]。新しい問題に直面したときの人間の思考過程は問題解決過程とみなすことができ、この解決過程をコンパイルすることによって、類似性、差異性に基づいた一般化などが自然に行える。

2. 知識コンパイラの枠組み

すでに述べたように知識コンパイラは種々の意味合いを持っている。知識コンパイラの一般的な枠組みを示すとすると図1のようになろう。広義の知識コンパイルには知識の獲得の仕方などのメタ理論の獲得(学習)を含めることもできる。事例を深く洞察することによって新たな概念を発見する過程も知識のコンパイルと考えることができる。太線の枠で囲んだものが狭義の知識コンパイルで、コンパイラーの入力が深い知識、出力が浅い知識に対応する。狭義の知識コンパイラは原則として記号レベルの知識を生成する。すなわち、コンパイルされた知識は与えられたタスクに対する有効性が向上するという意味で価値が付加されたと考えられることができるが、本質的に新しい知識が創造されたわけではない。深い知識として領域理論を考えた場合、これを出力とするコンパイラーも考えることが出来る。この場合は知識レベルの学習が行なわれる。インタビューの自動化などによって新しい知識を獲得することはこれに対応する。

3. 狭義の知識コンパイラ

通常、深い知識から浅い知識をコンパイルするというときのコンパイラは狭義の知識コンパイラである。狭義の知識コンパイラの入力対象となる知識は深い知識と呼ばれ、その出力は浅い知識と呼ばれる。ここで、「浅い」、「深い」という概念は相対的なものであり、コンパイルされて生成された知識はコンパイルされる前の知識に比べて「より浅く」なっているという意味である。従って、知識コンパイラにより「浅い知識」から「より浅い知識」を生成するという言い方も可能である。また、知識コンパイルは連鎖的であり得る。

最も深い知識は、対象の挙動に関する物理法則など原理・原則に関する知識や、対象がどのようなものかを表現するその構造や機能などに関する知識、つまり、対象を表現する世界モデル(Domain Model)ならびに支配原理(First Principle)を意味する。生成される知識の性質はコンパイルの目的によって異なる。

狭義の知識コンパイラは上述したように基本的には記号レベルの知識しか生成しない。しかし、コンパイルの際にエキスパートの知識を利用することにより、この枠を越えた知識を生成することも可能である。

4. 診断、設計への応用事例

筆者ら[Motoda 84]はプラントの診断用エキスパート・システムを開発した際に、深い知識から診断用知識を生成する方法を提案した。これは後にRESIDUEとして発表された手法[Finger 85]と類似の方法を採用しており、故障原因の絞り込みに必要なテスト用の知識を自動的に生成するものである。対象システム(原子力発電プラント)の性格上、数値シミュレータを深い知識の一部に使用している。

山口ら[山口 87, 88]は定性推論を診断用知識のコンパイルに応用し、汎用的な知識コンパイラを開発し、自動車や空調装置の診断に適用している。手法の特徴は深い知識を1)対象装置の構造と機能、2)物理原理、3)故障頻度や観測容易性、4)故障・微候と物理状態の対応、5)故障メカニズムに関する5種類に分類し、微候からこれらの知識を原因(仮設)側へたどる逆行推論と仮設から微候へたどる順行推論を組合せ、最終的に微候と仮設を結びつける知識をプロダク

ション・ルールとして生成するものである。最近、故障個所が同定された後、さらに故障モードまで同定するために故障モードを仮定した定性推論による挙動予測ができるように本手法を拡張している[林 89]。

設計用の知識コンパイルに関しては、まだ研究は多くないが取扱の対象となる深い知識（モデル）を持つという点で診断と共通点がある。しかし、設計と診断の最大の相違点は診断は対象モデルから出発するのに対して、設計は仕様を満足する対象モデルを生み出す創造的な性質を持っていることである。現在のところ、まったく独創的な発想に基づく設計を計算機によって実現することは望めないと考えるのが妥当である。

しかし、設計変更用の知識のコンパイルに関してはMurthyら[Murthy 87]のPROMPTにその方向性が示唆されている。エキスパートは既知の知識だけでは仕様を満足する設計案を生成することが出来ないとき、そのときまでに得られている設計案に基づき、深い知識を使い何がまずいのかを解析し、もとの物理方程式と照らし合わせ、どのような構造にすれば良さそうかを推論し、設計案を修正し、再び解析、評価する。PROMPTでは中間的なアプローチとして修正オペレータとモデルグラフという手法を使って、部品の構造と挙動との関係を再解析し、要求にあった規格外の部品を新たに考案する。

今、設計の仕様として「長さ L 、重さ W_1 以下、ねじり剛性 T_1 以上のはり」が与えられ、部品ライブラリには断面が円形の棒しかないとする。棒の重さ W 、ねじり剛性 T_0 は、材料の密度を D 、剪断弾性係数を G とするとそれぞれ次式で与えられる。

$$W = \pi R^2 L D, \quad T_0 = \frac{\pi}{2L} G R^4$$

従って、 G/D が大きな材料を選択する必要があるが、どんな材料を選んでも $T_0 \geq T_1$ なる最小の R に対しても $W \geq W_1$ となってしまうえば部品ライブラリの知識では設計できないという結果が得られる。しかし、断面が円形という条件を緩め一定であるとする、上式は次式のように変形される。

$$W = L \int D dA, \quad T_0 = \frac{1}{L} \int G r^2 dA$$

この式から dA は W に対しては r に無関係であるが、 T_0 に対しては r^2 に比例して効くことがわかるので、 r の小さい所から大きい所へ質量を移動すれば W を変化させないで T_0 を増加させることが出来る。PROMPTではこのような解析はエキスパートに任せ、結果を修正オペレータという形で与えている。このような知識を具体的な断面が円形である部品に適用すれば、例えば中空の棒にすれば仕様を満足させることが出来るとの結論が得られる。さらに、曲げ剛性に対して強い制約が追加された場合には、曲げ剛性に関する同様な解析により得られた深い知識を用いることにより、中空の円を中空の楕円にすればよいとの結論も得られる。

このような推論を機械化することは定性推論の技術の延長線上にあり、知識コンパイルの新しい技術として進展が期待されるものの一つである。

本報告では、知識コンパイルの新しい試みとして現在研究中の「階層的定性推論による深い知識からの機能知識のコンパイル」[Yoshida 88]と、「Frustration Based Learningによる深い知識からの戦略用知識のコンパイル」[Suwa 88]を紹介する。

いずれの方法も深い知識の変換による効率向上という点で、広い意味の記号レベルの知識の生成に関するものであるが、前者の場合には、取り出された知識が意味のある概念（例えば側路コンデンサ付き抵抗といった固まり）を表現しているという点で、後者の場合には、顕在化されていない情報を連想によって補完し、行間の意味を読みとるといった知識を生成しているという点で、単なる知識の変換ではない。

5. 階層的定性推論による深い知識からの機能知識のコンパイル

エキスパートが問題を階層的に理解していく過程で、断片的であった部分的な知識が、意味を持つまとまりとして再構成されるが、このような知識は一種のマクロオペレータのようなもので、適切な場面ですぐに使用すれば非常に効率のよい推論ができる。ここで述べる機能、構造知識の階層的コンパイラはこのような再構成された知識を計算機で半自動的に作り出そうとするものである。

この際、重要なことは階層度の違う記述レベルの間を人間のように自由に渡り歩けるような仕組みを考えることである。そのためには、知識の記述形式が同型であることが望ましい。

1) 階層的定性推論の枠組み

階層的定性推論による知識コンパイラの枠組みを図2に示す。深い知識として対象の機器を構成する部品に関する物理法則を与え、機器を組み合わせて使用した場合の代表的挙動に関する知識を浅い知識として出力する。前者は対象の設計図などから機械的に収集可能であるが、後者はCausal Patternと同様もれなく収集することは困難な知識であり、この意味でこのコンパイラは従来収集が困難であったヒューリスティックスを半自動生成する手法を提案しているとも云える。

図2で、Interpretation RuleとTranslation Ruleは、物理法則を記述した方程式（深い知識）の使用方法和変形方法を与えるもので、数学に関する知識など、やはり機械的に収集可能な知識である。

以下では、図2の各部の内容と機能を順に説明する。

2) 知識の記述

対象の機器を構成する部品に関する物理法則など深い知識は、物理法則を直接表現した方程式を用いて記述する。機器を組み合わせて使用した場合の代表的挙動に関する知識（浅い知識）は、部品の組み合わせ・使用条件毎に物理法則とは別途記述する。

図3にトランジスタと抵抗と側路コンデンサより構成される回路についてコンパイルされた後の浅い知識例を示す。『抵抗とコンデンサーを並列に接続し(式1, 2, 3, 4)、直流を流すと(式10)、コンデンサーに流れる電流の変化は無視でき(式2及び式4の1c項)、電流は抵抗に流れる(式11)』ことを表わしており『側路コンデンサー付抵抗』という仮想的な素子の代表的挙動に関する知識を表現している。

3) 機器の挙動解析

図2のRecognizerが、素子の名称などを付け替えることにより浅い知識のIf部と一致する方程式

(深い知識)の組み合わせを検索し、名前を付け替えてThen部の方程式に対応する方程式を作成する。また、式を無視することが指定されている場合は、別途記憶しておく。この段階で、浅い知識も深い知識と同様に、方程式の形で表現される。図2のIntended Taskにより深い知識として入力した式と、Recognizerが作成した式の2種類の式を使って値の変化の伝播を計算する。Recognizerにより無視することが指定された式は使用しないが、それ以外の式は全て使用する。計算に必要な四則演算の規則は、図2のInterpretation Ruleにより記述する。

4) 浅い知識の生成 (コンパイル)

機器の代表的挙動に関する知識(浅い知識)は、EBL(Explanation Based Learning[Dejong 86][Minton 87][Mitchell 86-1][Mitchell 86-2])を応用することで、物理法則(深い知識)と挙動例の説明から以下の3ステップによりコンパイルして生成することができる(図3)。この手法を暗黙の説明(図3の式1, 2, 3)を一般化するという意味で、Implicit-Explanation Based Learning (IEBL)と云うことにする。

(1) 深い知識から明示的な挙動例の説明を作成する。

ここで明示的な挙動例の説明とは、挙動例で初めに变化したとされる値(図3で $1E \uparrow$)から最後に变化したとされる値($1R \uparrow$)まで挙動解析を行うために必要な知識(図3 で方程式(4))と無視する項($1c$)の情報である。

(2) 挙動解析により挙動例の説明を拡張する。

具体的には、例題で最後に变化したとされる値(図3で $1R \uparrow$)を使い、挙動解析を実施し、解析結果が明示の説明に現われる変数の値になった解析過程(例えば図3の式1, 2, 3)を挙動例の説明に付け加える。ここで、挙動解析は、使える深い知識がなくなるか、全ての解析結果が明示の説明に現われる変数の値になるまで続ける。また、結果が明示の説明に現われる変数の値になる解析過程があれば、その過程に現われる知識のうち、無視すべき知識があるか否か使用者に問い合せ、無視すべき知識を記憶する。

(3) 2番目のステップで拡張された挙動例の説明を一般化して浅い知識を合成する。

一般化の方法はEBLと同様である。

5) Interpretation/Translation Ruleの役割

EnvisioningとIEBLの過程で必要となる推論(この場合は数学)に関する知識はInterpretation RuleとTranslation Ruleとして定義する。

例えば、Envisioningのためには、 $1R$ などの値と方程式以外に式を使用して計算する場合の計算規則をInterpretation Ruleとして定義しておく。IEBLのためには、式4を変形して式11を作るための変形規則をTranslation Ruleとして用意しておく。

また、一般化の過程で変数にするデータ名の判定にも2つのルールの情報を利用する。例えば、図3で ω 、 C 、 R の値がif部で言及されているが、これはInterpretation/Translation Ruleにより記述された式の計算・変形規則が、それらの値を直接利用しているためである。すなわち、抵抗

