



Title	問題解決の基本操作に基づく知識獲得支援システムに関する研究
Author(s)	瀧, 寛和
Citation	大阪大学, 1991, 博士論文
Version Type	VoR
URL	<a href="https://doi.org/10.11501/3054497">https://doi.org/10.11501/3054497</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

問題解決の基本操作に基づく  
知識獲得支援システムに関する研究

平成3年1月

瀧 寛 和

大阪大学 博士論文

問題解決の基本操作に基づく  
知識獲得支援システムに関する研究

平成3年 1月

瀧 寛和

## 要旨

本論文は、知識ベースを対話的に構築するための知識獲得支援システムについてまとめたものである。本研究は、第5世代コンピュータプロジェクトの一環として、論理型推論マシンのための知識ベース構築・利用技術の研究としてなされた。

知識ベースの構築は、専門家などの知識源から知識を獲得することにより行われるが、この作業は非常に困難な作業であり、知識獲得ボトルネックと呼ばれている。知識獲得支援の目的は、この問題を解消することである。本研究において提案する新しい方法は、問題解決向きの知識表現を用いることにより、効率的な知識の抽出と知識の洗練とを実現するものであり、次に示す4項目に集約される。

1. 問題解決における操作の型 (operation type) の分類結果に基づく知識表現「専門家モデル」。
2. 問題解決で使われる操作の型を事前知識として知識ベースを構築する対話的知識抽出の方法 (プリポスト法)。
3. 事例から知識獲得を行う際の帰納的一般化 (inductive generalization) による知識推定と、知識推定を用いた提案誘導型知識獲得のためのアーキテクチャ。
4. 操作の型という枠組みを用いた操作機能実現性の観点からの知識の不備 (矛盾と不足) の検出、および、その不備を解消する知識の洗練 (refinement) 法。

以下、簡単にこの4項目について説明を行う。

1. 知識ベース化されたルール表現の解析と分類を行った。この分類結果から得られたのが操作の型 (operation type) である。さらに、操作の型を使って問題解決を表現する知識表現「専門家モデル」を開発した。
2. 操作の型を利用した知識抽出では、「その型の実現にどのような情報が揃っている必要があるか」をシステムの事前知識として利用可能である。この利点を活かして、知識獲得支援システムは知識抽出に有効な質問を生成できる。
3. 専門家が事例しか思い出せない場合の知識抽出方法として、事例からより一般的な知識を推定する方法を開発した。また、その際に、操作の型を利用して、帰納的一般化に必要な情報の提供を実現することにより、推定される知識候補を制限できた。また、候補の提案により、専門家に関連知識の連想を促すことが可能となった。
4. 抽出した知識は、そのままでは矛盾や不足などの不備が含まれているため、知識の洗練が必要である。操作の型を利用した知識の洗練では、抽出した知識が操作の型の機能を満足するかどうかを確認することにより、論理的な矛盾や問題解決に不足している知識の検出と解消を実現した。

これら4項目は、知識獲得支援システムにおける基本機能の実現に関する提案であり、知識の洗練をのぞく1, 2, 3項目は、逐次型推論マシン PSI-II 上に実現され、1, 2項目は、実験的にその機能の検証を成し得たと考える。

# 目次

要旨

目次

1	序論	1
2	知識獲得支援システムの歴史と課題	7
2.1	緒言	7
2.2	知識獲得支援の概要と諸問題	9
2.3	知識表現と知識の抽出問題	10
2.4	知識の整理・体系化問題	12
2.5	知識の洗練問題	12
2.6	知識獲得支援システムの歴史	13
2.6.1	ルール表現を利用した知識獲得支援システム	13
2.6.2	格子型知識表現を利用した知識獲得支援システム	15
2.6.3	ドメインモデルによる知識獲得	17
2.6.4	汎化タスクによる知識獲得	20
2.6.5	知識獲得支援と学習システム	23
2.7	結言	24
3	知識獲得の為の知識表現：専門家モデル	25

3.1	緒言	25
3.2	専門家モデル	26
3.2.1	専門家モデルの基本的アイデア	26
3.2.2	専門家モデルの基本構造	29
3.2.3	基本操作 (generic operation)	31
3.2.4	専門家モデルの処理対象要素	37
3.2.5	知識専門家モデルの推論制御知識	38
3.3	関連研究との比較	38
3.4	結言	39
4	専門家モデルを利用した知識抽出方法	41
4.1	緒言	41
4.2	知識獲得過程における知識抽出	41
4.3	対話による知識抽出方法の分類	42
4.4	知識抽出方法：プリポスト法	43
4.4.1	プリポスト法の概要	43
4.4.2	プリポスト法の知識抽出手順	44
4.5	知識獲得支援システム EPSILON/One	48
4.5.1	知識の抽出モジュール	49
4.5.2	知識の変換・推論モジュール	50
4.6	関連研究との比較	52
4.7	システムの評価と拡張について	52
4.8	結言	53
5	提案誘導型知識獲得支援システム	55
5.1	緒言	55
5.2	専門家モデルと帰納推論による知識獲得	56

## 目次

5.2.1	専門家モデルと知識獲得	56
5.2.2	操作の型に対応した帰納的な知識推定アルゴリズム	57
5.2.3	選択・分類操作の推定	59
5.2.4	順序付け操作の推定	64
5.2.5	値入換え操作の推定	66
5.2.6	複数属性を持つ評価基準の知識の推定	67
5.2.7	ヒューリスティックスの利用	70
5.3	提案誘導型知識獲得支援システム	71
5.3.1	提案誘導型知識獲得支援による知識獲得	71
5.4	関連研究と課題	73
5.5	結言	75
<b>6</b>	<b>操作の型を利用した知識の洗練</b>	<b>77</b>
6.1	緒言	77
6.2	知識の洗練	78
6.2.1	操作知識と洗練	78
6.2.2	洗練戦略	79
6.3	操作の型を利用した洗練	81
6.3.1	分類と選択操作	81
6.3.2	順序付け操作	87
6.3.3	値入換え操作(変換操作)	89
6.3.4	その他の操作	94
6.4	関連研究	95
6.5	結言	97
<b>7</b>	<b>専門家モデルの操作の型の数学的な観点からの再構成</b>	<b>99</b>
7.1	緒言	99

7.2	関係代数と専門家モデルの関係	99
7.2.1	関係代数	100
7.3	専門家モデルと関係代数の比較	102
7.3.1	処理対象要素と関係データベースの違い	102
7.3.2	操作の型と関係代数の操作の対応付け	103
7.4	設計問題に対する操作の型の追加	105
7.5	操作の型の再構成	106
7.5.1	従来の操作の型との相違点	109
7.6	結言	110
8	操作を用いた知識獲得支援の総合評価	113
8.1	緒言	113
8.2	知識獲得支援における知識表現の水準	113
8.3	知識獲得支援における問題解決のモデルの役割	114
8.4	結言	119
9	結論	121
	謝辞	123
	参考文献	125
	研究業績	139
A	減速機の設計知識の獲得	143
B	コンタクトレンズの選定知識の専門家モデルでの表現	147
C	ベアリング選定知識の推定	149

# 第 1 章

## 序論

人工知能研究の初期は、特定の問題を解く知能レベルの高さよりも幅広い問題に対応できる問題解決の一般的仕組み、つまり、推論機構の研究が中心であった。これに対して、エキスパートシステムは、特定の対象分野の知識を持つことで、その分野では人間の専門家に匹敵する問題解決を持つシステムである。つまり、エキスパートシステムの研究では、知識と知識ベースの構築方法に主眼が置かれている [Abe 86]。

DENDRAL[Buchanan 69] に始まるエキスパートシステムの研究では、スタンフォード大学のファイゲンバウム (Feigenbaum) 教授の提案した「知識工学 (knowledge engineering)」の基本思想である

「知識は、力なり (Knowledge is power)」の考え方により、知識の研究が中心となってきた。エキスパートシステムは、推論機構と知識ベースを分離することにより、知識ベースを入れ換えることで、色々な問題に対応できる構成を特徴としているが、その推論機構は演繹推論を行うためのものであり、問題解決の能力は、知識に大きく依存するものとなっている。当然ながら、知識ベースはシステムの利用対象ごとに関係しなければならぬ。つまり、知識ベースを構築するために必要な知識獲得 (knowledge acquisition) は、新しいシステムの開発のたびに行わなければならない重要な作業となっている。ところが、知識ベース構築は非常に困難な作業で

あることが、エキスパートシステムの開発とともに認識され、知識獲得ポトルネックと呼ばれるようになった。初の知識獲得支援システム TEIRESIAS [Davis 79] は、医療診断エキスパートシステム MYCIN [Buchanan 83] の知識を洗練する目的で開発された。また、上記の DENDRAL のためには、Meta-DENDRAL [Buchanan 78] が知識の生成の目的で開発された。

知識ベースの構築には、専門家などの知識源から知識を獲得する対話による知識獲得の方法と、事例から知識を生成する帰納的学習 (inductive learning) による方法がある。帰納的学習は、帰納的一般化とも呼ばれ、豊富な事例が準備できる分野において有効である。上記の Meta-DENDRAL は、帰納的学習の機能を備えたシステムである。一方、対話型の知識獲得は、専門家の協力が得られる環境において、専門家の知識が十分にあり、その経験的知識が問題解決に利用できる分野で有効な方法である。

しかしながら、第2章で詳述するように、対話型の知識獲得の実現には、知識獲得に向けた知識表現、知識の抽出法と知識の洗練技術開発が必要である。エキスパートシステム構築ツールが提供するものは、プロダクションルールのような知識表現であり、専門家の問題解決の知識を表現するには、あまりにも一般的すぎた。そのため、知識処理技術者 (knowledge engineer) は、問題解決の仕組みが実現できるように、ルールの記述を工夫する必要があった。ルール表現だけでは、知識を抽出することが難しいため、初期の知識獲得支援システム TEIRESIAS や SEEK [Politakis 84] は、ルール型の知識を洗練することに主眼がおかれた。その後、特定の問題解決を想定して、その問題解決に利用される知識を特別の表現で抽出や洗練を行うシステム ETS [Boose 86] や MORE [Kahn 85] が開発された。これらのシステムに欠けていたのは、問題解決を表現する知識表現とその獲得方法であった。本論文の第3章では、この問題解決を表現できる知識表現とし

て、専門家モデルを提案している。専門家モデルは、いくつかの問題解決の単純なモデルと実際のルールで記述された知識ベースの分析結果から作られたものであり、7つの基本的な型を持つ「操作(オペレーション)」の集まりにより問題解決の過程を表現するものである。操作が型を持つということは、その操作を実現するのにどのような情報が必要であるかを予め知識獲得支援システムに持たせることができることを意味する。また、問題解決の知識が決まるとそれに利用される知識が持つべき情報を決めることができる。第4章では、この専門家モデルを利用した知識獲得方法として開発したプリポスト法を説明する。この方法では、問題解決の知識を操作の集合と操作間の関係として獲得する。そして、各操作の型を詳細化する情報を得るための質問を生成して、詳細な問題解決の知識を獲得する。さらに、この問題解決の知識を利用して、その問題解決の対象となる知識を獲得する。

また、この方法の実証のために逐次推論マシン PSI-II[Mitsubishi 88] 上に実現した知識獲得支援システム EPSILON/One[Taki 87c,89c] について論じる。専門家モデルは、診断型の知識ベースの分析から得られた操作の型を利用しているが、その利用範囲を調べるため、設計問題への適用した際の評価結果についても報告する。

知識獲得支援システムが獲得しなければならない知識は、問題解決に利用できる水準に一般化されたものである必要がある。しかしながら、ときには、事例しか獲得できない場合もある。事例からより一般的な知識を生成するには、帰納的学習の方法が必要となる。従来の知識獲得支援システムでは、対話による知識獲得と帰納的学習を併用したシステムはなく、直接一般化水準の高い知識のみの獲得を行っていた。また、帰納的学習の大きな欠点は、学習結果の候補が多く作られることと、その候補の多さ故に候補生成の計算量が莫大になることである。第5章では、専門家から操

作の知識を抽出する際に、事例から操作の知識を推定する方法と推定した知識を知識獲得支援システムの提案として専門家の知識の連想を促す知識獲得支援システムの枠組みについて論じる。この章での要点は、専門家モデルの操作の型を帰納的一般化の候補の形式を制限する情報に利用することで、候補の数の増加を抑えることが可能となり、帰納的学習を知識獲得支援システムに融合できたことである。

獲得された知識には、矛盾などの不備が内在している場合が多い。この不備(矛盾、冗長や不足など)を検出して、不備を解消するのが、知識の洗練である。従来の知識獲得支援システムにおける知識洗練は、洗練の対象となる知識を使用して推論を行い、システムの推論結果と専門家の推論結果の差から知識を洗練することが中心であった。これは、プログラム開発における特定の入力・出力を例題としたデバッキングと同様の方法であるため、試された例題に対しての不備の検出・解消しかなかった。しかし、知識ベースの使用目的が分かれば、知識が本来どのような情報を提供できなければならないかは、推論の実行以前に検出できるはずである。問題解決の枠組みを特定の診断方法に限定して、知識の洗練を実現したのが知識獲得支援システム MORE である。しかし、種々の問題解決の知識を洗練する効率的な方法を持った知識獲得支援システムは開発されていなかった。このような機能の実現には、問題解決を表現できる知識表現を用いた知識の洗練方法の開発が必要である。そこで、第6章では、専門家モデルの操作の型の情報を利用して、操作の型の実現に不足および、矛盾を生じる状態を検出することによる知識の洗練について論じる。操作の型を用いた知識の洗練方法の特徴は、その知識を問題解決に利用する以前に、その操作の型の機能実現に情報が十分であるかを検出できるばかりでなく、問題解決の過程で操作が対象の知識(データ)に対して操作の型の機能を発揮できたかを調べることで、対象領域に対する知識の不備を検出できる。

また、第7章では、関係代数と専門家モデルの比較と、設計問題での知識獲得支援システム EPSILON/One の評価結果に基づいて、専門家モデルの操作の型の再構成を論じる。

第8章では、操作を用いた知識獲得支援の総合評価として、知識獲得支援における知識表現の水準と知識獲得支援における問題解決のモデルの役割について論じる。

本研究において、知識獲得支援システム研究への大きな貢献を果たしたのは、知識表現「専門家モデル」と操作の型の開発である。この知識表現は、知識抽出方法と知識の洗練方法に有効に利用され、操作の型を利用した知識獲得法(プリポスト法)、操作の型を利用した帰納的一般化と知識の提案による知識抽出の枠組み、操作の型を利用した知識の洗練方法を生み出すことが出来た。



## 第 2 章

### 知識獲得支援システムの歴史と課題

#### 2.1 緒言

近年、活発に専門家の行っている種々の知的な情報処理をエキスパートシステムとして実現する試みが行われている。エキスパートシステムの基本要素は、推論機構と知識ベースであるが、エキスパートシステム構築ツールには推論機構は演繹推論機能としてシステムに組み込まれているものの、知識ベースはエキスパートシステムの用途ごとに専門家等の知識源から獲得しなければならない。ところが、この知識ベースの構築の為の知識獲得は非常に困難な作業となっており、知識獲得ボトルネックと呼ばれている。知識獲得と言っても、人間の専門家からの知識獲得と教科書（テキスト）や手引書（マニュアル）からの知識獲得では、その方法や技術も異なっている。人間の専門家からの知識獲得では、人間と対話するインターフェイス、専門家の問題解決を表現できる知識表現、知識連想法と知識の洗練の技術などが必要である。知識獲得に必要な技術を明かにする目的で行なわれた「知識処理技術者による実際のエキスパートシステムの構築」に関する調査の結果、知識ベースの構築過程は次のような過程で構成されていることが分かった [Suwa 86][ICOT-JIPDEC 87]。

1. 問題の設定: 開発の意義と実現可能性の検討。

2. 既存技術の評価: エキスパートシステム技術の必要性の検討。
3. 知識源の同定: 知識の所在とその質・量の分析。
4. 専門家の問題解決モデルの同定: 知識源が人間のときに、問題解決と推論方法の分析。
5. 利用者の使用形態の同定: システムの利用目的と利用方法の分析。
6. 知識表現の選択: 知識表現の選択と組合せの検討。
7. 知識の抽出: 決められた知識表現での知識抽出。
8. 知識の変換: 獲得された知識からエキスパートシステム構築ツール用の知識表現への変換。
9. 知識ベースの評価: 知識ベースの性能や不備の解消。
10. 知識ベースの管理: 知識の整合性の検証や維持。

この過程の内 1. 2. 3. は、システム開発の決定者のシステム分析作業である。知識処理技術者の主な作業はこの過程の内 4. から 9. までである。この過程に含まれる作業の支援には次のものが考えられる。

- 知識の抽出支援: 専門家の問題解決過程を表現し易い知識表現、知識の連想支援や対話型インターフェイスの提供。
- 知識の整理・体系化支援: 断片知識の整理と高次の知識表現への変換支援。
- 知識の洗練支援: 獲得された知識内の不足・冗長・矛盾の検出と解消支援。

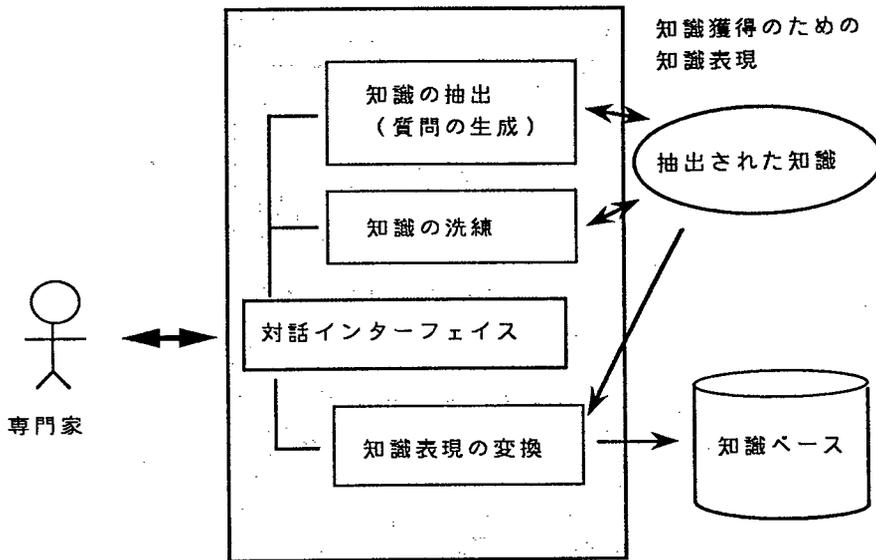


図 2.1: 基本的な知識獲得支援システムの構成

## 2.2 知識獲得支援の概要と諸問題

対話型による基本的な知識獲得支援システムの構成を図 2.1 に示す。知識獲得支援システムの主たる機能は次の様になる。

- 専門家に知識を連想させる刺激（質問）を与える知識抽出機能
- 抽出した知識を表現する知識獲得のための知識表現
- 目的とする知識のモデル（事前知識）との差異を検出し、質問を生成する知識の洗練機能
- 獲得した知識をエキスパートシステム構築ツールで利用する知識表現に変換する知識表現の変換機能

専門家に知識を連想させる為には、それなりの刺激が必要となる。この分野はもともと心理学の分野であり、心理学的方法を使った知識獲得支援システムもある（後述のパーソナル・コンストラクト心理学 (personal

construct psychology)[Kelly 55] は心理学的知識抽出方法の一例)。知識獲得支援システムには、抽出した知識を表現する為の知識表現が必要である。知識表現が特殊であれば、特定の問題の表現が容易になる反面、その利用できる対象領域は限られたものとなる。一方、一般的な知識表現は個々の問題解決の知識を記述する技術を必要とする。つまり、知識表現が知識抽出の方法に強く影響する。獲得した知識には、矛盾などの不備が含まれていることが多い。その知識の不備を除く技術が必要である。これは、知識の洗練（精練、リファインと呼ばれることもある）の方法として研究されている。次に、知識獲得を困難にしている問題について述べる。

### 2.3 知識表現と知識の抽出問題

知識抽出は専門家から必要な知識を効率良く抽出する問題である。一般に、専門家は自分の知識の内容と利用方法（推論方式）について、あまり意識をしていない。そのため、専門家に知識を連想させる方法が必要となる。また、専門家の表現と知識表現の格差も知識抽出を困難にしている、その為、専門家に自分の知識を表現できる枠組みが必要である。

エキスパートシステムの構築に必要な知識は「問題解決過程を表す知識」「問題解決に利用される対象知識」に分けられる。たとえば、電子回路の診断であれば、電子回路の素子や結線情報が対象知識であり、回路の診断手順や故障の判定知識が問題解決過程を表す知識である。問題解決全体のモデルは、問題解決モデルと呼ばれ、問題解決モデルを構成する要素はタスク (task) と呼ばれている。タスクはさらに操作 (operator, operation) に分解できる。プロダクションルールはさらに水準の低い知識として位置づけられ、実装水準 (implementation level) の知識表現と呼ばれている。知識獲得支援システムにおいてどの知識表現の水準で知識獲得を行うかは重要な問題である。

たとえば、ルール表現 (IF-THEN 型のプロダクションルールなど)、フレーム表現や意味ネット表現がエキスパートシステム構築ツールによって提供されているが、これらは、非常に基本的な知識表現であり、実装水準の知識表現にすぎない。タスク水準の知識表現としては、汎化タスク (generic tasks)[Chandrasekaran 86] が提唱されている。これはエキスパートシステムに良く現れる問題解決のタスク構造の知識表現である。

問題解決のモデルは、解析型問題と合成型問題では大きく異なっている [Sawamoto 88a, 88b]。解析型問題とは、診断型問題に代表されるように予め探索すべき空間とその解の候補が分かっている問題であり、その探索空間はあまり大きくない。しかし、合成型問題は、設計・計画型問題に代表される問題でその探索空間は推論の実行時に生成され、探索空間は非常に大きい場合が多い。前者に比べて、後者はより効率的な問題解決の知識を必要とする。

知識の抽出では、知識獲得支援システムが獲得対象についての情報を予めどの程度持っているかでその効率が左右される。システムが特定対象の問題解決モデルを持っていれば、後は対象知識のみを獲得することになるがその他の問題の知識獲得ができない。システムがほとんど獲得対象に関する知識 (問題解決モデルも含む) を持っていなければ、プロダクションルールなどの汎用的な知識表現で知識獲得を行うことになるが、汎用的な表現で特定の問題を表現する記述技術が必要となる。また、事前の知識が無い場合、効率的な知識獲得は難しい。

このような理由から知識表現は知識獲得において重要な役割を果たしていると言える。

## 2.4 知識の整理・体系化問題

断片的に抽出された知識を整理して、特定の知識表現で表す問題である。整理の方法における整理の観点は1つではないが、よく利用されるものに知識の階層化がある。まず、同意語を整理して、取り扱う概念(単語で表現されるものと考えて良い)を決める。次に、同じ概念を集合(グループ)化して、それを代表する概念を決め、その新しい概念について、さらに集合化を進めると、概念のクラス階層ができる。KJ法[Kawakita 67]は、発想のための知識整理には非常に有効であることが知られているが、上記の概念のクラス階層作成にも利用できる。このKJ法を利用した知識獲得支援システムにCONSIST[Shinohara 86]がある。この他に、概念間の関係整理には、表やネットワークが利用される。以上は、主に獲得する知識の問題解決モデルが不明な場合に利用される。獲得する知識の構造(汎化タスクなど)が、分かっている場合は、その知識の構造に合うように知識の抽出した方が効率が良い。このような断片的な知識からの知識ベースの構築は知識獲得としては効率が悪く、また、いつ知識ベースが十分になるかの判定ができない欠点がある。

## 2.5 知識の洗練問題

知識の洗練問題は、知識ベース内の矛盾・冗長・不足を検出して、その不備を除く処理を行う問題である。論理型知識表現での無矛盾性の維持は、一慣性の管理と呼ばれており、ある項目の肯定と否定を同時に導かないように、知識ベースを保つことである。この論理的矛盾以外の矛盾は、陽にどのような状態が矛盾になるかを知る必要がある。たとえば、対象が物理世界であれば、「同時に同じ場所に2つのものが存在すると矛盾である」や「1つのものが同時に2箇所に存在すると矛盾である」などはその

例である。つぎに、冗長性の除去であるが、同じ知識の検出や知識の包含関係を調べる機能が必要となる。知識の不足の検出は、知識の使用目的に関する情報なくしては困難である。その知識の使用目的に対して、どのような知識が足りないかを調べなくてはならない。知識ベースの管理についても、知識の追加・変更・削除時に洗練問題と同じ問題を抱えている。

## 2.6 知識獲得支援システムの歴史

本節では、実際に計算機上にして開発されている知識獲得支援システムについて説明し、その特徴と問題点を明かにする。知識獲得支援システムの歴史は、そのシステムに利用される知識表現の歴史であると言っても過言ではない。そこで、知識表現の観点からシステムを分類して、プロダクションルールを利用した知識獲得支援システム、格子型知識表現（レイティング・グリッド、rating grid）を利用した知識獲得支援システム、ドメインモデル（対象知識のモデル）を利用した知識獲得支援システム、タスクや操作（operator）を利用した知識獲得支援システムの順に説明する。

### 2.6.1 ルール表現を利用した知識獲得支援システム

初期の知識獲得支援システムでは、知識の抽出よりも既に獲得されている知識ベースの洗練の機能に重点が置かれていた。医療診断システム MYCIN の推論構造と知識表現を元に作られたエキスパートシステム構築ツール EMYCIN [van Melle 79] に組み込まれているプロダクションルールの洗練システムが TEIRESIAS である。このシステムは、専門家の判断と推論結果が異なる場合に、推論過程を辿ることで専門家と対話しながら不適切なルールの修正を支援する。具体的には、「ルールが余分な結論を持っている」「結論に不足した部分がある」「ルールの条件が厳し過ぎる（条件の一般化が必要）」「ルールの条件が緩過ぎる（条件の特殊化が

必要)」「誤ったルール」「ルールが不足している」場合に対して対話的に質問を行ないながら洗練を行う。専門家の判定事例(事例と数種類に限定された専門家の判定結果)を用いたルールの修正を行うシステムがSEEKである。ルールの誤りを複数の事例から統計的に判断して、ルールの洗練の案を生成する。この洗練案は、ルールの一般化と特殊化の方法を提供する。

TEIRESIASの欠点は、ある問題での専門家との推論結果の違いに注目しているため、推論過程を辿りながら知識の修正を行うので、推論の深さが深いと修正可能箇所が多くなりどの修正が有効かの判定が難しい、また1つの事例に対して知識を洗練しても他の事例に有効であるかが保証されない。

SEEKの方法では、ルールの一般化案は専門家の判断事例から生成することは比較的容易であるが、特殊化を行なう場合に判定結果だけでは情報が不足しているため対話的にルールを特殊化する情報(ルールの条件部)を抽出する必要がある。また、ある程度、専門家の判定事例が収集されている問題にのみ有効である。

### プロダクションルールの問題点

プロダクションルール型の知識の洗練では、知識自体が一般的であるためその知識がどのような役割を問題解決過程で果たすのかが分からないため、専門家の判定結果が知識の洗練に必要である。また、プロダクションルールが一般的な知識表現であるがゆえに、その知識表現を利用した知識抽出方法は開発されていない。

CTAS (version 1.0)				
RATING GRID				
1	2	3	4	
5	5	1	4	: tawny_color/not_tawny_color
5	2	4	3	: run_fast/run_slow
5	1	5	4	: group/not_group
1	:	cheetah		
2	:	tiger		
3	:	wolf		
4	:	fox		

図 2.2: 格子型知識表現の例

## 2.6.2 格子型知識表現を利用した知識獲得支援システム

格子型知識表現（レイティンググリッド、rating grid）は、項目と特徴の関係を表わした知識表現である。図 2.2は、CTAS の格子型知識表現の一例である。

この知識表現を利用している知識獲得支援システムには、ETS、AQUINAS[Boose 87]、CTAS[椿 88]、KSS0[Gaines 81]がある。これらのシステムは、分類問題のエキスパートシステムの知識ベース構築を支援する。分類問題は、与えられた特徴を満たす、または、その特徴に近い項目を選ぶ問題である。この問題に必要な概念は、項目、特徴と項目・特徴間の関係の強さである。格子型知識表現は、この概念を含んだ知識表現である為、分類問題の知識獲得の知識表現として利用されている。知識獲得には知識表現だけでは不十分である、つまり、その知識表現に合った知識の連想を促す心理学的方法が必要である。パーソナル・コンストラクト心理学 (personal construct psychology) と呼ばれる知識連想方法が抽出する知識表現の構造と格子型知識表現は類似の表現である。パーソナル・コンストラクト心理学は、異なる 3つの項目を提示して、2つに共通で他

