

## 解説



## 定性推論の知識獲得への応用†

元田 浩†† 吉田 健一††

## 1. はじめに

今日エキスパート・システムは各分野に浸透し、有効性が示されている。しかし、その開発はまだまだ芸術に近く、工学的に確立された方法論は模索中であると言ってよい。なかでも、知識獲得が重大なボトルネックになっている。定性推論は知識の表現・現象や機構の理解・モデルの自動構築・学習など知識獲得に関わる本質的な話題と関係があり、最近これらの研究発表も急速に増加しつつある。本稿では定性推論の技術が挙動の理解や知識獲得にどのように使用され得るかを、内外の最近の研究を紹介しつつ解説する。とくに、知識コンパイルという観点から 1) 診断など、特定のタスク遂行に有用な知識を獲得する方法、2) 複雑なシステムを理解する際に、あるいは問題を解決する際に、人間が通常行っている抽象化したレベルでの思考形態や、そのような抽象化を支援するための人間の理解の仕方に整合した階層的な知識表現とその構築に、定性推論がどのように貢献するかを分析する。

## 2. 知識の階層化とコンパイル

われわれは通常、問題を解決しようとする際に、過去の経験により有効であることを、あるいは結果が正しいことをすでに知っている知識を最初に使おうとする。そして、これらの知識では問題が解けないときに初めて原理原則に戻って熟考し、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を利用して問題の解決にあたらうとする。すなわち、知識を階層的にとらえ、必要最小限の最も簡単なモデルで問題を解決しようとする。こ

のような思考過程は、過去の経験を使って知識を整理・階層化し、類似の問題解決を効率よくしようとするプロセスととらえることができ、タスク依存の知識コンパイルとして計算機上での実現が研究されている。

また、われわれは複雑なシステムを理解する場合にも対象の構造に着目し階層的に把握することにより、目的に応じたレベルに注視の焦点を合わせて考えようとする。たとえば、図-1 はアナログ回路として見ればアンプであるが、デジタル回路として見ればインバータであると考えられる。つまり、インバータはアンプを組み合わせて実現されていると階層的に理解し、どちらに解釈すべきかは、そのときの目的（文脈）に依存して決定する。さらに、いったん、理解すると、類似の構造に対しては、分かっている解釈をそのまま適用する。この過程も、過去の経験を使って知識を整理・階層化し、類似の問題解決を効率よくしようとするプロセスととらえることができるが、前の例とは異なり、アナログ・デジタルといった概念構造も変化している。この場合、知識の階層構造がそのまま概念構造となっている。

さらにこれらの思考過程の多くは定性的であり、とくに現象の振る舞いを規定する物理原理から演繹によって世界を予測する思考形態が、いわゆる定性推論におけるエンヴィジョンング (envisioning) に対応し、一度理解したことを記憶する操作が知識獲得に相当する。しかし、定性的な関係を用いた推論だけで有為な結論が得られること

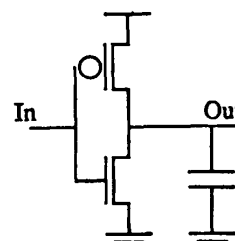


図-1 回路の解釈：アンプ／インバータ†

† Application of Qualitative Reasoning to Knowledge Acquisition by Hiroshi MOTODA and Ken'ichi YOSHIDA (Advanced Research Laboratory, Hitachi Ltd.).

†† (株)日立製作所基礎研究所

は少なく、定量的な情報を用いた数値計算や解析的な手法を用いた記号処理も必要となり、実際の知識獲得には、これらすべてが併用されている。

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、抽象度の高い記述や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ことを以下では知識コンパイルと言う。上の議論からコンパイルの軸は二つ考えられ、階層化された知識を階層化の程度により、「深い」、「浅い」と呼ぶことにすれば、図-2のように整理できる。

縦軸は機能に関する抽象度の違う知識（記述）のコンパイル、横軸は特定のタスク実行に特化した知識のコンパイルを表している。この図において機能知識はいずれのレベルにおいてもタスクに特化した知識に対しては深い知識となっている。いずれの軸の浅い知識も深い知識から基本的には演繹操作で生成できる。ただし、その過程で不要な情報を落としている。縦軸に関しては注目している現象を特徴づける代表的な機能に関係のない情報は落としており、横軸に関してはタスクに関係のない情報を落としている。たとえば図-2において、縦軸方向の知識コンパイルでは「変数“C”は十分小さな値で無視可能である」との仮定のもとに“C”に関する情報を落としており、深い知識では可能であった“C”に関する推論が浅い知識ではできなくなっている。同様に横軸方向のコンパイルを行うと、元の知識では可能であった設計・挙動予測などへ利用することができなくなっている。

知識コンパイルを機械化することは人間が対象を理解して得た知識や特定タスクの実行に有効な知識の収集に重要な意味をもっている。対象の挙動に関する物理法則などの深い知識から、対象の仕様に関する知識や診断・設計など特定タスク向けの知識を作成すれば、各種問題解決用のエキスパート・システムの知識ベースとして利用できる。これらの知識ベースは同一の深い知識から生成されるので、深い知識のみ管理しておけば、それらの間の整合性は自動的に維持される。