に流れる電流が増加した時に、抵抗にかかる電圧が上昇することなどは、抵抗の値が正であることを利用している。

6) 本知識コンパイル方式の特徴

(1) ここで対象としている『機器の代表的挙動に関する知識』は、いわゆる浅い知識全てをカバーするものではない。しかし、浅い知識を上記のように表現することにより、深い知識である物理法則との関係を、『どの項／式を無視した』という情報として明確に記述することができる。また、『浅い知識』より『さらに浅い知識』を考えることが容易であり、複雑な対象を扱うために必要な多重の階層化の表現形式として利用できる。さらに、同時に使うと矛盾を起こす深い知識は無視することが浅い知識に記憶されている。このため、記述レベルの異なる知識が同時に使用可能となっている。即ち、記述レベルの異なる知識を同時に使用しても、矛盾の発生を避けられる。

(2) 浅い知識の生成過程は、『例題をベースとなる知識（ここでいう深い知識）で解釈した過程を一般化し、新しい知識（ここでいう浅い知識）を作る』という点でExplanation Based Learningの1種である。但し、今までに提案されたExplanation Based Learningの手法とは次の点が異なっている。

(a) Explanation Based Learningは、公理から定理をコンパイルするような場合に利用される手法である。従って、そのままでは、ここで扱っているような、『ベースとなる素子の挙動を現わす方程式』から『側路コンデンサ付抵抗といった仮想的な素子の挙動を現わす方程式』のコンパイルには適用できない。すなわち、EBLでは特定の条件を無視することはしていないので、ベースとなる知識が正しい場合、コンパイルされた知識の正しさも保証されている。一方、ここで用いている方法は、一般には無視してはいけない条件を無視している可能性があり、生成した知識の正しさは保証されていない。しかし、階層的に対象を理解するために必要な浅い知識を生成するには、特定の条件を無視する機能（正しさの保証はないが過去の経験から本質に関係ないと判断したものを無視する機能）は必須と考える。

(b) EBLでは、例題を説明するのに必要な深い知識の連鎖（例えば図3の式4）だけを浅い知識の合成に利用している。一方、ここで用いている方法は、例題を説明するには直接関係のない深い知識の連鎖も浅い知識の合成に利用している。これは、説明に表われる深い知識の連鎖だけから浅い知識を生成すると、浅い知識を用いた推論結果に矛盾を生じる場合が多いからである。例えば、図3の例では、説明に表われる深い知識の連鎖だけを利用したのでは、式(4)のみ一般化され、抵抗とコイルが並列に接続された回路などにも同じ知識を利用しようとして妥当な推論結果が得られなくなる（図4）。

ここで用いている浅い知識の合成方法は、このような矛盾の発生を、処理対象の物理システムが持つフィードバックを解析することにより避ける仕組みを備えており、この意味でEBLの改良になっている。

6. Frustration Based Learningによる深い知識からの戦略用知識のコンパイル

エキスパートは新しい問題を理解する過程で問題解法の戦略を学習する。学習した戦略を用いる

ことにより類似の問題に対しては、たとえ同じ問題を解いたことがなくても解決策を見いだすことができる。このような理解に伴う学習も知識コンパイルと考えることができる。ここで述べる Frustration Based Learning による深い知識からの戦略用知識のコンパイルはこのような戦略知識を計算機で半自動的に作り出そうとするものである。

1) Frustration Based Learningの枠組み

Frustration Based Learning による知識コンパイルの枠組みを図5に示す。深い知識として領域知識を与え、問題要素に関連して想起すべき戦略用知識を浅い知識として出力する。想起すべき浅い知識は問題全体の構造を反映するものではないため、構造が違ふ類似の問題に対しても有効に適用される。基本となる考えは、問題解決の過程で既存の領域知識だけでは先に進めなくなった個所で Frustration が蓄積し、その解消欲求が学習の強い契機となるということである。学習された知識は Frustration とその解消に貢献した知識のペアという形で整理される。従って、コンパイルされた知識は、別の問題で類似の Frustration に遭遇したならば、対応する知識を想起せよといった意味を有しており、連想用知識として活用される。Frustration は領域知識の不完全性に起因して発生するわけではない。コンパイルされた知識は一種のヒューリスティックスであり、必ずしも適用が成功するという保証はない。

以下では例題として初等幾何学の補助線問題を取り上げ、Frustration Based Learning の枠組みを具体的に説明する。

2) 補助線問題

補助線問題とは、新たな線分もしくは点を付け加えなければ解くことの出来ない平面幾何学の問題を言う。図6に典型的な補助線問題を2題示す。これらの問題が補助線を必要としない問題と違う点は、本来必要な点が欠如しているために与条件と証明すべきゴールの間に証明パスの成立を遮るギャップが横たわり、ギャップの両側で定理を前向きにも後向きにも適用できない不連続点が存在することである。妥当な補助線を引く行為は、証明パスの成立を可能にするようにギャップに橋を架けることに相当する。このようなギャップの存在は領域知識としての基本定理の欠如に起因するものではなく、必要な解法戦略の欠如に起因するものである。知識コンパイルの目的はこのようなギャップを埋めるための知識を生成することである。

3) 補助線生成の基本メカニズム

補助線生成の基本メカニズムは問題中の特定の条件もしくはサブゴールに対して条件反射的に図形パターンを連想することであるという仮説を立てる。そのための連想用知識（図形パターン戦略と呼び、これがコンパイラの出力）は次のように表される。

IF 特定の条件および（もしくは）サブゴールがあれば THEN 或る図形パターンを連想せよ。

そのIF部は連想を喚起する問題要素であり、THEN部は連想の仕方を示す手続きである。連想による補助線生成のプロセスの一例を図7に示す。生徒が問題2において $BP=PC$ という条件に注目したとしよう。もし

IF there exists 条件($XY=YZ$) (X, Y, Z は変数) THEN 図形パターン (図7) を連想せよ。

という図形パターン戦略を持っているとすれば、問題図形と連想された図形パターンの一部を図7の様に重ね合わせることによって図形パターンに存在して問題図形にない点及び線分が補助線として見えてくるであろう。

幾何学的情報は、図形を構成する構成情報と、構成された対象間に成り立つ拘束情報の2つに大別できる（以下では、それぞれ、構成情報、拘束情報と呼ぶ）。この区別は学習において重要な役割を持つ。具体的には、構成情報は点及び線分の存在を表す情報から成り、一方、拘束情報は線分・角・線分の比が等しいという情報、2つの線分が平行であるという情報、2つの三角形が合同・相似であるという情報から成り、いずれも述語で記述できる。

問題図形は、引き数部の変数が問題中の点を表す定数でインスタンス化された述語の *association* 集合として論理表現する。一方、図形パターン戦略は、

```
IF      ?-condition(問題条件を表す述語) or/and ?-subgoal(サブゴールを表す述語)
THEN   ?-make_pattern(図形パターン名, 関与する点のリスト)
where
  make_pattern(図形パターン名, 関与する点のリスト)
      :- (図形決定条件) -> (図形に成り立つ条件).
```

と表現する。図形決定条件 (if-part) とは、図形パターンの性質を一意に決定する条件である。従って、make_pattern 述語はある図形パターンを連想するための具体的な手続きを示すマクロオペレータとみなせる。そして、if-part と then-part を合わせた全情報はその図形パターンが持つすべての性質を表している（全図形パターン情報と呼ぶ）。

4) 浅い知識の生成（フラストレーションに基づく学習）

学習アルゴリズムは以下の4段階のプロセスからなる。まず、第一のプロセスで推論における困難点を発見することにより特定の問題要素に注目する。次に、第二、第三のプロセスにおいて完全な証明木を構築する。最後に、第一のプロセスで注目した問題要素を参照しながら証明木の中から図形パターン戦略を抽出する。

(1) フラストレーション同定プロセス

領域知識としては初等幾何学の基本定理のみを与える。しかし、ギャップがあるため補助線を引かない限り、基本定理を条件もしくはゴールに適用できなくなる個所（推論停止位置）が現れる。これをフラストレーション要素と呼び、それらを同定するのがこのプロセスの役目である。前向き・後向き推論に対応しフラストレーション要素はフラストレーション条件とフラストレーションサブゴールに細分される。

(2) 補助線の生成

図形パターン戦略を利用して補助線を生成するか、又は、適用できる図形パターン戦略を持たない場合はユーザーにその引き方の教示を受ける。

(3) 証明プロセス

基本定理を駆使して再び完全な証明木の形成を試みる。その過程で新たにF条件が発生した場合、補助線生成プロセスに戻って補助線を引き直す（補助線がユーザーの教示により生成された場合にはこのような事態は起こらない）。

(4) 学習プロセス

このプロセスにおいて、証明木の中から複数の図形パターン戦略を抽出する。以下の4つのサブプロセスを経て、木のどの部分を抽出すべきかが決定される。

a) フラストレーション木分析サブプロセス

このプロセスでは、F木と証明木を比較しF木の中で証明に貢献した部分を切り出すと共に、解消されたF要素を同定する。これにより得られる木を精製F木と呼ぶ。

b) フラストレーション解消サブプロセス

前サブプロセスで同定されたF要素が、補助線の生成によって如何に解消されたかを調べ、証明木から図形パターンを抽出するプロセスである。そのために、精製F木の中の各F要素ごとに以下の2つのタスクを行う。

1. F要素を解消するのに直接貢献した基本定理を、問題中の定数でインスタンスエートされたまま抽出する。F条件を解消するのに貢献した定理とは、その条件をボディ部に含む定理であり、Fサブゴールを解消するのに貢献した定理とは、そのサブゴールをヘッド部とする定理である。
2. 1で抽出された定理に含まれている構成情報を抽出する（それらの構成情報で形成される図形パターンを取り出すことに相当する）。

上記の2つのタスクを各F要素に関して行うことにより、フラストレーションの解消の仕方だけでなくそれが解消された状況をも憶えることができる。

c) 図形内限定推論プロセス

このサブプロセスで、抽出した図形が持つすべての情報を前向き推論により収集する。解消プロセスで抽出した情報のみを使って（その図形の中だけで）推論を行うため、「限定」推論と呼ぶ。

図形内限定推論は次の3つの重要な役割を持つ。

1. 当問題の証明木の成立には貢献しなかった情報を含めて、抽出された図形パターンが持つあらゆる情報を調査すること。なぜなら、貢献しなかった情報（例えば角newBP=角DCP）は将来この図形パターン戦略が連想される時には有用な情報であるかもしれないからである。大切なことは正しい証明のシーケンスを記憶するのではなく、連想の対象となる図形パターンを記憶することである。

2. この前向き推論において「言及」されない構成情報を削除することにより、抽出した図形パターンを洗練すること。「言及される」とは、それ自身がこの推論による証明木のノードになっているか、又は証明木中の少なくとも1つの拘束情報を構成する情報であるかの状況を指す。

3. 得られた図形パターンを一般化するための基準となること。

解消プロセスと図形内限定推論プロセスによって、各F要素に対して抽出された図形パターンの全図形パターン情報が得られる。

d) 図形パターン一般化プロセス

全図形パターン情報を図形パターン戦略として将来の問題にも適用できるように一般化する手法は、説明に基づく学習において説明木を一般化する方法と同じで、限定推論プロセスの証明に使用した定理中の変数の束縛関係を保存したまま証明木の各ノード（幾何学的情報）を変数化するという方法を採用する。図8に問題2のフラストレーション条件BP=PCに対して上記のプロセスで得られた全図形パターン情報を一般化して得られる木を示す。

e) 図形パターン戦略の形成

戦略の形成の仕方を図8にて説明する。図形パターンを抽出する動機になったF要素を戦略のIF部とする。そしてTHEN部（make_pattern述語の定義）は全図形パターン情報により構成する。make_pattern節のボディ部のif-partは、図形パターンを形作る構成情報と解消プロセスで抽出した基本定理のボディ部の情報から構成する。また、then-partは全図形パターン情報の残りの情報から構成する。問題解決の経験の中で注目を集めたF要素が学習の契機と基準を与えているので、この学習法をフラストレーションに基づく学習法と呼ぶ。この学習法は、人間が悩み抜いた事柄ほどその解決法をその時の状況と一緒によく記憶するという傾向を反映するものである。