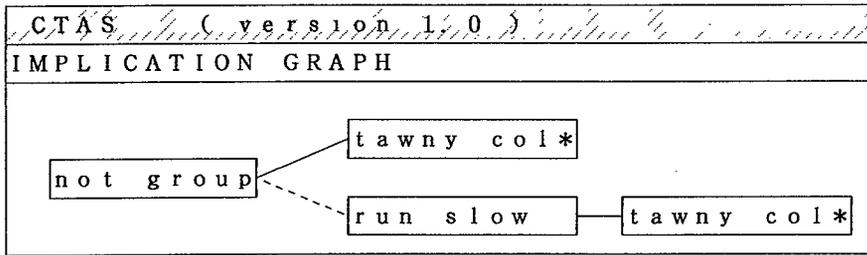


図 2.3: 特徴間の導出グラフ

の1つにない特徴を連想させる方法である。連想された特徴とその反対の特徴の対をコンストラクト (construct) と呼ぶ。ETSでは、分類対象の項目を専門家に述べさせた後、パーソナル・コンストラクト心理学によって、分類に利用できる特徴を抽出する。その後、各項目と特徴間の関係を度合い表現で抽出する。この関係度は、1から5の5段階で表現され、5が項目と特徴間に強い関係があることを示しており、1は、その項目が反対の特徴を持つことを示している。3は無関係を示し、4は弱い関係、2は負の弱い関係を示している。この関係度の抽出をレイティング (rating) と言う。次に ETS は格子型知識表現の洗練 (不足・矛盾・冗長) を行う為に、格子型知識表現を別の形式の表現で専門家に提示する。1つは、特徴の導出関係を表す導出グラフ (implication graph) であり、もう1つは、項目間の類似度を表す類似分類木 (cluster tree) である。図 2.3は、CTASの導出グラフの例である。これは、格子型知識表現の横の値の列を特徴に関するベクトルと見て、ベクトル間の距離が近いものは導出関係ありとして表現している。図中導出関係が強いものは、実線で示されている。極端な例として、同じベクトルを持つ特徴は等価と見なされることになる。類似分類木も同様に格子型知識表現の値の列を特徴ベクトルとして、類似度を計算する。これは項目の類似度だけでなく、特徴の類似度も表現可能である。専門家はこの導出関係や類似度を見て、関係が不適當なところをさ

Dialog manager						
ETS Repertory grid tools	Hierarchical structure tools	Uncertainty tools	Internal reasoning engine	Multiple scale type tools	Induction tools	Multiple expert tools
Object-oriented DBMS						
CommonLoops/CommonLisp						

図 2.4: AQUINAS のツールセット

がし、「関係度の修正」「新しい項目や特徴の追加」などを行う。ETSでは、分類タスクの作業を特徴群が入力されて、その特徴に合う項目を適合順に示すものとしている。ETSは、格子型知識表現の知識表現を上記タスクを実行できる確信度付きプロダクションルールに変換する。

AQUINASは、このETSに格子型知識表現の階層表現と関係度の拡張(5段階値から名義尺度、順序尺度、間隔尺度と比率尺度に拡張)を行った知識獲得支援システムである。図2.4は、知識獲得の為のワークベンチを目指したAQUINASのシステム構成を示している。

### 格子型知識表現の問題点

格子型知識表現での知識獲得は、獲得できる知識が分類対象の項目と分類する特徴の間が直接関係付けられる単純な分類問題に限られている点である。また、分類のための問題解決知識(たとえば、どの特徴から調べるべきかの戦略)は獲得できないため効率的な分類を必要とするエキスパートシステム向きではない。

### 2.6.3 ドメインモデルによる知識獲得

エキスパートシステムが処理する問題領域に含まれる概念と概念間の関係をネットワーク構造で示したものがドメインモデルである。各ドメイ

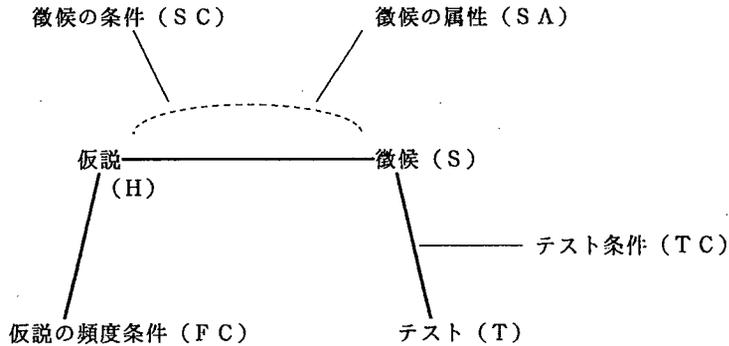


図 2.5: MORE のドメインモデル

ンごとに概念(ネットワークのノードで表す)やそのネットワーク構造が異なる。格子型知識表現では、項目と特徴が概念として表され、関係度が概念の関係を表していると見ることができ、その項目や特徴に関する制限はない。ドメインモデルでは、たとえば、徴候、原因やテストなどが概念であり、徴候になりえないものはそのモデルでの徴候に登録できない。つまり、格子型知識表現より特殊化された知識表現と言える。

MORE のドメインモデルの例を図 2.5 に示す。このドメインモデルは、掘削泥水診断システム MUD の知識ベースのモデルとなっている。ここでは、徴候・仮説・テストが主な概念として扱われている。また、確信度をモデルに付加することもできる。MORE はこのドメインモデルを洗練するために 8 つのインタビュー戦略を持っている。その戦略は以下の 8 つである。

1. 仮説群を区別する徴候の決定 (differentiation)
2. 仮説の発生に関する条件の決定 (frequency conditionalization)
3. 共通の徴候を持つ仮説群を区別に関する特徴の決定 (symptom distinction)

4. 徴候の発生に関する条件の決定 (symptom conditionalization)
5. ある仮説からある徴候へのパス間への新しい徴候の決定  
(path division)
6. 共通の徴候を持つ仮説群の区別に関する新たな徴候の決定  
(path differentiation)
7. より正確に徴候を発見する検査方法の決定 (test differentiation)
8. 検査の精度に関する条件の決定 (test conditionalization)

これらの戦略では、どのようなドメインモデルの関係が良くないかを探して、その不備を補う情報に関する質問を生成する。図 2.6 は MUD の知識をドメインモデルで表したものである。たとえば、「徴候：密度の減少」から「仮説：水の流入」と「仮説：油母頁岩の混合」が導けるそこで徴候の区分戦略 (symptom distinction) 戦略の質問により「減少速度が遅い」「減少速度が速い」の情報を抽出して仮説を特定するように知識を洗練する。

cover-and-differentiation 戦略と呼ばれる特定の問題解決において、この MORE のドメインモデルは利用される。MORE の拡張として、ドメインモデルの確信度の自動生成・修正を特別なヒューリスティックスを使って支援する知識獲得支援システム MOLE[Eshelman 86] も開発されている。設計仕様を評価する知識を獲得する知識獲得支援システム KNACK[Klinker 86] もドメインモデルを利用している。ドメインモデルとその洗練戦略を記述できる知識獲得支援システムの構築ツール SIS [Kawaguchi 87] も開発されている。

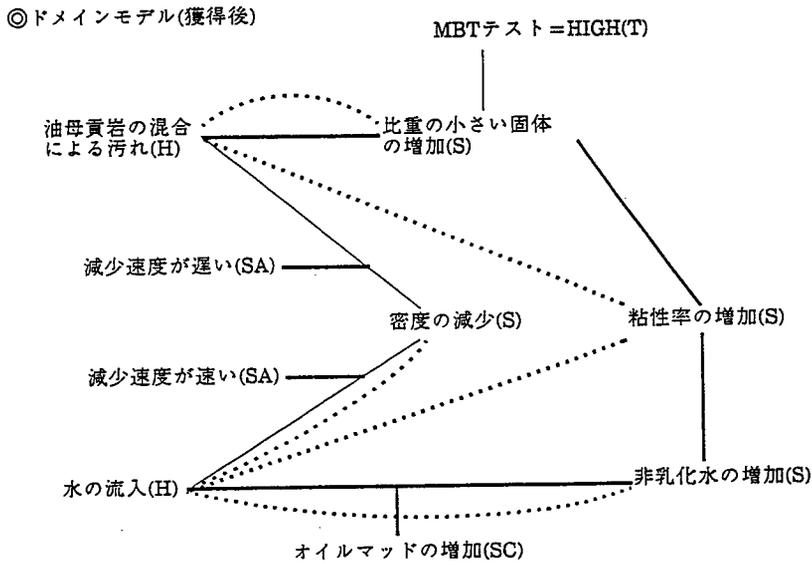


図 2.6: ドメインモデルの例

### ドメインモデルの問題点

ドメインモデルは、問題解決のモデルが最初に有って、その問題解決に利用される知識の概念の関係を表すモデルである。つまり、MOREは診断問題の問題解決での知識の洗練に特化したシステムであり、問題解決の知識自体を獲得する支援は行わない。

#### 2.6.4 汎化タスクによる知識獲得

汎化タスクは、エキスパートシステムの問題解決の汎化構造に注目した知識表現である。現在、汎化タスクとしては以下のものが発見されている。

1. 「階層化分類 (classification)」
2. 「プラン選択と詳細化による設計  
(design by plan selection and refinement)」
3. 「知識データの検索 (knowledge-directed data passing)」

4. 「仮説生成 (abductive assembly of hypothesis)」
5. 「状態のアブストラクション (state abstraction)」
6. 「仮説照合 (hypothesis matching)」

この内、「階層化分類」「仮説照合」が診断型問題に向くことが分かっている。これら2つのタスクをサポートするエキスパートシステム構築ツールに CSRL[Bylander 86]がある。また、CTASは、「階層化分類」と「順序付け (仮説照合の一種)」のタスクに基づいて知識獲得を行う知識獲得支援システムである。CTASは、分類対象の項目を専門家から抽出後、項目を2つの群(グループ)に分ける特徴を抽出する。各グループを分ける特徴が連想されなくなるまで、2つのグループへの分割が続けられる。ここで抽出される知識は、分類木(項目を二分する特徴と項目の集合からなる二分木)であり、階層化分類を行う知識となる。分割が難しくなった似たものの項目集合は、順序付けタスクの処理対象となる。CTASは上述のETS、AQUINASと同様に格子型知識表現を各似たもの項目集合ごとに獲得し、洗練する。CTASは、プロダクションルールから成る階層化分類ルールと順序付けタスクのルール(項目と特徴間の関係度を表すルールと特徴間の導出関係のルール)を知識ベースとして生成する。ルーチン設計の為のエキスパートシステム構築ツールとして DSPL[Brown 86]が開発されている。この DSPL のタスク表現に従った知識獲得を支援するのが、DSPL-ACQUIRER[Chiang 87]である。このシステムは、DSPL のタスクモデルを詳細化しながら知識獲得を行う。獲得は、「設計対象の(部品の)階層知識」「各部品の設計管理知識 (specialist)」「設計プラン知識 (plan)」「プラン内のタスク知識 (task)」「タスクステップ知識 (step)」について行われるが、この質問順序を設定できる。

しかし、この DSPL-ACQUIRER には、知識の連想支援や洗練支援と言っ

た機能はない。汎化タスクのライブラリーを作って、知識獲得支援を行うシステムとして MULTIS [Tijerino 89] が研究されている。

この他に汎化タスクではないが、特定の問題のタスクや特定の問題解決の知識獲得支援を行うものには、次の2つがある。知識獲得支援システム ROGET [Bennett 85] は、医療診断システムなどの知識ベースを分析して得られた診断問題における問題解決の概念構造 (conceptual structure) に基づいて、知識獲得を支援する。

知識獲得支援システム OPAL [Musen 86] は、癌治療計画エキスパートシステム ONCOCIN [Suwa 82] の治療プラン知識獲得を支援する。しかしながら、このシステムは、特定の問題解決モデルにのみ対応しているため汎用的な知識獲得の機能はない。知識獲得支援システム SALT [Marcus 87] は propose-and-revise の問題解決モデルのための対象知識と後戻り知識を獲得する。

ヨーロッパの人工知能研究プロジェクト ESPRIT では知識獲得の方法論として KADS [Breuker 89] がある。この方法論では知識獲得を問題解決モデルの構築 (building conceptual model) と処理系の実現 (implementation) に分けて捉えて、そして、問題解決モデルの構築を支援する方法を提供している。操作水準の概念 (concept) を準備して、知識処理技術者が予めその操作水準の概念で問題解決の枠組を作る。この枠組に従って知識獲得を行う方法である。ただし、KADS の操作水準の概念はタスクより少しだけ低い水準の知識表現であるため、非常に多くの種類を用意しているが、十分な量に達しているかは不明である。また、方法論を提供しているが、その計算機上での処理系は実現されていない。

### 汎化タスクの問題点

汎化タスクは非常に高い水準の知識表現を提供するが、個々のタスク表現は、特殊なもので汎用性は低い。そのため、色々な問題解決の知識獲得を行うためには、数多くの汎化タスクを準備する必要がある。しかしながら、現在、発見されているタスクの数は十分とは言えない。エキスパートシステムの構築においては、タスクの分析と汎化タスクの発見は重要な課題の1つである。

#### 2.6.5 知識獲得支援と学習システム

知識獲得支援システムとして、専門家と対話しながら知識を獲得するシステム(インタビューによる知識獲得支援システム)について述べたが、対話による知識抽出が困難な場合には、専門家の作業を観測 [Taki 85,88a] したり、別の知識から必要な知識を生成する技術が必要となる。前者には例題から知識を生成する学習システムが対応し、後者には、いわゆる知識コンパイル [Araya 87] が対応する。定性モデルを利用して、物理的な深い知識から専門家が経験により獲得する知識に相当する浅い知識を生成する知識コンパイルシステム [Yamaguchi 87] がある。知識ベースの学習では、説明に基づく学習 (EBL:Explanation-Based Learning) [Mitchell 85, 86]、大規模な先行知識 (百科辞典) を利用し、類推で学習をめざす CYC [Lenat 86]、決定木の学習を行う ID3 [Quinlan 86] や知識のチャンキング (Chunking) を行う SOAR [Laird 84] などがある。このなかで特に説明に基づく学習 (explanation based learning) が問題解決の効率化のための学習方法としては有効である [Kobayashi 88]。

## 2.7 結言

知識獲得支援の基本技術について説明した。また、知識獲得支援システムの歴史をその扱う知識表現の観点から特徴と問題点を説明した。知識獲得支援システムに求められている主要な技術は、次の3点である。

1. 問題解決モデルの構築できる知識表現であり、知識の抽出と洗練に役立つ情報が提供できる知識表現
2. その知識表現を利用して、問題解決に必要な知識を極力無駄な質問をしないで抽出できる抽出方法
3. その知識表現を利用した知識の洗練であり、推論の実行以前にも有効な洗練機能を持つ洗練方法

## 第 3 章

### 知識獲得の為の知識表現：専門家モデル

#### 3.1 緒言

対話型知識獲得に必要な技術には、「専門家からの知識の抽出」、「知識の整理・体系化」、「知識の不備(不足・矛盾・冗長)の検出と解消」がある。これらすべてに共通の技術要素として、知識獲得のための知識表現がある。現在、知識システムの知識ベース構築のために、エキスパートシステム構築ツール(OPS5[Brownston 85]など)で用意されている知識表現は、プロダクションルールやフレームと言った非常に一般的な表現である。これらの知識表現は、一般的であるがゆえに広い範囲の知識を表現できるが、表したい知識とその知識表現の記述が1対1に定まらない。そのため、専門家は知識の記述技術なくしては、自分の知識をうまく表現できないのが現状である。この問題に注目して、問題解決の観点から知識表現を見直したのが、汎化タスク(generic tasks)である。しかしながら、汎化タスクは、かなりまとまった問題解決のタスク構造を表現しているために、現在、確認されている7つの汎化タスクでは、表現しきれない問題解決もある。そこで、知識獲得には、プロダクションルールよりも特殊であり、汎化タスクよりも詳細に問題解決を表現できる知識表現が必要と考えられる。また、知識獲得の知識表現は、知識の抽出や不足・矛盾・冗

長の検出に向けたものである必要がある。この知識獲得のための知識表現として、問題解決過程を操作単位で表す知識表現を提案する。専門家の問題解決過程を表現することから、この表現を「専門家モデル」[Taki 87a, 87b, 90a]と呼ぶことにする。

## 3.2 専門家モデル

### 3.2.1 専門家モデルの基本的アイデア

専門家モデルは、専門家の問題解決の過程に注目した知識獲得のための知識表現である。我々は、この専門家モデルの表現を決定するにあたり、単純タスクモデルと実際の知識ベースの分析を行った。

#### (1) 単純タスクモデル

問題解決過程を表現し易いモデルを探すために、基本的な問題解決過程の幾つかを単純なモデルで表すことを試みた。ここでは、診断問題と組合せ問題の単純なモデルの検討結果を示す。

#### [診断問題]

診断問題は、予め与えられた「故障の候補」を絞り込んでいく過程としてモデル化できる。それを図示すると、次の図 3.1 のようになる。図 3.1(a) は、候補の絞り込みの一操作(フィルタリング操作、filtering operation)を表している。図 3.1(b) は、段階的に故障の候補を絞り込んで行く過程のモデルである。故障の候補は、絞り込みのための操作を通過するごとに、その数を減らし、最後に残った結果(result)が故障の箇所の候補となる。

#### [組合せ問題]

組合せ問題は、複数の要素を組合せて、ある条件を満足するものを選

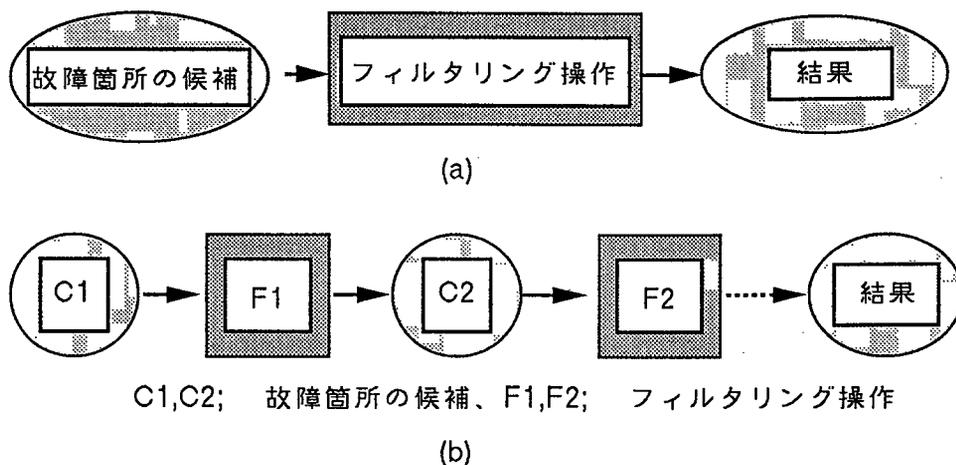


図 3.1: 単純タスク診断モデル

択する問題である。図 3.2 は、ある物質と体積の候補から、目的とする質量を持つ候補の組合せを求める問題である。ここでは、要素集合としては、体積の集合 (a set of volume) =  $\{5\text{cm}^3, 6\text{cm}^3, 7\text{cm}^3, \dots\}$  と物質の集合 (a set of material) = {物質 A, 物質 B, ...} を考える。まず、この 2 つの要素の組合せを作り、次に、物質の密度と体積から質量を計算する (mass calculation)、そして、その質量を検査して条件 (たとえば、 $50\text{g} < \text{質量} < 100\text{g}$ ) を満足したものを選択 (selection) する。本節では、問題解決の過程を単純にする表現方法として、要素とそれに対する操作 (オペレーション) による表現を示した。

## (2) 操作の基本型

本来、効率的な知識ベースの構築には、その対象とする問題解決を表現し易い知識表現が望ましい。しかしながら、現在、エキスパートシステム構築ツールが、知識ベースの構築に準備している知識表現は、プロダクションルールなどの非常に一般的な知識表現である。このような事実にもかかわらず、知識処理技術者 (knowledge engineer) は、プロダクション

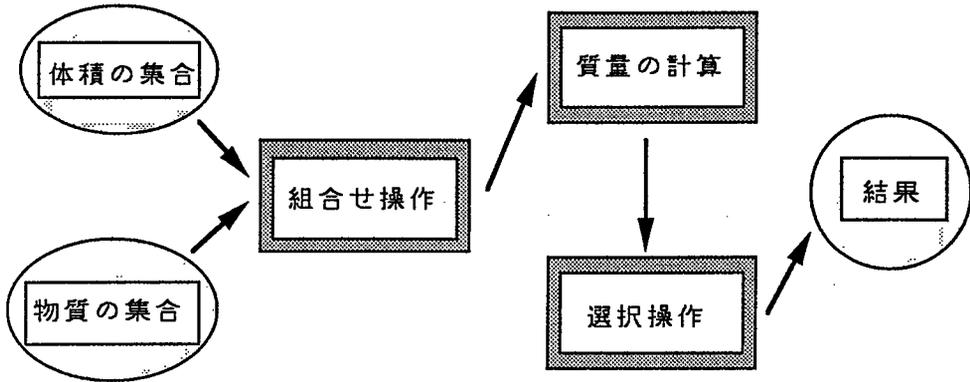


図 3.2: 単純タスク組合せモデルの一例

ルールによる知識ベース構築を行っている。我々は、この点に注目して、以下の仮説をたてた。

(仮説)

実知識ベースには、知識処理技術者の知識の表現技術が含まれている。

この仮説に従って、診断型知識ベース ([Wada 88a, 88b] など) の分析を行いプロダクションルールの機能の型分類を行った。抽出された基本操作の型は、「選択」、「分類」、「順序付け」、「組合せ」、「変換」、「入力」と「出力」の7つである。この基本操作と単純タスクモデルの考えを融合したのが、専門家モデルである。なお、プロダクションルールの機能とそれが扱うデータの型の組合せからも同様の操作の型が得られることも分かっている [Taki 89b]。また、データベース操作の関係代数の基本操作との対応をとることもできる (第7章参照)。また、いくつかの診断問題をこの専門家モデルで表現した結果、変換操作は、「数値演算操作」「値入換え操作」「要素分解操作」に細分類できることが分かった。

知識表現の水準をより、一般的にすると汎用性は増すが、特定の機能を持たせるには、その知識表現での記述技術 (知識) が必要になる。また、

逆に、特定の問題解決に特殊化した知識表現では、その問題の表現は容易であるがその他の問題の知識が表せない。つまり、知識獲得にある程度広い問題に対応でき、また、後述の知識抽出や知識の洗練に向けた水準の知識表現が必要となる。そこで、操作の型を抽出する際には、次の方針でその型分類を行った。

- (1) 操作の型の表現水準についての分析方針：1つのプロダクションルールで実現されている機能は、極力、それ以上分解しない。これは、あまり細かく機能を分割すると問題解決過程の単位が小さくなり、逆に知識抽出が難しくなるためである。
- (2) 操作の型の種類の制限についての分析方針：診断知識を表現しているプロダクションルールの構文的な記述を重視する。つまり、プロダクションルールは、汎用的な知識表現であるから、どのような操作の型も実現できるため、操作の型は実際の知識ベースに実現されているものに限定する。

以上の方針から、選択操作の型の他に、分類操作が準備されており、交換操作の一部に選択機能が含まれる型分類になっている。診断型問題以外の分野（具体的には、設計問題）での操作の型の評価と拡張については、第4章 4.7 節（システムの評価と拡張について）で議論する。また、数学的観点での操作の型についての考察は、第7章で論じる。

### 3.2.2 専門家モデルの基本構造

上記の基本的アイデアを具現化した専門家モデルについて説明する。専門家モデルでは、問題解決の過程を操作の集まりにより表現する。その操作は、7つの基本操作のいずれかで表される。各操作で処理されるも

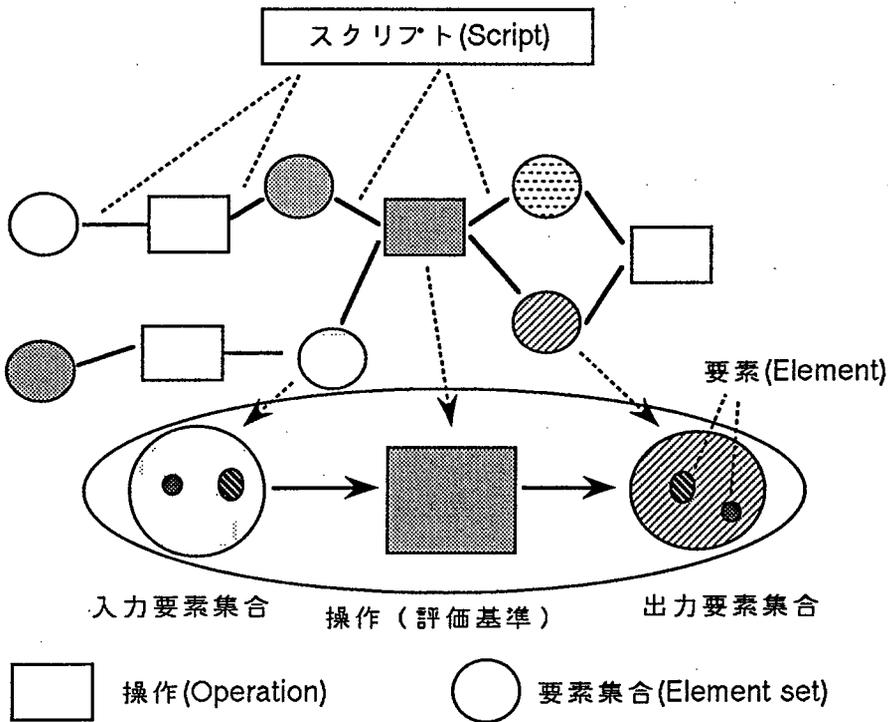


図 3.3: 専門家モデルの構成

のを処理対象要素 (以下、要素) と呼ぶことにする。また、操作の実行を制御するメタ知識をスクリプト (問題解決のシナリオ) と呼ぶことにする。図 3.3 に専門家モデルの基本構成 (EXPERT MODEL basic structure) を示す。この図では、専門家モデルは、操作 (operation) 群とスクリプト (script: 操作の実行優先順序、要素の流れと操作の起動条件) で表される。1つの操作は、型 (type) を持つとともに入力要素集合 (input element set: 処理対象要素の集合)、出力要素集合 (output element set: 処理結果の要素の集合) と評価基準 (evaluation criteria: 操作の基準を表す) で構成されている。評価基準は、7つの基本操作のいずれかを論理演算・数値演算で実現する。要素集合の各要素は、属性と属性値の対の集合で表される。

### 3.2.3 基本操作 (generic operation)

知識ベースの分析から求められた個々の基本操作について説明する。各操作が扱う要素集合、評価基準の内容、及び、プロダクションルールでの表現について言及する。

#### 1. 選択操作 (selection operation)

この操作は、入力要素集合から選択基準を満足する要素を選択して出力要素集合を生成する。プロダクションルールでは、作業記憶 (working memory) 内の条件を満足した要素に指標 (flag) を設定する (たとえば、ある属性の値を "1" にする) などの方法で実現している。

選択操作は、基本的には入力要素集合から出力要素集合を選択する集合の演算である。選択基準は、「単一または、複数の評価属性とその属性のとり得る値の集合 (複数の条件間は、AND 条件である)」または、「2つの評価属性とその値の関係」で表される。つまり、選択対象の入力要素集合を  $X$ 、選択基準を表す要素の集合  $A$  が定義できるなら、出力要素集合は、 $X$  と  $A$  の積集合  $X \cap A$  で表される。選択基準は、単一の属性に注目した場合には、文字列から成る取り得る値の集合は、外延的に表す。

例 (1): 「属性: 大きさ」に対する選択基準の文字列の属性値は、  
{非常に小さい、少し小さい、小さい} で表す。

また、数値は、文字列のように外延的表現だけでなく、次のように、内包的表現も可能とする。

例 (2): 「属性: 大きさ」に対する選択基準の数値の属性値は、  
{ $x \mid x \leq 100$  かつ  $20 < x$ } のように表す。