## Functional Description

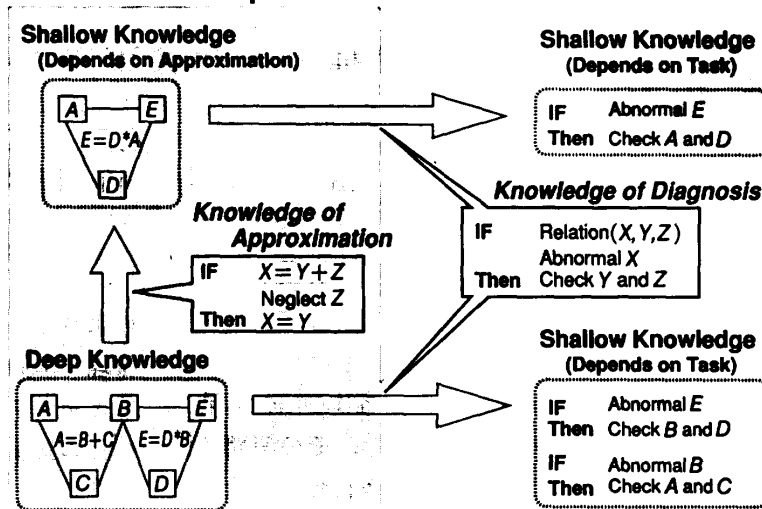


図-2 深い知識から浅い知識への知識コンパイル

### 3. タスク依存知識の獲得

タスクに依存した知識を獲得する場合の深い知識は基本的にはドメインにのみ依存し、タスクには依存しない客観的な知識であるので、タスクに関する知識を与えることで、同じ演繹的な枠組みで診断、設計・計画など幅広い領域の知識をコンパイルすることができる。また、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<sup>2)</sup>。以下では、分析的なタスクの代表例として診断を、合成的なタスクの代表例として設計を取り上げ、いくつかの定性推論による知識獲得法を紹介する。

#### 3.1 診断用知識の獲得

深い知識から診断用知識を獲得する試みは多い<sup>3), 4)</sup>。筆者たち<sup>5)</sup>も深い知識から診断用知識を生成する方法を内蔵したプラント診断用エキスパート・システムを開発したことがある。しかし、生成に使用した推論手法はいわゆる記号処理で、通常の意味での定性推論を利用したものではなかった。定性推論により診断用知識を生成した最近の代表的な研究に山口らの知識コンパイラ<sup>6)~8)</sup>がある。

彼らは深い知識として 1) 対象装置の構造と機能 (Device World: DW), 2) 物理原理 (Physical World: PW), 3) 故障頻度や観測容易性 (Control World: CW), 4) 物理状態を故障や徴候に対応づける解釈知識 (Interpretation World: IW), 5) 故障メカニズムに関する汎化知識 (Failure Mecha-

nism World: FMEW) の5種類の世界の知識を同定している。DW と PW はドメインのみに依存する深い知識 (PW はドメインにも依存しないという見方もある) であるが、CW と IW はドメイン、タスクの両方に依存する。これらは知識をコンパイルする際にコンパイラが参照する知識である。FMEW は故障メカニズムに関するヒューリスティックスであるが、ドメインに独立した形で記述されている (たとえば、異物の存在は新たな生成かほかからの移動のいずれかであるといった知識)。

これらの5種類の知識を用い、徴候から原因(仮説)側へたどる遡行推論と仮説から徴候へたどる順行推論を組み合わせ、最終的に徴候と仮説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(図-3. 文献7)より)。徴候が与えられると、それを対応する物理パラメータの異常(定性値として正常値より大, 小)に変換し、遡行推論により DW と PW の知識を用いて定性値を逆向きに伝播する。注目している物理パラメータと左辺のパラメータが一致する物理式のうち適用条件の一致度が最大の物理式が、そのパラメータを決定する物理式であるとして自動的に選択される。右辺への伝播は、各パラメータの異常は同時には発生しないとして、おのおののパラメータにそれぞれ異常を伝播する。すなわち、 $n$ 個のパラメータのうち、 $n-1$ 個のパラメータの値には変化がないとして残りの1個のパラメータにのみ値を伝播する。したがって、ここで定性推論が分岐し、単純な遡行推論では分岐の再帰的な繰り返しによる組合せ的爆発が起こる可能性があるが、まれにしか故障しないようなものについては CW の知識を用いて推論処理を中止するなど、不要な推論を抑制している。この推論は IW の知識によって故障仮説とみなせる要素に到達するまで繰り返す。こ

で、FMEW の知識が使える場合には、さらに徴候として現れていない真の故障仮説(たとえば、なぜパイプが詰まったかの原因)を探索する。この過程で上記の定性推論が再度用いられる。このように DW/PW/IW/CW の知識による推論と FMEW の知識による推論を交互に組み合わせることにより、より深い真の原因が追求されていく。故障仮説の候補が同定されると、次に順行推論により、深い知識を遡行推論のときとは逆向きに適用し、各故障仮説に対応する全ての徴候を求め、徴候と故障仮説を対応させ診断用知識を生成する。

この手法は非常に簡単にみえるが、これによりエキスパートでも容易には思いつかないような診断用知識を生成することができている。実際、空調装置に適用した結果、たとえば「熱膨張率の差により冷媒がもれ、このため部品 A がオーバーヒートし、部品の一部が融解して経路に流れ出し、経路を流れている冷媒に冷却されて凝固してパイプを詰まらせ、これが原因となって圧縮器の回転数が減少した」というような知識が生成されている。