(5) フラストレーションに基づく学習の意味

図9に見られるように、問題1からは4つの戦略、問題2からは5つの戦略が学習される。これは、「フラストレーションに基づく学習を行う行為を通して、問題1を4つの図形（各戦略のTHEN部の図形）の重ね合わせとして認識し、問題2を5つの図形の重ね合わせとして認識している」ことを意味する。これらの戦略の中で、2つの戦略が両問題に共通していることに注意しよう。共通する戦略（従って共通する図形）があるからこそ両問題に類似性を感じられるのであろう。これは学習に際して図形を意識することの正当さを語るものである。従って、フラストレーションに基づく学習は、連想用知識である図形パターン戦略を学習する方法であるのみならず、問題を大局的に把握するという人間の能力の一面を捉えた「理解の枠組み」ともみなせる。

5) 本知識コンパイル方式の特徴

IF（特定問題要素）THEN（連想行為）という形式を持つ連想用知識をコンパイルし生成する方法－フラストレーションに基づく学習法－は以下の特徴を有す。

- (1) 特定問題要素はF同定プロセスで発生したF要素であり、これが学習の契機になり、更にこの連想を喚起する条件になる。フラストレーションの導入は連想用知識を学習するに必要不可欠である。
- (2) 連想行為部は図形パターンを問題中に創り出す（あるいは発見する）ためのマクロオペレータである。
- (3) 幾何学的情報は、図形を構成する低次元の情報（学習者の目に訴える情報）と構成された対象間の拘束を示す高次元の情報に分類され、この分類は解消プロセスにおいて重要な役割を果たす。
- (4) 解消プロセスにおいては、解消に貢献した基本定理だけでなく関与する構成情報のすべての候補をも取り込んで図形パターンの候補を形成する。この原則が証明木の中のどの部分を抽出すべきかを決定する尺度になる。
- (5) 図形内限定推論プロセスの役割りは、連想される対象物の全図形パターン情報を収集すること、無関係な構成情報を取り除いて図形パターンを洗練することにあり、これは証明プロセスとは独立したものである。
- (6) 図形パターンの一般化は、図形内限定推論プロセスで使用した基本定理に基づいて行われる。この手法はEBLの一般化法を踏襲したものである。
- (7) フラストレーションに基づく学習を通して問題を解くことは、学習によって得た戦略のTHEN部に当たる図形パターン（連想対象物）の重ね合わせとして問題を理解することを意味する。

参考文献

[Anderson 83-1]

Anderson, J. R., "Knowledge Compilation: General Learning Mechanism", Machine Learning Vol.2, Morgan Kaufmann Publishers, Inc., pp.289-310, 1983

[Dejong 86]

Dejong G., Mooney R.:Explanation-Based Learning : An Alternative View. Machine Learning, 1. 2, pp.145-176, 1986

[Finger 85]

Finger, J. J and Genesereth, M. R., "RESIDUE: A Deductive Approach to Design Synthesis", STAN-CS-85-1035 (HPP-85-1) (1985)

[Hart 82]

Hart, P. E.: Direction for AI in the eighties, SIGART Vol. 79, pp.11-16 (1982)

[林 89]

林、山口、溝口、角所：深い知識に基づく知識コンパイラ（KCⅢ）におけるモデル生成と深い推論、人工知能学会研究会資料SIG-KBS-8805-7 (1989)

[Minton 87]

Minton S., Carbonell J. G., Etzioni O., Knoblock C. A., Kuokka D. R.: Acquiring

Effective Search Control Rules: Explanation-Based Learning in the PRODIGY System.
Proceedings of the 4-th International Workshop on MACHINE LEARNING, pp.122-133,
1987

[Mitchell 86-1]

Mitchell T. M., Keller R. M., Kedar-Cabelli S. T. : Explanation-Based
Generalization: A Unifying View. Machine Learning, 1. 1, pp.47-80, 1986

[Mitchell 86-2]

Mitchell T. M., Utgoff P. E., Banerji R. : Learning by Experimentation: Acquiring
and Refining Problem-Solving Heuristics. In R. S. Michalski, J. G. Carbonell, T.
M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach Volume
2. Morgan Kaufmann, 1986

[Motoda 84]

Motoda, H., Yamada, N. and Yoshida, K. : A Knowledge Based System for Plant
Diagnosis. Proc. of FGCS, pp.583-588 (1984)

[Murthy 87]

Murthy, S. S. and Addanki S. : PROMPT: An Inovative Design Tool. Proc. of AAAI-87,
pp.637-642 (1987)

[Suwa 88]

諏訪正樹、元田浩、" フラストレーションに基づく学習"、人工知能学会研究会資料SIG-
FAI-8801, 1988

[山口 87]

山口、溝口、田岡、小高、野村、角所: 深い知識に基づく知識コンパイラの基本設計、人
工知能学会誌、Vol.2、No.3、pp.77-84 (1987)

[山口 88]

山口、溝口、角所: エキスパート・システムにおける深い推論機構の統一的枠組み、人
工知能学会知識ベースシステム研究会資料SIG-KBS-8801-3 (1988)

[Yoshida 88]