図 3.4: 選択操作の例

(ただし、一つの内包的表現の集合内では数値の区間条件は、AND 条件とする。)

2つの評価属性とその値の関係に注目した選択基準では、2つの属性とその値間の比較(=、≠、>、≥、<、≤)で表す。

例(3): 「属性：縦の長さ」より「属性：横の長さ」の方が長い要素を選択するには、

(属性：縦の長さ) < (属性：横の長さ) で評価基準を表す。

複数の属性に注目した基準では、「属性とその取り得る値の集合」または、「2つの評価属性とその値の関係」を複数用意して、それらを AND 結合したものが選択基準となる。

例(4): 「属性：大きさ」と「属性：重さ」の2つの属性を考慮する基準は、

大きさ: {非常に小さい, 小さい} ∧ 重さ: {軽い} で表現し、両方の属性がその取り得る値に含まれる要素を選択する。

例(5): 取り得る値の集合の各要素には否定の表現を付加できる。

{ (≠ 非常に小さい) }

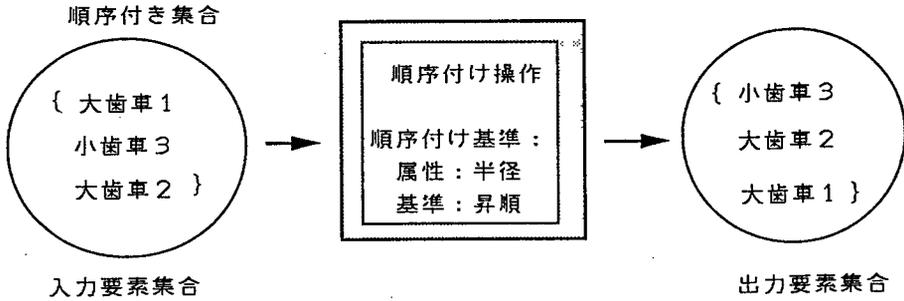


図 3.5: 順序付け操作の例

図 3.4は選択操作の例を示す図である。入力要素集合 {大歯車 1、大歯車 2、小歯車 3} が選択基準 {歯数が 50 以上} により出力要素集合 {大歯車 1、大歯車 2} が選択される操作である。

## 2. 分類操作 (classification operation)

この操作は、分類基準に従って、要素を複数の要素集合に分ける操作を行う。分類先の出力要素集合ごとに決められた選択基準に従って、属性の値が判定されることで、要素の分類を行う。この分類作業には、1つの要素を複数の要素集合に属させる操作も含む。プロダクションルールでは、ほとんどの場合、分類先ごとにルールを用意する。各ルールは、選択を行うルールであり、分類の識別値を作業記憶内の要素が持つ特定の属性に書き込む方法で実現されている。分類操作も複数の選択操作で実現できる。

## 3. 順序付け操作 (ordering operation)

この操作は、要素集合内の要素の集合内での順序を決める。要素の順序は、操作の処理順序や解の可能性の順序を表すことに利用される。プロダクションルールでは、要素に評価順序を決める値や確信度を持たせるなどの方法で要素の順序を表現している。順序付け基

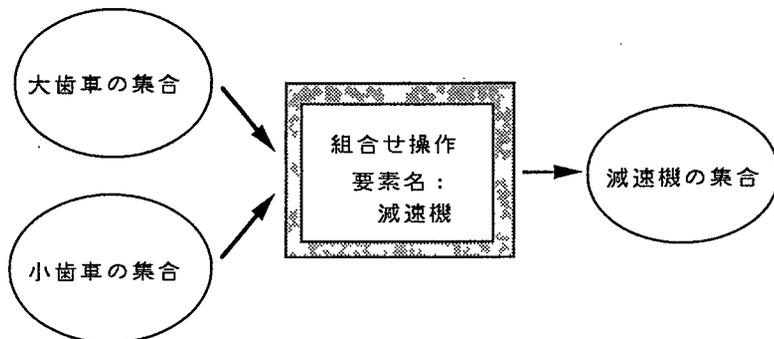


図 3.6: 組合せ操作の例

準は、評価属性と「数値の昇順・降順」と「文字列の順列」で表される。

例 (6): 順序付け 基準 1: (属性: 大きさ, 基準: 昇順)

基準 2: (属性: 大きさ, 基準: [小さい、普通、大きい])

図 3.5は順序付けの操作の例で歯車が半径の大きさに並べられる操作である。

#### 4. 組合せ操作 (combination operation)

この操作は、2つ入力要素集合の要素を各1つずつ選び、それらを組合せて、新しい要素を生成する。新しい要素は、その元となった要素の属性を持つ(継承する)。集合演算の直積に相当する。ただし、同じ名前の属性名があるとどちらか一方の入力要素集合を優先することとして、その集合に含まれる要素の属性値を採用する。

プロダクションルールでは、複数の要素の属性とその値をコピーした新しい名前を持つ要素を作業記憶に登録することによりこの機能

を実現している。この操作は、評価基準を持たない。

図 3.6は 2つの歯車の組合せから減速機構を生成する操作である。

## 5. 変換操作 (translation operation)

この操作は、次の 3つの型に分けられる。

### (a) 値入換え操作

この操作は、評価属性の値を別の値で入換える操作である。評価基準は、変換規則の形式であり、属性値判定部と変換後の属性と属性値 (固定値である文字列または、数値) で表す。属性値判定部は、基本的には選択操作と同じである。また、1つの値入換え操作には、複数の変換規則を OR 関係で表現できる。固定値でない値を入換える場合は、選択/分類操作と次の数値演算操作の組合せで実現できる。

例 (7):

属性: 大きさの値が 10 以上は、"大きい" に入換える。

属性: 大きさの値が 10 未満は、"小さい" に入換える。

プロダクションルールでは、要素のある属性値の判定後、その値に対応する値に属性値を修正 (modify) することで実現している。

### (b) 数値演算操作

この操作は、複数または、1つの属性が持つ数値を数値計算することで、属性値を決定する操作である。たとえば、(属性: 質量)の値を (属性: 体積)\*(属性: 物質の密度)の計算で求める操

作が、一例として挙げられる。参照される属性は、同一の要素にすべて含まれるものとする。要素間の演算は、組合せ操作と併用して行なうことが出来る。評価基準は、計算式、代入先属性名と参照属性名（計算式の変数として扱われる）で表される。計算式が演算を伴わない場合には、単なる代入操作になる。代入操作だけであれば、参照される属性値が数値でなく、文字列であってもよい。プロダクションルールでは、要素の属性値を参照した後、その値を元に数値計算 (compute) することで、代入する値を求めて、要素の属性値の修正または、追加によりこの機能を実現している。

#### (c) 要素分解操作

この操作は、要素をより細かい複数の要素に分解する操作である。たとえば、電子装置の診断において、初めは、どの回路が故障しているかを診断し、回路が特定できると、その回路に含まれる部品の診断を行う場合を考える。このとき、初めは、回路が、要素であり、次に、回路に含まれる部品が、要素となる。評価基準は持たないが、その要素に要素の構成を示す属性 (consist-of) が必要である。プロダクションルールでは、作業記憶に新しい要素を追加するか、予め作業記憶に用意してあった要素のある属性にフラグをたてることで、その要素を利用できるようにする (活性化させる) 方法でこの機能を実現している。この操作の例を図 3.7 に示す。

#### 6. 入力操作 (input operation)

この操作は、要素のいくつかの属性に、エキスパートシステムの外部から対話などの方法により、問題解決中に値を与える操作である。

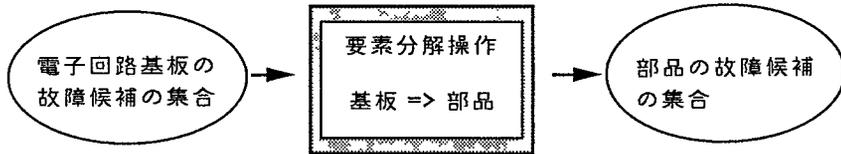


図 3.7: 要素分解操作の例

たとえば、回路の診断を例にすれば、ある抵抗の電流値や発熱の度合いなどの測定結果を与える操作である。プロダクションルールでは、メッセージを出力した後、入力 (read 関数など) 操作を行うことで実現している。評価基準は、値を代入すべき単一または複数の属性である。

#### 7. 出力操作 (output operation)

この操作は、問題解決過程で、メッセージや処理結果を出力する操作である。問題解決の状態や結果の表示に利用されることが多い。見方を変えると、要素の複数の属性から特定の属性とその値を抽出する操作と言える。評価基準は、値を出力すべき単一または複数の属性である。プロダクションルールでは、メッセージや属性値を出力 (write 関数など) 操作で実現している。

### 3.2.4 専門家モデルの処理対象要素

専門家モデルの操作は入力要素集合に対して行われ、その操作処理の結果は、出力要素集合となる。この入力・出力の要素集合を構成する要素が操作の処理対象である。1つの要素 (処理対象要素) は、属性名と属性値の対を要素とする集合である。また、入力・出力要素集合内では、要素間に順序がある。この順序は、操作における処理順序を表す。知識獲得時には、必要に応じて、要素集合に名前を付けることができる。例 (8) の

歯車カタログが要素集合名である。

- 例(8): 要素集合：歯車カタログ {大歯車1、大歯車2、小歯車3}  
要素：大歯車1 { (属性：歯数、120)、 (半径：20) }  
要素：大歯車2 { (属性：歯数、110)、 (半径：15) }  
要素：小歯車3 { (属性：歯数、30)、 (半径：5) }

### 3.2.5 知識専門家モデルの推論制御知識

「操作と要素集合の関係」と「操作と別の操作の関係」を表す知識である。「操作と入力要素集合の関係」には、OR参照関係とAND参照関係がある。このOR参照関係は、複数の入力要素集合を和集合として扱う。入力要素集合にAND参照関係を持つ操作が処理を実行できる条件は、複数の入力要素集合の各々に少なくとも1つの要素が必要である。「操作と別の操作の関係」は、ある操作の前に実行すべき操作とその操作の後に実行すべき操作を表す。また、この関係をプリポスト関係と呼ぶことにする。ある操作の後に実行すべき操作が複数あるときは、その操作をすべて実行すべきときをAND実行関係、いずれかの操作が実行されるべきときをOR実行関係と呼ぶ。これらの知識は、計算機上で専門家モデルの表現で獲得された知識を用いて問題解決を行うときに利用される。

## 3.3 関連研究との比較

一般的な知識表現であるプロダクションルールでは、問題解決の過程を記述するための知識を必要とする。その為に、プロダクションルールを知識表現とする知識獲得支援システム TEIRESIAS や SEEK は、知識抽出には向かず、ルールの洗練のみを支援している。格子型知識表現やドメ

インモデルを利用した知識獲得支援システム ETS や MORE は問題解決モデルは固定して、そこで利用される対象知識の獲得を行うのみであり、問題解決知識の獲得は難しい。汎化タスクのようなタスクでの知識獲得では、数多くのタスクの準備が必要である。専門家モデルは、問題解決過程を表現できる知識表現であり、タスクよりも小さい単位の操作の集まりとして、問題解決の過程を表現できたため、事前のタスクの準備は必要ではない。また、問題解決過程で使用される対象知識の抽出にも操作の型の情報を利用できる。操作の型の情報を利用することで、プロダクションルールの記述技術に相当する知識は不要となる。

### 3.4 結言

問題解決の過程を表現できる知識表現：専門家モデルを提案した。専門家モデルは、問題解決の過程を操作の集まりにより表す知識表現である。また、診断型の知識ベースのプロダクションルールの分類から求めた操作の型について論じた。専門家モデルやその操作の型の知識獲得における有効性の実証は、それを利用した効率的な知識の抽出方法と知識の洗練方法を通して、以下の章において論じる。

次章（第4章）では、専門家モデルの枠組みと操作の型を有効に利用した知識抽出について論じる。また、知識獲得支援システムの処理系の実現と評価についても合わせて説明する。第5章では、操作の型を用いて、知識の候補を例題から帰納的一般化により生成する方法を論じる。さらに、第6章では、操作の型を利用した知識の洗練方法を示す。第7章では、専門家モデルの評価結果と数学的な観点から、操作の型の考察と再構成を行う。



## 第 4 章

### 専門家モデルを利用した知識抽出方法

#### 4.1 緒言

知識抽出は専門家の知識を抽出して、その専門知識を特定の知識表現で表す技術である。知識抽出に必要な技術は、抽出すべき知識に関する質問生成および、知識を連想させる技術である。知識獲得支援システムには、ほとんど知識を獲得していない段階では、非常に一般的な質問で知識を抽出し、知識の抽出が進んだ段階では、既に獲得した知識を利用してより詳細な質問を行える対話技術の開発が必要である。

#### 4.2 知識獲得過程における知識抽出

知識獲得支援システムが何を抽出すべきかを知っていれば、専門家に対するインタビュー(質問)を容易に行うことができる。しかし、知識獲得以前に、何を獲得するかを詳細に知ることは、極めて難しい。知識獲得支援システム ETS/AQUINAS では、問題解決を「ある特徴群を持つ項目の適合度による順序付け」として、それに必要な知識を獲得する。この問題解決実現に必要な知識は、項目の集合、特徴の集合と、項目と特徴間の関係度である。ETS では、この 3つの情報を格子型知識表現 (rating grid) で表し、連想手法に パーソナルコンストラクト心理学を用いて、項

目からそれらを区別するのに利用し易い特徴の対(ある特徴とその反対の特徴)を抽出する。つまり、このシステムでは、獲得すべき問題解決のモデルを仮定しているために、必要な知識を効率的に獲得できる。しかし、知識獲得以前に、詳細な問題解決構造を知ることは困難である。この対策としては、問題解決モデルに対する依存性の低い知識表現と知識の抽出方法が必要である。この知識の抽出方法に、専門家モデルの操作の型を利用した知識獲得方法「プリポスト法」[Taki 87a, 87b]を開発した。

本章では、専門家モデルの操作の型の情報を利用して、その型の実現に必要な知識の抽出に関する質問を生成することを特徴とした知識抽出方法(プリポスト法)について説明する。まず、対話のための質問生成の分類について議論し、次に、専門家モデルを利用した知識抽出手順を説明する。さらに、専門家モデルとプリポスト法を計算機上に実現した知識獲得支援システム EPSILON/One とその評価について説明する。

### 4.3 対話による知識抽出方法の分類

対話型知識獲得においては、専門家から効率的に知識を抽出できる質問の生成が重要な技術である。現状の知識獲得に用いられている質問生成方法(インタビュー方法)は、以下のように分類できる。

#### 1. 定型的質問の生成

状況や抽出できた知識に無関係に、決まりきった質問を行って、情報を得る方法である。効率的ではないが、事前情報がまったくない場合に、わずかでも情報を得たいときに行う。他の質問生成方法では、この方法で得た情報を知識抽出の手掛りに利用できる。

## 2. 連想を促す質問の生成

心理学的な方法により、専門家に知識の連想を促す質問の生成方法である。パーソナルコンストラクト心理学が、一例として挙げられる。

## 3. 知識表現の整理や変換による方法

獲得した断片的知識を整理・体系化することや抽出した知識表現の変換により、専門家自身が自分の知識を再確認して、不足や矛盾などを自らが気がつくようにする方法である。

## 4. 知的な質問の生成（知的インタビュー）

知識獲得支援システムが、予め獲得すべき知識がどのようなものであるかを知っていて、その知識に従って、効率的な質問の生成を行う。MOREの洗練戦略や油圧回路の知的インタビュー [Kawaguchi 88] が、これに当たる。

プリポスト法は、「連想を促す質問の生成」、「知識表現の整理や変換による方法」と「知的な質問の生成」を併用した知識抽出方法である。知的な質問の生成は、専門家モデルの操作の型を利用して行われる。

# 4.4 知識抽出方法：プリポスト法

## 4.4.1 プリポスト法の概要

プリポスト法の知識獲得作業は、次に示す3つの段階に大別できる。

### 1. 問題解決過程のモデル化

専門家の問題解決過程を専門家が、日常利用しているタスク（作業）名で抽出し、その作業の流れで表す。その後、各タスクを、7つの基本操作で表される水準になるまで分解する。

## 2. 問題解決過程モデルの詳細化

問題解決を表現する操作の詳細な知識を抽出する。操作の型に対応した評価基準を抽出する質問を生成することで知識の抽出を行う。

## 3. 問題解決で処理される知識(要素)の抽出

各操作で処理される要素に関する知識を抽出する。既に抽出済みの操作の評価基準の属性とその取り得る値などを利用して、各要素の属性と属性値の抽出を行う。

### 4.4.2 プリポスト法の知識抽出手順

プリポスト法の主たる知識抽出戦略は、3つある。1つ目は、ある操作に関係する前後に実行する操作の連想を専門家に促す。これは、上記の「連想を促す質問の生成」の方法に属する。専門家にとって、ある操作の前後にどのような操作が、必要であるかを述べることは比較的易しい。たとえば、車が動かないときに、「エンジン検査の前に何を行うか?」の質問に対して、人間は、「ガソリンの残量を調べる」や「バッテリーを調べる」などを容易に答えることができる。この操作の抽出過程からこの方法をプリポストと命名した。2つ目は、抽出した操作とその関係をグラフ表示することにより、知識の整理した結果を専門家に示すことである。これは、「知識表現の整理や変換による方法」と言える。最後の方法は、操作の型を利用した質問生成である。これは、上述の「知的な質問の生成」に属する。以下、プリポスト法の知識抽出手順を示す。

#### 1. 知識獲得の出発点となる操作の収集

専門家作業の問題解決の一部の処理を操作として幾つか収集する。各操作間には関係がなくてもよく、操作は、無作為に抽出してよい。

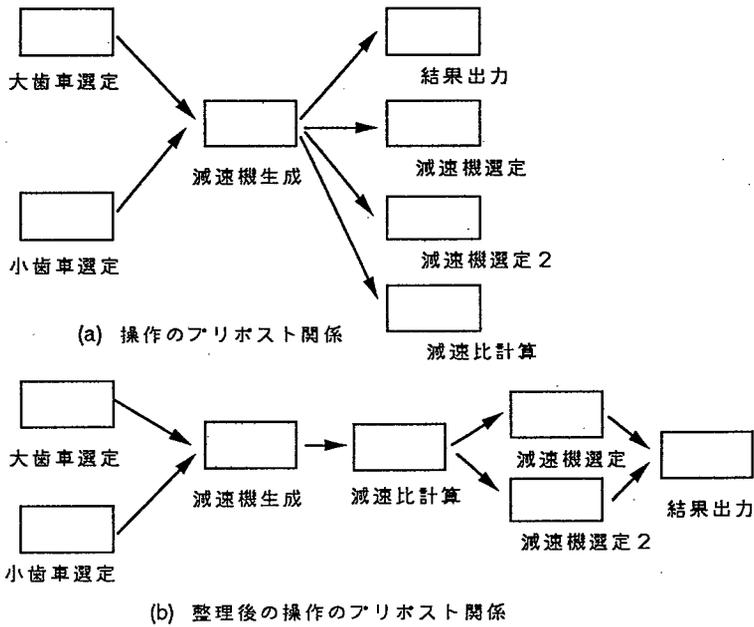


図 4.1: 操作のプリポスト関係

## 2. 前後の操作の抽出

抽出した操作についてその前後に行う操作を質問する。ある操作の後に実行する操作には、「成功した場合に実行される操作」と「失敗した場合に必要な操作」が抽出される。前後の操作は、直前・直後の操作でなくてもよい。この抽出作業は、専門家から新しい操作が出なくなるまで続けられる。さらに、新しく抽出した操作についてもその前後の操作の抽出作業を行う。操作の処理順序に関する情報をプリポスト関係と呼ぶことにする。

## 3. プリポスト関係の検証・評価

プリポスト関係の不備を調べるために、この関係をグラフで表示して、専門家が検証する方法がある。図 4.1は、プリポスト関係の情報が追加されるにつれて、初期のプリポスト関係 (a) が、明確化された関係 (b) になる例を示している。

#### 4. 操作の型の決定

専門家は、操作ごとに、基本操作の7つの型のいずれに対応するかを決定する。抽出されている操作が、大きなタスクである場合には、いくつかのサブタスクを含んでいることがある。このタスクをサブタスクに分解するには、そのサブタスクの幾つかを専門家に述べてもらう。そして、そのサブタスクを新たな操作にして、「前後操作の抽出」を行う。

#### 5. 評価基準の抽出

操作の型に従って、次の様な知識を抽出する。

- 選択: 選択基準を抽出する。
  - a. 評価属性とその評価属性の取り得る値
  - b. 2つの評価属性とその属性間の値の関係
  
- 分類: 分類先と各分類先の分類基準を抽出する。  
この基準は、選択操作の基準と同じである。
  
- 順序付け: 順序付け基準を抽出する。
  - a. 評価属性とその取り得る値の優先順を示す順列
  - b. 評価属性と数値の昇順・降順
  
- 組合せ: 新しく作る要素名を抽出する。  
評価基準は持たないが、生成する要素名を抽出する。生成された要素の名前は、要素名に整数を付加した文字列となる。
  
- 変換 (値の入換え): 変換規則を抽出する。  
変換条件 (選択基準と同じ)、変換対象の属性と代入する値

- 変換（数値演算）：計算式を抽出する。  
計算に使用される属性（変数に相当）と代入先の属性および、  
計算式
- 変換（要素分解）：分解される要素と生成される複数の要素の  
対応関係を抽出する。  
実際には、この情報は要素に付加される。つまり、この操作の  
処理系の実現においては評価基準を持たないことになる。また、  
要素の階層知識が、たとえば、フレーム表現のクラス階層で獲  
得されていればその情報を利用することができる。

#### 6. 要素集合と要素の抽出

この段階では、各操作と入力要素集合の対応関係を抽出する。入力要素集合は、問題解決を行う以前に集合として獲得できるものと問題解決中に生成される集合つまり、他の操作の出力要素集合の2種類に分けられる。後者の場合は、要素が操作をどのように流れるか（要素のデータフロー）を表すことになる。知識の抽出は、各操作に対して「どのような要素を処理するか」または、「どの操作の結果を処理するか」を質問することで行われる。前者の入力要素集合が抽出されると、その集合に属する個々の要素の抽出を行う。この要素の集合は、たとえば故障診断の問題であれば、最初に用意する故障可能な箇所（故障仮説）の集合であり、部品の組合せ問題であれば、部品の集合に相当する。

#### 7. 要素の属性と属性値の抽出

入力要素集合の各要素について、属性と属性値を抽出する。要素の流れからどの操作で各要素が処理されるかの検討をつけることが可能である。各操作の評価属性と判定に利用される属性値は、すでに

抽出済みである。要素が処理される可能性のある操作から、評価属性と属性値を集めて一覧表を要素ごとに作り、そこから必要な属性と値を専門家が選択する方法を提供する。

#### 4.5 知識獲得支援システム EPSILON/One

専門家モデルとプリポスト法に基づき推論マシン PSI-II 上に ESP 言語 [Chikayama 84] で開発した対話型知識獲得支援システム EPSILON/One について、システムの構成、専門家モデルの処理系の実現とシステムの評価結果について論じる。EPSILON/One は、図 4.2 の様に、対話インタフェース (interview interface)、知識の抽出モジュール (knowledge elicitation module) と知識の変換・推論モジュール (knowledge translator and EXPERT MODEL inference module) で構成されている。専門家の知識

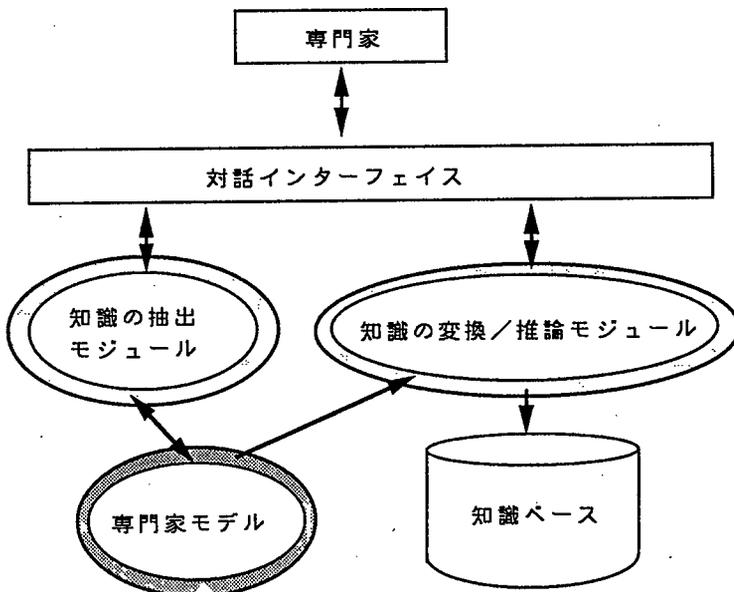


図 4.2: EPSILON/One の構成

は、マルチウインドウとマウスを使った知識獲得の対話を支援する対話インタフェースを通じて、知識の抽出モジュールにより、専門家モデルと

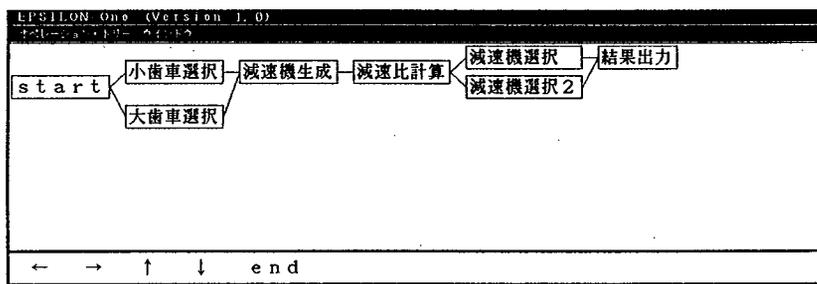


図 4.3: 操作の抽出画面の例

して獲得される。知識の変換・推論モジュールは、抽出された専門家モデルを評価可能形式の知識表現に変換する。また、このモジュールは専門家モデルで表された知識を使って問題解決を行う。

以下、知識の抽出モジュールと知識の変換・推論モジュールについて説明する。獲得された知識の不備（矛盾と不足）の検出と解消を行う知識の洗練の方式については、第6章で詳しく説明する。

#### 4.5.1 知識の抽出モジュール

知識の推論モジュールは、プリポスト法に従って、専門家の知識を専門家モデルの形式で抽出する。操作、プリポスト関係と要素の流れは図4.3操作の抽出 (operation elicitation) のように、画面 (ウインドウ) 上にグラフィック表示され、抽出内容の確認が容易にできる。

図 4.4評価基準の抽出 (evaluation criteria elicitation) のように、操作の型ごとに評価基準抽出ウインドウを用意して抽出の効率化を実現している。

また、要素の要素属性と属性値は、図 4.5要素属性の抽出 (element attribute selection) のように、ウインドウ上に示される知識の候補の表からマウスで選択することで抽出する。

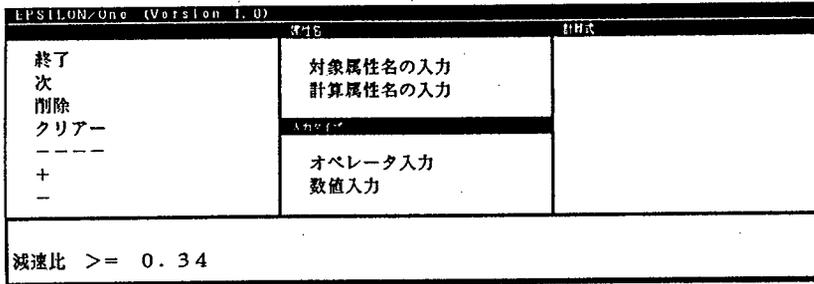


図 4.4: 評価基準の抽出画面の例

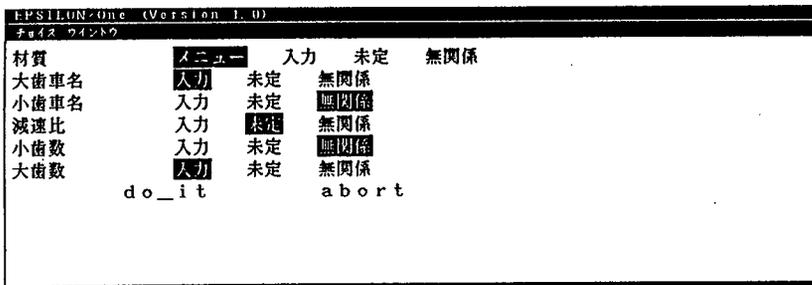


図 4.5: 要素属性の抽出画面の例

#### 4.5.2 知識の変換・推論モジュール

専門家モデルで表される知識を逐次推論マシン PSI-II 上で実行するために評価用の知識表現形式に変換する。変換の際に、専門家モデルの操作で表された知識やスクリプトの情報も合わせて変換する。これは、実行評価時に、専門家が獲得のときと同じ知識表現(専門家モデル)で知識の検証と洗練を行い易くするためである。問題解決は、操作単位で動作確認ができる。図 4.6 は、実行終了(図中反転表示)および、実行中の操作(実行中は点滅表示)を示している。また、実行を終了した操作の処理結果を参照できる。推論の実行は、指示した操作を起動(実行要求)することで開始される。