### 3.2 設計用知識の獲得

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いの対象となる深い知識(モデル)をも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。

深い知識を用いて設計知識を獲得する試みの一つに松田らの最近の研究がある<sup>9)</sup>。これは油圧回路の設計知識を獲得するもので、理解に基づく知

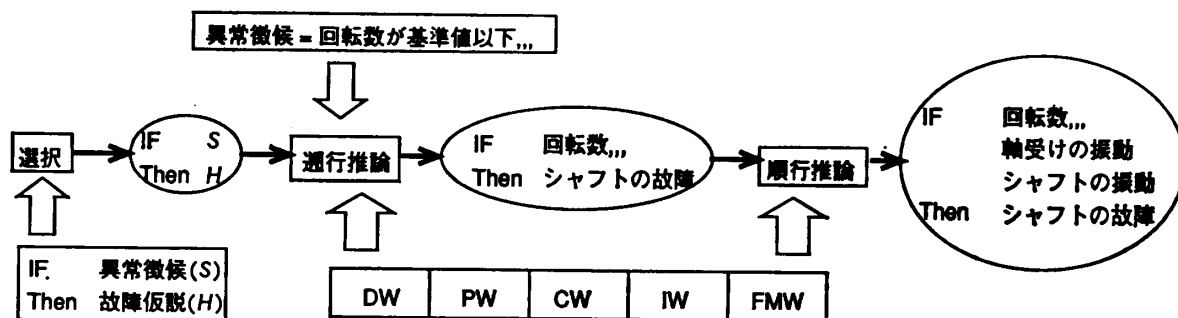


図-3 診断用知識のコンパイル

識獲得法と言えるものである。具体的な設計事例を深い知識（領域知識）で解析し、その事例が設計仕様を満足していることを説明しようとし、説明可能であれば、その説明を基に具体的な機器の存在理由と仕様との関係を設計知識として抽出し、説明不可能であれば、失敗を分析し、領域知識あるいはヒューリスティクスをエキスパートから抽出するものである。説明の生成の過程でシミュレーションが必要になる。ここでは記号処理を用いており、通常に用いられている意味での定性推論を用いてシミュレーションしているわけではないが、深い知識からの設計知識の獲得という点で注目すべき研究である。

設計とは機能から構造へのマッピングである。Goel ら<sup>10)</sup> は機能と構造のおおのをお互に参照可能な形でスキーマを用いて明示的に階層表現し、設計変更問題を診断と修正の二つの問題に分け、さらに前者を 1) 望ましくない挙動を検知したセンサに対する因果関係の同定、2) そのような挙動を示す原因の同定、3) 原因に対応する構造の同定、後者を 4) 修正戦略の選択、5) 修正案の提案、6) 検証に細分し、変数の値の伝播、変数間の関係の評価に定性推論を用いている。NASA の宇宙望遠鏡の姿勢制御装置を例題にして、軸受けベアリングのセンサ温度上昇対策を取り上げている。具体的に診断部の 1) でセンサ温度の上昇は、ベアリング温度上昇であると同定し、2) で変数間の因果関係をたどり、原因はベアリングの負荷の上昇であると同定し、3) でベアリングに成立する関係式を評価して、ベアリングのサイズ小が原因であることを同定する（図-4 に 1~2 のステ

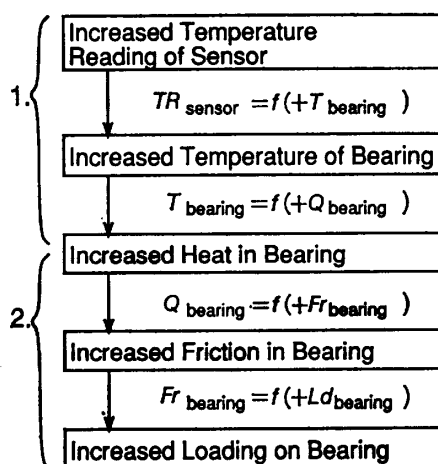


図-4 設計変更用の知識コンパイル

ップを示す。図は文献 10) の Fig. 5 から関連した説明部分を抜き出したもの。修正部の 4) では部品交換戦略を選定し、5) でサイズの大きいベアリングを提案し、6) でベアリング大にすることにより、変数間の因果関係を逆にたどり、各変数が制約内に入っているかどうか、つまり、構造を変更する必要があるかどうかを評価する。最終的に「軸受けベアリングのセンサ温度上昇対策としてはベアリングのサイズを大きくせよ」との設計変更用知識を獲得する。現在のところ、空間に関する推論を行っておらず、6) の処理が十分ではない。

以上の例で明らかのように、定性的な値の伝播は設計変更に対する指針を得るのには有益であるが、それからそれ以上のことを期待するのは難しい。最初に指摘したように、本当に必要な知識を獲得するには定量情報を用いた値の伝播や数式処理など、いわゆる知的な思考に必要な多くの技術を併用しなければならない。しかし、このことは定性推論の意義を否定するものではなく、逆に、単独の技術のみをうんぬんすることに建設的な意義を見いだせないと感じている。