吉田健一、元田 浩、" 階層的定性推論"、人工知能学会研究会資料SIG-KBS-8801-4,
1988

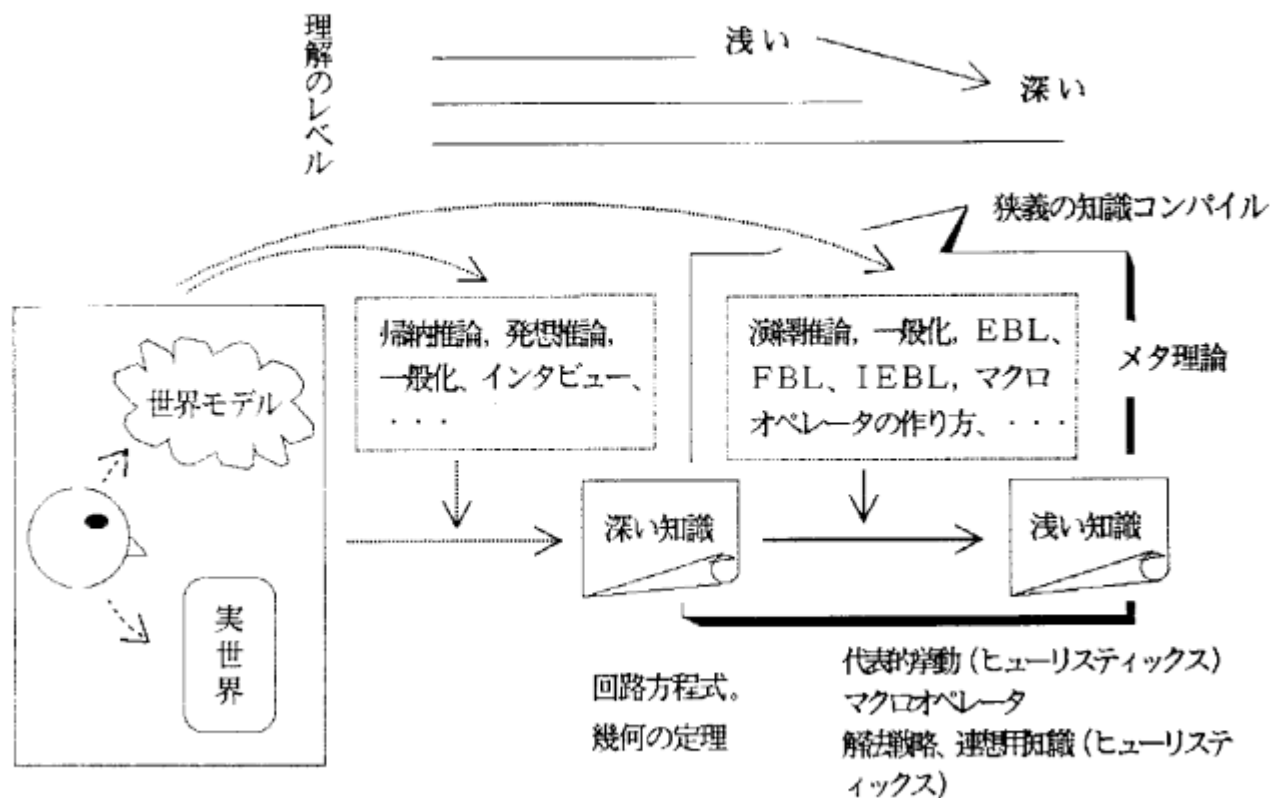


図1 知識コンパイルの枠組み

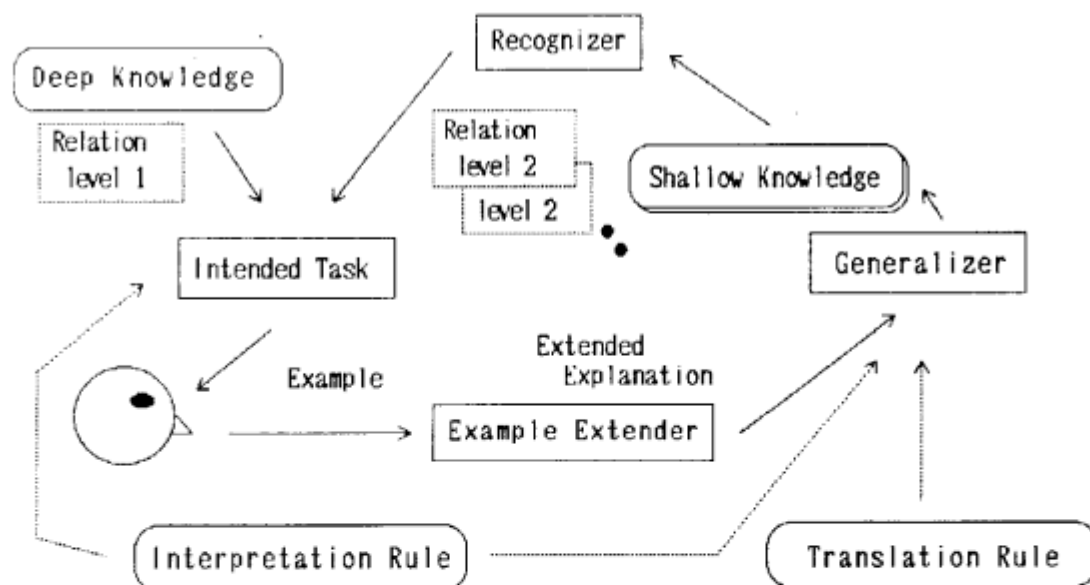


図2 階層的定性推論による深い知識からの機能知識のコンパイルの枠組み

Deep knowledge

Extended Explanation

Shallow Knowledge

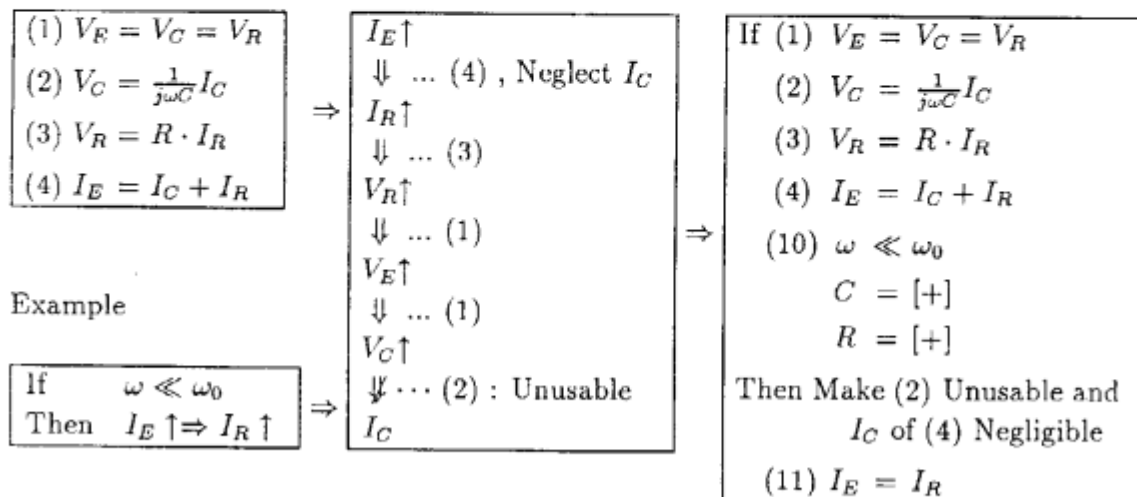


図 3 浅い知識の生成例

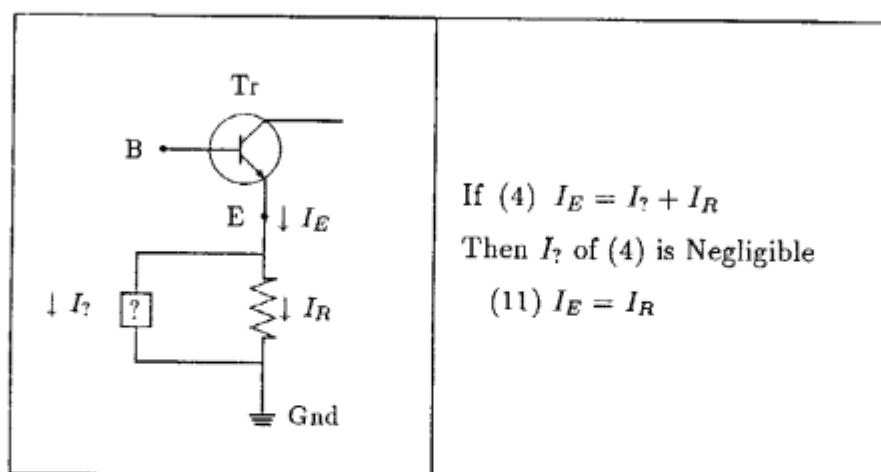


図 4 明示的説明から得られる浅い知識

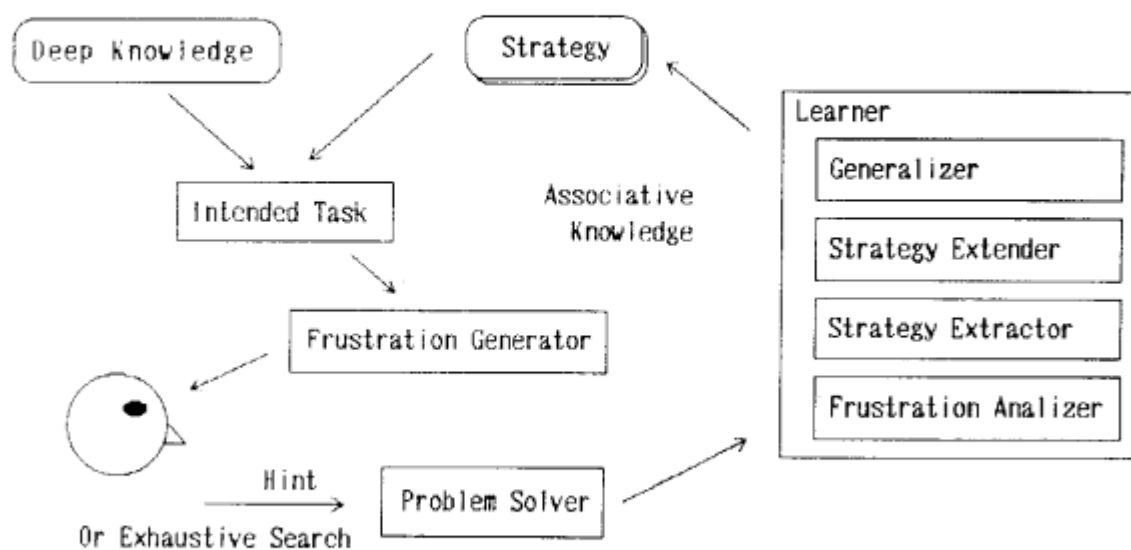


図5 Frustration Based Learningによる深い知識からの戦略用知識のコンパイルの枠組み

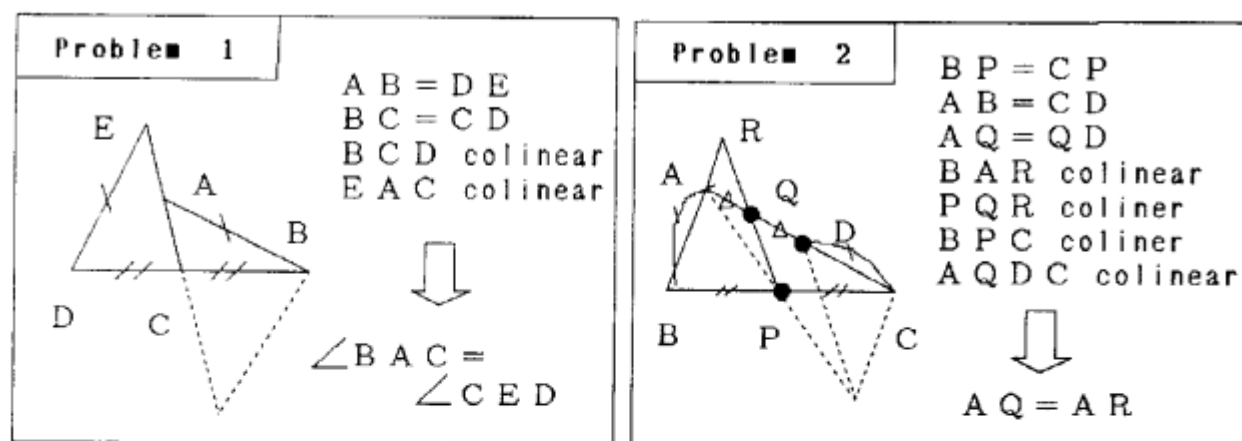


図6 典型的な補助線問題

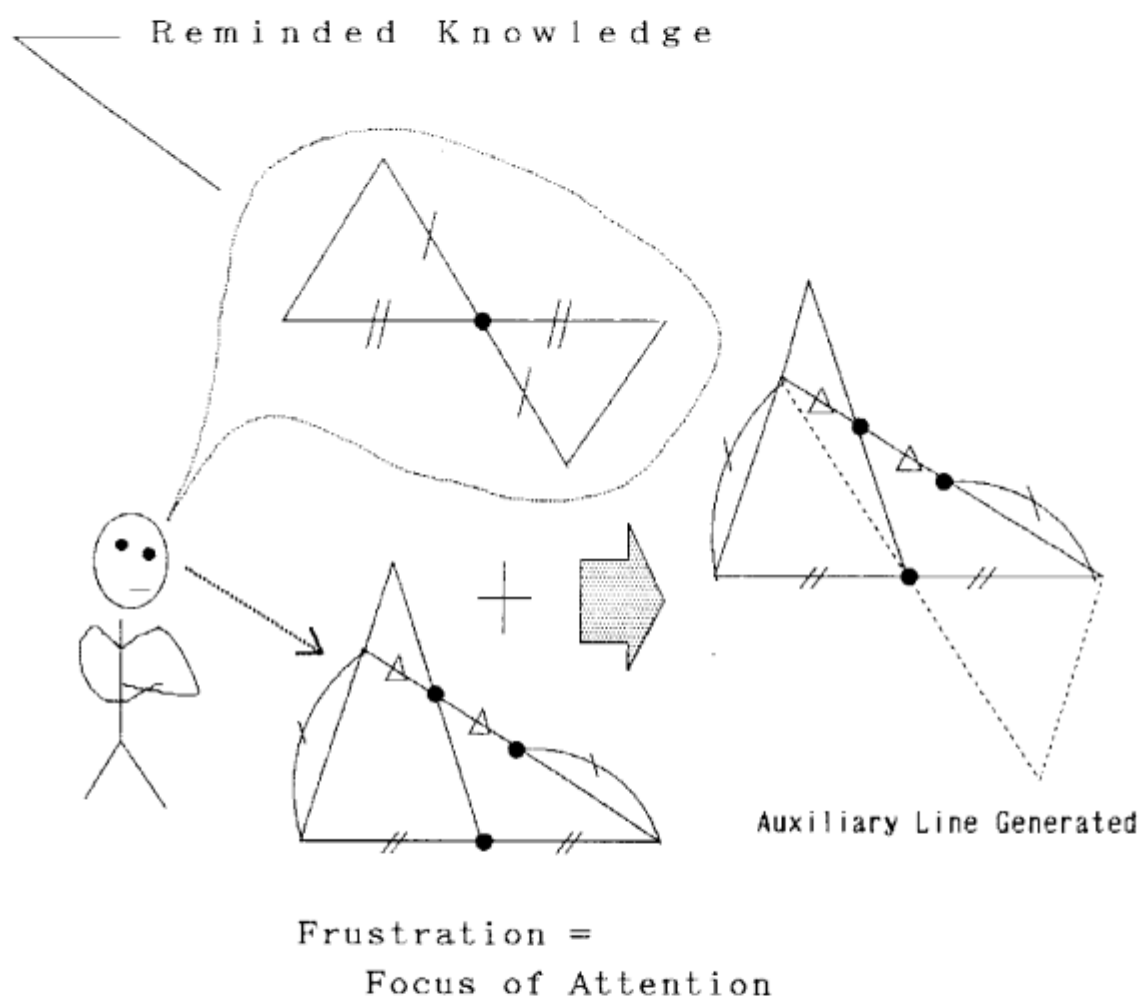


図7 補助線生成の基本メカニズム

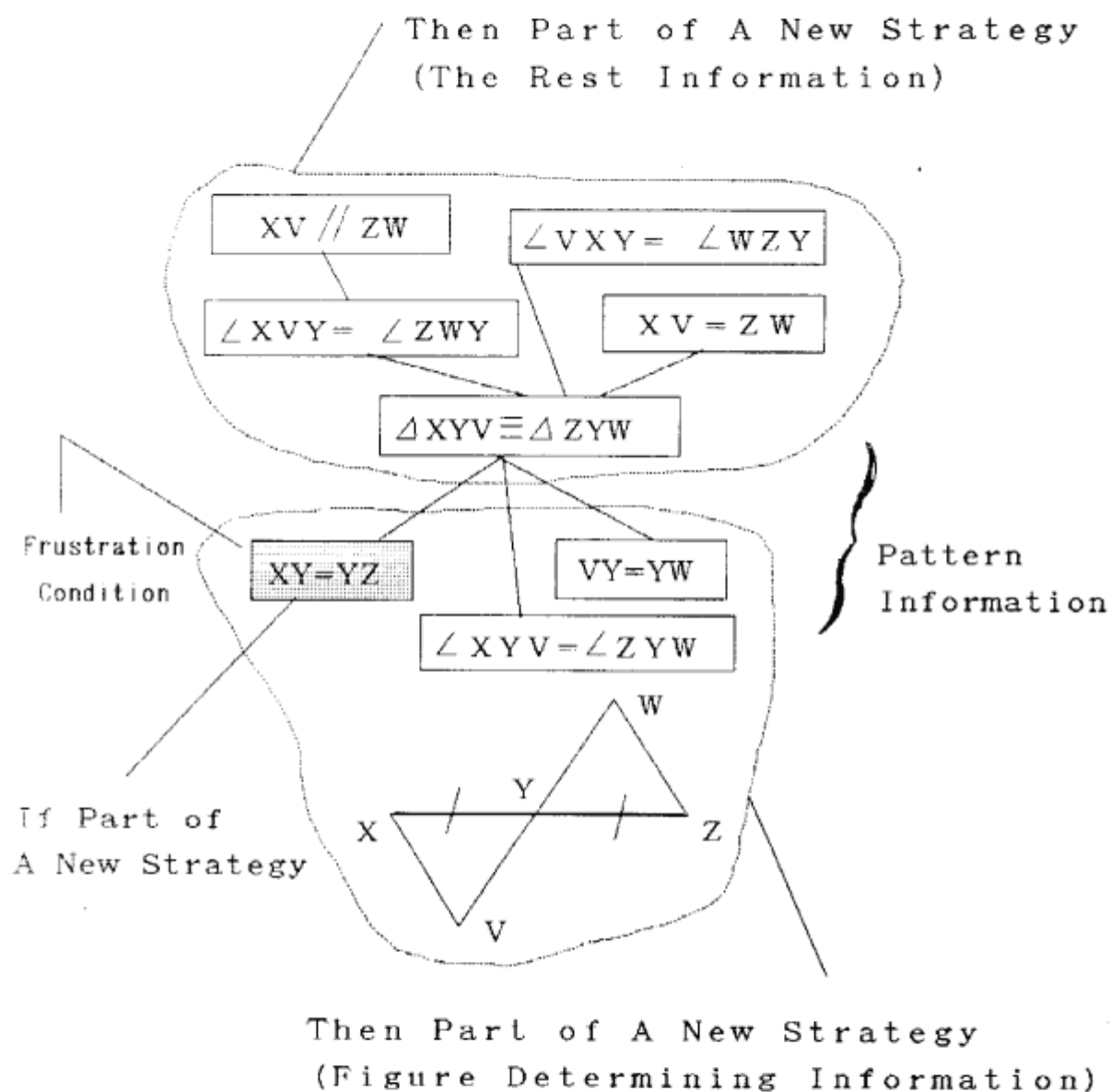


図8 図形パターン一般化 (図12の問題2のBP=PCのフラストレーション条件)