要素の流れとプリポスト関係を利用して、次々に、操作が起動され問

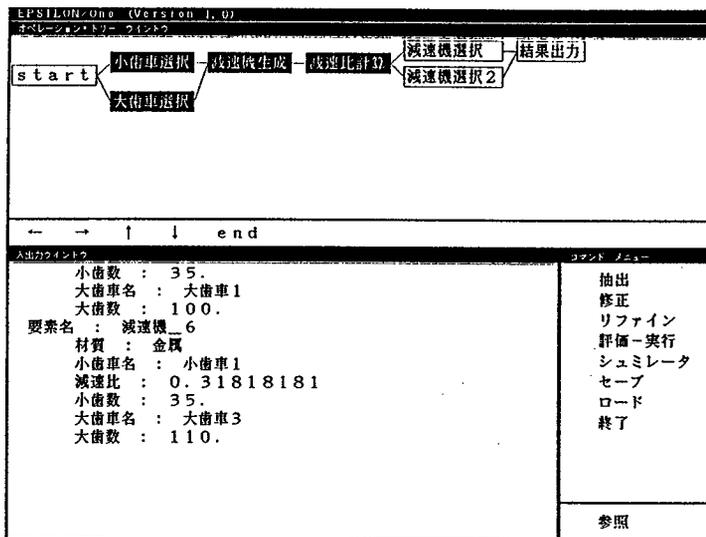


図 4.6: 実行評価画面の例

題解決が行われる。基本的な操作起動条件を次に示す。

- (1) 入力要素集合に要素が準備された場合 (データ駆動方式)
- (2) 出力要素集合生成の要求がきた場合 (要求駆動方式)

起動要求が来たにもかかわらず入力要素集合が空集合のときは、要素集合の生成要求を他の操作に出す。また、要素が複数の操作に流れる (処理される) 場合、つまり、操作が OR 関係の場合には、一つの操作が選択されて処理が進むが、その処理の流れが中断すると (たとえば、出力要素が一つも生成されない場合には)、別の操作が起動される。これは、Prolog 言語などのバックトラックキング (backtracking) 機能に相当する。

ルール表現の知識ベースへの変換機能では、専門家モデルの形式で獲得した知識をエキスパートシステムで利用できる知識表現に変換する。専門家モデルの基本操作は、プロダクションルールの記述から抽出したもの

であるから、プロダクションルールへの変換は、容易である。

## 4.6 関連研究との比較

知識抽出を支援する機能を持つ知識獲得支援システムに ETS がある。ETS は、心理学的な知識抽出と知識表現の整理と変換による知識抽出法を備えている。心理学的方法は、上述のパーソナルコンストラクト心理学であるが、知識表現の整理方法では、類似分類木と導出グラフを用意している。しかしながら、ETS で抽出できる知識は分類対象の知識のみであり、問題解決の知識は抽出できない。問題解決過程の抽出できる知識獲得支援システムに OPAL があるが、これは、癌治療計画システム ONCOCIN の診断過程を診断の小さなタスクの組合せで抽出する。しかし、このシステムでは、ONCOCIN の問題解決の枠組以外の問題には対応できない。

プリポスト法では、「操作の前後関係の連想方法」、「操作の型を利用した知的な質問生成」、「複数の操作の関係の整理結果のグラフ表示」と「獲得された操作の評価基準の情報で対象処理要素の属性抽出の支援」による知識抽出を行っている。このように、プリポスト法では複数の知識抽出方法を組合すことと、操作の型の情報を有効に利用することで効率的な知識獲得を実現している。EPSILON/One は、操作の型で表現できる解析型問題と一部の合成問題（組合せ問題）の問題解決知識の抽出に対応できる。合成型問題に対応できる操作の型については、第7章で議論する。

## 4.7 システムの評価と拡張について

本システムの専門家モデルにおける操作の型は、診断型の知識ベースの分析に基づいているため、知識獲得対象は解析型の問題である。解析型問題での機能の確認と、合成型問題への拡張上を目的として、2つの例題によるシステム評価を行った。解析型問題として、「計算機 (PSI)

ネットワークの整備診断問題」 [Tsubaki 89b]、合成型問題として、「工作機械の主軸設計問題」 [Inoue 88] を例題として知識獲得の評価を行った。例題における診断知識の良構造部分 (well-structured, well-defined) は本システムで獲得できたが、悪構造部分 (ill-structured, ill-defined) は操作の評価基準を専門家が十分に意識している場合のみ抽出でき、専門家が意識していない場合には、操作の具体的な処理例以外獲得できなかった。このような場合の評価基準の抽出方法として、少数の入力と出力の要素の例から操作の評価基準を推定して、その推定結果を専門家に提案することが考えられる。この提案による知識抽出は、次章 (第5章) で論じる。設計知識の獲得では、操作の型の不足や状況により操作の評価基準が変わる知識 (制約知識) があり、合成型問題に向けた操作の型の追加や制約表現の知識に対応した専門家モデルの拡張が必要であることが分かった。

操作の型の拡張については、専門家モデルの操作の型と関係代数の比較および、合成型問題に必要な操作の型の考察を中心に第7章で議論する。

操作の型が状況により変わる知識を獲得するには、専門家モデルの拡張として、各操作の評価基準を可変とすることが考えられる。この場合には、評価基準を決めるメタ知識が必要である。この機能をこの専門家モデルの枠組で実現するには、操作の評価基準を選択できるようにする。

また、この専門家モデルを用いた推論では、まず、評価基準を要素とするメタ水準の専門家モデルの推論が行われ、各操作の評価基準が決定された後、通常専門家モデルを利用した問題解決を行う [Osaki 88]。

## 4.8 結言

専門家の問題解決の過程に注目した知識獲得の実現を目指して、問題解決のモデルを操作単位で表現する専門家モデルの抽出方法 (プリポスト法) を提案した。今後の課題としては、設計型問題に向けた知識獲得があ

る [Mizoguchi 89]。設計の知識の多くは、制約であらわされることが多い。設計問題においては、対話により獲得できる知識には、限界がある。なぜならば、設計者は、その設計作業において、「解の候補が、多過ぎる」や「解が得られそうにない」などの設計の状況を見ながら、設計の要求仕様や設計の知識などの制約を追加や緩和しながら設計作業を遂行している。つまり、知識の想起には、このような状況が必要となる。そして、設計者自身も、設計が成功するとその設計過程を学習し、失敗や知識の操作(制約の緩和など)についても学びながら問題解決を行っている。設計問題の知識獲得に対して、「設計向きの操作の型の追加」「説明に基づく学習などを応用した問題解決過程での知識獲得」についての研究が必要である [Taki 89d, 89e, 89g][Kobayashi 88b]。知識獲得支援システム EP-SILON/One の計算機上での実現課題は、未実装の知識の洗練機能である。操作の型に対応した効率的な洗練方法の利用が必要であるが、この機能については、第6章で論じる。

## 第 5 章

### 提案誘導型知識獲得支援システム

#### 5.1 緒言

知識抽出の作業を支援するシステムとしては、問題領域に関する一般的な知識を直接、対話により専門家から抽出する対話型知識獲得支援システムと、具体的な例から一般的な知識の生成を行う帰納的推論を利用した学習システムがある。

現在の知識獲得支援システムと帰納推論システムでは、伴に、長所と短所がある。帰納推論の基本機能は、例から一般的知識を生成することであるが、その短所は、帰納的に作られる概念の候補(仮説)が非常に沢山生成されることと、計算量が非常に多いことである。帰納推論の利用において、計算量と概念の候補を減らして推論の効率を上げるには、生成される概念の言語(形式)を制限することと、学習対象の領域固有の知識を利用する必要がある。対話的知識獲得は、可能なかぎり一般的な知識を専門家から直接抽出するが、その問題点は専門家が一般的な知識を意識していない場合には、非常に個別的な例や不十分な知識しか抽出できないことである。知識獲得支援システム EPSILON/One は、知識表現に専門家モデルを利用して対話型知識獲得を行うが、この知識抽出の際に、具体例から知識の候補を推定する機能の必要性が見い出された。[Tsubaki 89b]

本章では、知識獲得支援システム EPSILON/One での知識表現である専門家モデルの特徴を旨く利用した帰納推論をこの知識推定に利用する方法を提案する。専門家モデルの特徴は、限定された7つの操作で問題解決の知識を表現することであるが、この制限を帰納推論の生成概念の言語制限に利用する。

対話型知識獲得支援と帰納推論を融合することで、専門家から抽出した情報を元に、新たな知識を仮説として推定でき、単なる知識抽出だけでなく、知識を提案する機能を持つことになる。このような知識獲得を提案誘導型知識獲得支援と呼ぶことにする [Taki 89a, 90b, 90d]。本章では、帰納推論を用いた操作知識推定と提案誘導型知識獲得支援システムを中心に述べる。

## 5.2 専門家モデルと帰納推論による知識獲得

### 5.2.1 専門家モデルと知識獲得

知識獲得支援システム EPSILON/One の知識獲得のための知識表現は、専門家モデルである。第3章で詳述したように、専門家モデルは、操作の集まりで専門家の問題解決の知識を表現する。操作は、7つのタイプ、すなわち、「分類」、「選択」、「入力」、「出力」、「順序付け」、「組合せ」と「変換」に限定されている。

専門家モデルを利用した知識抽出は、プリポスト法により行われる。プリポスト法による知識獲得は、(1) 操作とスクリプトの抽出、(2) 操作の型を利用した操作の内容(評価基準)の抽出、(3) 要素の抽出の3つの段階に分かれる。この内、(2)の段階では、操作の型が獲得すべき知識の型を限定することを利用して、効率的な対話型知識獲得を行う。しかしながら、獲得すべき知識は一般的な知識であるため、専門家がその操作内で利

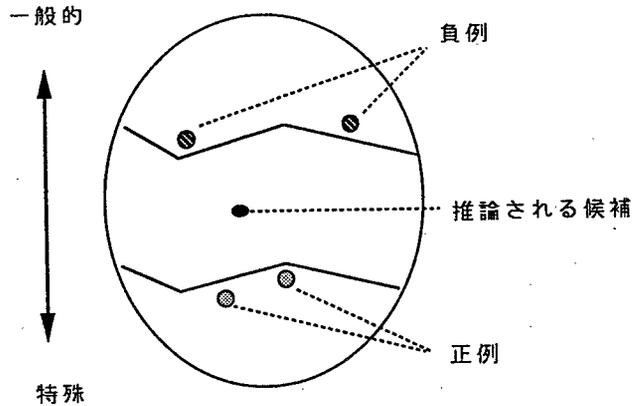


図 5.1: バージョン空間の構造

用する一般的な評価基準を意識していない場合には知識獲得は易しくない。一方、専門家にとって、その操作の具体的な処理例を示すことは、比較的易しい。このような場合における知識獲得支援には、その操作の具体的な入出力からその操作の一般的な内容を推定する機能が必要となる。

### 5.2.2 操作の型に対応した帰納的な知識推定アルゴリズム

知識獲得支援システム EPSILON/One の専門家モデルの操作の型を利用して、帰納推論の帰結を制限した知識（操作）の推定方法について述べる。帰納推論には、次に説明するバージョン空間法 [Mitchell 78] の枠組を使用している。

#### バージョン空間法

帰納推論は具体例から一般的な知識を生成する方法であるが、通常、推論の帰結として得られる知識の候補は多数存在する。この候補の空間を管理する方法がバージョン空間法である。図 5.1 の空間の上方が一般的な候補を表し、下方が特殊な候補を表す。この空間は求める概念の例（正例）と候補の概念を表す言語（形式）および、一般化の階層（バイアス、bias）

から作られる。また、正例で空間の下限が決まり、求める概念ではない例（負例）により空間の上限が決まる。

### 知識推定に必要な情報

例からの帰納的一般化、つまり、上述のバージョン空間を生成するために、「求める概念を表現する言語」、「概念についての正例・負例」、「概念の一般化空間を規定する知識（バイアス）」と「帰納アルゴリズム」が必要であるが [Genesereth 86]、操作推定で準備しているこれらの基本的情報を次に示す。

- 「求める概念を表現する言語」：操作の型毎に決められた操作実行の為の評価基準の表現。属性と属性値による評価基準を基本とする。
- 「概念についての正例・負例」：操作の入出力の要素の例。
- 「概念の一般化空間を規定する知識」：操作の型の評価基準毎の評価の一般化方法。
- 「帰納アルゴリズム」：操作の型ごとに決められた評価基準推定アルゴリズム。

このように、操作の型を活用して、言語の制限、例の扱い、概念の一般化空間を規定する知識の設定を行い帰納的一般化から得られる候補を制限し、評価基準の推定を効率の良いものとしている。

以下、操作型知識の推定方法として、「属性値の入換え」（変換）操作、「順序付け」操作、「選択」操作と「分類」操作について論じる。ただし、「入力」と「出力」の操作及び、「要素分解」（変換）操作では、専門家が注目している属性は比較的明確に抽出できるので、知識の推定は行っていない。「組合せ」操作は、2つの要素を組合せて1つの要素を作るアルゴ

リズムを固定してあり、評価基準は不要となっているため、知識の推定は行わない。また、「数値演算」(変換)操作は、数式を多項式とすると、入出力の例から作られる式を連立方程式として解く問題となるが、ここではそのアルゴリズムを論じない。

知識の候補の提案は、単純な評価基準から序々に複雑な評価基準を提示することとした。つまり、仮説の言語の制限を序々に緩和しながら候補を求める推論戦略をとっている。専門家は、単純な基準の方が判断し易く、連言型で表される基準では単純な方が、生成される候補の数が少なく帰納推論の計算量を低くできるためである。

概念の一般化空間は、属性の連言型の一般化階層に基づいて生成する方法をとっている。(例(9):  $a \wedge b$  よりも  $a$  の方が一般的である) 推定対象となっている操作毎に次の項目に従って説明する。

- 操作の実行アルゴリズム
- 操作の実行に必要な評価基準(獲得対象と推定対象)
- 例の扱い
- 評価基準の推定アルゴリズム

評価基準の推定アルゴリズムは、単一属性の評価基準を求めるものを示す。複数属性の評価基準は、属性の組合せと属性値の組合せを、それぞれ新たな属性と属性値として扱い同じアルゴリズムで推定する方法をとっている。

### 5.2.3 選択・分類操作の推定

選択操作は、与えられた要素の集合から選択基準を満足する要素を選ぶ。分類操作は、与えられた要素の集合を分類基準により、複数の集合に分割する。分類操作は、選択操作を複数個組合せたものと言える。

## (a) 選択アルゴリズム

ある要素が与えられたとき、その要素の属性と属性値が選択基準を満たすときに、その要素が選択される。選択基準は、評価属性とその属性値の取りえる集合（または、複数の属性の組についての可能な属性値の組の集合）で表される。選択基準を満たすとは、評価基準が規定している属性に関して、評価される要素のその属性値が、属性の値の取りえる集合に含まれることをいう。

## (b) 選択基準

単一属性から成る選択基準は、次のように限定する。評価属性とその取り得る値（外延的表現）の集合とする。

(属性名、属性値の集合)

また、属性値の集合要素には、否定の表現を含める。

## (c) 例の扱い

複数の要素を例として、操作の入力および、出力要素集合を与える。選択操作で、選択された要素は、選択基準の正例と言える。また、選択されなかった要素は、負例と言える。要素は、操作とは独立の知識として定義できる。また、要素は、属性と属性値の対の連言型で表わされるものとする。

例 (10): 入力要素集合と出力要素集合を次のように与えると

入力要素集合 =  $\{e_1, e_2, \dots, e_i, e_{i+1}, \dots, e_n\}$

出力要素集合 =  $\{e_1, e_2, \dots, e_i\}$

(ただし、 $e_j$  ( $j=1 \dots n$ ) は、要素を表す)

正例は、出力要素集合そのものであり、 $\{e_1, e_2, \dots, e_i\}$ となる。

また、負例は、入力要素集合から出力要素集合の要素を除いたもので、 $\{e_{i+1}, \dots, e_n\}$ となる。

#### (d) 選択基準の推定アルゴリズム

**Step1:** まず、正例、負例を問わず、例すべてに含まれる属性  $a$  を求める。

例 (11):  $\{a_1, a_2, \dots, a_m\}$  (ただし、 $a_j$  ( $j=1 \dots m$ ) は、属性を表す)

**Step2:** 正例・負例毎に、各要素に対して、上記の属性集合の各属性に対して値を求める、要素がある属性を持たない場合には、その属性値を undefined (未定義) とする。

**Step3:** 求める選択基準が単一属性であると仮定して、属性単位で正例・負例の可能な属性値の集合を求める。これは、属性間の従属性に関する一般化である。

例 (12): 要素を  $e_k$  ( $k=1 \dots n$ )、属性を  $a_j$  ( $j=1 \dots m$ )、属性値を  $v_{jk}$  ( $k=1 \dots n, j=1 \dots m$ ) とすると正例・負例毎に要素と属性に関するマトリックスができる。

<i>Positive</i>	<i>a1</i>	<i>a2</i>	<i>a3</i>	...	<i>am</i>
<i>e1</i> :	<i>v11</i>	<i>v21</i>	<i>v31</i>	...	<i>vm1</i>
<i>e2</i> :	<i>v12</i>	<i>v22</i>	<i>v32</i>	...	<i>vm2</i>
		⋮			
<i>ei</i> :	<i>v1i</i>	<i>v2i</i>	<i>v3i</i>	...	<i>vmi</i>
<i>Negative</i>					
<i>ei + 1</i> :	<i>v1i + 1</i>	<i>v2i + 1</i>	<i>v3i + 1</i>	...	<i>vmi + 1</i>
<i>ei + 2</i> :	<i>v1i + 2</i>	<i>v2i + 2</i>	<i>v3i + 2</i>	...	<i>vmi + 2</i>
		⋮			
<i>en</i> :	<i>v1n</i>	<i>v2n</i>	<i>v3n</i>	...	<i>vmn</i>

正例と負例について、属性ごとに属性値の集合を作る。属性 *aa* に関する正例・負例の属性値集合をそれぞれ  $P(aa)$ 、 $N(aa)$  とする。

$$\begin{aligned}
 P(a_1) &= \{v_{11}, v_{12}, \dots, v_{1i}\} \\
 P(a_2) &= \{v_{21}, v_{22}, \dots, v_{2i}\} \\
 &\quad \vdots \\
 P(a_m) &= \{v_{m1}, v_{m2}, \dots, v_{mi}\} \\
 N(a_1) &= \{v_{1i+1}, v_{1i+2}, \dots, v_{1n}\} \\
 N(a_2) &= \{v_{2i+1}, v_{2i+2}, \dots, v_{2n}\} \\
 &\quad \vdots \\
 N(a_m) &= \{v_{mi+1}, v_{mi+2}, \dots, v_{mn}\}
 \end{aligned}$$

**Step4:** 正例と負例で積集合を持つ属性は、単一属性では選択に利用できないので除く（次のルール参照）。

if  $P(a_j) \cap N(a_j) = \emptyset$  (空集合)

then 属性  $a_j$  を選択する

else 属性  $a_j$  は選択しない ( $j = 1 \dots m$ )

選ばれた属性の集合が、 $\{aa, ab, ac\}$ であれば、 $P(aa)$ 、 $P(ab)$  または、 $P(ac)$ が選択基準となる。

(注： $P(a_j) \cap N(a_j) = \emptyset$ であるから  $\neg N(aa) \cap \neg N(ab) \cap \neg N(ac)$ も選択基準である。ただし、 $\neg N(aa)$ は、 $N(aa)$ の補集合であり、属性  $aa$ が、属性値として、集合  $N(aa)$ の要素を取らないことである。)

**Step5:** 求められた選択基準を提示する。複数ある場合には、順に提示する。

上記の例では、 $P(aa)$ 、 $P(ab)$ 、 $P(ac)$ 、 $\neg N(aa) \cap \neg N(ab) \cap \neg N(ac)$ の順に提示する。

### (e) 選択基準の一般化の誘導

属性値の集合を一般化することで、選択幅を広げられることを示す。

例(13):	$\{1, 2, 3, 4, 5, 6, 7, 8\}$	$\rightarrow$	$\{>= 1, <= 8\}$
	$\{56, 78, 34, 22, 10\}$	$\rightarrow$	整数
	$\{大歯車, 小歯車\}$	$\rightarrow$	$\{歯車\}$

さらに、正例か負例か分からない場合があることを提示して、より一般化できることを専門家に示す。

例(14):  $P(aa) = \{1, 2, 3\}$   
 $N(aa) = \{5, 6, 7\}$

例(14)の場合、属性  $aa$ の属性値が、4のとき、正例か負例か不明である。選択基準が、 $P(aa)$ なら、この要素は、負例として扱われる。また、選択基準が、 $\neg N(aa)$ なら、この要素は、正例として扱

われる。つまり、 $\neg N(aa)$  を選ぶ方が選択幅は広がり、より一般的な選択基準となる。

$P(a_j) \cap N(a_j) = \emptyset$  となる属性  $a_j$  がない場合は、属性間の従属性を無視できなくなり、複数の属性の関係を考慮した判定基準が必要となる (5.2.6 節参照)。

## 5.2.4 順序付け操作の推定

### (a) 順序付けアルゴリズム

与えられたすべての要素について、ある属性に注目して、その属性値の全順序基準に従って、要素を並べ替える。この順序は、操作 (選択や出力操作など) での要素の評価順序を意味する。

### (b) 順序付け基準

単一属性から成る順序付け基準は、次のように限定する。評価属性とその値の優先順序で表す。

(属性名、属性値リスト) または、(属性名、昇順) または、

(属性名、降順)

属性値リストの文字列または、数値の順序が、上位から下位への順序を表すものとする。また、昇順と降順は、属性値が数値であるものに限る。

### (c) 例の扱い

複数の要素を例として、操作の入力要素集合および、出力要素集合を与える。順序付け操作では、出力要素集合の要素の順序が順序基準の正例と言える、また、入力要素集合での 2 要素の順序関係が出力要素集合で異なる順序となるものは、負例となる。

例 (15): 入力要素集合と出力要素集合を次のように与えると

入力要素集合 = {e1, e2, e3, e4}

出力要素集合 = {e1, e3, e4, e2}

a よりも b を優先する関係を  $b \gg a$  で表すとすると、

正例は、 $e1 \gg e3 \gg e4 \gg e2$  である。

また、負例は、 $e2 \gg e3$  と  $e2 \gg e4$  となる。

#### (d) 順序付け基準推定アルゴリズム

**Step1:** 入力要素集合と出力要素集合のそれぞれについて、要素間の優先順序に関する 2 項関係を求める。出力要素集合についての要素の 2 項関係は、すべて正例である。また、入力要素集合についての要素の 2 項関係で、出力要素集合での関係と矛盾するものは、負例である。順序の関係に矛盾があるとは、 $a \gg b$  か  $b \gg a$  の関係が同時に成立することをいう。

**Step2:** 属性値に undefined を含まない属性を 1 つを選び、出力要素集合の要素の対について、その属性値の順序を決める。これをその属性に関する順序基準と呼ぶ。推定対象となる属性がない場合には、推定を終了する。

**Step3:** 順序の関係に、矛盾がないかを調べる。矛盾があれば、属性を変更して、Step2 を実行する。

**Step4:** 属性値が数値のときは、昇順か、降順かを調べる。その他の関係(偶数を優先するなど)であれば、数値であっても、文字列と同じように扱う。

**Step5:** このような条件を満足する属性が、複数ある場合には、属性とその順序基準を順に提示する。

なお、順序付け操作での入力要素の負例は、正例と共通部分がない(順序に関して反対の関係にある)ため、正例から求められた評価基準を制限できないので、順序基準の推定には使用されない。

### 5.2.5 値入換え操作の推定

#### (a) 値入換えアルゴリズム

ある要素が与えられたとき、ある1つの属性に注目して、値入換え基準に従って、元の属性値に対応する別の値に入換える。入換え基準は、変換ルール形式で表され、元の値と変換後の値の対からなる集合である。

#### (b) 値入換え基準

単一属性から成る入換え基準(規則)は、次のように限定する。評価属性とその取り得る値および、変換後の値とする。条件が検査される属性と値の変更される属性は、同じとする。

(属性名、(元の属性値、変換後の属性値)の集合)

(注:これは、第3章の値入換え操作の基準を限定したものである。より複雑な変換条件を扱う場合は、選択基準の推定と同じ方法が利用できる)

**(c) 例の扱い**

複数の要素を例として、操作の入力および、出力要素集合を与える。属性値の入換え操作で、入力と出力で値の異なる要素は正例と言える、また、値の変わらない要素は負例とする。

**(d) 値入換え基準推定アルゴリズム**

**Step1:** 値が変更されている属性を1つ探す。これを注目属性とする。

**Step2:** 正例について、注目属性の元の属性値と変更後の属性値の対を作る。

**Step3:** 元の属性値が同じで、変更後の属性値が異なるものを探す。この変更後の属性値の候補が複数あるものを不明確な基準とする。

**Step4:** 負例について、注目属性の属性値を取り出す。

**Step5:** Step2 で得られた属性値の対の中で、元の属性値が Step4 で得られた負例の属性値と等しい属性値の対を不正確な基準とする。

**Step6:** 不明確でなく、不正確でない属性値の対を値入換え基準として提示する。

**Step7:** さらに、不明確な基準と不正確な基準を参考情報として、提示する。

**5.2.6 複数属性を持つ評価基準の知識の推定**

上記に示した帰納推論（帰納的一般化）アルゴリズムは、基本的には、操作の型（評価基準の表現）により制限されたバージョン空間のもっとも一

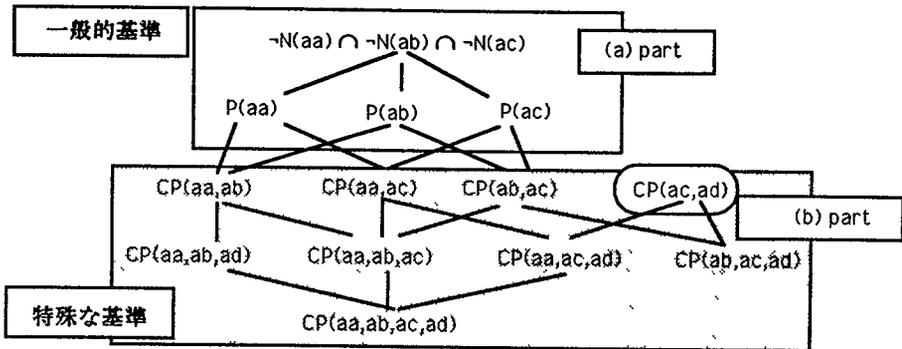


図 5.2: 例題のバージョン空間

一般的な部分を決定する処理を行っていた (図 5.2 参照)。この節では、選択・分類操作の基準が、複数の属性からなる場合を論じる。ただし、ここで述べる方法は、仮説として考慮するバージョン空間を徐々に、特殊な方向に広げながら、上記 5.2.3 節のアルゴリズムを適用しながら知識推定を行うものである。ただし、帰納推論を効率化するために、専門家のヒューリスティックス (問題領域の知識) を利用して、考慮するバージョン空間の広げ方を制御する (5.2.7 節参照)。