定性推論が効果を発揮するのは、種々の関係式から新たな構造を演繹的に発見する場面ではないかと想像する。この点に関し、Murthy<sup>11)</sup> らの 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部品に関する知識だけでは、どう部品を組み合わせても仕様を満足する設計案を生成することができないとき、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分析し、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あらかじめ用意した修正オペレータとモデルグラフを使って、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。修正オペレータはエキスパートが与えているが、これを定性推論によって作り出すことは不可能ではないと思われる。具体的には

$$\begin{aligned}
 W &= L \int D dA \\
 T_s &= \frac{1}{L} \int G r^2 dA
 \end{aligned} \quad (1)$$

のような式（これは断面形式が一定の棒の重さ  $W$ , ねじり剛性  $T_s$  を与える式で,  $D$  は材料の密度,  $G$  は剪断弾性係数) を見て, 「 $dA$  は  $W$  に対しては  $r$  に無関係であるが,  $T_s$  に対しては  $r^2$  に比例して効くことが分かるので,  $r$  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  $T_s$  を増加させることができる」というようなことを推論することである。

最近, Williams<sup>12)</sup> は, 深い知識を直接自動設計に結びつける「相互作用 (interaction) に基づく発明」と称する手法を提案している。これは部品に関する深い知識 (挙動を導出するのに必要な関係式) から設計仕様を満足する部品の組合せを自動的に求めようとするもので, 結局, 新しい設計は, 既存の設計の理解過程と通ずるものがあるという発想である。深い知識は定性推論の演算子を含む  $Q_i$  代数と呼ばれる記法で記述され, 値の伝播に定性推論を併用している。しかし, 1) 使用する部品をあらかじめ指定していること, 2) 一つの部品がなるべく多くの機能を満足するように配慮するなど探索量を減らす工夫はしているが, 各部品の機能を明示的に表現せず網羅的な探索を実施しているなど, 創造的な思考を模擬した設計というレベルには到っていない。

4. 機能知識の獲得

われわれは, たとえば, ある設計物を理解するのに, まず個々の部品の機能と部品間の構造との関係を解釈しようとし, そのために個々の部品の果たす役割を考え, 起こり得る挙動のうち適切なもののみを抽出し次へ進む。そして, 部品の幾つかをまとめて一つの機能をもつグループと認識し, 階層的に理解していく。いったん, 理解すると, 類似の構造に対しては, 分かっている解釈をそのまま適用する。このようにして, ある設計物に対する階層的な知識構造が獲得される。

図-5 は加算回路の一部である桁上げ連鎖回路<sup>13)</sup>に関する知識を階層的に示したイメージ図である。最下位のレベルはアナログ回路としての特徴に着目した場合の知識で, 112 個の回路方程式で表現できる。中間に 2 階層, スイッチ・レベルと論理レベ

ルの記述を挟んで最上位のレベルの知識は桁上げ連鎖回路の入出力関係を表現した一つの論理式で表現できる。各階層はそれぞれ異なった意味をもち, 表現する言葉も推論形態も違う。たとえば論理レベルでは, アナログレベルで 112 個の回路方程式で表現されていた回路が 8 個の論理方程式で表現されており, 論理的な性質に関する推論がしやすい表現となっている。

なお, この例では一つの回路 (デバイス) を, 抽象度の違う種々の見方で眺めているので, オントロジ (物事をどう捉えるかの存在論的な立場: たとえばデバイスの集合として捉えるか, あるいはプロセスの集合として捉えるかなど) としては一つであると考えるほうが自然であるが, 言葉が違えば領域知識も違うので, 各レベルは異なった領域知識を用いているという意味で, 別々のオントロジに対応しているという見方もある。

このように階層的な形で獲得された知識を解釈し直すと, 各階層間の上のレベルと下のレベルの知識の関係を, 仕様とその実現方法とみなすことができる。たとえば最上位の桁上げ演算回路 (仕様) は, 中間層の NOR, NOT などの論理回路に