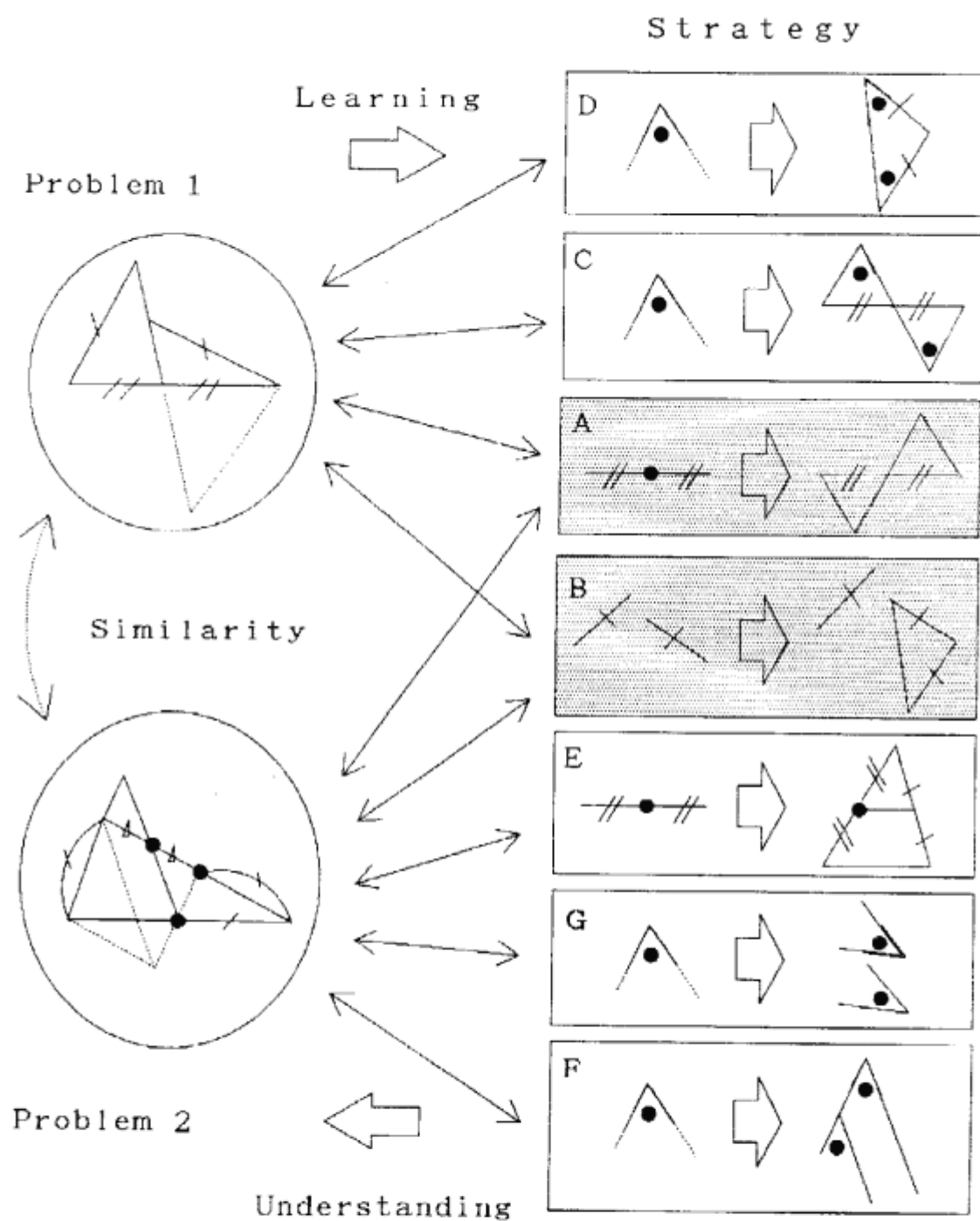


図9 フラストレーションに基づく学習の意味

説明学習に基づく統合化学習アーキテクチャ

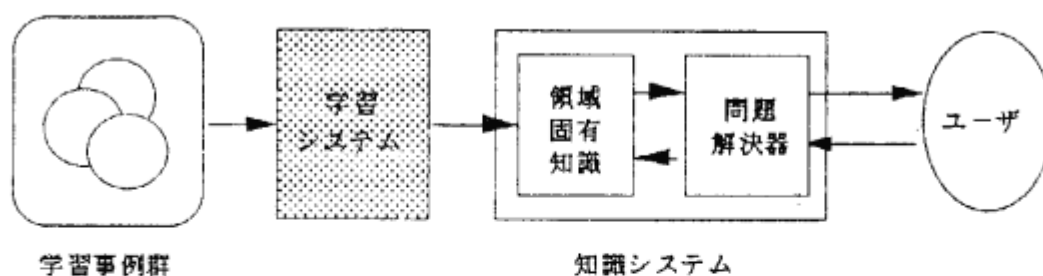
榎木 哲夫

(京都大学 工学部 精密工学教室)

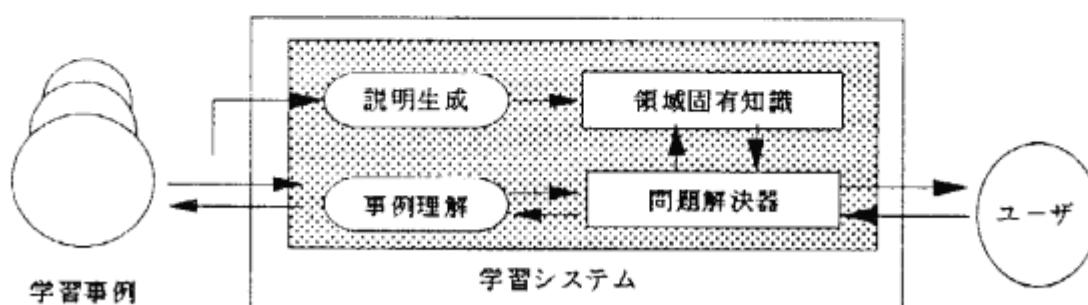
1. 学習 = 適応的問題解決

予め決められたシナタックスによる表層的な事例記述間の類似性に基づいて発見的に一般化規則を生成する帰納学習に代わり、近年の学習研究は、その事例の正負判断をつけるに至った専門家の推論プロセスをも考慮した上での、事例の理解に立った学習を進めるものである¹⁾。このような、いわゆる『認識的理解を通しての学習』へと移行してきた背後には、従来の帰納学習の理論が学習者をブラックボックス化した上で、学習者に与えられる刺激(事例)と学習者の反応(事例の帰属する概念)の間の相関として学習を捉えるものであったのに対し、人間本来の学習は、外的な刺激によってのみ決定づけられるような受動的・無目的な情報処理ではなく、学習者自身の内的状態、すなわち、学習のニーズや既得の知識・概念の状態に応じて学習内容が決まるという、行動主義心理学の流れにその根拠を見いだすことができる。そこで学習者は、外的環境からの特定の刺激を選択的に抽出・解釈し、たとえ既有知識による解釈がつかない場合でも、仮想的な仮説の設定や更新を繰り返しながら外界の中に整合性を見いだすような「内への働きかけ」と、外部から与えられた事例に対してある特定の視点から変更修正を加えた新たな事例を生成し外部環境(教師や教材)に判断を求めるといった「外への働きかけ」の両者による能動的かつ目的指向的な適応のプロセスが介在している。

この意味から学習システムは、図1に示すように、従来のような複数事例からのバッチ処理形式で知識ベースの構築を自動化するための一方向的な独立したモジュールとしてではなく、外部環境との双方向のインタラクションを伴いながら目的に指向した適応を行うための問題解決システムとしてのアーキテクチャーが要請される²⁾。すなわち、学習システム自らが、与えられた事例の内部に既得の知識と整合しない問題箇所を学習ニーズとして見いだすと、自身の既有知識を再編成したり、過去の既知事例を参照したり、あるいは適確な注視点を設定して外部環境にその解決の糸口を能動的に求めたりすることによって問題箇所を解消し、このようなサイクルを通して新たな知識の獲得を行うと同時に事例記憶の再編を行い、きたるべき新たな事例処理に備えることになる。



(a)



(b)

図1 (a)従来の学習システム
(b)認知的理解に基づく学習システム

2. 適応的問題解決システムとしてのEBLの課題

Mitchell et al.³⁾の「説明に基づく一般化学習 (Explanation-Based Generalization/Learning、以後EBLと呼ぶ)」の枠組みにおいて定義された学習の諸要素、すなわち、訓練事例 (training sample)・目標概念 (goal concept)・領域知識 (domain theory)・説明構造 (explanation structure)・操作性規範 (operationality criteria) の概念は、「認知的理解に基づく学習システム」を構築する上でいずれも不可欠な核となる概念であり、近年の学習研究はこれらの諸要素により大きな自由度と柔軟性をもたせるための研究や、他種学習手法との融合形態についての理論的展開、また各種領域問題への応用研究や、学習知識による問題解決パフォーマンス向上の実証研究など、理論・実用の両面からのEBLの拡張法が模索されている。ただし、EBLの枠組みは、上記のような、適応的問題解決システムとみなすには未だ不十分であり、数多くの課題を克服せねばならない。

本来「理解のための説明」のタイプとしては、Schank⁴⁾によると、

(1) Canned Explanation:

既知のことを対話の相手に伝えるだけの行為で、説明を行うことによって説明者自身にはなんら新たな創造は生まれない。

(2) Explain-Away Explanation:

説明対象が一事例と見なせるような、既知の仮説世界を見いだすことを意味するが、説明を生成することによって説明者自身が知識構造の変容を加えられるものではない。

(3) Additive Explanation:

既有知識の中に、入力に該当するものが見いだせない場合に、ある一連の手段を講じてその理由付を行い、より深い理解に到達しようとするもので、説明生成を通じて以前には未知であったことが新たに付加される。

のように分類される。そして、説明が要請されるニーズは、(1)の外部供与的なものから(3)の内部発生的なものへと変化し、後者は説明生成の困難さと複雑さが増すことになる。従来のEBLの枠組みがカバーしてきたのは、第一・第二の説明タイプによる知識の編纂(knowledge compilation)とこれによる問題解決効率の向上を目的としたものであるが、真の意味での創造的な knowledge-levelでの学習が可能な適応的問題解決を行う学習システムを実現するには、この第三のタイプの説明を実現しなければならない。そのためには以下のような諸点を統合化したシステムアーキテクチャが要請される。

(1) インパス駆動型学習

学習は、いくつかの事例間の類似性・差異点を認識するような受動的な情報処理では達成され得ないし、また既有の一般的な知識をやみくもに当てはめてみるだけの説明づけであってもそこから新たな知識の生成を望めるものではない。特に、EBLのように予め事例理解のための完全な領域知識が与えられているという前提は非現実的な要請であり、むしろ、学習における知識の役割は、既有の知識の範囲内では首尾一貫した理解ができないとか、一つの解釈に決めかねて思い悩むといった「説明しかねること(インパス(impassé))」を認識するためにこそ既有知識の存在価値があるのであり、このことが学習システムに学習のニーズ・目的を与え、創造的な学習を達成させるためのトリガーとなる。