新しく仮説の候補の形式を次のように定める。

複数属性から成る選択基準は、属性の組合せとそれに対応する属性値の組合せを選択基準とする。

つまり、選択基準は、属性の組 (属性名 1、属性名 2、...) に対して、取り得る属性値の組 (属性値 1、属性値 2、...) を要素とする集合で表す。

5.2.3 節で示した例の基準の一般性・特殊性の関係は、図 5.2 で表される。ただし、図中の  $CP(aa, ab)$  は、すべての正例について、属性  $aa$  と属性  $ab$  について、同一の正例から得られる属性値の組合せを要素とする集合を表す (負例についても同様に、 $CN(aa, ab)$  で表す)。この図で、5.2.3 節 (d) のアルゴリズムでは、図 5.2(a) の部分 ( $P(aa)$ ,  $P(ab)$ ,  $P(ac)$ ),

$\neg N(aa) \cap \neg N(ab) \cap \neg N(ac)$ のみを対象としていた。複数属性を考慮する場合には、図 5.2 (b) の部分を検討することになる。

たとえば、単一属性では、選択基準とならない属性  $ad$  を含む  $CP(ac, ad)$  が選択基準の候補として検討対象になる。候補となるのは、 $CP(a1, a2, \dots) \cap CN(a1, a2, \dots) = \emptyset$  (空集合) を満足する属性に限る。

ここで、例題として、下記の値を持つ正例と負例の場合を説明する。

Positive	at1	at2	at3	at4
e1:	1	1	1	0
e2:	1	1	0	1
e3:	1	1	0	0
Negative				
e4:	0	1	0	1
e5:	1	0	0	0

推定の候補となる属性の連言形に対応するように属性値のリストを各例(要素)ごとにつくり、その属性の連言形を1つの属性、その属性値リストを1つの属性値として、5.2.3 節のアルゴリズムを適用する。たとえば、2つの属性  $at1$  と  $at2$  を一組とした場合には、与えられた要素  $e1$  の属性と属性値は、次のように表される。

属性	(at1, at2)	(at1, at4)	(at2, at4)
e1:	(1, 1)	(1, 0)	(1, 0)

この例で求められた選択基準の候補は、

$(at1, at2) = \{ (1, 1) \}$  (つまり、 $CP(at1, at2) = \{ (1, 1) \}$ ) となる。

### 5.2.7 ヒューリスティックスの利用

生成される仮説の言語を制限しても、属性の組合せ数が多い場合には、仮説の数は増加する。効率良い推論のためには、獲得対象に依存した領域固有の知識を利用して、可能性を減らさなければならない。ところが、この知識は専門家自身が持っている。そこで、図 5.2(b) の部分を検討するにあたり、例に含まれる例の類似度 (属性値が等価である属性の数で例間の距離を規定したもの) を示して、専門家から属性の重要性 (優先順位) を推論のためのヒューリスティックスとして獲得する。ここでは、どの属性から注目すべきかがヒューリスティックス知識となる。特に、正例と負例の中で距離の近いものの対を専門家にみせることで、その例を分離するのに必要な属性を思いつかせる。

上記例の場合には、単一の属性では、正例と負例は、分離できない。そこで、正例と負例の要素間の距離 (distance) を次の式で求める。

$$\text{distance}(e1, e4) = |v11 - v41| + |v12 - v42| + |v13 - v43| + |v14 - v44|$$

(ただし、 $|a - b|$  は、属性値  $a$  と  $b$  が等しいとき、1 それ以外は、0 とする。)

正例と負例間で、もっとも、近い要素の対は、 $e2$  と  $e5$ 、及び、 $e3$  と  $e5$  (いずれも距離は 1) である。これらが、近い例であることを専門家に示して、この 2 つを分離するのに、重要と思われる順に専門家に属性を並べさせる。専門家が無関係と思う属性は、外すことができる。いま、 $at3$  が無関係で、 $at2, at1, at4$  の順に重要と指示されたとすると、仮説の生成・検査で考慮される属性の組合せ順序は、 $at2 \wedge at1, at2 \wedge at4, at1 \wedge at4, at2 \wedge at1 \wedge at4$  となる。

## 5.3 提案誘導型知識獲得支援システム

### 5.3.1 提案誘導型知識獲得支援による知識獲得

提案誘導型知識獲得の基本的知識獲得の形態は、専門家から与えられた知識にもとづいて、仮説を生成し、その仮説を専門家にフィードバックすることで知識の抽出を行うことである。「専門家が知識獲得支援システムに与える情報」、「知識獲得支援システムが生成する情報」、「専門家から知識獲得支援システムへの要求」を列挙すると次の(1)から(7)になる。

**専門家が与える情報:**

- (1) 例(操作の入出力の要素の例) (ヒューリスティックスの情報を含む)
- (2) 専門家がシステムが提案したものを修正した知識

**知識獲得支援システムが生成する情報:**

- (3) 例に含まれる性質(選択・分類操作では、例の類似度情報)
- (4) 例から帰納的に推論される一般的知識
- (5) 専門家が修正した知識の検証結果  
(与えられた入力要素と修正された知識による操作の実行とその結果である出力要素集合を表示)

**専門家から知識獲得支援システムへの要求:**

- (6) 次候補(推定した知識)の提示要求
- (7) 専門家が修正した知識の検証

この専門家と知識獲得支援システムの関係を図示したのが、図 5.3であ

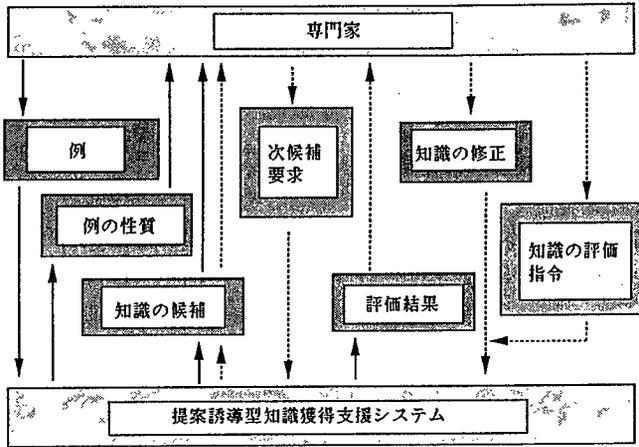


図 5.3: 提案誘導型知識獲得における対話

る。対話の過程の一例を説明する。専門家の提示した例 (1) に対して、知識獲得支援システムが一般的知識の候補 (4) を提示して、これを専門家が修正する (2)、また、その修正が例 (1) を満たしているかをシステムが判定する (5)。

提案誘導型知識獲得支援システムは、知識獲得支援システム EPSILON/One に、操作推定の機能を独立したサブモジュールとして追加した形で実現されている (図 5.4 参照)。

知識獲得支援システム EPSILON/One は、逐次推論マシン PSI-II 上に論理型言語 ESP で、操作推定サブモジュールは、並列論理型言語 KL1[Miyazaki 88] で実現されている。操作推定サブモジュールでは、選択・分類操作推定アルゴリズム (単一属性と複数属性の評価基準を対象)、順序付け操作推定アルゴリズム (単一属性の評価基準を対象)、値入換え操作推定アルゴリズム (単一属性の評価基準を対象) が実現されている。

図 5.5 は、ベアリング選定知識を獲得する過程で、ベアリングのカタログ知識 [Nippon-seikou 86] と専門家の選択順序から順序付け基準の推定をおこなった結果の一例の表示である。(詳しくは付録 C を参照)

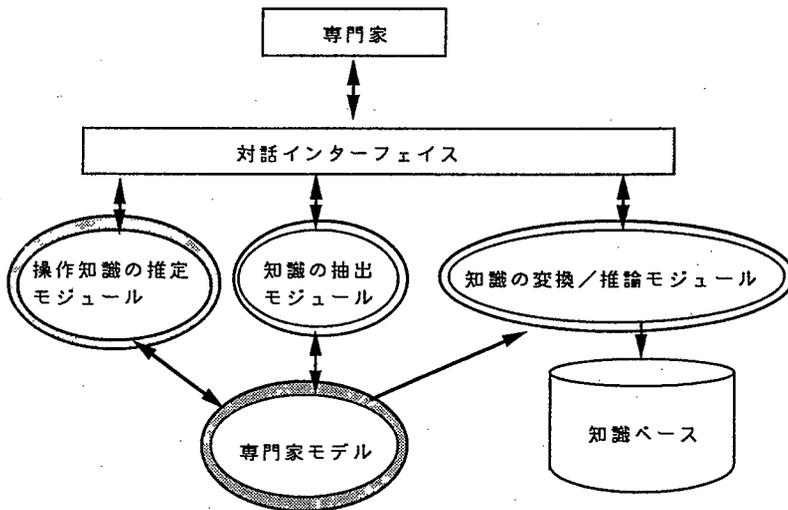


図 5.4: 拡張版 EPSILON/One の構成

## 5.4 関連研究と課題

ID3[Quinlan 86] は、分類木をつくるのに適した帰納推論アルゴリズムである。分類の評価基準としては、その分類木の探索の効率を考慮している。操作推定の問題では、分類木の葉 (leaf) を与えて、分類基準を求める問題であるが、ID3 は、分類基準となる情報を与えて、分類過程 (分類木) と分類木の葉を求めるアルゴリズムである。

MIS[Shapiro 81] は、Horn 節を帰納的に求めるシステムである。Yes/No の質問をオラクル (Oracle) と繰り返しながら、Horn 節を特殊化する。Horn 節の候補は、精密化操作 (オペレータ) により、その探索空間が規定される。この探索空間は、バージョン空間に相当する。また、質問の出し方を工夫することで、効率的な探索の枝刈りを行っている。操作推定で扱っている推論候補よりも一般的な Horn 節を扱っている点が勝っているが、提案誘導型知識獲得支援システムでは、専門家との対話は、単純な Yes/No 質問だけでないため、推論を効率化させるための情報をより多く得ることができる。

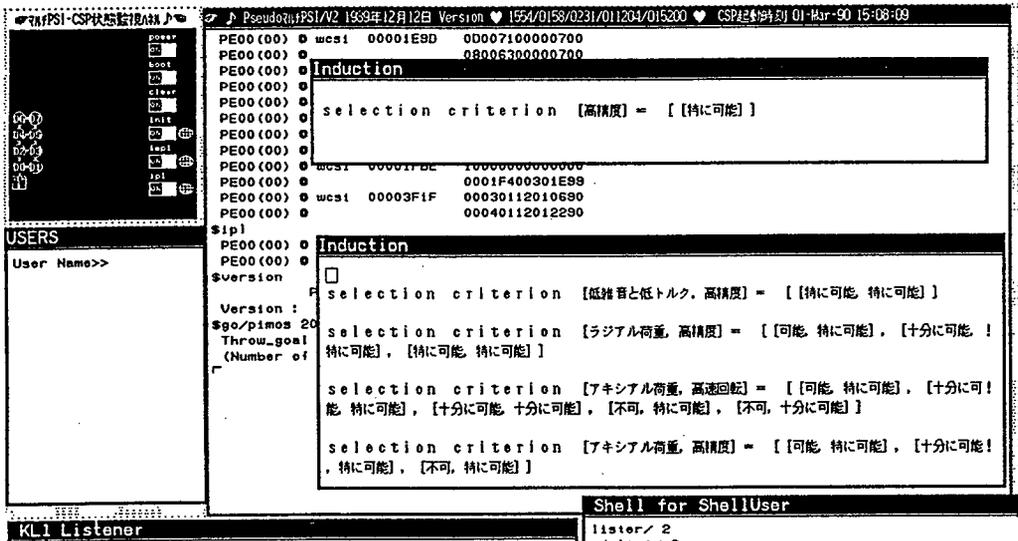


図 5.5: 評価基準の推定画面の例

ETS/AQUINAS/CTAS は、分類問題に適した対話型の知識獲得支援システムである。知識は、分類項目、特徴および、項目と特徴間の関係度 (rating 値) を表す格子型知識表現 (rating grid) の形式で獲得される。この知識の洗練は、知識表現を変形することにより行われるが、導出グラフ (implication graph) は、特徴間の含意関係を仮説として生成する。専門家は、不適当な含意関係を見つけると、その関係を導かないように、元の知識 (rating grid) を修正する。また、仮説生成は、特徴間の含意関係を関係度から生成している。これは、本論文で言及している提案誘導型知識獲得支援システムの単純な実現例と言える。

現在の操作推定では、生成する仮説の言語を制限しているが、複雑な組合せ概念 (排他的論理和など) を考える構成的帰納推論 (constructive induction)[Genesereth 86] を行う場合には、計算量が増加する。このような場合にも対応できるように、専門家からヒューリスティックスをうまく引き出せるような情報の提示と対話方法が課題である。これには、5.2.7

節で示した例の性質を利用したヒューリスティックスの抽出方法をより発展させる必要がある。CTAS の例の類似度を木構造で表す類似分類木 (cluster tree) の利用が有効と考えられる。

## 5.5 結言

知識獲得における知識抽出過程において、帰納推論（帰納的一般化）で生成した知識の候補を専門家に提示することで、知識を獲得する提案誘導型知識獲得の枠組とその実現方法を示した。特に、操作の型により帰納推論が生成する仮説を制限することで、知識の候補を効率的に求める方法を示した。この研究は、工学的な知識獲得方法と数学的・論理的な帰納的学習方法の1つの適切な融合を示した例といえる。



## 第 6 章

### 操作の型を利用した知識の洗練

#### 6.1 緒言

知識抽出により集められた知識には、矛盾や不足などの不備が含まれていることが多い。この不備を除くのが知識の洗練である。知識の不備はその使用目的が分かって初めて、その不足や矛盾が発見されることが多い。従来の知識の洗練法は、TEIRESIASのように専門家の判断結果とシステムの推論結果の相違点から知識ベースの不備を推論過程を辿って、検査・修正する方法が主流であった。MOREは、問題解決モデルを定めて、それに必要なドメインモデル（対象知識の構造）の理想的な状態を想定して、その状態と抽出された知識の差異から知識を洗練する方法であり、知識の使用目的を決めた上で知識の洗練を行っている。個々の知識を使用目的に合うように洗練するには、その知識の役割を知る必要がある。MOREは単純な問題解決モデル（cover-and-differentiation）での知識の役割を定めて対象知識の洗練を行っているが、種々の問題解決モデルに対応するのは容易ではない。また、複雑な問題解決モデルであれば知識の役割も複雑となりその洗練方法も個々に準備しなければならない。本章では、操作の型を利用して効率的に知識の洗練を行う方法を論じる。

## 6.2 知識の洗練

知識の洗練は、獲得した知識の不備（矛盾・不足・冗長など）を除く機能である。この知識の不備には、論理レベルの不備、対象領域に依存した不備と使用目的から見た不備がある。

**論理レベルの不備** とは、 $a$  と  $\neg a$  が存在する場合の論理的矛盾

$(a \wedge \neg a \rightarrow \perp)$  などである。

**対象領域に依存した不備** とは、その対象領域固有の制約を満足しない場合などを言う。たとえば、歯車の利用において、「歯車は、その回転軸まわりに、同時に1方向にしか回転できない」などの物理的制約がある。「同時に2方向に回転する歯車」や「この歯車の存在を前提にした知識」は上記の制約を満たさない。このように対象領域の制約に違反する知識には不備がある。

**使用目的から見た不備** とは、たとえば、診断問題のための知識ベースにおいて、故障箇所を特定するという目的があるにもかかわらず、観測した兆候から故障の候補が複数推論されてしまうことである。

知識ベースの使用目的や使用環境がはっきりしない場合や多目的の汎用知識ベースになると、論理レベルの洗練しか行えない。

そこで、“操作”と呼ぶ知識単位で抽出された知識の洗練を、数種類に分類された操作の型を利用して、不備の検出と不備の解消を行う方法を開発した。以下、操作知識とその洗練について説明する。

### 6.2.1 操作知識と洗練

効率的な洗練を行うには、使用目的および、対象領域の情報を旨く利用する必要がある。しかしながら、問題解決のタスクが複雑であれば、知

知識の使用目的も複雑となり、洗練が難しくなる。そこで、使用目的を分解して、局所的な用途についての知識の洗練が考えられる。つまり、問題解決の過程を操作でとらえることで、操作の型を使用目的の部分問題（サブゴール）化されたものとして利用できる。また、対象領域の例を操作の処理対象要素として与え、その操作の処理結果を調べることで操作（知識）の洗練を行うこともできる。

操作は、それぞれ型を持っている。つまり、その役割が決まっている。この役割をどの程度満足しているか、満足していないかによって、その不備を判定する。この役割に対する不備を見ると、いかなる入力要素に対しても不備になる場合と特定の入力要素に対してのみ不備になる場合がある。前者は、操作の知識、すなわち、評価基準の知識のみに内在する不備であり、推論を行う以前に知識を洗練できることから、静的洗練と呼ぶことにする。これに対して、後者は、処理対象の要素（知識の適用対象の情報）が与えられた場合での知識の洗練となり、動的洗練と呼ぶことにする。

### 6.2.2 洗練戦略

不備の解消の方法には、いくつかの水準がある。不備を起こす知識の修正（一般化や特殊化など）を行う水準、知識を削除する水準、新たに知識を抽出する水準などがある。また、操作の評価基準（属性値の集合など）の修正水準と評価属性の変更がある。そこで、操作の不備の解消戦略を以下のように分類する。

#### 1. 評価基準の特殊化

- (a) 直接修正：システムが検出した不備を含む基準を専門家が特殊化する

- (b) 選択修正：評価基準に対して、システムが提案した値の追加による特殊化を行うか・行わないか、その追加をどの基準に行うかを専門家に問う。

## 2. 評価基準の一般化

- (a) 直接修正：システムが検出した不備を含む基準を専門家が一般化する。

- (b) 選択修正：評価基準に対して、システムが提案した値の削除による一般化を行うか・行わないか、削除をどの基準から行うかを専門家に問う。

- 3. 評価基準の削除：システムが提案した評価基準の削除を行うかどうかを専門家に問う。

- 4. 評価属性と基準の提案：操作の入出力の例があれば、その入出力例から帰納推論により、操作の評価基準を推定し、これを評価基準の候補として提案する。（第 5 章参照）

- 5. 警告：不備ではないが、操作の処理が片寄っている場合（たとえば、分類処理で「ほとんど分類されない分類先」があるとき）に警告を行う。

- 6. 再抽出：明らかに知識が不足している場合にプリ・ポスト法により、知識の再抽出を行う。

## 6.3 操作の型を利用した洗練

本節では、各操作の型毎の知識の不備の定義、評価基準の例に対する不備とその検出、不備の解消戦略について説明する。なお、動的な洗練が対象としている不備は、静的な洗練で検出・解消可能でない不備を対象としている。また、各操作共通の不備及び、与えられた要素の不備については、まとめて6.3.4節で述べる。

### 6.3.1 分類と選択操作

分類操作は、要素を分類基準により複数の分類先に分ける。選択操作は、分類先が1つの場合の分類操作である。分類対象全体の集合を  $X$ 、分類先を  $m$  個としたときの各分類先の分類基準を満たす要素の集合を  $A_k$  ( $k=1, \dots, m$ )、動的洗練の際に与えられる分類対象の集合を  $B$  とする。ただし、 $B \subset X$  とする。分類操作は選択操作の複合操作であるため、分類操作の洗練は選択操作では検出できない複合的な不備を洗練できる。

#### 分類基準の例

分類基準は、評価対象の属性  $at$  とその取りえる値の集合で表される。取りえる値は、関係 ( $=, \neq, <, >, \leq, \geq$ ) と値 (文字列, 整数, 実数) の対で表される。各集合要素間の関係は、OR関係である。複数属性の組合せを分類基準として扱う場合には、「属性の組」に対応する「取りえる値の組」を要素とする集合で、分類基準は表される。

#### 分類操作の静的な不備

1. いかなる要素も割り振られない分類先 ( $A_k$ ) がある。(選択操作では、いかなる要素も選択しない選択基準の不備に相当する。)

$$\forall x_i \in X, x_i \notin A_k \text{ i.e. } X \cap A_k = \emptyset \text{ (空集合) となる。}$$

このとき、任意の集合  $X$  について、 $X \cap A_k = \emptyset$  となるのは、  
 $A_k = \emptyset$  となるときである。

また、分類基準が基準  $a$  と  $\neg a$  を基準として含み、かつ、 $a \wedge \neg a$  を含む場合には、 $a \wedge \neg a \rightarrow \perp$  (矛盾) であり、評価基準は、論理的矛盾を起こす。この場合にも、 $A_k = \emptyset$  となる。(図 6.1(a) を参照)

## 2. いかなる要素の同じ分類先 ( $A_k$ ) に分類する。

(選択操作では、いかなる要素も選択する選択基準の不備である。)

$$\forall x_i \in X, x_i \in A_k \text{ i.e. } X \subseteq A_k$$

また、分類基準が基準  $a$  と  $\neg a$  を基準として含み、かつ、 $a \vee \neg a$  を含む場合には、いかなる集合  $X$  についても  $X \subseteq A_k$  となる。

(図 6.1(b) を参照)

## 3. 複数の分類先に分類される要素がある。

$$\forall x_i \in X, x_i \in A_k \text{ かつ } x_i \in A_l$$

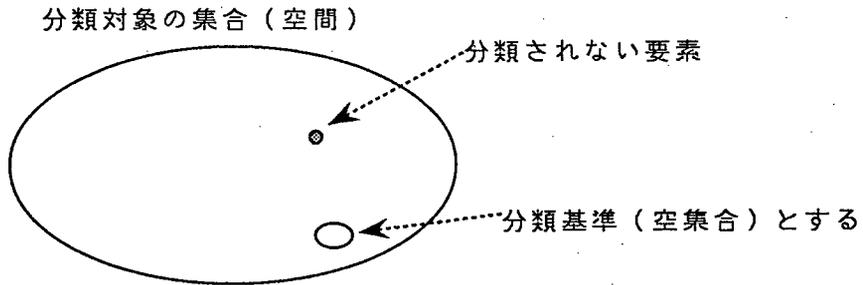
$$\text{i.e. } A_k \cap A_l \neq \emptyset$$

図 6.2 のように分類基準の集合に共通部分がある。

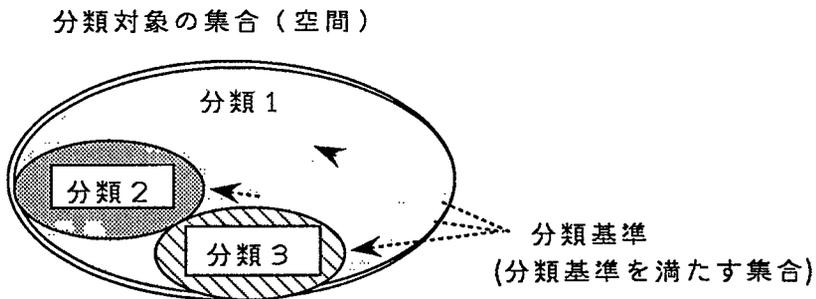
## 分類操作の静的な不備の検出と解消

1. 分類先の中で取り得る値の集合が空集合の場合には、その分類先には、すべての要素が分類されない。評価属性  $at$  に対して、取りえる値の集合が  $\emptyset$  (空集合) であるかどうかを検査すればよい。

解消戦略は、特定の分類先についての分類基準が「未獲得」であることを提示する。分類基準の再抽出を 6.2.2 節の戦略 6 を適用して質問を行う。



(a) いかなる要素も分類基準を満たせない



(b) いかなる要素も分類 1 に分類される

図 6.1: 分類操作の静的な不備 (その 1)

2. 取りえる値に、肯定と否定の情報が存在する場合には、その分類先にすべての要素を分類する。

評価属性  $a_t$  に対して、取りえる値の集合要素に、 $(= a_i)$  と

$(\neq a_i)$  ( $a_i = 1, \dots, n$ ) が両方とも存在するかどうかを検査すればよい。解消戦略は、矛盾する値（たとえば、 $(= a_i)$  など）の対を示して、戦略 1 (a)(b) を適用して、その値を削除するか、修正するかを質問する。

3. 取りえる値に、すべての数値の範囲を表す情報が存在する場合には、その分類先にすべての要素を分類する。

評価属性  $a_t$  に対して、取りえる値の集合要素の数値の区間が、 $-\infty$  から  $+\infty$  を示すかどうかを検査する。

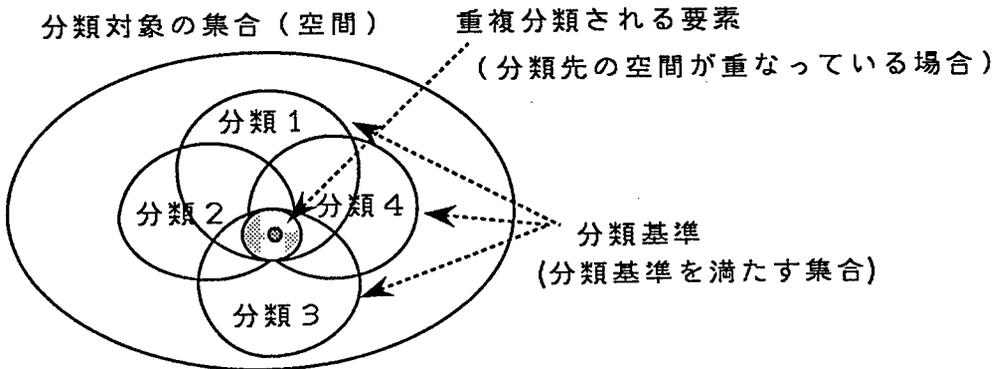


図 6.2: 分類操作の静的な不備 (その 2)

たとえば、 $(\geq a)$  と  $(\leq b)$  があり、

$a \leq b$  ならばこの条件を満たす。解消戦略は、 $-\infty$  から  $+\infty$  の区間を定義する値 (条件) の組 (この例では、 $a, b$  になる) を列挙して、どの値 (条件) を削除するか、修正するかを戦略 1 (a)(b) を適用して、質問する。

4. 取りえる値の要素間に、同じものや一方が他方に含まれる場合には、冗長な評価基準である。2 要素間を比較して、等価性と包含性を検査する。

解消戦略は、等価なものや包含関係にあるものを提示して、「等価なものの方」・「他のものに包含されるもの」を削除してよいかを戦略 3 を適用して、質問する。

5. 分類先の取り得る値の集合間に共通集合が存在する。

取り得る値の集合間の共通集合を調べ、共通集合が空集合でない集合の組合せは不備となる。解消の方法は、その共通集合の要素がどちらの集合に含まれるかを戦略 1 (b) または、2 (b) を適用して、質問する。

## 分類操作の動的な不備

1. 与えられたいかなる要素も割り振られない分類先 ( $A_k$ ) がある。

言い替えると集合  $B$  のいかなる要素も集合  $A_k$  に含まれない場合である。

$$\forall b_i \in B, b_i \notin A_k \text{ i.e. } B \cap A_k = \emptyset \text{ (空集合)}$$

(注:  $A_k = \emptyset$  は、静的な不備の解消で検出済み)

集合  $B$  が図 6.3 の領域 (a) の部分に対応する場合である。

2. 与えられたいかなる要素の同じ分類先 ( $A_k$ ) に分類する。

言い替えると集合  $B$  のいかなる要素も集合  $A_k$  に含まれる場合である。

$$\forall b_i \in B, b_i \in A_k \text{ i.e. } B \subseteq A_k$$

集合  $B$  が図 6.3 の領域 (b) の部分に対応する場合である。

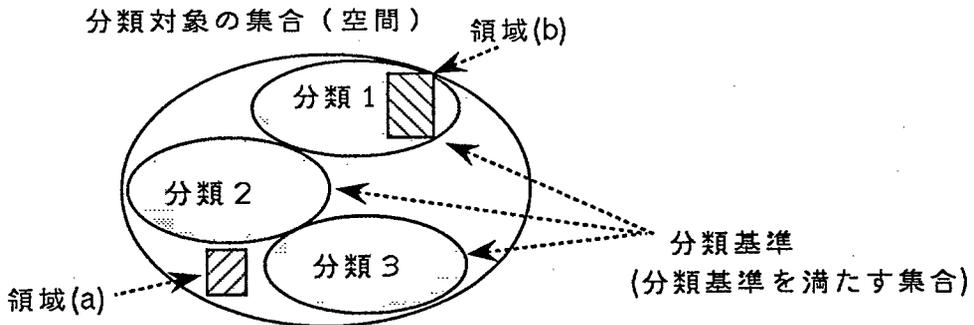


図 6.3: 分類操作の動的な不備

3. 与えられた要素に対する分類処理が片寄る場合

集合  $B$  が  $A_k$  と  $A_l$  ( $k \neq l$ ) に分類されるときに、 $A_k$  と  $A_l$  の要素の数 (cardinal number) が極端に異なる場合には、知識の不備の可能性はある。