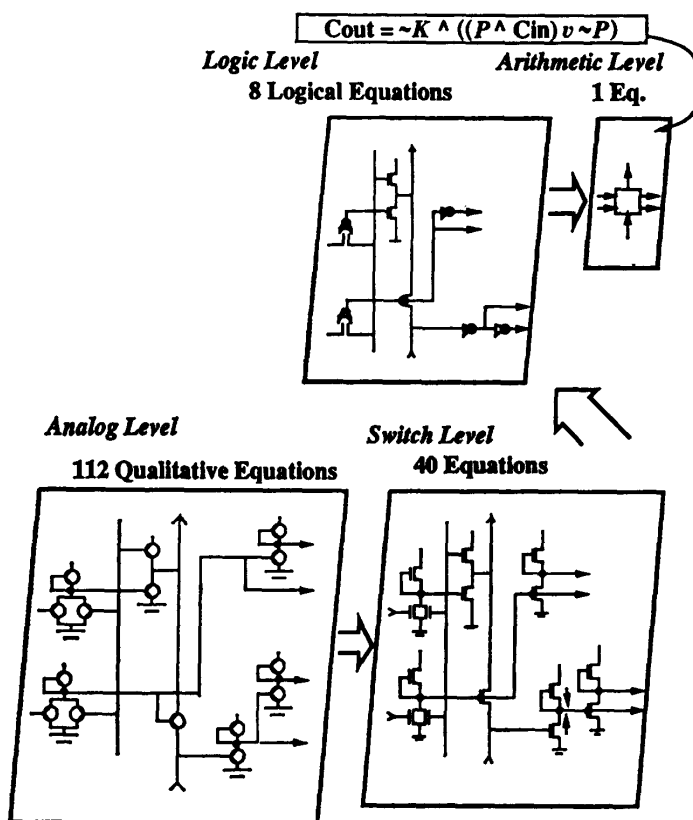


図-5 桁上げ連鎖回路の知識階層

より実現されていると考えることができる。個々の論理回路を仕様とみなした場合、最終的には MOS トランジスタにより実現されているとみなせる。

このような階層的な機構を用いれば、問題に合わせて使用する知識のレベルを変化させ、効率的に問題を解決することができる。たとえば、回路の算術演算に関する診断・挙動予測であれば、図-5 の最上位の知識を使用すればよい。三つの MOS トランジスタが NOR 回路として動作するということを説明するには最下位あるいは一つ上のレベルの知識が必要となる。最下位の知識を直接使って図-5 の回路が桁上げ演算回路であることの説明を試みるのは現実的ではない。

#### 4.1 専門家による知識の階層構造の抽出

筆者たち<sup>14)</sup>は、上位レベルの知識（機器の仕様と相当）と下位レベルの知識（その実現方法に相当）を入力し、両者の挙動が一致するために必要な仮定を見つけ、この仮定と両者の対応関係を機械学習の1手法である EBL<sup>15),16)</sup>により一般化して記憶する方法を提案している。この方法では、上位レベルと下位レベルの知識をつなぐために必要な仮定を明示的に記憶し、「近似」を「適当な仮定と論理的に正しい変換操作の組合せ」として扱っている。

この方法によれば、たとえば、図-5 に示した桁上げ連鎖回路全体の場合、スイッチ・レベルと論理レベルの関係「1個の Pull Up トランジスタと2個の Pull Down トランジスタで NOR 回路になる」を扱うには「Pull Down トランジスタの動作時は、Pull Up トランジスタが無視できる」が、算術演算レベルとの関係「全体として桁上げを計算する」には「C<sub>out</sub> がプリチャージされている、クロック信号がオフになっていること」などの「仮定」が階層構造を抽出するために必要であり、これらを見つけてくれる。

#### 4.2 機能階層知識の自動合成

上述した著者らの研究は与えられた階層間の知識の整合性保持に必要な仮定（近似）を抽出するもので、階層知識を自動生成するものではない。個々の事例から類似な特徴を取り出し一般化することにより概念を形成 (concept formation) する研究は、機械学習の一分野として従来より盛んに行われている。たとえば、Lebowitz の UNIMEM<sup>17)</sup>

は、そのような研究の代表例である（ただし、定性推論は使用していない。）

UNIMEM は多くの事実を入力し、初めに事実の中から共通した特徴（新しい概念に相当）をやはり機械学習の1手法である SBL<sup>18)</sup> の手法を用いて抽出する。たとえば、議会の投票記録から、「MX ミサイルの削減に反対する議員は防衛費の削減に反対する傾向が強く、軍関係の支出の多い選挙区であることが多い」ことを見つけてくる。次に領域知識（深い知識）を用いて、これらの傾向の因果関係を説明しようとする。このとき、抽出された特徴の中でほかの概念にあまり現れないものをほかの特徴の原因と考え、推論を制御する。説明に成功した特徴のみを意味あるものとして取り出し、EBL で一般化する。説明に失敗したものは単なる同時発生として学習結果から除去する。UNIMEM は SBL の学習結果の精度を EBL を用いて向上したシステムとみなすこともできるし、SBL で意味のありそうな固まり（概念）の候補を見つけた後、その概念を EBL で学習するシステムとみなすこともできる。

しかし、概念を生成するだけでは階層知識を構成できたことにはならず、生成された概念を用いた効率的な推論を実現するために、論理演算や桁上げ演算に相当する新しい領域知識をも、下のレベルの領域知識が抽象化されたものとして生成する必要がある。残念ながら UNIMEM では、意味のある固まりまでは認識するが、新しい上位レベルの領域知識までは生成していない。

### 5. 知識獲得と概念構造

知識の階層構造（概念構造）の生成は、自動化の程度により以下の三つに分類することができる<sup>19)</sup>。

#### 自動的な概念構造の生成：

1番下のレベルの知識だけから階層構造・階層ごとの領域知識を自動生成するシステム。たとえば、トランジスタの挙動に関する知識と、回路方程式の情報だけから、NOR 回路や、桁上げ連鎖回路を、意味のある固まりとして認識し、論理演算規則を抽出するもの。

#### 領域知識を限定した概念構造の生成：

上記のほかに、「論理回路になっているはず」というヒントを、上位レベルの領域知識（この場

合、論理演算規則)として与えるもの。新しい機械をリバース・エンジニアリングで理解していくことに相当し、上記の方法に比べ、探索空間を削減できる。

#### 専門家の知識を用いた概念構造の抽出:

さらに、どこが NOR 回路であるかまで、例を与えるもの。教科書による学習に相当する。

UNIMEM は完全に自動化した 1 番目のシステムの一機能のみを満たしたもので、著者らのシステムは自動化していない 3 番目のシステムの全機能を満たしたものと考えることができる。このような、計算機による概念構造の学習の自動化を考える上では、以下の 3 点を、よく考察しなければならない。