このようなインパス駆動型学習の推論ステップは図2に示すような各段階からなる。

① インパスの認識とその分類

EBLにおいては、学習ニーズは目標概念という形で外部から与えられ、これに対する既有知識(領域知識)のトップダウン適用により説明が生成される。これに対し統合化学習システムでは、システム

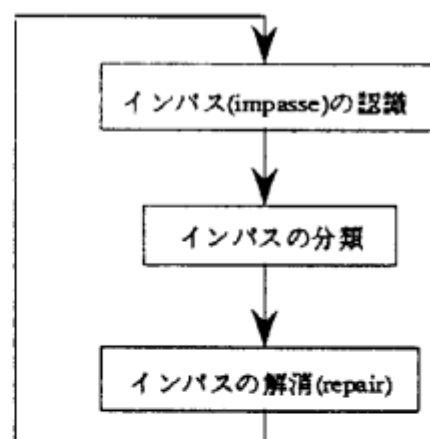


図2 インパス駆動型学習の基本サイクル

自身が与えられた事例内部に説明ニーズを見いださなければならない。そのためには学習システムが、事例のおかれている状況世界に関する「モデル」を生成できることが必要になる。このようなモデルとしては、対象世界での因果則が連鎖された因果モデルや事例クラスから帰納的に導出されたカテゴリー概念の原型（プロトタイプ）⁵¹等が考えられる。システムは学習事例が与えられると同時に、システムの有する領域知識に基づいて事例の断片情報から、本来はその事例はどのようなべきかについての予期構造・モデルを事例処理とは独立に並行して構築し、その結果と、与えられた（観測された）事例との間で整合性のとれない部分、あるいはモデルの結果から逸脱するような異常（anomalous）な部分を学習ニーズを示唆する箇所として認識する。

不完全な領域知識に起因する説明の失敗の形態としては、知識が欠如しているが故に説明生成に必要な演繹を行うことができない場合（the incomplete theory problem）があるが、これはさらに必要な知識が完全に欠如している場合（the first type of incompleteness）と知識が備わってはいるものの十分に詳細な知識が欠落しているが故に仮説生成を余儀なくされたり相互に相矛盾する演繹を行ってしまう場合（the second type of incompleteness）に細分類される。もう一方は、知識自身が整合性を欠く（the inconsistent theory problem）がために首尾一貫しない結論に導いてしまうような演繹を行う場合で、この原因として誤った知識を含んでいる場合（the first type of inconsistency）と誤った演繹を排除するのに必要な知識に欠ける場合（the second type of inconsistency）に分類される⁵¹。

このような不完全知識に基づいて説明生成を行う場合、インパスは以下のような形となって現れる。

(1) Broken Explanation:

知識が欠如しているが故に説明間にギャップが存在して説明を完成することが出来ない場合（incompleteness - type I）

(2) Contradiction:

互いに相矛盾するような結論に導く説明を生成する場合で、誤った知識の存在による場合（inconsistency - type I）と知識の欠如による場合（inconsistency - type II）

(3) Multiple Explanations:

本来単一の説明しか存在しえないところが、説明の善し悪しを識別するための知識が欠如しているが故に複数の説明が生成される場合（incompleteness - type II）

(4) Resource Exceeded:

学習システムに許容された計算時間や計算容量の限界から説明生成が行えない場合で以下の2つの場合がある。

- (4-1) Large Search Space Problem: 説明生成時に広大な状態空間探索を必要とするにも拘らず、パス優先則等のヒューリスティックスを持たない場合（incompleteness - type II）

(4-2) Small Links Problem: 説明構成要素となる単位があまりにマイクロレベルの詳細多岐にわたるために、完全な説明生成が行えない場合

このように認識されたインバスの分類を行うことにより、各インバスのタイプに応じて、学習対象の固有領域には独立に定義され、かつそこでの問題解決タスクのタイプに応じた一般的な「メタ認知知識」として、インバス解消のための方策をパターン化することが可能になる。またこのことは、KE (knowledge Engineer) が、専門家から man-to-man のインタビューによる知識獲得を行う際の、質問生成戦略あるいは generic operation として活用することが可能になる⁷⁾。

② インバスの解消 (repair)

上述のようなインバスの解消のための具体的方策としては以下に示すような様々な形態が考えられる。これらの処理を経てインバスが解消されると、この解消のためにとられた一連の手続きが既存の不完全知識を修正する新たな知識単位として、学習獲得されることになる。

(1) 仮説生成によるインバス解消

何か新しい情報が与えられた場合、我々は通常自分のもつ知識に何らかの仮説を付け加えてその情報を説明しようとする。そしてその説明に矛盾がなければ与えられた情報が説明できたとみなす。勿論ここでは、仮説を生成・選択する方法や、無矛盾の説明を行える複数の仮説から一つを選択する方法、そしてやはり説明づけられないと解った時点でどこまでバックトラックして仮説を棄却すればよいか、についての解決策が必要になる。

(2) 過去事例からの類推によるインバス解消

発生したインバスの解消方策を知識として規定したり、演繹や解析によって実行が不可能なような問題対象においては、その解消は過去の事例からの類推 (analogy) に依存せざるを得ない。旧事例 (base) を参照し、その中の特定の関係を写像することによって、新事例 (target) で成立することを推論する類推の定式化においては、事例中のどの関係に焦点を絞るのか、そして多数ある過去事例の中からどれを base として採用するのかが常に問題になるが、これは過去事例を共通のインバス毎に、あるいはインバス解消に取られた共通の手だて (インバスに対する説明構造) 毎に組織化しておくことにより、新事例処理のインバス発生時に効率よく過去事例を検索することが可能になる⁴⁾。

(3) 実験の創案によるインバス解消

以上までに述べたインバス解消が、既存の知識・記憶構造に対する学習システム内部への交渉によるものであったのに対し、システム外部の教師や教材に交渉を持つことによりインバスを解消することが考えられる。例えば、学習事例として与えられたものが既存知識の範囲内では矛盾が発生して理解ができない場合に、自らの知識の一部を疑問視する仮説の生成を行う。そしてこの仮説の正当性を実証するために、学習事例中のパラメータを修正した新たな実験事例を生成し、この実験事例に対する外部環境 (教師や教材等) の

判断に基づいて元の学習事例の理解を達成し、そこから新たな知識獲得を目指すというものである⁸¹⁾。

(2) 事例記憶に基づく学習

① 事例記憶に基づく説明——直感的理解のメカニズム——

E B Lにおける事例理解は、事例を意識的に構成要素に分解し、高次知識構造のもとでの論理的な整合性を追跡していくことにより達成するのであるが、問題対象によっては、必ずしもこのような形で事例の理解が行われるものばかりではない。実際、状況を分解して一般的な知識と照らし合わせながら構成要素を識別する分析的な行動は、エキスパートの熟練度レベルからいうならばビギナーレベルのタスク実行の特徴であり、これが経験を蓄積したプロのレベルに達すれば新しい状況の中に過去の経験との類似性を認め、事例全体としてのパターンを直感的に捉えながら、「全体観的な類似性認知」を身につけるわけである。このような「直観による理解」すなわち過去の経験との類似性を認めたときに苦もなく何かを理解する力は、具体的な状況を全体として体験し状況全体を記憶できるようになって初めて備わるものである。ここで直観とは論理的な整合性を事例の各部分に亘って調べ上げるというものではなく、ある特定の状況下での典型例を記憶して現在の状況パターンがその事例と似ているか否かイメージ的に見比べることに相当する。

ここで生成される説明は、E B Lに見られるような演繹による論理的証明木が生成できるか否かによる二分的な説明可能性判断ではなく、人間のもつカテゴリー概念に見られるような境界が明確でなく連続的に変化するようなフエジィな説明可能性判断であり、またその説明構造からなされる一般化も、E B Lのような定数の変数置換による画一的な一般化手法から、当該事例を取り巻く周囲事例に依存する形で決定されるような一般化手法の開発が要請されることになる。このような非分析的な事例指向の戦略により学習者は分析的戦略では対処できないような複雑な問題に取り組むことができると考えられる。

ここで理解に必要な知識は、E B Lのように一般的な知識というよりはむしろ具体的な事例が概念的に結び付いた集合体であり、過去の経験は何が有意な特徴かという点で既に記憶構造内部で組織化されている。すなわち、具体的な状況を全体として記憶しているからこそ次にどうなるかの予想が当然記憶に残り、このような予想が実現することによってその事実は意味あることとして強く意識されることになるし、逆に予想が実現しなければしないで、そのようなずれが真の創造的な学習達成への引金となる。表1にこのような事例記憶に基づく説明学習とE B Lに見られる演繹による説明学習との対比を示す²⁾。

② 事例記憶の組織化

事例記憶を構築する上で最も重要なことは、新しい事象を理解する助けとなるような過去の事例をいかにうまく見つけ出すかであり、そのためには既知の事例群の効果的なカテゴリー化や事例の組織化が重要な課題となる。このような事例

	証明木生成による説明学習	事例記憶に基づく説明学習
学習事例	構成要素を分析的に照合識別	全体として直観的に活用
学習に必要な知識	領域固有な一般的普遍的知識	事例の集合体とメタ認知知識
説明構造	論理的証明木の生成	既有記憶構造への同化操作
自己組織性	コンパイル知識生成による 知識量の線形的増大 ミクロな適応性立	記憶構造の再編による 非線形的増大 構造的な適応性

表 1 論理的証明木生成・事例記憶に基づく説明学習

の組織化手法としては、既に1980年代前半に事例を属性とその値の集合として記述し、これらの記述間の類似性に基づいて階層構造化された事例クラスを組織する一般化された記憶構造（Generalization-Based Memory）であるMOP（Memory Organizing Packets）の考え方が提唱されている⁹⁾¹⁰⁾¹¹⁾。しかしながらより柔軟な事例の検索を実現するには、このような陽に表現された事例記述上での類似性に加え、一見共通点や関連のないように見える事例を組織化するための一般的原理を確立する必要がある、そのためには事例記述の中のどの属性が些細なことでありどれが重要なことであるかを識別できるだけの事例理解が前提となる。ここで上述のような予測の失敗に対する説明、インパスの解消にとられた説明が、事例理解にたった組織化を行う上で表層的な（あるいは事例記述に陽に表された）特徴だけからでは結び付けようのないような過去の類似事例を想起・連想させるインデックスとして有効に機能すると考えられる（Explanation Based Indexing, EBI）⁴⁾¹²⁾。

3. 統合化学習アーキテクチャー

従来のEBLの考え方を基盤にして、前節で述べた「インパス駆動型学習」及び「事例記憶に基づく説明学習」をハイブリッドに組み込んだ、「適応的問題解決システム」として統合化された学習システムのアーキテクチャーを、図3に示す。本システムは、学習モードとして従来のEBLの考え方に立つ「演繹による説明学習」と「事例記憶による説明学習」の2つの学習モードをもち、両者は相互のインターアクションを通じて補填しながら、全体として「インパス駆動型学習」のもとに推論プロセスの制御された学習システムとなっている。

まず、専門家により学習対象として与えられた訓練事例が入力されることからシステムは起動される。システムはこの事例の部分的な情報から能動的に対象世界モデルを自身の既有知識に基づき生成する。そしてこのモデルに基づくシステ