## 分類操作の動的な不備の検出と解消

1. 評価属性についての与えられた入力要素の値の集合 B と分類先の取りえる値の集合の積集合が存在しない場合には、すべての要素がその分類先に分類されない。操作の実行後に、要素を含まない分類先があれば、この状態であることが分かる。不備の解消には、戦略 2 (a)(b) または、正しい分類が与えられれば、戦略 4 を適用する。
2. 与えられた入力要素の値の集合 B が評価属性について、取りえる値の集合に含まれるか、また、入力要素の属性値が数値であり、その値すべてが取りえる値の集合の条件を満たすならば、すべての要素が分類される分類先がある。操作の実行後に、入力されたすべての要素を含む分類先があれば、この状態であることが分かる。不備の解消には、戦略 1 (a)(b) を適用して、この分類先の分類基準を特殊化する。
3. 分類結果の分類先の要素数を比較することで、分類の片寄りを評価する。片寄りがあれば、戦略 5 を適用して、警告を出す。この警告について、専門家が不備と判定すれば、分類基準の変更または、分類結果の変更を質問する。分類結果の変更質問においては、分類先の分割、分類先の統合、要素の分類先の変更を尋ねる。分類先の分割であれば、分類基準の取り得る値の分割を促す。つまり、戦略 1 (a) を適用して、分類基準を特殊化する。分類先の統合では、分類基準の取り得る値の集合を和集合とする。つまり、戦略 2 (a) を適用して、分類基準を一般化する。  
要素の分類先の変更では、分類基準の各分類先の分類属性の取り得る値の変更を行う必要があるため、戦略 1 (a), 2 (a), 6 を適用して、取り得る値の変更を専門家に促すか、戦略 4 の操作推定の方法によ

り、分類基準を分類先ごとに提案する。

なお、評価基準の実現方法によって、検出や解消の方法は、異なる。たとえば、評価基準の集合に  $\{ (= a_i) \cap (\neq a_i) \}$  のような論理式を表現できるようにした場合には、 $a_i \wedge \neg a_i \rightarrow \perp$  からこの集合は、 $\emptyset$  (空集合) となり、論理的な取りえる値の集合要素の検査が必要となるが、 $\neq$  (否定) を扱わない基準であれば、この不備の検査は必要はない。

### 6.3.2 順序付け操作

順序付け操作は、1つの属性に注目して、その属性値についての順序基準に従って、与えられた要素を順序付けする。

#### 順序付け基準の例

順序付け基準は、評価属性についての半順序または、全順序関係を表す値の優先順序のリストとする。

b よりも a を優先する関係、つまり、半順序関係を  $a \gg b$  で表す。半順序関係が複数ある場合には、各半順序関係間は AND 関係とする。

#### 順序付け操作の静的な不備

##### 1. 順序基準が順序関係を表さない場合

獲得した順序基準が、順序関係を表さないものとしては、獲得した順序基準が、

$$a_1 \gg a_2 \gg \dots \gg a_n \text{ のときに、}$$

すべての  $a_i$  ( $i = 1, \dots, n$ ) が等しい場合である。

##### 2. 半順序関係が矛盾する場合

獲得した順序関係に相反するものがある場合には、これを矛盾とし

て扱う。

$$(a \gg b) \wedge (b \gg a) \rightarrow \perp$$

### 3. 順序関係の情報不足で、全順序関係を構成できない場合

2項関係で表される半順序関係で抽出された順序基準において、この半順序関係から全順序関係を作るのに情報が不足している場合である。

例 (16):  $a \gg b, b \gg c, d \gg c$  の関係がある場合には、

次の3つの全順序の可能性がある。

$$(a \gg b \gg d \gg c) \vee (a \gg d \gg b \gg c) \vee (d \gg a \gg b \gg c)$$

## 順序付け操作の静的な不備の検出と解消

### 1. 順序基準が順序関係を表さない場合

順序関係リストが空リストの場合であるか、リストの要素が全部同じ場合である。戦略6を適用して、値のリストまたは、評価属性の抽出を行う。

### 2. 半順序関係が矛盾する場合

矛盾する2項関係を検出し、戦略2 (b)を適用して、どちらの関係を選択すべきかを質問する。

### 3. 順序情報が不足していて、全順序関係を構成できない場合

戦略1 (a)(b)を適用して、順序関係の不明な半順序関係のリストの要素を比較することにより、新たな関係を抽出する。たとえば、例(16)において  $b \gg d$  かどうかを質問する。

## 順序付け操作の動的な不備

### 1. 順序基準の知識が不足している場合

与えられた要素が順序基準に含まれない値を持つ場合である。つまり、ある属性について、

$a_1 \gg a_2 \gg \dots \gg a_n$  の関係が順序基準のときに、与えられた処理対象の要素がその属性値として、 $b$  を持つときに、

$$b \notin \{ a_1, a_2, \dots, a_n \}$$

であれば、順序基準は、 $b$  に関する知識不足となる。

## 2. 順序基準の属性が不適切な場合

与えられた要素の評価属性の値がすべて同じ場合には、その属性では、順序が付けられない事になる。

### 順序付け操作の動的な不備の検出と解消

#### 1. 順序基準の知識が不足している場合

与えられた要素の順序基準の属性の属性値集合と順序基準の値リストの集合との積集合が空集合ならばこの状態である。検出は、実行時に、与えられた要素の評価属性の値が値のリストの要素であるかどうかを調べることで判る。不備の解消は、戦略1 (b) を適用して、その値を値リストのどの位置（順序位置）に入るかを質問する。

#### 2. 順序基準の属性が不適切な場合

与えられた要素の評価属性での値がすべて同じ場合である。戦略6により、専門家から別の属性での順序基準を獲得する必要がある。要素の並べるべき順序が分かっている場合には、戦略4の操作推定の方法を利用して、順序基準の候補を生成する。

### 6.3.3 値入換え操作（変換操作）

値入換え操作では、値入換え基準（変換ルール・変換規則）に従って、ある属性の値を変更する。変換対象全体の集合を  $X$ （変換属性の値の集合）、

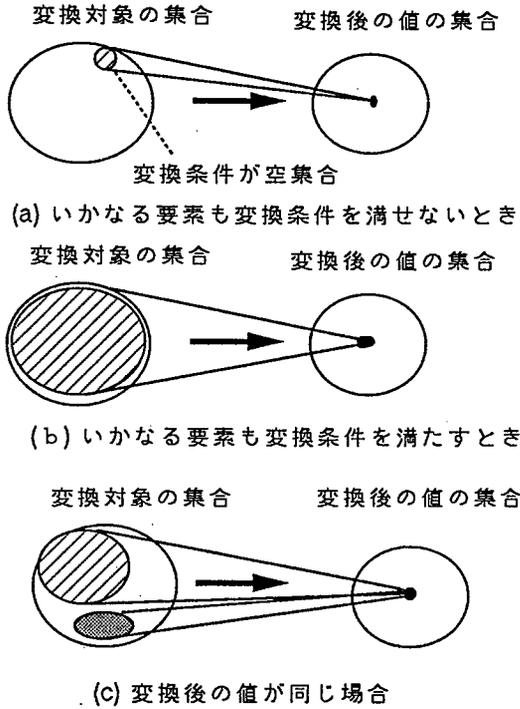


図 6.4: 値入換え操作の静的な不備 (その 1)

変換ルール (規則) の集合を  $R$ 、各ルール  $(r_i (i=1, \dots, n))$  の条件部は変換基準の属性値が取り得る値を  $c_i$  で表し、ルールの実行部の属性の変更後の値を  $d_i (i=1, \dots, n)$  で表す。  $c_i$  を要素とする集合を  $C$  で表す。動的洗練の際に与えられる変換対象の集合を  $B$  とする。(注: 変換ルールの条件部は選択操作に相当するため、ルールの条件部は選択操作の洗練方法を流用可能である)

### 値入換えの基準の例

各ルールの条件部は、否定を含まない基準とし、単一の属性  $at$  に対して1つの値 (文字列か数値) の等価性判定または、数の区間 (1つ または、複数の不等号の AND 関係) 判定とする。変換後の値は、文字列また

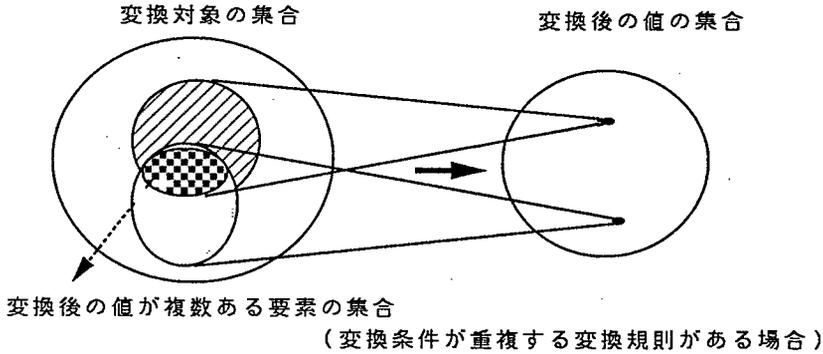


図 6.5: 値入換え操作の不備 (その 2)

は数値とする。各ルール間は、OR 関係とする。

### 値入換え操作の静的な不備

#### 1. いかなる要素の値も入換えない

$X \cap C = \emptyset$ 、i.e.  $C = \emptyset$ 、これは、変換ルールが全くないか、個々のルールの変換条件  $c_i$  に、条件  $a$  と条件  $\neg a$  があり、 $a \wedge \neg a$  が存在する場合である。(図 6.4 の (a) を参照)

#### 2. いかなる要素の値も同じ値に入換える

$\forall x_i \in X$  について、 $x_i$  は、変換ルール  $r_i$  の条件  $c_i$  を満たす。つまり、変換条件  $c_i$  に、条件  $a$  と条件  $\neg a$  があり、 $a \vee \neg a$  が存在する場合である。(図 6.4 の (b) を参照)

また、 $X \subseteq C$  かつ、変換後の値  $d_i$  が同一の場合である。つまり、すべてのルールの変換後の値  $d_i$  が同じ場合である。(図 6.4 の (c) を参照)

#### 3. 値入換え候補が複数ある場合

$\exists x_i \in X$ 、 $x_i$  が複数の変換ルール  $r_i, r_j (r_i \neq r_j)$  の変換条件  $c_i, c_j$  を満たし、その変換後の値  $d_i, d_j$  が異なる場合である (図 6.5 参照)。

### 値入換え操作の静的な不備の検出と解消

#### 1. いかなる要素の値も入換えない

変換ルールがないときは、戦略 6 により、知識の再抽出を行う。個々のルールの条件は、単一の文字列、数値または、数値の範囲で表されるため、矛盾に相当するものは、例 (17) のような数値の存在しない範囲を示すものである。この場合には、戦略 2 (a)(b) を適用して、取り得る数値の範囲を一般化するための質問をする。

例 (17):  $\{ a \mid a > 10 \text{ かつ } a < 9 \}$

#### 2. いかなる要素も同じ値に入換える

例の基準では、変換条件が任意の文字列であることはない、また、不等号の OR 条件は表現できないので任意の数値であることもない、よって、変換条件  $c_i$  に、条件  $a$  と条件  $\neg a$  があり、 $a \vee \neg a$  が存在することはない。このような場合になるのは、すべてのルールの変換後の値  $d_i$  が同じ場合である。すべてのルールの変換後の値  $d_i$  が同じ場合は、ルールの変換後の値を比較して、すべてが同じであれば、戦略 6 により知識の再抽出か、戦略 1 (a), 2 (a) により値の修正を行う。

#### 3. 値入換え候補が複数ある場合

変換条件が文字列の場合には、その文字列に同じものがある場合である。このときに、変換後の値が同じものであれば、ルールが冗長であり、異なれば、一方を削除する必要がある。戦略 2 (b) を適用して、どちらを削除するかを専門家に確認する。

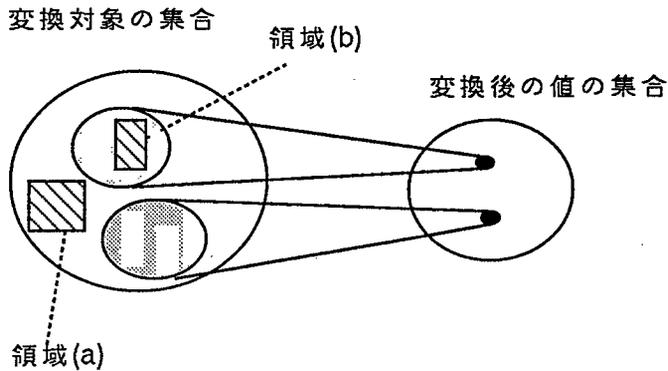


図 6.6: 値入換え操作の動的な不備

### 値入換え操作の動的な不備

1. 与えられたいかなる要素の値も入換ええない

$\forall b_i \in B, b_i \notin C$  の場合、つまり、 $B \cap C = \emptyset$

集合  $B$  が図 6.6 の領域 (a) の部分に対応する場合である。

2. 与えられたいかなる要素の値も同じ値に入換える

$\forall b_i \in B$  について、 $b_i$  は、特定の変換ルール  $r_i$  の条件  $c_i$  を満たす。

集合  $B$  が図 6.6 の領域 (b) の部分に対応する場合である。

### 値入換え操作の動的な不備の検出と解消

1. 与えられたいかなる要素の値も入換ええない

与えられた要素の変換数を数え上げて、1 つも入換えられなければ、この状態である。この解消には、ルールの条件緩和か、新たなルールの追加が必要である。変換条件の緩和で可能なものは、数値の範囲の変更（範囲の拡大）であり、与えられた要素の評価属性が数値であれば、戦略 2 (a) により、数値の範囲の拡大を専門家に質問する。あるいは、戦略 6 により、新たなルールの追加または別の属性を評

価するルールの可能性を専門家に質問する。

## 2. 与えられたいかなる要素の値も同じ値に入換える

与えられた要素の変換後の値がすべて同じかどうかを確認することで検出できる。利用されたルールが1つのときを考えると、変換条件が文字列のとき、このルールの削除か評価属性を変更しないかぎり、この状態の解消はできない。数値の範囲が変換条件であれば、戦略 1 (a) により、これを特殊化、つまり、範囲を狭めることで解消できる。利用されたルールが複数の場合には、上記の解消方法に加えて、戦略 6 により、変換後の値をルール毎に異なるように知識の再抽出を行う。

### 6.3.4 その他の操作

#### 操作の不備

本節では、EPSILON/One で実現している操作の型について、前節までに説明しなかったものの不備を扱う。

**組合せ操作** は、2つの要素を組合せて、両方の属性とその属性値を持つ1つの要素を作る操作である。その要素の対に対する不備は、動的洗練において、与えられた2つの要素が同じ属性を持つ場合に、新しい要素でどちらの属性値を選ぶかが不明であることである。この場合には、戦略 6 を適用して、一方の要素の属性名を変更する。この操作の処理アルゴリズムは固定であるので、操作に対する洗練の必要はない。

**要素分解操作** は、ある要素を複数の要素に分解するが、分解対象の情報は、その要素の構成属性 (consist-of) で規定される。与えられた要素にこの属性情報が不足している場合には、要素の不備となる。この操

作も処理アルゴリズムは、固定であるので操作に対する洗練の必要はない。

数値演算操作、入力と出力操作 では、演算に利用される属性または、入出力情報に関する属性がない場合には、要素の不備となるか、参照または利用する属性の変更が必要である。つまり、戦略6により、別の評価属性を持つ知識の抽出を行う。

#### 共通の不備の検出と解消

動的洗練において、専門家の判定と操作の処理が異なる場合には、その正しい要素の入出力からその処理を満足する操作の評価基準を推定する戦略4の操作推定の方法が解消戦略として利用できる。

また、与えられた要素が評価属性を持たない場合、つまり、処理対象の要素に関して、その要素が評価基準が扱う属性を持たない場合には、「その要素の不備として、属性が不足している」か、評価基準が別の属性を評価すべきであることを示している。別の評価基準を抽出するには、戦略6を適用する。

#### 6.4 関連研究

知識の洗練に関しては、2種類の方法が研究されている。ひとつは、誤った解答を知識ベースを利用した問題解決システムが出した場合に、その誤答を生成した原因（箇所）の究明と修正を支援する動的な洗練方法である。誤りを生成した推論過程を辿り対話的にルールの修正を支援するのが、TEIRESIASである。このシステムは、推論過程を逆に辿りながらの対話的にルール型知識の修正を支援するため、修正箇所の候補が多い場合には、修正すべき箇所の特定が困難であり、組合せ的修正箇所の爆発問題

も発生する。正しい解答と誤った解答の生成比率から修正すべきルールを検出して、そのルールの修正を支援するのが、SEEKである。

SEEK2[Ginsberg 85]は、SEEKの修正機能をメタプログラム化できるようにし、その知識の修正を自動化したシステムである。このシステムは、ある程度の事例がないと適切な洗練の指示ができない。MOLEは、後述のMOREに、確信度の洗練機能などを追加したシステムであり、専門家の判断と異なる知識の確信度の修正を行う。これらのタイプシステムでは、事前に、事例とその事例に対する正しい解答（専門家の解答）が必要である。つまり、動的な洗練を支援する型である。

もうひとつは、洗練対象となっている知識ベースの使用目的からその不備を検出し、その不備の解消を支援する型である。MOREは、診断のためのドメインモデルを持ち、診断目的に合うようにドメインモデルで現されている知識を8つの洗練戦略で対話的に修正する。つまり、静的な洗練を支援するシステムである。このシステムでは、静的な洗練を中心に行っていたので、動的な洗練を可能としたMOLEが開発された。

本システムの知識の洗練は、後者の知識ベースの使用目的に相当する個々の知識の役割（操作の型で表される役割）に合った静的な洗練を行い、知識をその使用目的に合うように修正できる。さらに、前者のように、事例を利用して、さらに、知識の対象適合性に関する洗練を動的に行う。

MOLEの問題解決戦略(cover-and-differentiation)と比較して、操作の型の方がより複雑な問題解決対象に適用できるため、より詳細な知識の洗練が可能である。

本方式は、確信度を扱わないが、MOLEは、選択操作と選択対象の要素の確信度の洗練を行っている。

動的な洗練においても、専門家の処理結果がなくとも、洗練できる点があるが、TEIRESIASとSEEKよりも優れている。専門家の解答が利用できる

場合には、EPSILON/Oneで利用されている方法（操作知識の推定の方法）により、操作の評価基準を提案できる。

本システムの課題は、要素の洗練である。現在は、属性値の不足の検出と解消のみを支援している。要素を修正すべきであるか、操作を修正すべきであるかの決定基準の検討も必要である。

その他の問題としては、操作の評価基準が静的には、不備であるが、使う対象が限られている場合には、不備にならない（動的には、不備でない）こともある。静的な不備の解消コストが高くなる場合には、この問題を考慮する必要があると考える。

## 6.5 結言

知識の役割に注目した知識の洗練方法として、操作の型が示す役割を利用した知識の不備の検出と解消方法を紹介した。知識の洗練に於いては、「どのような対象にどのように使用されるのか」の情報が非常に重要である。これらの情報が知識獲得支援システムの知識表現にうまく表されているならば、効率的な知識の洗練が可能である。静的な洗練のためには、これらの情報が洗練以前に分かっていることが必要である。また、動的な洗練においても、これらの情報を利用して、専門家の解とエキスパートシステムの推論結果を比較しなくても一部の洗練が可能となる。また、動的な洗練では、知識の一般化と特殊化を行うため、帰納推論との関係が深い。そのため、本方法では、操作推定（第5章参照）を知識の候補生成に利用している。

本論文では、確信度やファジー表現などの曖昧性のある知識の洗練には言及しなかったが、確信度やメンバーシップ関数のチューニング方法との関係を明らかにしていくことも今後必要と考えている。



## 第 7 章

# 専門家モデルの操作の型の数学的な観点からの再構成

### 7.1 緒言

専門家モデルの操作の型は、プロダクションルールで記述された診断型知識ベースの分析結果に基づいている。そのため、設計型問題を表現するには操作の型が不足していることが第3章の評価でも判明している。また、操作の型は、1つのプロダクションルールで表現できる機能をより細かい機能に分解しない方針で求められているため、数学的には機能重複がある。本章では、関係代数との比較から専門家モデルの操作の型の過不足を確認し、さらに、操作の型の再構成を行う。

### 7.2 関係代数と専門家モデルの関係

専門家モデルの操作の型の充分性を検討するのに、関係データベースの操作を規定した関係代数 (relational algebra) との比較を行った。なぜなら、プロダクションシステムにおける作業記憶の要素は、関係データベースと似た属性と属性値から成る構造を持っているからであり、関係代数は関係データベースの基本処理に必要な操作を提供しているからである。

まず、関係代数について説明し、次に、専門家モデルとの比較を論じ

る。

### 7.2.1 関係代数

関係データベースは、コッド (E.F. Codd) が、1970 年に関係モデル (relational model of data)[Codd 70] を提案したことに始まる。彼は、データベースをその操作プログラムから独立させる目的で、現実世界のデータを把握するモデルとして、数学における関係 (relation) に注目した。関係は集合の基礎的な概念であり、次のように定義される [Matsumura 90]。

#### [集合における関係の定義]

定義 1 :  $M$  の直積  $M \times M$  の部分集合を  $M$  上の関係という。

定義 2 : 集合  $M$  上の関係  $\delta$  とは、 $M$  の任意の 2 元  $x, y$  に対して、 $x \delta y$  であるかないかをきめる規則のことである。

コッドは、関係の集まりからなる関係データベース上を操作して目的の情報を得るために、関係代数 (relational algebra) を提案した。

関係代数では、データベースに問い合わせる処理を関係の集合から新しい関係の集合を作る集合演算操作と捉える。関係代数には、通常の集合演算 (以下の 1 から 4 の演算) と関係代数独自の演算 (以下の 5 から 7 の演算) がある [Uemura 79][Ullman 82]。

1. 和集合 (union) を作る演算 :  $\cup$

$$R \cup S = \{x \mid x \in R \vee x \in S\}$$

2. 積集合 (intersection) を作る演算 :  $\cap$

$$R \cap S = \{x \mid x \in R \wedge x \in S\}$$

## 3. 差集合 (difference) を作る演算：－

$$R - S = \{x \mid x \in R \wedge x \notin S\}$$

## 4. 直積 (Cartesian product) を作る演算：×

$$R \times S = \{(r,s) \mid r \in R \wedge s \in S\}$$

## 5. 射影 p (projection)

関係を構成する属性のうちから、必要な属性だけを取り出して、新しい関係を作る演算を射影という。m項関係 R の組 r ( $r \in R$ ) の属性  $A_j$  の値を  $r[A_j]$  ( $j=1, \dots, m$ ) と記す。複数の属性について考えると

$$r[A_1, A_2, \dots, A_k] = r[A_1], r[A_2], \dots, r[A_k] \text{ となる。}$$

属性の並び  $A = (A_1, A_2, \dots, A_k)$  に対する  $r[A]$  は、

rの属性  $A_1, A_2, \dots, A_k$  のそれぞれの値 (成分) を並べたものである。

このとき  $R[A] = \{r[A] \mid r \in R\}$  を R の A への射影という。

## 6. 結合 (join)

2つの関係を共通の定義域を手がかりとして結合し、新しい関係を作る演算を結合という。θ は、=, ≠, <, ≤, >, ≥ のいずれかの比較演算子とする。関係 R の属性 A と関係 S の属性 B を次のように定義する。

$$R[A \theta B]S = \{(r,s) \mid r \in R \wedge s \in S \wedge (r[A] \theta s[B])\}$$

## 7. 制約 (restriction) (選択 (selection) とも言う)

1つの関係における2つの属性 A, B について結合と同様の演算を考えることができる。関係 R の2つの属性 (または、属性の並び) A と B との間の θ 制約を次のように定義する。

$$R[A \theta B] = \{(r,s) \mid r \in R \wedge (r[A] \theta r[B])\}$$

### 8. 割算 d (division)

ある関係の属性（群）とべつの関係の属性（群）とを比較して、一方がもう一方のすべての要素を完全に含んでいるかどうかを調べる演算である。

関係 R の属性の並びを A とし、C を A の補ならびとする。A と C を合せて、関係 R の属性とする。

$x=r[A]$ ,  $x \neq r[C]$  となる  $r[C]$  の集合を  $gR(A,x)$  とする。

$$gR(A,x) = \{r[C] \mid r \in R \wedge r[A]=x\}$$

$$gR(C,x) = \{r[A] \mid r \in R \wedge r[C]=x\}$$

関係 R の属性並び A と関係 S の属性並び B で  $R[A], S[B]$  が比較可能なとき関係 R を関係 S で割る割算を次のように定義する。

$$R[A \div B]S = \{r[C] \mid r \in R \wedge S[B] \subseteq gR(C,r[C])\}$$

## 7.3 専門家モデルと関係代数の比較

### 7.3.1 処理対象要素と関係データベースの違い

関係代数が関係データベースを操作する場合には、属性と属性値の組から成る表（処理対象要素、つまり要素に対応）ばかりでなく、その組の一部を抜き出して新しい表を生成する。これに対して、専門家モデルの操作は、属性と属性値の組から成る要素に操作を行うが、その操作結果は要素であり、一部の属性と属性値の組を抜き出して新しい要素を作ることはしない。つまり、関係代数は、表（要素に対応）とその部分を扱うが、専門家モデルの操作単位は、要素である。（注：専門家モデルも要素の一部の属性の属性値に操作を加えるが、操作の出力は要素である。また、要素分解の操作で分解された要素は、元の要素の属性の一部を持つものではない。

く、より詳細な属性を持つものである。さらに、出力操作は、要素の一部の属性を取り出すが、その出力要素は、元の要素であり、新しい要素は生成しない)

専門家モデルの1つの要素は、要素名と「属性名と属性値の対」の集合構造であり、要素集合は、要素に順序を持つ外延的表現集合である。

### 7.3.2 操作の型と関係代数の操作の対応付け

#### 通常の数集合演算

- 和集合は、専門家モデルでは、操作に入力される要素集合の OR 関係で実現している。
- 積集合は、集合間の要素を比較することで行われるが、専門家モデルの原型の操作の型 (第3章で説明した7つの型) にはこの演算はない。ただし、比較する一方の集合の特徴を評価基準で表すことで、類似操作を選択操作で実現できる。
- 差集合も積集合と同様に集合間の要素を比較するので、専門家モデルの原型の操作の型にはこの演算はない。ただし、積集合と同様に、比較する一方の集合の特徴を評価基準 (否定を扱う評価基準) で表すことで、類似操作を選択操作で実現できる。
- 直積は、作られた要素が元の要素の属性と属性値を持つものと見ること、組合せ操作に対応付けられる。

#### 関係代数特有の演算との比較

射影: 専門家モデルの操作は、要素の一部の属性を持つ要素の生成はできない。値の入換え (交換) で不要な属性値 (属性名は、既知である必要がある) を undefined などの記号で置き換えることで、そ

の属性が存在しないことと等価にすれば、同様の機能を実現できる。ただし、射影が、冗長な関係を除く機能を持つが、原型の操作の型には冗長な要素を除く機能はない。また、出力操作が特定の属性の値を計算機端末の画面などに出力する機能を持つが、これが射影機能の一部を実現していると言える。

**結合:** 関係  $R$  を持つ集合と関係  $S$  を持つ集合を入力要素集合として、組合せ操作により新しい集合を作る。属性  $a, b$  を与えてその間の値が、たとえば、同じなら ( $\theta$  の指定された関係なら) 属性  $c$  の値を ”等価” (別にどのような文字列でもよいが指定された関係を満足したか否かが分かる値) に値入換え操作で書き換える。

属性  $c$  が ”等価” である値を持つ要素を選択操作で選択する。(自然な結合 (natural join) では、その後、属性  $b$  の値を undefined に値入れ換え操作で変換する。) これで、結合が実現される。ただし、属性  $a$  と  $b$  が分かる必要がある。結合は、操作の型よりも水準の高い表現であり、操作の型の複合記述 (macro) で表現できる。

**制約:** 属性  $a, b$  を与えて要素内でその属性間の値が同じなら ( $\theta$  の指定された関係なら) 属性  $c$  の値をたとえば ”等価” に値入換え操作で書き換える。属性  $c$  が ”等価” である値を持つ要素を選択操作で選択する。これで、制約が実現される。ただし、属性  $a$  と  $b$  が分かる必要がある。この制約も操作の型の複合記述になっている。