#### 5.1 学習すべき概念単位・概念構造の生成

何を学習すべきかという研究は、機械学習の研究の一環として盛んに研究されている。前述の UNIMEM も、その 1 例である。

UNIMEM では、初めに、事例から概念の候補を見つけた後、その概念を領域知識で確認し一般化していたが、逆に最初に領域知識を利用し事例を一般化し、その結果から概念構造を見つけようとした研究もある<sup>20), 21)</sup>。

Drastal らの MIRO<sup>20)</sup> は、Version Space における学習システムの 1 種である。食器類の特徴記述 (descriptor) とそれが学習すべき概念に相当するか否かの 2 値情報を入力した MIRO は、学習すべき概念を特徴記述の選言または連言形式で記述しようとする。このとき、概念を記述するのに利用する特徴記述として、個々の事例の記述に使われている特徴記述を全て使用するのではなく、その中で領域知識によりほかの特徴記述から証明されうるもの (maximal proven descriptor, MIRO では本質を表しているとして扱っているもの) を優先している。このように MIRO では概念記述に用いる特徴記述を最初に洗練することで、結果として簡潔な概念構造を効果的に生成している。

また、因果知識の生成を試みた研究もある<sup>21)</sup>。Pazzani らの OCCAM<sup>21)</sup> は、「異なった特徴をもつ対象に対する行動が異なった結果をもたらしたときには、異なった特徴を異なった結果の原因と推定せよ」といった知識 (Generalization rule) を使い、例題の解析結果から因果知識を推測し、後から入力した例題で推定した因果知識の検証を行

う機能をもっている。Generalization rule は学習すべき因果知識を探索するための仕組みと捉えることができる。因果知識がなんらかの概念に相当しているとみなせば、Generalization rule により概念の候補を生成し、生成した概念を後の例題で検証しているとみなせる。

以上の研究は、何を新しい概念単位・概念構造として学習すべきかという点において、示唆する内容を含んだ研究であるが、例題が単純、探索空間の量が創造的な発明を行うのに必要と思われる量に比較し少ないなどの理由で、「新しい概念を創造する」ところまでは至っていない。今後、さらに複雑な例題などを通じて研究を進めていく必要がある。

#### 5.2 概念構造の評価

作成した概念構造の評価に関しては Bennett<sup>22)</sup> や Ellman<sup>23)</sup> らの研究がある。どちらの研究も、領域知識に近似を入れ、与えられた問題の近似解を効率よく求め、最後に近似解の誤差を評価している。これらは、近似により単純化された知識を、もとの領域知識の中から重要な関係だけを抽出したものとみなせば、一種の概念構造を生成・評価した研究とみなせる。

Bennett<sup>22)</sup> は変数の大きさの情報をもとに、領域知識として与えられた数式を単純化し近似解を求めた後、誤差を計算し許容範囲内の回答を与えた近似の方法を EBL で学習するシステムを提案している。

たとえば、水中のイオン濃度を求める問題では、電荷の平衡に関する領域知識は “[H<sup>+</sup>] = [Ac<sup>-</sup>] + [OH<sup>-</sup>]” といった、方程式の形で与えられる。この式はイオン濃度の大小関係に関する定性的な知識 ([Ac<sup>-</sup>] ≫ [OH<sup>-</sup>]) を用いれば、“[H<sup>+</sup>] = [Ac<sup>-</sup>]” という形で近似できる。この近似式を用いてイオン濃度を計算した後、近似前の式を用いて値を検算すれば、誤差を見積もることができる。文献 22) では、この誤差が一定値 (5%) 以下であれば、近似の方法 (ルール) を EBL で学習している。

Ellman の開発した hearts というトランプ・ゲームをするシステム POLLYANNA も<sup>23)</sup> 領域知識を単純化し評価しているという点で Bennett の研究と対応させて考えることができる。

POLLYANNA はあらかじめ与えられた評価関

数をもとに、最も有利となるカードを次の手として選ぶ。評価関数は「手の善し悪し」という抽象的な概念を表す領域知識とみなせるが、正確に計算するには膨大な先読み計算を必要とする。そこで POLLYANNA は「ある値は定数とみなす／簡略式で代用」といった知識の書き換え規則を使って評価関数を単純化し、単純化した評価関数を使ってゲームを実行する。単純化した評価関数は「手の大雑把な善し悪し」という抽象度のレベルが変化した概念とみなせる。また、文献 23) で触れられている“contextual knowledge”を「どの程度の誤判定率・効率でゲームをするか」というタスクの定義とみなせば、単純化した評価関数は、もとの評価関数を特定タスク用にコンパイルした知識とみなすこともできる。Bennett のシステムと POLLYANNA の差は、獲得した概念の評価方法である。前者は計算誤差を用いていたが、後者は実際に獲得した概念（評価関数）を用いてゲームを実行し、手の善し悪しの誤判定の率と必要な計算時間の 2 点から評価している。