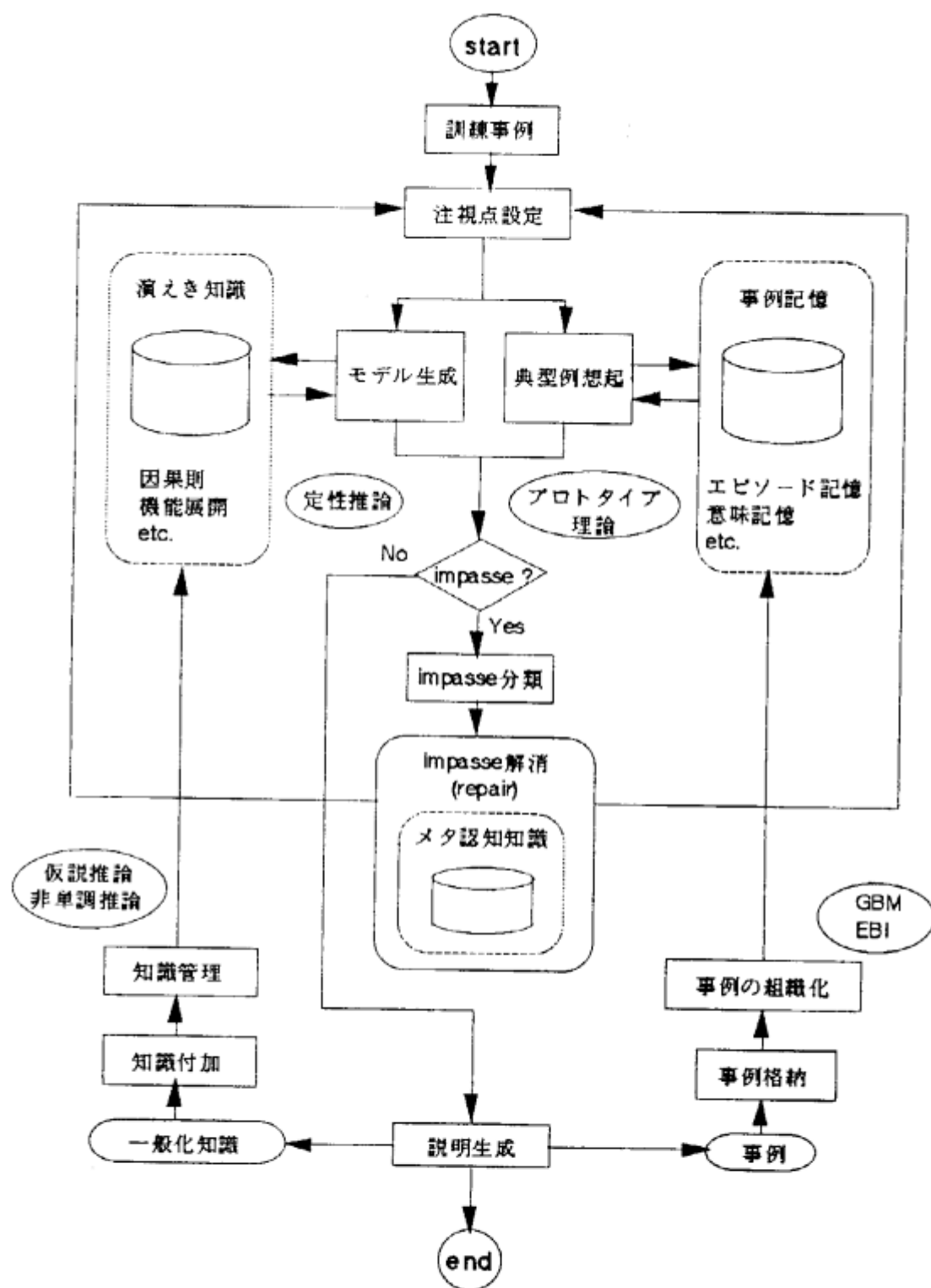


図3 統合化学習システムアーキテクチャー

ムの予期する事例解釈の結果と、実際に与えられた事例記述の間に不整合が生じないかをチェックすることになる。ここでの世界モデルの構築法としては、対象システムに関する因果則が明確に与えている場合には定性推論などのメンタルシミュレーション技術が、また既知事例の蓄積がある場合には、これらから帰納的に導出したり、人間の概念生成プロセスのメカニズムに則ったエピソードからのスクリプト学習¹³⁾¹⁴⁾¹⁵⁾や、その下での典型例の想起による手法¹⁶⁾が有効になると考えられる。

この段階でもし不整合が生じなければ、与えられた事例は既有知識の常套的な適用で説明がつくことになる。すなわち演繹学習の場合には、その説明構造から従来のEBLの考え方に従って訓練事例固有の知識が抽出獲得されることになるし、訓練事例が事例記憶から検索された典型例とのアナロジーで解釈のつくものであれば、近年の事例推論(Case-Based Reasoning)で行われているようにここでの類比を通して一般的な知識が生成されることになる。

しかしながら、通常こうして抽出される知識は訓練事例の一部を書き換えた(rephraseした)知識にすぎず、このような知識はその後の問題解決のパフォーマンス向上の点では有意義なものであるものの、knowledge levelでの新規知識の獲得を行うものではない。実際、専門家により知識学習の対象として与えられる訓練事例は、常套的な既有知識の適用では説明のつかない部分を含んでいるからこそ訓練事例として与えているのであり、このような部分こそが新規な知識獲得を示唆する部分となるものである。従ってこのような不整合(インバス)が認識されたなら、メタのレベルでインバス-解消方策を定義したメタ認知知識を参照し、自身の知識の修正や知識の適用に関する注視点を変更した上で再度事例解釈を試みることになる。このような注視点の変更としては、演繹学習においては、仮説的知識の生成付加と変更や既有知識の詳細化などが考えられ、事例記憶に関しては事例検索のインデックスの変更による対象となる事例群の再定義やインバス発生の原因となっている異常属性(unusual features)に着目した事例群の特定を行うことにより、新たな視点のもとでの概念生成とそのもとでの典型例の想起が行われることになる¹⁷⁾。この意味で、インバスの認識は訓練事例中のどの部分に焦点を当てて説明生成を試みるかを示唆するとともに説明生成の方略をも決定することになり、説明生成プロセスの制御策を与えることになる。

こうしてインバスが解消され事例に対する説明が完了したならば、以下のような手続きが必要になる。まず、説明構造から一般化された知識は単純に既有知識に付加するということだけではなく、その付加が既有知識との間で矛盾や不整合を生じないかについて、知識ベースの真実性の管理がなされねばならない。ここでは各種の仮説推論や真理の維持管理に関する方法論が有効に機能することになる。また説明構造からの一般化に際しては、従来のEBLのような画一的な定数の変数置換による一般化に代わり、説明の導出プロセスを考慮にいれた一般化やコンテキストとしての周囲事例を考慮に入れた一般化¹⁸⁾を考える必要がある。いま一つは、説明のついた訓練事例を捨ててしまうのではなく、これを既有の事例記憶内の適切な箇所に格納して置く必要があるが、ここでは来るべき新規事例の解釈

時に有効に連想されるように事例間の組織化・インデックスづけが重要課題となる。これには当該事例に関して発見されたインバスの分類やこの解消に用いられた方策など、説明構造自身をインデックスとして組織化することにより部分的かつ表層的な事例記述の類似性のみに従う従来のGBMに代わり、事例の全体観的な類似性を考慮した専門家の直感に即した事例の想起を可能にするインデキシングが可能になると考えられる。

統合化学習システムでは、EBLのように単一の事例処理のみを考えるのではなく、インクリメンタルな事例処理を想定しなければならないことは既述の通りであるが、事例から獲得され領域知識に付加される知識の管理、さらに説明処理の終了した事例管理など領域知識自身のダイナミックなアップデートも重要課題となってくる。これは、演繹知識については適応的プロダクションシステムの学習形態として従来から提唱されているバケツリレーアルゴリズム (Bucket Brigade Algorithm) や遺伝アルゴリズム (Genetic Algorithm) など知識の使用に伴う信憑性の更新 (credit assignment) に関する手法¹⁹⁾が、事例記憶については事例記憶のダイナミックな再編 (dynamic memory) に関する手法²⁰⁾が有用な示唆を与えものと考えられる。

4. 新たな今後の学習対象

最後に、今後期待される学習対象領域について若干ふれておく。これまでの学習研究の大部分が、いわゆる記号世界を対象にした問題解決を扱ってきたのに対し、実世界の制御やパターン認識・分類の問題となると、フロントエンドにおいてはやはりアナログ量による実観測データを扱わざるを得ないし、また説明学習に必要な領域固有知識も、決して well-structured な形で宣言的に与えられないのが現状である。このような領域においては、実際に専門家がやってみせたり、専門家判断の最終結果のみを訓練事例として与えることは容易であっても、これから直ちに他の事例に対しても適用可能な operational な記述の一般化知識の形でその背後にある知識を抽出することは困難な作業になるが、このようなプロセスの自動化は、命題的知識のみならず技能的知識をも組み込んだリアルタイム処理の可能な知識システム構築に当たり不可避の課題であると考ええる。

近年、認知科学を始めとしてマンマシンシステムあるいはヒューマンインターフェイスの研究領域において、専門家パフォーマンスのソフト (= 認識) ・ハード (= 技能) の両面に互る情報処理モデルが、制御タスクや診断タスク、分類タスク等について個別に提唱されている。これらのモデルは未だ直ちにシステム実装が可能なまでに操作的 (operational) なものではないが、これらの定性的かつ記述的なモデルのもとで、実際の専門家のパフォーマンス事例を解釈し何故に専門家が訓練事例に見られるような認知判断を下したかを説明づけることによって、その背後にある専門家のスキル (skill) 知識 = 技能的知識 (behavioral knowledge) を具象化 (操作化) して抽出することが可能になる。勿論、このようなタスク毎の専門家モデルは決して正確なものではなく、あくまで総括的な近似的なモデルに過ぎないが、本報告で提唱した統合化学習システムアーキテクチャーのも

とでの学習システムを構築することにより、事例処理に伴って随時精練化され、各個別タスクに応じて真の意味での適応性を有したシステムの実現が可能になると考えられる²¹⁾²²⁾¹⁸⁾。

【参考文献】

- 1) 榎木：問題解決と知識学習に関する一般的考察、ICOT Technical Memo.、TM-577、3-1-1/3-1-21、1988。
- 2) 榎木：学習における最近の話題、計測と制御、27-10、931-934、1988。
- 3) Mitchell, T.M. et.al.:Explanation-Based Generalization: A Unifying View, Machine Learning, 1-1, 47-80, 1986。
- 4) Schank, R.:Explanation patterns: Understanding mechanically and creatively, Lawrence Erlbaum Assoc., 1986。
- 5) Rosch, E. and Lloyd, B.B.:Cognition and Categorization, Hillsdale, N.J.:Lawrence Erlbaum Associates, 1978。
- 6) Rajamoney, S. and DeJong, G.: The Classification, Detection and Handling of Imperfect Theory Problems, Proc. of IJCAI-87, 205-207, 1987。
- 7) Falkenhainer, B. and Rajamoney, S.: The Interdependencies of Theory Formation, Revision, and Explanation, Proc. of the 5th Int. Conf. on Machine Learning, 353-366, 1988。
- 8) Rajamoney, S. et al.: Towards a Model of Conceptual Knowledge Acquisition through Directed Experimentation, Proc. of IJCAI-85, 688-690, 1985。
- 9) Schank, R.: Language and Memory, Cognitive Science, 4, 243-284, 1980。
- 10) Kolodner, J.K.:Maintaining Organization in a Dynamic Long-Term Memory, Cognitive Science, 7, 243-280, 1983。
- 11) Lebowitz, M.:Generalization from Natural Language Text, Cognitive Science, 7, 1-40, 1983。
- 12) Barletta, R. et al.:Explanation-Based Indexing of Cases, Proc. of AAAI-88, 541-546, 1988。
- 13) 榎木、他：社会現象の深層構造把握のための因果連鎖知識の多重構造化、計測自動制御学会論文集、23-9, 977-984、1987。
- 14) Sawaragi, T. and Iwai, S.:Cognitive Simulation for Intelligent Decision Support, in Ras, W.R. and Zemankova (eds.), Methodologies for Intelligent Systems, Elsevier Science Pub. Co., 362-369, 1987。
- 15) 榎木：人間の認知プロセスモデルに基づく意思決定支援と知識情報処理、第18回OR学会シンポジウム講演論文集、1987。(also in 榎木：経営決定における認知情報処理と知識学習、関西経済研究センター資料、89-3、1989。)

- 16) Sawaragi, T. et al.: Conceptual Clustering: Finding Patterns from Chaotic Reality and Pattern-Directed Decisionmaking, Prep. of the 3rd IFAC/IFIP/IEA/IFORS Conf. on Man-Machine Systems, Analysis, Design and Evaluation, 1988, Vol. 2, 387-392, 1988. (Also to be appeared in the Proceedings)
- 17) Sawaragi, T. et al.: Control of Learning Process for Intelligent Decision Support Based on Episodes Memory, Proc. of the International Workshop on Fuzzy System Applications, 232-233, 1988.
- 18) 榎木、他: 分類問題のためのプロトタイプ理論に基づく分散記憶モデル、計測自動制御学会第9回知識工学シンポジウム講演論文集、129-134、1989.
- 19) Holland, J.H. et al.: Induction: Process of Inference, Learning, and Discovery, The MIT Press, 1986.
- 20) Schank, R.: Dynamic Memory: A Theory of reminding and Learning in Computers and People, Cambridge Univ. Press, 1982.
- 21) 榎木、他: ファジィ事例学習による制御戦略知識の生成、第4回ファジィシステムシンポジウム講演論文集、167-174, 1988.
- 22) Sawaragi, T. et al.: Skill Acquisition for Intelligent Fuzzy Controller by Analyzing Operator's Control Action Sequences, (To be appeared in Proc. of the 28th SICE Annual Conf. SICE'89), 1989.