**割算:** 入力要素集合に対して、割りたい属性とその取り得る値を評価基準として持つ選択操作で選択を行う。ただし、評価基準の属性は、要素に含まれる属性を使用する。選択結果に対して、次の射影と同じ操作、つまり、不要な属性の値を未定義とする値 (undefined) に値入換え操作で書き換える。ただし、射影同様に冗長な要素を除く機

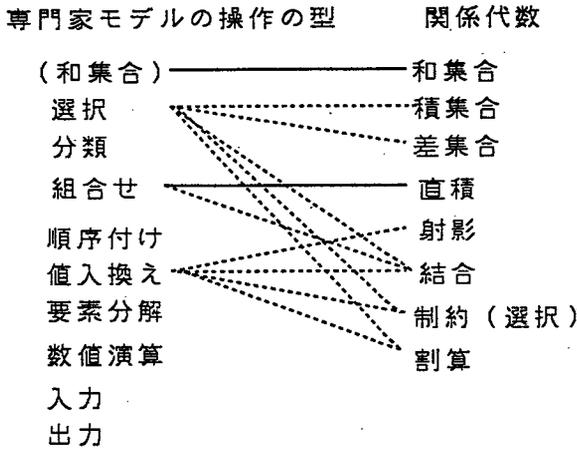


図 7.1: 専門家モデルの操作の型と関係代数の対応

能は専門家モデルにはない。

関係代数との比較を総合的に見ると、積集合演算と差集合演算、また、冗長な要素を除く機能が専門家モデルには欠けていることが分かった。逆に、データベース操作以外の数値演算操作、入力操作、出力操作と要素分解操作を専門家モデルでは供給している。

関係代数と専門家モデルの機能の対比は図 7.1 のようになる、実線は対応を、破線は部分的対応を表す。

#### 7.4 設計問題に対する操作の型の追加

診断問題に代表される解析型の問題は、予め与えられた解の空間から効率的に解を選び出す問題である。これに対して、設計問題に代表される合成型の問題では、解の空間は問題解決過程で作られるものであり、与えられる情報は、解が満足すべき制約や部分要素である。解析型問題は、解空間が固定されているので、外延的な要素集合を扱う操作で解ける。しかし、合成型問題の解空間は、組合せ的または制約から生成されるため大きい場合が多い。この問題の知識を表現するには、制約などを表せる内包的

な集合を扱う操作が必要である、また、大きな解空間の探索のために問題解決の効率化も重要である。

設計問題での評価から追加の必要な操作の型は生成操作であることが判明した。生成は、初期値、変分量（増減量）、終了条件（終了値）を評価基準として持ち、生成すべき値（属性値）を計算する。たとえば、

属性：長さの値が制約  $\{x \mid 1 \leq x \leq 20 \text{ かつ } x \text{ は整数}\}$

で表されるときに、属性：長さの値の生成の基準が

（初期値：1、変分量：1 終了条件：20 より大きい場合に終了）で表現できる。言い替えると、内包的な制約から外延的な集合を生成する操作である。

関係代数との比較および、問題解決の効率からの操作の追加すべきものに、積集合と差集合の実現と冗長要素の除去操作がある。

## 7.5 操作の型の再構成

ここでは、操作の型を集合の操作の観点から再構成について論じる。操作は、要素の比較の観点を分類すると次の4つになる。

- (1) 2つの要素集合の要素間の比較
- (2) 1つの要素集合の要素間の比較
- (3) 1つの要素集合の1つの要素の属性間の比較
- (4) 1つの要素集合の1つの要素と操作の評価基準の属性間の比較
- (5) 要素集合を持たないもの

(1) は集合間の演算であり要素間の比較を行う、要素の持つ属性は比較しない。和集合、積集合、差演算、直積の4つの集合演算が代数学では定義されている。

そこで、「積集合」「差演算」を新たに操作の型に追加する。「和集合」の処理は、従来通り入力要素集合の OR 関係で実現する。専門家モデルでは、直積をそのままは利用しないが、それに似た「組合せ操作」で対応する。

- (2) は1つの集合内での要素間の比較である。属性の比較までは行わない。この演算には、冗長な要素を除く「冗長除去」操作がある。この冗長要素の除去操作を操作の型に追加する。
- (3) は集合の1つの要素に注目して、その要素が持つ2つの属性間でその値を比較する。これは、専門家モデルでは、「選択（属性間比較）操作」である。また、値入換え操作でもこの機能を持っているが、重複を避けるため、値入換え操作は、選択（属性間比較）操作と代入操作に分離する。
- (4) は集合の各要素の特定の属性の属性値と操作が持つ評価基準の属性値集合との比較により、操作を行う演算である。これには、「選択（属性値評価）操作」「順序付け操作」がある。属性値の値は評価しないで属性を操作するものに「入力操作（値の代入操作の対話版）」「出力操作」がある。これに、値入換え操作を分解した「値（固定値）の代入操作」を追加する。また、異なる属性間で値を伝播させる「参照値の代入操作」を追加する。評価基準のない操作には、1つ要素を複数の要素に置き換える「要素分解操作」がある。
- (5) は要素を生成する操作である。生成される要素間は、すべて同じ属性を持つが、属性値が異なるものである。この他に、数値演算操作がある。

以上をまとめると評価基準を持つ操作の型が6種類と評価基準を持たない操作の型が6種類（この内1つは、操作外の演算である和集合）に整理される。

### 評価基準を持つ操作の型

1. 選択: 従来の選択操作と同じであり、1つの要素内の属性間の関係（関係は、評価基準で表される）を判定する「選択（属性間）」と1つの要素の特定の属性の属性値と評価基準の取り得る値を比較判定する「選択（属性値評価）」がある。
2. 値の代入操作: 従来の入力操作と値の入換え操作を見直した操作である。1つの要素の特定の属性に値を代入する。代入する値が固定の場合、他の属性の値の場合と対話による値入力の場合から「値（固定値）の代入操作」「参照値の代入操作」「入力」の3種類がある。
3. 出力: 従来の出力操作と同じであり、メッセージと属性値を対話装置に出力する。ただし、出力要素集合に入力要素集合と同じ要素を出力する操作だけでなく、指定された属性以外の属性を削除した要素を出力する操作を追加した。
4. 生成: 新しく追加された操作であり、特定の属性の属性値を生成する。出力要素集合には、順次、その属性に新しい属性値をもつ要素が追加される。
5. 数値演算: 従来の数値演算操作と同じであり、1つの要素に対して、単一または複数の属性の値を使って演算を行いその結果を指定した属性に代入する。

6. 順序付け：従来の順序付け操作と同じであり、入力要素集合全体について、特定の属性に注目してその属性値の優先順に要素を並べて、順序付きの出力要素集合を作る。

#### 評価基準を持たない操作の型

1. 組合せ：従来の組合せ操作と同じであり、2つの入力要素集合の各要素を組合せて新しい要素を生成する。集合の直積演算に相当する。
2. 要素分解：従来の要素分解操作と同じであり、1つの要素を低い水準の複数の要素に置き換える。
3. 冗長除去：新しい操作である。入力要素集合の中の同一の要素の冗長性を除去する。
4. 積集合：従来の選択操作の機能では一部しか実現できていなかった操作である。2つの入力要素集合の共通要素から新しい集合を出力要素集合として作る操作である。
5. 差集合：従来の選択操作の機能では一部しか実現できていなかった操作である。1つの入力要素集合の要素からもう一方の入力要素集合に含まれない要素から成る新しい集合を出力要素集合として作る操作である。
6. 和集合（操作外の演算）：従来から操作ではなく、操作に入力する要素集合のOR関係で実現されていた演算である。OR関係にある入力要素集合は和集合として扱われる。

#### 7.5.1 従来の操作の型との相違点

分類操作は、複数の選択操作（AND 実行関係）で実現できるので省略した。値入換え操作は、属性値の条件判断（選択操作）と値（固定値）

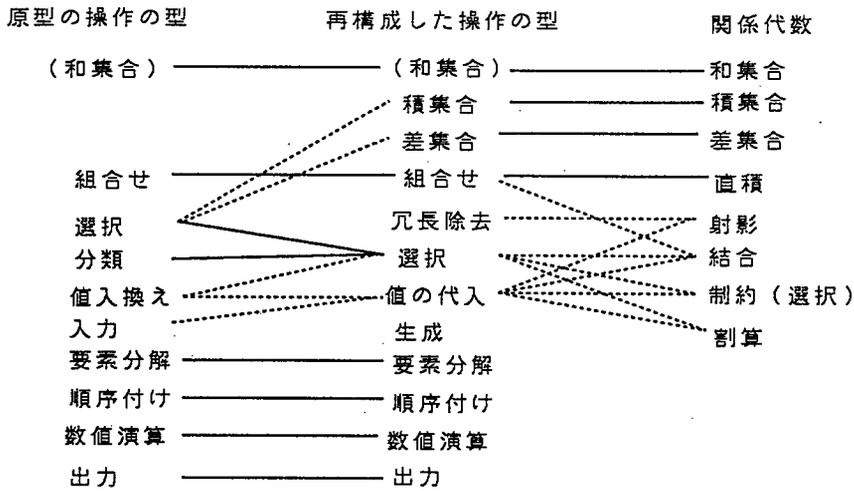


図 7.2: 原型および再構成した操作の型と関係代数の対応

の代入に分解した。従来属性値判断の選択に加えて、集合間の要素比較を行う積集合操作と差集合操作を追加した。また、冗長要素の除去の操作を追加した。選択（属性値評価）操作では、複数属性の評価を廃止して、単一属性の評価のみに変更した。これは、選択操作を直列につなぐことが、選択基準を AND 結合したものに等しいからである。この選択基準の表現から否定の表現を廃止した、否定は元の集合から肯定的な要素を持つ集合を差集合により生成することで実現できる。

値の代入操作では、属性間の値を伝播する機能（参照代入）を追加した。合成型問題に対応するため生成操作を追加した。従来の専門家モデルの操作と新しい操作の型の対応関係を図 7.2 に示す。

## 7.6 結言

専門家モデルの操作の型を関係代数との比較および、設計問題での知識獲得支援システム EPSILON/One の評価結果から再構成した。この新しい操作の型での知識獲得の抽出が従来の操作の型に勝っているとは言え

ないが、不足していると型の追加と重複の除去を行ったことにより、操作推定や洗練に対してそれらの処理アルゴリズムの改善に役立つと考えている。



## 第 8 章

### 操作を用いた知識獲得支援の総合評価

#### 8.1 緒言

本研究の知識獲得支援は、まず「専門家モデルによる問題解決の知識獲得」を行い、次に「その問題解決過程での対象領域の知識獲得」を行う。前者の問題解決の過程を表現するのに、他の問題解決の表現と比較して操作の表現が適切であるかを評価する。後者の知識獲得を特定の問題解決過程を固定した知識獲得支援システムと比較することで知識獲得支援機能を評価する。

#### 8.2 知識獲得支援における知識表現の水準

専門家モデルの操作はプロダクションルールの分析から決定されたルールの機能に近い比較的小さな単位の問題解決の構成要素である。これに対して、汎化タスクは比較的まとまった単位の問題解決の構成要素である。問題解決を定めた方が、その問題解決に利用する対象領域の知識を詳細に定義できるため詳細な知識獲得支援が行える。

しかしながら、色々な問題に対応するには、数多くの汎化タスクを準備する必要がある。あるいは、汎化タスクを修正できる機能が必要である。また、汎化タスクごとに知識獲得支援の方式を開発する必要もある。専門

家モデルの利点は操作の型を限定することで、操作対応の知識の推定と洗練の方式を決定できることである。

一般的な問題解決の枠組として「生成・検査・修正のモデル」がある。この3つの問題解決の基本要素を用いて知識獲得支援を行うことを考える。このモデルの基本要素は3つであるが、各要素はさらに詳細化できる。「定められた集合から候補を選ぶこと」、「いくつかの値から数値計算により、候補の値を計算すること」、「少しずつ値を変更して次々に値を生成する」などに、生成は細分化できる。つまり、この細分化した生成機能ごとに、知識獲得の対象の知識は異なる。「生成・検査・修正のモデル」の問題解決の機能を知識獲得に利用できる水準に詳細化すると、専門家モデルの操作の型と同じ水準での表現が必要と考えられる。生成は、操作の型の「選択」、「順序付け」、「組合せ」、「数値演算」、「要素分解」と「値の入換え」に対応する（第7章で説明した「生成（少しずつ値を変更しながら候補を生成する）」も対応する）。修正は、機能的な表現というよりも役割的性格のもので機能的には生成と同等である。生成が独立した入力要素集合を対象にできるのに対して、修正の入力要素は検査の結果に問題のある解（要素）を対象にしているため、これは、「選択」、「数値演算」および「値の入換え」といった操作の型に対応する（第7章で説明した「生成」も対応する）。検査には、操作の型の「選択」と「分類」が対応する。操作の型の選択には、候補の集合から候補を選ぶ（候補を生成する）という役割と条件に合わない要素を除く（選択しない）という役割とがある。

### 8.3 知識獲得支援における問題解決のモデルの役割

特定の問題解決のモデルを定めるとその問題解決に必要な対象領域の知識を詳細に定義できる。対象領域の詳細な定義を利用することによって

効率的な知識抽出と洗練が行える。一方、本研究の方式は問題解決のモデルを固定していないので、広い範囲の問題解決に対応できる。しかしながら、対象領域の知識を獲得する場合には前者に比べて一般的な方法での知識抽出と洗練を行わざるをえない。本節では、対象領域の知識獲得において、この2つ方法の差異を議論する。比較対象は、分析型の問題解決のモデルを利用する ETS と MORE（第2章で説明したシステム）、及び、MOLE とする。

一般的に、知識抽出と洗練の過程では、対象領域の基本的な要素を抽出し、次に、属性と属性値を抽出する。その後、要素、属性と属性値を洗練する。

#### 1. ETS の対象領域の知識抽出と洗練過程

ETS の問題解決のモデルは「与えられた特徴による項目の順序付け」であり、対象領域の知識は、格子型知識表現（項目と特徴、および、項目と特徴との関係度）である。

ETS は項目（要素）の抽出には特別な支援を行わない。また、ETS は項目からパーソナルコンストラクト心理学により、項目を分類する特徴（属性）を抽出する。そして、各項目と特徴間の関係度（属性値に相当）を抽出する。しかし、関係度抽出には特別な支援を行わない。

ETS の知識の洗練方法は、既に獲得した知識（格子型知識表現）を別の形式で専門家に提示することで、専門家が不備に気づいて自ら修正する方法をとっている。導出グラフは項目（要素）と項目（要素）の関係の強さを提示する。これにより、項目（要素）間関係に同意できない（専門家が不備を検出した）場合には、反例となる項

目（要素）の追加、特徴（属性）の追加や関係度（属性値）の変更を専門家が行う。要素間の関係の強さの確認は類似分類木でも行える。

## 2. MORE と MOLE の対象領域の知識抽出と洗練過程

MORE の問題解決のモデルは "cover-and-differentiation"、つまり、可能な仮説をできるだけ求めて、その中から仮説を特定する問題解決である。対象領域の知識は、ドメインモデルである。

MORE の 8 つの洗練戦略には、次の 2 種類がある。1 つは、ドメインモデルを構成する知識を抽出するために、仮説の頻度条件（属性と属性値）、徴候の頻度条件（属性と属性値）、検査（要素）と検査の条件（属性と属性値）の抽出を行う。もう 1 つは、徴候から仮説を特定するのに不足している知識の検出を目的としている。抽出されるのは、新たな徴候（要素）と徴候の属性（属性値を含む）が中心である。新たな徴候の抽出には、3 つの戦略 (differentiation, symptom distinction, path division) がある。この不備の検出はシステムが行ない、専門家が徴候または、徴候の属性を追加する。

MOLE は、MORE の洗練に加えて、要素が仮説か徴候かを仮定する機能、確信度を仮に決める機能、確信度の自動修正、専門家の診断結果に合ない推論結果を導くドメインモデルの仮説や徴候の部分の検出とその不備の洗練方法の提案機能といった特徴を持つ。

## 3. 操作を用いた知識獲得での対象領域の知識抽出と洗練過程

本研究の方式は、要素集合名に属する要素の抽出には特別な支援は行わないが、問題解決の知識（操作の評価属性）により各要素の属性の抽出を支援する。具体的な属性を問題解決の知識から導く点は、問題解決のモデルを固定している場合に対象領域の知識の属性が既知であることに相当する。属性値も同様にその候補を操作の評価基準から導くことができる。本研究の方式は、評価基準の属性とその取り得る値（属性値）、および、要素を洗練する。洗練の戦略は、属性と属性値の一般化と特殊化を行うことが中心である。

ドメインモデルと専門家モデルを比較するには、「ドメインモデルの仮説と徴候を要素に対応させる」「仮説を要素に、徴候を仮説の属性に対応させる」「仮説を要素に、徴候を選択操作の評価基準に対応させる」の3通りが考えられるが、ここでは、2つ目の対応関係で比較する。また、ドメインモデルでは、仮説、徴候、徴候の属性と検査、及び、それぞれの条件を扱うが、ここでは、仮説、徴候、徴候の属性に関して比較を行う。

まず、静的な洗練での不備の検出を比較する。ETSでは、専門家が特徴の関係をみることで、分類に役だっていない特徴を検出する。MOREやMOLEは仮説を特定する知識が不足していることをシステムが検出する。本研究の方式では、分類操作であれば分類基準（ETSの特徴と関係度、ドメインモデルの徴候と徴候の属性）の不備をシステムが検出する。この不備の解消では、ETSは専門家に修正（反例の項目の追加、特徴の追加、関係度の修正）を委ねている。MOREとMOLEも同様に徴候と徴候の属性を追加を専門家に委ねている。本研究の洗練では、操作の評価基準と要素に対して、属性や属性値の削除と一般化の候補は提案できるが、特殊化については、専門家

に委ねている。しかし、新たな例（要素）の追加がある場合には、操作推定で評価基準の属性や属性値の追加を提案できる。

動的な洗練では、MOLE は専門家の診断結果とシステムの診断結果が適合するように確信度の修正を行い、修正しきれない場合に徴候と仮説の追加を専門家に質問する。この方式は、TEIRESIAS のルールの洗練と同様に、導かれた仮説と導かれるべきでない仮説に注目して、その条件（徴候）の追加や変更を専門家に促す。本研究の方式では、選択と分類操作の動的な洗練において、選択されるべき要素が選択できない場合には、評価基準を一般化する質問を行う。この際に、一般化は属性と属性値の両方に適用でき、操作推定を利用することで、属性と属性値を提案できる。選択されるべきでない要素が選択された場合には、評価基準を特殊化する質問を行う。操作推定でこの基準を推定できない場合には、新たな属性が必要であることが分かる。この場合には、専門家にその属性の追加を促すことになる。また、操作推定の結果を専門家に提示することで、ETS と同様に、反例（要素）の抽出支援も行える。すでに、要素に関して多くの属性が分かっている場合には、本研究の方式では、必要な属性を操作推定により選ぶことができるが、ETS、MORE と MOLE では、事前に準備されている特徴や徴候の中から必要な知識を選ぶ支援はできない。

洗練の内容はいずれのシステムも非常に類似している。問題解決のモデルを固定している知識獲得支援システムでは、具体的な概念名（特徴、徴候など）を使って、質問できるが、本研究の方式では、要素、属性、属性値といった一般的名称で質問することになるので、連想作用は前者と比較すると弱い。

## 8.4 結言

知識獲得支援に適した知識表現の評価では、問題解決の知識を獲得する必要がある問題に対しては、「生成・検査・修正のモデル」と比較して、本研究の「専門家モデル」が適切な水準の問題解決の知識表現であると言える。

獲得対象には、問題解決モデルが分かっているものと不明なものがあるため、準備されている汎化タスクが適用できる問題には、汎化タスクによる知識獲得を行い、問題解決の知識の不明確な部分では「専門家モデル」の操作で知識獲得するような、汎化タスクと専門家モデルの長所を補完的に利用する知識獲得支援システムも考えられる。

操作を用いた知識獲得支援でも操作の評価基準の属性を利用することで、問題解決を固定した知識獲得支援システムに対応した対象領域の知識獲得を効率的に行える。しかしながら、要素の抽出と洗練においては、要素のより具体的な概念名で質問を生成できない点が劣っている。専門家モデルの要素に型を持たせることが新たな研究課題である。



## 第 9 章

### 結論

本論文は知識獲得支援システムにおいて、次の 4 項目の新しい研究成果を示した。

1. 問題解決の知識を操作の集合と操作間の関係で表現する専門家モデルを提案した。この専門家モデルの操作の型は、実際の知識ベースの分析から得られた 7 つの基本型を持つものである。知識獲得におけるこの知識表現の有効性は知識抽出方法（プリポスト法）と共に知識獲得支援システム EPSILON/One の開発と評価により確認した。また、操作の型の充分性は、関係代数との比較により確認した。
2. 専門家モデルの操作の型を利用した知識抽出の方法としてプリポスト法を提案した。プリポスト法は、「連想法」、「知識の整理結果の提示による知識抽出法」と「操作の型を利用した効率的な質問の生成方法」を合せ持つ知識抽出方法である。この方法の有効性は処理系の開発と評価により確認した。
3. 対話型の知識獲得において、一般的な知識だけでなく例から帰納的に知識の候補生成を行う方法を提案した。この知識の生成方法は、操作の型を利用して、帰納的一般化の欠点である候補生成過多問題を緩和して効率的に知識の候補を求めることができる。また、この

知識の生成機能を持つ知識抽出方法の実現可能性をその処理系の開発により確認した。

4. 知識の洗練に対して、操作の型の情報を利用することで、知識の役割に注目した知識の洗練方法を提案した。知識の不備を操作の型に基づいて集合操作の観点から定義し、その検出と解消の戦略を示した。

以上の4項目は知識獲得支援の研究の発展に充分貢献できるものとする。

## 謝辞

本研究をまとめるに当たり、懇切丁寧なる御指導と御鞭撻を賜わりました大阪大学基礎工学部制御工学科 辻 三郎教授に衷心より感謝の意を表しますとともに、厚く御礼申し上げます。

本論文をまとめる過程で、適切な御指導と御助言をいただきました大阪大学産業科学研究所 北橋 忠宏教授、溝口 理一郎教授、安部 憲広助教授に心から感謝いたしますとともに、深く御礼申し上げます。

本研究の機会を頂いた歴代の ICOT 第5研究室長 岩下 安男室長、藤井 裕一室長、生駒 憲治室長ならびに、ICOT 古川 康一次長、淵 一博所長に感謝いたします。ICOT のワーキンググループで貴重な御意見を頂いた大阪大学 溝口理一郎教授、大阪大学 安部 憲広助教授、東京工業大学 小林 重信教授、京都大学 片井 修助教授、京都大学 樫木 哲夫博士、電子総合技術研究所 諏訪 基博士ならびに 各委員に感謝いたします。

本研究に協力頂いた ICOT 第5研究室 椿 和弘研究員、EPSILON/One の処理系の実現に協力頂いた (財) 日本情報処理開発者協会 大崎 宏氏に感謝いたします。本研究にコメント頂いた以下の ICOT 研究員 井上 克己氏、佐藤 健氏、有馬 淳氏、寺崎 智氏、永井 保夫氏、岡 夏樹氏、招聘研究員の Marc Linster (GMD) に感謝いたします。

本論文の作成に激励と協力を頂いた大阪大学 山田 誠二博士と三菱電機 (株) 情報電子研究所 AI グループマネージャ 辻 秀一博士に感謝いたします。



## 参考文献

- [Abe 86] 安部, 滝 (共著) : エキスパートシステム入門, 共立出版 (1986).
- [Araya 87] Araya, A. A. and Mittal, S.: Compiling Design Plans from Descriptions of Artifacts and Problem Solving Heuristics, Proceedings of the 10th IJICAI (1987).
- [Bennett 85] Bennett, J.: ROGET: A Knowledge-Based System for Acquiring the Conceptual Structure of a Diagnostic Expert System., Journal of Automated Reasoning 1, 49-74 (1985).
- [Boose 86] Boose, J. H.: A Knowledge acquisition program for expert systems based on personal construct psychology, J. of Man-Machine Studies, Vol.23, pp.495-525 (1986).
- [Boose 87] Boose, J.H., et al.: Expertise transfer and complex problem: using AQUINAS as a knowledge-acquisition workbench for knowledge-based systems, J. of Man-Machine Studies, Vol.26, No.1, pp.3-28 (1987).
- [Breuker 89] Breuker, J. and Wielinga, B: Models of Expertise in Knowledge Acquisition, Topics in Expert System Design (Guida, G. and Tasso, C. eds.), pp.265-295 (1989).

- [Brown 86] Brown, D. C. and Chandrasekaran, B.: Knowledge and Control for Mechanical Design Expert System, IEEE Computer, pp.92-100, July (1986).
- [Brownston 85] Brownston, L., et al.: Programming Expert System in OPS5, Addison-Wesley (1985).
- [Buchanan 69] Buchanan, B.G., Sutherland, G.L. and Feigenbaum, E.A.: Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. In B.Meltzer and D.Michie, eds., Machine intelligence, vol.4. Edinburgh: Edinburgh Univ. Press, pp.209-254 (1969).
- [Buchanan 83] Buchanan, B.G. and Shortliffe, E.H.: Rule-based expert systems: The MYCIN experiments of Heuristic Programming Project, Reading, Massachusetts: Addison-Wesley (1983).
- [Buchanan 78] Buchanan, B.G. and Feigenbaum, E.A.: DENDRAL and Meta-DENDRAL: Their applications dimension. Artificial Intelligence 11, 5-24 (1978).
- [Bylander 86] Bylander, T. and Mittal, S.: CSRL: A Language for Classification Problem Solving and Uncertainty Handling, The AI Magazine, pp.66-77 (1986).
- [Chandrasekaran 86] Chandrasekaran, B.: Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design, IEEE Expert, pp.23-30, Fall (1986).

- [Chiang 87] Chiang, T. Y-L. and Brown, D. C. : DSPL ACQUIRER - A System for ACquisition of Routine Design Knowledge, AI in Engineering: Tools and Techniques, pp.95-110 (1987).
- [Chikayama 84] Chikayama, T.: ESP Reference Manual, ICOT Research Center (1984).
- [Chikayama 88] Chikayama, T., Sato, H. and Miyazaki, T.: Overview of the parallel inference machine operating system(PIMOS), Proc. of the International Conference on Fifth Generation Computer Systems, pp.230-251 (1988).
- [Codd 70] Codd, E.F.: A Relational Model of Data for Large Shared Data Banks, CACM, vol.13, No.6, pp.377-387 (1970).
- [Davis 79] Davis, R.: Interactive transfer of expertise: acquisition of new inference rules, Artificial Intelligence, vol.2, pp.121-157 (1979).
- [de Kleer 86] de Kleer, J.: An Assumption-based TMS, Artificial Intelligence 28, pp.127-162 (1986).
- [Doyle 79] Doyle, J: A Truth Maintenance System, Artificial Intelligence, Vol. 24 (1979).
- [Eshelman 86] Eshelman, L. and McDermott, J.: MOLE: A Knowledge Acquisition Tool that Uses its Head, Proc. of AAAI-86, pp. 950-955 (1986).
- [Gaines 81] Gaines, B.R. and Shaw, M.L.G.: New directions in the analysis and interactive elicitation of personal construct systems, Shaw,

- M.L.G. eds. Recent Advances in Personal Construct Technology, London: Academic Press, pp.147-182 (1981).
- [Genesereth 86] Genesereth, M. and Nilsson, N. : Logical Foundations of Artificial Intelligence, Los Altos, CA: Morgan-Kaufman (1986).
- [Ginsberg 85] Ginsberg, A., Weiss, S.M. and Pilitakis, P.: SEEK2: A Generalized Approach to Automatic Knowledge Base Refinement, Proc. of IJICAI-85, pp.367-374 (1985).
- [ICOT-JIPDEC 87] ICOT-JIPDEC AI センター: "知的情報処理システムに関する調査研究報告書 - 知識システム開発方法論 -" (1987).
- [Inoue 88] 井上, 他: 工作機械の設計手法の解析 - 旋盤の回転機能部品の設計 -, ICOT Technical Memorandum No.494 (1988).
- [Ishida 87] 石田 好輝: 定性推論を利用したプロセス診断知識獲得, 日本ソフトウェア科学会第4会大会論文集, D-4-1 (1987).
- [Kahn 85] Kahn, G., et al.: Strategies for Knowledge acquisition, IEEE Trans. on PAMI, Vol. PAMI-7, No.5, pp.511-522 (1985).
- [Kawakita 67] 川喜多 次郎: 発想法, 中央公論社 (1967).
- [Kawaguchi 87] Kawaguchi, A., Mizoguchi, R., Yamaguchi, T. and Kakusho, O.: SIS: A Shell For Interview Systems, Proc. of IJICAI '87, vol.1, pp.359-361 (1987).
- [Kawaguchi 88] 川口, 溝口, 角所, 山田, 野村, 馬場, 中島: 設計時の仕様獲得を支援する知的インタビューシステム I2S/D- 油圧回路の設計に関し

- て-, 情報処理学会, 知識工学と人工知能研究会資料, 59-16, pp.139-146 (1988).
- [Kelly 55] Kelly, G. A.: The Psychology of Personal Constructs, New York: Norton (1955).
- [Klinker 86] Klinker, G., Bentolia, J., Genetet, S., Grimes, M. and McDermott, J.: KNACK-Report-Driven Knowledge Acquisition, Proceedings of the Knowledge Acquisition for Knowledge-Based Systems Workshop, 23-0 (1986).
- [Kobayashi 88a] 小林 重信: 「学習工学」の確立に向けて, 計測自動制御学会, 知識工学部会, 知識獲得と学習に関する特別講演会資料 (1988).
- [Kobayashi 88b] 小林: 知識システム技術の現状と将来, 計測と制御, Vol.27, No.10, pp.859-868 (1988).
- [Laird 84] Laird, J., Rosenbloom, P.S. and Newell, A.: Towards chunking as a general learning mechanism. Proceedings of AAAI-84, Austin (1984).
- [Lenat 86] Lenat, D., Prakash, M. and Shepard, M.: CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks, The AI Magazine, 6(4) (1986).
- [Linster 89] Linster, M.: Report on my Visit to ICOT in August 1989, ICOT guest researcher report, (1989).
- [Marcus 87] Marcus, S: Taking backtracking with a grain of SALT, Int. J. Man-Machine Studies, vol.26, pp.383-398 (1987).