### 5.3 概念構造を利用した問題解決

矛盾のない対象の階層的な記述が、さまざまな問題解決を強力に支援することは明らかであるが、利用方法にも工夫が必要である。

Falkenhainer ら<sup>24)</sup>は、対象の一般的なモデルから解決すべき問題に応じて、適切な詳細度（問題を解くのに不必要に詳細でないもの）や視点（特定のパラメータの変化のみに興味があるなど）からながめたモデルを自動生成する方法を提案している。具体的にはモデルに用いた仮定 (Simplifying Assumption) を明示的に表示する枠組 (CONSIDER 述語) を準備し、各仮定に対応した部分モデルを取り出すことによって詳細度と視点を制御している。

どの仮定を選択すべきかは、質問中の語彙から CONSIDER 述語の引数に関する仮説を立て、回答を得るのに必要最小限の仮定（どの CONSIDER 述語が必要か）を ATMS で同定し、たとえば、システムの効率にはどんな因子が影響するか、バーナの煙が黒くなるのは何が原因かといった質問に答えられる適切なレベルのモデルを自動生成している。

Liu<sup>25)</sup>、Williams<sup>1)</sup>らも、解決すべき問題に応じて、適切な詳細度や視点からながめたモデルを生

成・選択する方法を提案している。

Liu<sup>25)</sup>らの方法は、物理システムを異なるオントロジに従って記述しておき、与えられた問題によって適切なオントロジを選択する定性推論システムを提案している。電気回路を例にとれば、一つの回路に対して、素子の抵抗や端子間に流れる電流・電圧間の関係の記述 (device ontology) と、電荷と電磁場の関係の記述 (charge-carrier ontology) を用意しておく。さらに二つの記述の関係も電流と電荷の流れの関係式などの形で与えておく。

どのオントロジを使用して問題を解くかは「あるオントロジに属する値の変化の影響を、そのオントロジにとっての公理も含めて説明するには、説明に関係するオントロジを選択しろ。」といった 4 種類のヒューリスティックスを利用している。たとえば、「抵抗の端子間の電圧を増やすと、流れる電流も増える」ことについて、オームの法則 (device ontology では公理) も含めた説明を求められれば、charge-carrier ontology の記述を利用して、「電圧が増えると、電場が強くなり、電子に働く力が増え、電子の速度が上がり、移動する電荷が増え、流れる電流が増える」という説明が生成できる。

どのモデルによって問題を解くかをヒューリスティックスによらずに選択する点で Williams<sup>1)</sup>の研究は興味深い。Williams の提案する方法では、初めに物理システムを物理量間の定量的方程式（たとえば水位と底部の水圧の関係式）の形で表現しておく。「底部をパイプで接続した二つの容器の水位が分かっているとき、水位の変化の方向はどうなるか？」といった質問に対して、1) 変数間の因果関係の情報を用いて、水位変化の方向に関係ある物理量と、その関係式を物理システムの記述から選び出す、2) 機器内部の状態を表し外部の変数と直接関係しない値は別の変数を使った形で解き直し消去する、3) 前 2 ステップの結果として得られた定量的方程式において、定性方程式にしても問題への解答が作成できるものを定性方程式に置き換える、の 3 ステップにより、解答を作成するために必要最小限の情報をもったモデルを作成し解答を作る。この方法によれば、上記の例では、水位変化からパイプの流量までは定性的、パイプの流量から底部の圧力・容器の水位ま



では定量的に表現したモデルが作成され、それに基づき解答が作成できる。

深い推論を実現するためには推論の基になる知識が完備していなければならない。この点に関連した野心的な研究として Stanford の Iwasaki らが現在進めている How Things Work Project<sup>26)</sup>がある。これについては本号別稿に詳述されている。

## 6. おわりに

定性推論を知識獲得のための要素技術として用いている幾つかの研究事例を紹介した。定性推論を狭く定義すれば、モデル（深い知識）を用いた推論の実現手法の一つに過ぎないので、本解説で紹介した手法のほとんどは他のモデルを用いた推論手法に対しても適用される。本解説でも、類似の考え方で、定性推論以外の推論手法で知識を獲得しているものも幾つか簡単に紹介した。逆に、定性推論の意味を最も広く捉えれば、人間の理解の形式は定性的であるから、すべての知的な行為の基礎をなすものであると位置づけてしまうことであろう。

定性推論をどのように定義するかは、人によって違うであろうし、本解説で取り上げる問題ではない。大切なことは理解の過程で知識がいかに構造化され、蓄積されていき、それが問題解決にいかにか寄与するかを追求していくことである。定性推論とは、このような試みの中で、従来の手法ではカバーしきれなかった部分をカバーするための新しい手法であると積極的に解釈すべきものなのである。