高次推論と計算量

國藤 進

(富士通(株)国際情報社会科学研究所)

帰納、類推、仮説推論、非単調推論といった演繹以外の高次の推論を、記号としてアルゴリズム化できることが次々に実証されてきた。しかしながら、新たに作成されたアルゴリズムの計算スピードの遅さというバリアー、すなわち計算量の壁、に気付く結果となった。その結果、知識表現のクラスと計算量との関連が各種学習アルゴリズムを素材に調べられている。計算の複雑さのクラスには、図1 (Miyano 88) に示されているように、NP、P、NCといったクラスがある。ここに注目すべきは、クラスNCという並列推論可能なクラスである。NP≠P と信じられているのと同様に、P≠NCと強く信じられている。宮野のサーベイ等によれば、AIの多くの問題がクラスNPに属し、P やNCに属さず、かつクラスP やNCで扱える知識表現(問題)のクラスは驚くべきほど小さい。それ故、このような計算量の壁を破る方策がないか、特に並列推論や近似推論 (Mayr 88) との関連で探究されている。

知識表現のレベルを記号処理に限定した場合、AIにおける学習研究の基本は、帰納推論にある。帰納推論に基づく学習スキーマ(learning scheme) (Kunifuji 87) という学習研究を統一的にみる視点を与える。学習スキーマ設定にあたっては、まず学習(推論)対象の領域を設定する。すなわち、どのような領域における概念を学習するのかを規定する。ついで、図2に示されるように、以下の構成要素を考慮していく。

(a)仮説の表現空間： 推論対象となる概念を表現するための方法を定める。形式言語の推論を例にとると、言語を表現する方法として、“形式文法”、“(抽象)機械”等が一般的に用いられるが、もちろん、プログラミング言語あるいは一階述語論理等で表現することも可能である。

(b)学習アルゴリズム(learning algorithm)： アルゴリズム論でいう演繹的なアルゴリズムと、たとえば確率的な要素を含む非演繹的なアルゴリズムとが考えられる。

(c)学習プロトコル(learning protocol)： 学習が行われる状況(前提条件)を規定する。例の提示法は、具体例の種類(正・負の文字列、正・負の構造体(項、木、グラフ等)、

計算過程例、確率分布付き文字列等々)、あるいは利用可能なオラクルの能力(所属しているかどうかの質問に答える能力、反例を提示する能力等)を規定する。

これらの要素を一つずつ具体的な設定に固定していくことにより、様々な帰納推論の問題が得られる。実際、これまでに行われている研究のほとんどは、この枠組みによって説明される。帰納推論による学習システムの研究には、これらの学習システムの各構成要素を適切な設定に選択する事が重要であり、また得られたシステムが(インタフェース部分も含めて)効率的なものであることが要求される。従って、効率の良いシステムが得られるような学習スキーマの設定条件を、いかにして見い出すかが重要な問題になる。

学習可能性(learnability)という概念を、計算量(complexity)の立場から再考してみる動きがある。従来、学習可能性概念の形式的定義としては、“極限における同定”(Angulín 83)という概念が支配的であり、事実これまでに知られている帰納推論に関する仕事のほとんどは、この考え方に基づいている。しかしながら、Valiant によって提案された学習可能性を一種の“確率的事象”としてとらえる学習可能性概念(Valiant 84)の定義に注目したい。図3に示されているように、考えられる可能な仮説全体のクラスKを仮説空間とする。また、推論しようとする概念の(一般には、正・負両方の)具体例が入っている壺Aを考える。この壺から具体例を一つずつ取り出して、それと矛盾する仮説をKから排除した後、この具体例を元の壺に戻す。このようないわゆるベルヌーイ試行を多数回繰り返すと、ブール関数(概念)の幾つかのクラスは、その概念のサイズの多項式回のベルヌーイ試行によって、ある誤差の範囲内で正しい仮説のクラスに絞り込むことができる。また、このようにしてベルヌーイ試行を限りなく繰り返していくと、誤差も限りなくゼロに近づいていく。したがって、この学習モデルは、従来のGoldによる“極限同定”(Gold 67)の拡張モデルであると見る事ができる。このような方向での計算量の壁を破る研究については様々の方向があり、文献(Laird 89, Mayr 88)を参照されたい。

上述のように、学習システムは「学習システム=学習プロトコル×学習アルゴリズム=(例の提示法×オラクル)×(演繹的アルゴリズム+非演繹的アルゴリズム)」で構成される。そこで、上記等式における各パラメタをいかに具体化して行って、得られた学習システムの具体例が効率良く動くようにするか、ということが今後の研究課題となる。ここに、“効率良く”は非常に重要な要請である。この効率の問題を、理論的にはいわゆる“多項式時間推論”(クラスP)を意味すると解釈する。従って、多項式時間で動作するようなアルゴリズムによる学習システムを構築することが、第一の目標ということになる。

そのための例の提示法、オラクル、そして最も重要なアルゴリズムをいかに見出すかが、今後の重要課題である。事実、このような方向でAngluin (Angluin 87) や榊原 (Sakakibara 88) は最近、優れた結果を導いた。榊原が示したように、オラクルとして正負の実例のみならず、正負の構造体といった構造情報をも与えることが計算量の遞減に大きく寄与することは、注目に値する。

以上のように、学習可能な表現のクラスが多項式時間で学習可能という性質を保持してどこまで拡張できるかという問題は理論的かつ実的に興味深い。しかし計算機が演繹的なアルゴリズムによって実地的な時間内で学習できる概念のクラスは、ある意味で非常に限られたものであり、いかなる生物システムも計算理論的に手におえないクラスは（実地的な時間内には）学習できないであろうと予想される。例えば、たかだか100個の可能な仮説のありとあらゆる組み合わせから、考えうる最高の記号処理コンピュータを用いて、仮説を検定していく問題すら、人類が生誕してから今までの計算時間をもったとしても対処できないことは明らかである。AIにおける学習研究の陥りやすい弱点は、AIツールの急激な進展を信奉し、おもちゃの問題のみに通用する安易なアルゴリズムを設計することである。その結果、そのアルゴリズムがNP完全という既存のアーキテクチャの手に負えないクラスであることを意識せず、実用的な規模の問題に適用して、初めてアルゴリズムの選さに気付く傾向がある。「どういう知識表現のクラスに対して、どういう学習アルゴリズムを適用したら、どういう計算量になるか」をきちんと押さえないと、実地的な時間に答えの戻ってくる実用的なアルゴリズムは得られないであろう。このことを理論的にもインプリメンテーション的にも保証しない限り、学習アルゴリズムが既存のソフトウェア環境の優れたツールとして組み込まれることはないであろう。

(参考文献)

(Angluin 87) Angluin, D.: Learning Regular Sets from Queries and Counterexamples, Information and Computation, Vol. 75, 1987, pp. 87-106.

(Gold 67) Gold, E. M., Language Identification in the Limit, Information and Control 10, 447-474, 1967.

(Kunifuji 87) 國藤 進、横森 貴: 知識獲得と演繹・帰納・発想、認知科学会、学習と対話研究分科会SIGLAL87-1-(6)、1987.

(Laird 89) Laird, P.: A Survey of Computational Learning Theory, NASA Ames Research Center, Report RIA-89-01-07-0, Jan. 1989.

(Mayr 88) Mayr, E.W.: Parallel Approximation Algorithms, Proc. of FGCS'88, Tokyo Prince Hotel, Dec. 1988.

(Miyano 88) Miyano, S.: Parallel Complexity and P-Complete Problems, Proc. of FGCS'88, Tokyo Prince Hotel, Dec. 1988.

(Sakakibara 88) Sakakibara, Y.: An Efficient Learning of Context-Free Grammars for Bottom-Up Parsers, Proc. of FGCS'88, Tokyo Prince Hotel, Dec. 1988.

(Valiant 84) Valiant, L.G.: A Theory of the Learnable, CACM, 27-11, 1984., pp.1134-1142.

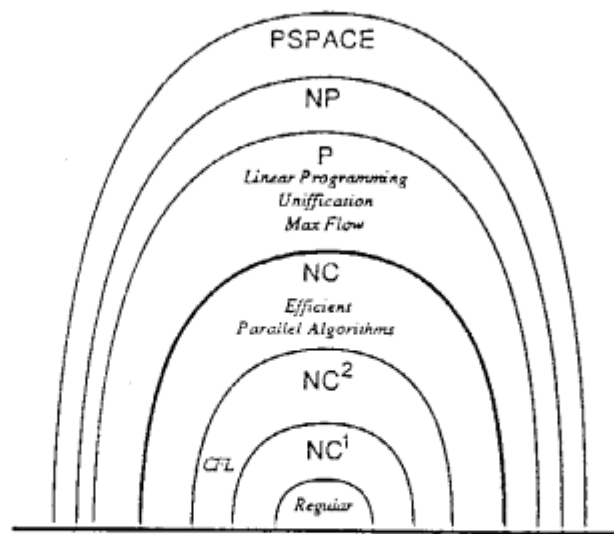


図1 計算の複雑さのクラス (Miyano 88)

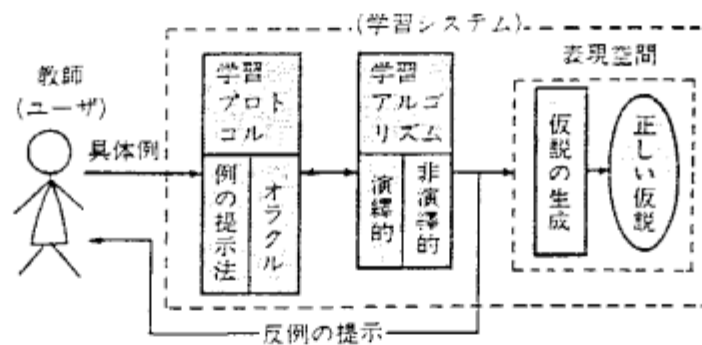


図2 帰納推論に基づく学習スキーマ (Kunifuji 87)

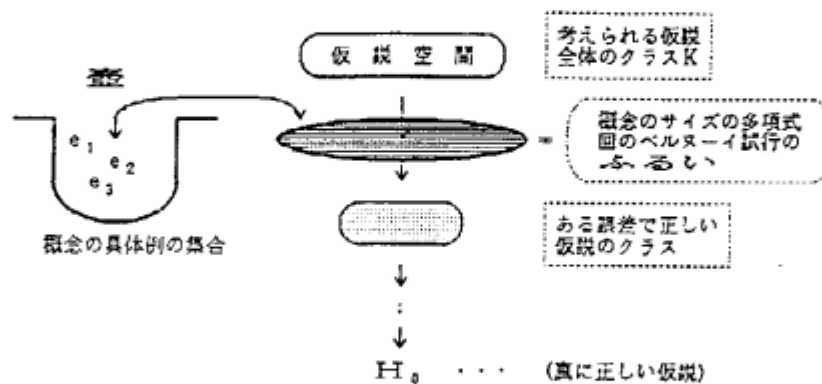


図3 確率的学習モデル (Kunifuji 87)