このような観点に立つと、本論文では紹介しなかったが、西田たちの最近の研究<sup>27)</sup>は注目すべきもののひとつである。

## 参考文献

- 1) Williams, B.C.: Capturing How Things Work: Constructing Critical Abstractions of Local Interactions. In *Working Notes of the AGAA-90 Workshop*, pp. 163-174 (1990).
- 2) Anderson, J.R.: Knowledge Compilation: General Learning Mechanism, *Machine Learning*, Vol. 2, pp. 289-310 (1983).
- 3) Chandrasekaran, B. and Mittal, S.: Deep Versus Compiled Knowledge Approaches to Diagnostic Problemsolving. In *The National Conference of Artificial Intelligence*, pp. 349-354 (1982).
- 4) Sembugamoorthy, V. and Chandrasekaran, B.: Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. In Kolodner, J. and Reisbeck, C. editors, *Experience, Memory, and Reasoning*, pp. 47-73, Lawrence Erlbaum Associates Publishers (1986).
- 5) Motoda, H., Yamada, N. and Yoshida, K.: A Knowledge Based System for Plant Diagnosis. In *FGCS*, pp. 583-588 (1984).
- 6) 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 (1987).
- 7) 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*, pp. 23-31 (1988).
- 8) 小沢, 野村, 山口, 溝口: 知識コンパイラ (KCII) の DST 化. In 人工知能学会研究会資料 *SIG-KBS-8905*, pp. 33-40 (1989).
- 9) Matsuda, K. and Mizoguchi, R. et al.: ISAK: Interview System for Acquiring Design Knowledge. In *JKAW 90* (1990).
- 10) Goel, A. and Chandrasekaran, B.: Functional Representation of Designs and Redesign Problem Solving. In *IJCAI-89*, pp. 1388-1394 (1989).
- 11) Murthy, S.S. and Addanki, S: PROMPT: An Innovative Design Tool. In *AAAI-87*, pp. 637-642 (1987).
- 12) Williams, B.C.: Interaction-Based Invention: Designing Novel Devices from First Principles. In *AAAI-90*, pp. 349-356 (1990).
- 13) Mead, C. and Conway, L.: *Introduction to VLSI Systems*, Addison-Wesley Publishing Company (1980).
- 14) 吉田, 元田: 階層的定性推論のための浅い知識の合成法, 人工知能学会誌, Vol. 4, No. 4 (1989).
- 15) Mitchell, T. M., Keller, R. M. and Smadar, T.: Kedar-Cabelli, Explanation-Based Generalization: A Unifying View, *Machine Learning*, pp. 47-80 (1986).
- 16) DeJong, G. and Mooney, R.: Explanation-Based Learning: An Alternative View, *Machine Learning*, pp. 145-176 (1986).
- 17) Lebowitz, M.: Integrated Learning: Controlling Explanation, *Cognitive Science*, Vol. 10, pp. 219-240 (1986).
- 18) Dietterich, T. G. and Michalski, R. S.: A Comparative Review of Selected Methods for Learning from Examples, In Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. editors, *Machine Learning*, pp. 41-81, Springer-Verlag (1983).
- 19) Yoshida, K. and Motoda, H.: Towards Automatic Generation of Hierarchical Knowledge Bases. In *Working Notes of the AGAA-90 Workshop, Boston*, pp. 98-109 (1990).
- 20) Drastal, G. Czako, G. and Raatz, S.: INDUCTION IN AN ABSTRACTION SPACE: A Form of Constructive Induction. In *IJCAI-89*, pp. 708-712 (1989).

- 21) Pazzani, M., Dyer, M. and Flowers, M.: The Role of Prior Causal Theories in Generalization. In *AAAI-86*, pp. 545-550 (1986).
- 22) Bennett, S. W.: Approximation in Mathematical Domains. In *IJCAI-87*, pp. 239-241 (1987).
- 23) Ellman, T.: Approximate Theory Formation: An Explanation-Based Approach. In *AAAI-88*, pp. 570-574 (1988).
- 24) Falkenhainer, B. and Forbus, K. D.: Setting Up Large-Scale Qualitative Models. In *AAAI-88*, pp. 301-306 (1988).
- 25) Liu, Z. Y. and Farley, A. M.: Shifting Ontological Perspectives in Reasoning about Physical Systems. In *AAAI-90*, pp. 395-400 (1990).
- 26) How Things Work: Knowledge-based Modeling of Physical Devices. In *Applied Artificial Intelligence Research: The Heuristic Programming Project*, pp. 5-18, Knowledge Systems Laboratory, Computer Science Department, Stanford University (1990).
- 27) Nishida, T., Fujisawa, H. and Doshita, S.: Abstracting Flow as Mapping. In *PRICAI-90*, pp. 384-389 (1990).

(平成2年9月3日受付)



元田 浩 (正会員)

昭和40年東京大学工学部原子力工学科卒業。同42年同大学院原子力工学専攻修士課程修了。同年、(株)日立製作所に入社。同社中央研究所、原子力研究所、エネルギー研究所を経て、現在、基礎研究所主管研究員。原子力システムの設計、運用、制御に関する研究、診断型エキスパート・システムの研究を経て、現在は人工知能の基礎研究、とくに定性推論、機械学習、知識獲得などの研究に従事。工学博士。昭和45年日本原子力学会奨励賞、同52、59年日本原子力学会論文賞、平成2年人工知能学会論文賞受賞。人工知能学会、日本ソフトウェア科学会、日本認知科学会、日本原子力学会、AAAI、IEEE Computer各会員。



吉田 健一 (正会員)

昭和55年東京工業大学情報科学科卒業。同年、(株)日立製作所に入社。エネルギー研究所にて、プラントの異常診断等の研究に従事。昭和61年より基礎研究所。定性推論、機械学習などの研究に興味を持つ。昭和59年日本原子力学会論文賞受賞。平成2年電気学会論文賞受賞。人工知能学会、ACM、AAAI各会員。