- [Matsumura 90] 松村: 代数学, 数理科学ライブラリー, No.8, 朝倉書店 (1990).
- [Mitchell 78] Mitchell, T.M.: Version Spaces: An Approach to Concept Learning, Stanford Technical Report STAN-CS-78-711, HPP-79-2 (1978).
- [Mitchell 85] Mitchell, T., Mahadevan, S. and Steinberg, L.: LEAP: A Learning Apprentice for VLSI Design, Proceedings of the 9th IJCAI, Los Angeles (1985).
- [Mitchell 86] Mitchell, T., Keller, R. and Kedar-Cabelli, S.: Explanation-Based Generalization: A Unifying View, Machine Learning 1, January (1986).
- [Mitsubishi 88] MELCOM PSI- II プログラミング and オペレーティングシステム 総合説明書, 三菱電機 (株) , (1988).
- [Miyazaki 88] 宮崎: 並列論理型言語 KL1 の実現方式と並列 OS の記述, 電子情報通信学会論文誌 D, Vol. J71-D No.8, pp.1423-1432 (1988).
- [Mizoguchi 88a] 溝口, 角所: 知識獲得支援システム, 人工知能学会誌, Vol.3, No.6, pp.732-740 (1988).
- [Mizoguchi 88b] 溝口, 滝, 他: 知識獲得支援システム (1988 年度 KSA-KAS-SUBWG 報告書), ICOT Technical Memorandum No.654 (1988).
- [Mizoguchi 89] 溝口, 滝, 安部, 長沢, 他: 設計問題向き知識獲得支援システムの構想, ICOT Technical Memorandum No.747 (1989).

- [Musen 86] Musen, M., Fagan, L., Combs, D. and Shortliffe, E.: Using A Domain Model to Drive an Interactive Knowledge Editing Tool, Proceedings of Knowledge Acquisition for Knowledge-Based Systems Workshop, 33-0 (1986).
- [Nagai 89] 永井, 滝, 寺崎, 横山, 井上: 設計問題向けツール・アーキテクチャ, 人工知能学会誌, Vol.4, No.3, pp.297-303 (1989), also in ICOT Technical Report No.455 (1989).
- [Nippon-seikou 86] 転がり軸受 (カタログ), 日本精工株式会社, (1986).
- [Osaki 88] 大崎, 滝, 椿: 知識獲得支援システム EPSILON/One(2), 第8回 知識工学シンポジウム資料, 計測自動制御学会 (1988), also in ICOT Technical Memorandum No.601 (1988).
- [Osaki 89] 大崎, 椿, 滝: 知識獲得支援システム EPSILON/One 操作マニュアル, ICOT Technical Memorandum NO.746 (1989).
- [Politakis 84] Politakis, P. and Weiss, S.M.: Using Empirical Analysis to Refine Expert System Knowledge Bases, Artificial Intelligence, Vol. 22, pp. 23-48 (1984).
- [Quinlan 83] Quinlan, J. R.: Learning Efficient Classification Procedures and Their Application to Chess End Games, in Machine Learning: An Artificial Intelligence Approach, R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), Tioga, Palo Alto, Calif. (1983).
- [Quinlan 86] Quinlan, J. R.: Induction of Decision Trees, Machine Learning 1: pp.81-106 (1986).

- [Sawamoto 88a] 澤本, 小林, 石塚, 滝, 他: 解析型知識システムにおける知識獲得支援 (1987年度 KSS-WG 報告書 No.1), ICOT Technical Memorandum No.581 (1988).
- [Sawamoto 88b] 澤本, 石塚, 他: 合成型知識システムにおける知識獲得支援 (1987年度 KSS-WG 報告書 No.2), ICOT Technical Memorandum No.588 (1988).
- [Shapiro 81] Shapiro, E. Y.: Inductive Inference of Theories from Facts, Technical Report 192, Yale University Computer Science Dept (1981).
- [Shinohara 86] 篠原 靖志, 寺野 隆雄: 関係の階層化を利用した知識ベース構築支援システム, 電子通信学会 (人工知能と知識処理研究会資料) AI86-32 (1986).
- [Suwa 82] Suwa, M., Scott, A.C. and Shortliffe, E.H.: An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System, the AI Magazine, Fall, pp.16-21 (1982).
- [Suwa 86] 諏訪, 小林, 國藤, 岩下: エキスパートシステム開発事例による知識獲得の諸相, 計測と制御, Vol.25, No.9, pp.801-809 (1986).
- [Taki 85] Taki, H. and Sakaue, Y.: A Programming System for Intelligent Robots with Force Feedback (Skill Acquisition), ROBOTS9 Conference Proceedings(SME), Vol.1, pp.11-29 - 11-41 (1985).
- [Taki 87a] 滝, 椿, 岩下: 知識獲得支援システム (EPSILON) における専門家モデル, 情報処理学会, 知識工学と人工知能研究会報告 52-4, pp.25-31 (1987), also in ICOT Technical Report No.253 (1987).

- [Taki 87b] Taki, H., Tsubaki, K. and Iwashita, Y.: Expert Model for Knowledge Acquisition, Proc. of IEEE 3rd Annual Expert Systems in Government Conference, pp. 117-124 (1987), also in ICOT Technical Report No.254 (1987).
- [Taki 87c] 滝, 椿, 藤井: 知識獲得のための専門家モデル-知識獲得支援システム EPSILON/One について, SICE 3 部会 合同研究会 資料, pp.297-304 (1987), also in ICOT Technical Report No.272 (1987).
- [Taki 87d] 滝, 椿, 藤井: EPSILON/EM: 専門家モデルを用いた知識獲得支援システム, 情報処理学会, 第 35 回全国大会資料, 1N-3, pp.1701-1702 (1987), also in ICOT Technical Report No.276 (1987).
- [Taki 88a] Taki, H.: Knowledge Acquisition By Observation, Proc. of The International Conference on Fifth Generation Computer Systems, Vol.3, pp.1250-1258 (1988), also in ICOT Technical Report No.387 (1988).
- [Taki 88b] 滝: 解釈型知識獲得システム, 情報処理学会, 知識工学と人工知能研究会資料, 56-5, pp.33-40 (1988), also in ICOT Technical Memorandum No.425 (1987).
- [Taki 88c] 滝, 椿: 知識獲得支援システムの技術動向, ICOT Technical Memorandum No.483 (1988).
- [Taki 88d] 滝: プロトコル解析の為の帰納的学習, 情報処理学会, 知識工学と人工知能研究会資料 (88-AI-59), 59-11, pp.95-102 (1988), also in Inductive Learning for Protocal Analysis, ICOT Technical Memorandum No.512 (1988).

- [Taki 88e] 滝: 並列推論におけるヒューリスティックス知識, ICOT Technical Memorandum No.580 (1988).
- [Taki 88f] 滝: 知識獲得支援技術, 計測と制御, Vol.27, No.10, pp.875-880 (1988), also in ICOT Technical Memorandum No.610 (1988).
- [Taki 89a] Taki, H. and Fujii, Y.: Operation Presumption: Knowledge Acquisition By Induction, Proceedings of Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems, pp. 34-48 (1989), also in ICOT Technical Report No.450 (1989).
- [Taki 89b] 滝: プロダクションルールの機能的側面, 計測自動制御学会, 第9回 知識工学シンポジウム資料, pp.95-98 (1989), also in ICOT Technical Report No.459 (1989).
- [Taki 89c] 滝: 知識獲得支援システム EPSILON/One, 第7回 第5世代コンピュータに関するシンポジウム 予稿集, pp. 17-18 (1989).
- [Taki 89d] 滝, 椿: 仮説を含む説明構造の生成による仮説的知識の獲得, 人工知能学全国大会 (第3回) 論文集, Vol.1, 2-12, pp.123-126 (1989).
- [Taki 89e] Taki, H.: Knowledge Acquisition by Abductive Explanation, Proc. of SICE'89, ESS3-5, pp.1381-1384(1989).
- [Taki 89f] 滝: 知識コンパイル, ジェネリックタスク, ICOT Technical Memorandum No.704 (1989).
- [Taki 89g] Taki, H.: Knowledge Acquisition By Abductive And Inductive Explanation, Proc. of the 4th AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop, pp.34-1 - 34-19 (1989)

- [Taki 90a] 滝, 椿: 知識獲得のための知識表現『専門家モデル』, 人工知能学会誌, Vol.5, No.2 (1990), also in ICOT Technical Report No.510 (1989).
- [Taki 90b] Taki, H. and Terasaki, S.: A Proposal Guided Knowledge Acquisition Support System, Proc. of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, pp.315-330 (1990), also in ICOT Technical Report No. 541 (1990).
- [Taki 90c] Taki, H. and Terasaki, S.: Knowledge Acquisition by Inductive Operation Presumption, ICOT Technical Report No. 546 (1990).
- [Taki 90d] 滝, 寺崎: 提案誘導型知識獲得支援システム, 日本ソフトウェア科学会, 知識プログラミング研究会資料 KP90 (1990), also in ICOT Technical Memorandum No. 866 (1990).
- [Taki 90e] 滝, 寺崎: オペレーション推定のためのKL1による並列帰納推論プログラム, ICOT Technical Memorandum No.878 (1990).
- [Taki 90f] 滝: 対話による知識獲得, ICOT Technical Memorandum No.880 (1990).
- [Taki 90g] 滝: 知識の役割に注目した知識のリファイン, 人工知能学会, 第14回, 知識ベースシステム研究会資料, (1990).
- [Terasaki 88] 寺崎, 滝, 他: 機械設計支援システム構築ツール MECHANICOT, 人工知能学会, 知識ベースシステム研究会資料, SIG-KBS-8803 (1988), also in ICOT Technical Report No.424 (1988).
- [Tijerino 89] Tijerino, Y.A., Mizoguchi, R., Kawaguchi, A., Kakusho, O. and Kitahashi, T.: A Generic Vocabulary And Generic Task Li-

- brary For Use In MULTIS, 人工知能学会研究会資料、SIG-KBS-8805-3, pp.21-30 (1989).
- [Tsubaki 87] 椿, 滝: 解析型知識獲得メタシステムの構想, ICOT Technical Memorandum No. 409 (1987).
- [Tsubaki 88a] 椿, 滝: 知識獲得支援システム CTAS2 の構想, ICOT Technical Memorandum No.511 (1988).
- [Tsubaki 88b] 椿, 滝, 大崎: 知識獲得支援システム EPSILON/One(1), 第8回知識工学シンポジウム資料, 計測自動制御学会 (1988), also in ICOT Technical Memorandum No.600 (1988).
- [Tsubaki 88c] 椿, 滝, 藤井, 山崎: 類型タスク構造に基づく知識獲得, 計測自動制御学会, 「知識獲得と学習に関する特別講演会」資料 (1988), also in ICOT Technical Report No.360 (1988).
- [Tsubaki 89a] 椿, 滝, 他: CTAS の評価のまとめ, ICOT Technical Memorandum No.669 (1989).
- [Tsubaki 89b] 椿, 滝, 大崎: 知識獲得支援システム EPSILON/One, 人工知能学会全国大会 (第3回) 論文集, Vol.2, 12-5, pp. 719- 722 (1989).
- [Uchihashi 88] 打橋, 滝, 堀内: 対話型 EBL システム, 情報処理学会第39回全国大会講演論文集, vol.1, 6C-8, pp.351-352 (1988), also in EBG 実験システム, ICOT Technical Report No.471 (1989).
- [Uemura 79] 植村: データベースシステムの基礎, オーム社 (1979).
- [Ullman 82] Ullman, J.D.: Principles of DATA BASE SYSTEMS, Second Edition, Computer Science Press (1982).

- [van Melle 79] van Melle, W.: A domain-independent production-rule system for consultation programs, Proc. of IJICAI 6, pp.923-925 (1979).
- [Wada 88a] 和田, 他: エキスパートシステムの構築事例 (SHOOTX), ICOT Technical Memorandum No.490 (1988).
- [Wada 88b] Wada, S. and et al.: An Expert System Architecture for Switching System Diagnosis, ICOT Technical Memorandum No.491 (1988).
- [Yamaguchi 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.333-340 (1987).
- [Yamazaki 87] 山崎, 滝, 椿: 類型タスク構造に基づく知識獲得支援—分類問題向け知識獲得支援システム CTAS —, 人工知能学会第1回人工知能ツールと知識システム研究会 (1987).



## 研究業績

### 学会誌掲載論文

1. 滝, 椿: 知識獲得のための知識表現『専門家モデル』, 人工知能学会誌, Vol.5, No.2 (1990).
2. 滝: 知識獲得支援技術, 計測と制御, Vol.27, No.10, pp.875-880 (1988)
3. 永井, 滝, 寺崎, 横山, 井上: 設計問題向けツール・アーキテクチャ, 人工知能学会誌, Vol.4, No.3, pp.297-303 (1989)

### 国際会議発表論文

1. Taki, H., Tsubaki, K. and Iwashita, Y.: Expert Model for Knowledge Acquisition, Proc. of IEEE 3rd Annual Expert Systems in Government Conference, pp. 117-124 (1987)
2. Taki, H.: Knowledge Acquisition By Observation, Proc. of The International Conference on Fifth Generation Computer Systems, Vol.3, pp.1250-1258 (1988)
3. Taki, H. and Fujii, Y.: Operation Presumption: Knowledge Acquisition By Induction, Proceedings of Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems, pp. 34-48 (1989).

4. Taki, H.: Knowledge Acquisition By Abductive And Inductive Explanation, Proc. of the 4th AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop, pp.34-1 - 34-19 (1989)
5. Taki, H. and Terasaki, S.: A Proposal Guided Knowledge Acquisition Support System, Proc. of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, pp.315-330 (1990)
6. Taki, H. and Sakaue, Y.: A Programming System for Intelligent Robots with Force Feedback (Skill Acquisition), ROBOTS9 Conference Proceedings(SME), Vol.1, pp.11-29 - 11-41 (1985)

#### 研究会発表論文

1. 滝, 椿, 岩下: 知識獲得支援システム (EPSILON) における専門家モデル, 情報処理学会, 知識工学と人工知能研究会報告 52-4, pp.25-31 (1987).
2. 滝, 椿, 藤井: 知識獲得のための専門家モデル - 知識獲得支援システム EPSILON/One について, SICE 3 部会 合同研究会 資料, pp.297-304 (1987).
3. 椿, 滝, 大崎: 知識獲得支援システム EPSILON/One(1), 第 8 回 知識工学シンポジウム資料, 計測自動制御学会 (1988).
4. 大崎, 滝, 椿: 知識獲得支援システム EPSILON/One(2), 第 8 回 知識工学シンポジウム資料, 計測自動制御学会 (1988).
5. 滝, 寺崎: 提案誘導型知識獲得支援システム, 日本ソフトウェア科学会, 知識プログラミング研究会資料 KP90 (1990).

6. 滝: プロトコル解析の為の帰納的学習, 情報処理学会, 知識工学と人工知能研究会資料 (88-AI-59), 59-11, pp.95-102 (1988)
7. 滝: 知識の役割に注目した知識のリファイン, 人工知能学会, 第14回, 知識ベースシステム研究会資料, (1990)
8. 滝: プロダクションルールの機能的側面, 計測自動制御学会, 第9回知識工学シンポジウム資料, pp.95-98 (1989)
9. 滝: 解釈型知識獲得システム, 情報処理学会, 知識工学と人工知能研究会資料, 56-5, pp.33-40 (1988)

#### 口頭発表論文

1. 滝, 椿, 藤井: EPSILON/EM: 専門家モデルを用いた知識獲得支援システム, 情報処理学会, 第35回全国大会資料, 1N-3, pp.1701-1702 (1987)
2. 椿, 滝, 大崎: 知識獲得支援システム EPSILON/One, 人工知能学会全国大会 (第3回) 論文集, Vol.2, 12-5, pp. 719- 722 (1989).
3. 滝, 椿: 仮説を含む説明構造の生成による仮説的知識の獲得, 人工知能学会全国大会 (第3回) 論文集, Vol.1, 2-12, pp.123-126 (1989)
4. Taki, H.: Knowledge Acquisition by Abductive Explanation, Proc. of SICE'89, ESS3-5, pp.1381-1384(1989)
5. 滝: 知識獲得支援システム EPSILON/One, 第7回 第5世代コンピュータに関するシンポジウム 予稿集, pp. 17-18 (1989).
6. 打橋, 滝, 堀内: 対話型 EBL システム, 情報処理学会第39回全国大会講演論文集, vol.1, 6C-8, pp.351-352 (1988).

7. 山崎, 滝, 椿: 類型タスク構造に基づく知識獲得支援—分類問題向け知識獲得支援システム CTAS —, 人工知能学会第1回人工知能ツールと知識システム研究会 (1987).
8. 椿, 滝, 藤井, 山崎: 類型タスク構造に基づく知識獲得, 計測自動制御学会, 「知識獲得と学習に関する特別講演会」資料 (1988).

### 著書

安部, 滝 (共著) : エキスパートシステム入門, 共立出版 (1986)

## 付録 A

### 減速機的设计知識の獲得

工作機械の設計知識 [Inoue 88] を工作機械の設計エキスパートシステム開発者から獲得した例を示す。ただし、ここでは、その知識の一部の獲得を紹介する。知識獲得は知識獲得支援システム EPILON/One により実施した。ここでは、その獲得過程を EPILON/One の表示画面の例と説明を行う。図 A.1 はプリポスト法により知識抽出された操作とそのプリポスト関係である。操作としては、大歯車選定、小歯車選定、減速機生成（歯車の組合せ）、減速比計算、減速機選定、結果表示の操作が抽出された。

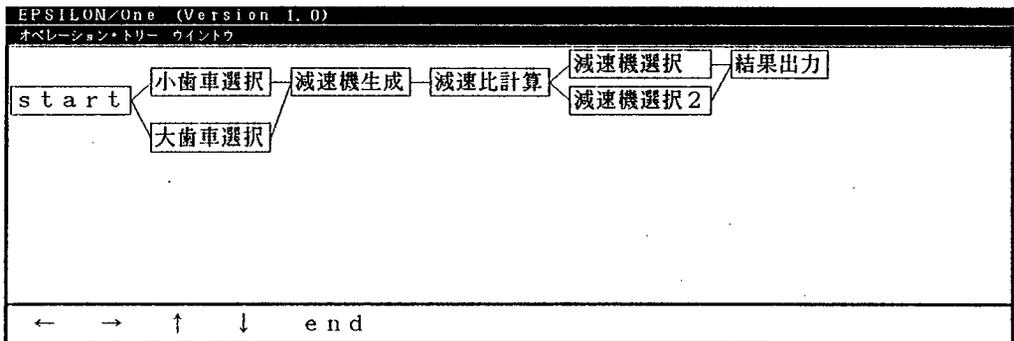


図 A.1 操作の抽出画面

図 A.2 は抽出された操作の型を専門家が決定するための画面（ウインドウ）表示である。各操作は7つの操作のいずれかの指定がなされなければならない。変換操作が選ばれると数値演算、要素分解、値入換えのいずれになるかが質問される。大歯車選定と小歯車選定は選択操作、減速機生成は組合せ操作、減速比計算は数値演算操作、減速機選定は選択操作、結果表示は出力操作である。

EPSILON One (Version 1.0)							
メニュー ウィンドウ							
小歯車選定	入力	出力	<b>選択</b>	分類	組み合わせ	ソート	変換
減速機生成	入力	出力	<b>選択</b>	分類	<b>組み合わせ</b>	ソート	変換
減速比計算	入力	出力	<b>選択</b>	分類	組み合わせ	ソート	<b>変換</b>
減速機選定 2	入力	出力	<b>選択</b>	分類	組み合わせ	ソート	変換
結果出力	入力	<b>出力</b>	<b>選択</b>	分類	組み合わせ	ソート	変換
減速機選定	入力	出力	<b>選択</b>	分類	組み合わせ	ソート	変換
	do_it				abort		

入出力ウィンドウ	抽出コメント
<p>*** オペレーション・タイプの修正 ***</p> <p>オペレーションのタイプを上記のメニューより選択してください。</p>	<p>オペレーション名</p> <p>プリーポスト関係</p> <p><b>オペレーションタイプ</b></p> <p>ソース要素グループ</p> <p>分類項目</p> <p>入力要素グループ</p> <p>エバリュエータ</p> <p>-----</p> <p>要素名</p> <p>要素属性</p>
	参照

図 A.2 操作の型の決定画面

図 A.3 と図 A.4 は操作の型に対応した知識の抽出画面（ウインドウ）を示している。図 A.3 は大歯車選定の選択基準の抽出である。図 A.4 は結果表示のメッセージと属性の抽出である。

EPSILON One (Version 1.0)		
操作名	属性	結果
終了 次 削除 クリアー ----- + -	材質 ----- 数値入力 文字入力	
材質 @== 金属		
<p>入出力ウインドウ</p> <p>オペレーション名 減速機選択 2 の入力要素グループを上記のメニューより選択してください。</p> <p>オペレーション名 結果出力 の入力要素グループを上記のメニューより選択してください。</p> <p>オペレーション名 減速機選択 の入力要素グループを上記のメニューより選択してください。</p> <p>オペレーション名 大歯車選択 の入力要素グループを上記のメニューより選択してください。</p> <p>入力要素グループ名の修正を上記のメニューよりおこなってください。</p> <p>入力要素グループの修正を上記メニューより行ってください。</p> <p>*** エバリュエータの抽出 ***</p> <p>小歯車選択は選択オペレーションですが、このエバリュエータに必用な情報を上記のメニューより入力してください。</p> <p>ヘルプ情報はラベル名：入出力ウインドウのラベルを右一回クリックしてください。</p>	<p>操作名</p> <p>属性</p> <p>結果</p>	<p>参照</p>

図 A.3 選択基準の抽出画面

EPSILON One (Version 1.0)		
操作名	属性	結果
エバリュエータ名変更 : default yes no 出力元 : モニターウインドウ 出力属性 : すべて 指定 なし 出力メッセージ変更 : yes no do_it abort		
<p>入出力ウインドウ</p> <p>*** エバリュエータの修正 ***</p> <p>エバリュエータをクリアーしますか? y/n n</p> <p>エバリュエータを修正したい場合には、上記のメニューよりおこなってください。</p> <p>( 終了時 : 終了 選択 )</p> <p>結果出力は出力オペレーションですが、このエバリュエータに必用な情報を上記のメニューより入力してください。</p> <p>上記のメニュー中の出力属性名以外の属性をあげてください。</p> <p>( 終了時 : end 入力 )</p> <p>出力属性名 &gt; 「</p>	<p>操作名</p> <p>属性</p> <p>結果</p>	<p>参照</p>

図 A.4 結果表示のメッセージと属性の抽出画面

図 A.5 は要素（歯車）の属性と属性値の抽出である。属性は画面（ウインドウ）上の表から選択する。属性値はメニューまたは、端末操作卓（キーボード）から入力する。

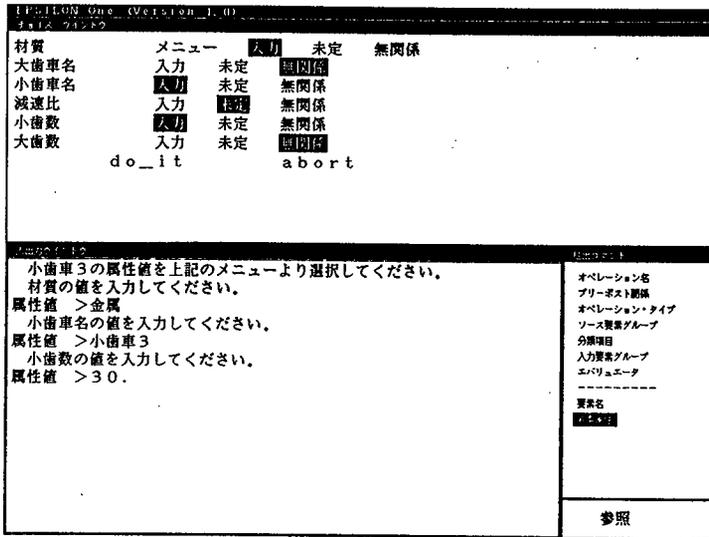


図 A.5 要素の属性と属性値の抽出画面

図 A.6 は実行画面である。画面上の操作表示で黒く反転しているのが処理の終了した操作である。

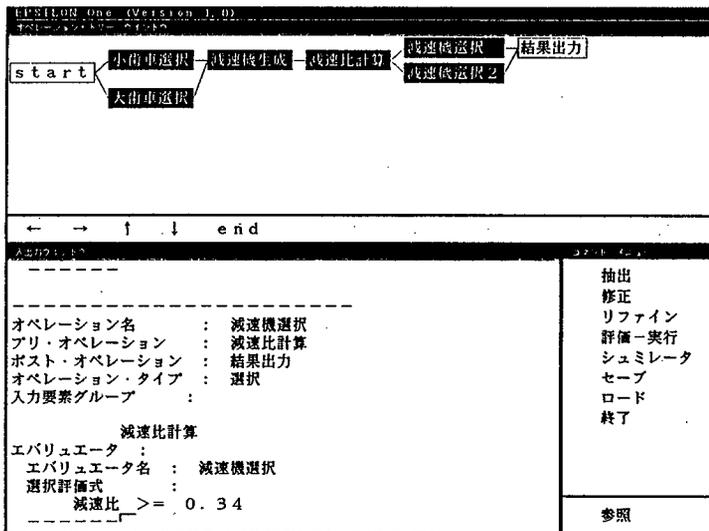


図 A.6 専門家モデルの実行画面

## 付録 B

### コンタクトレンズの選定知識の専門家モデルでの表現

この例題は、カルガリー大学のゲインズ (Brian Gaines) 教授とドイツの研究機関であるマーク・リンスター (Marc Linster) 氏から頂いたコンタクトレンズ選定知識である。この知識は知識獲得支援システム KSS0 が知識獲得した知識であるが、同じ問題を知識獲得支援システム EPSILON/One で知識獲得を行った。この知識は年齢や目の性質に合わせて、コンタクトレンズの種類を選定する [Gaines 89] [Linster 89]。各操作は次のような知識である。

**iAGE:** 年齢の入力操作, **iPRE:** 近視かどうかの入力操作

**iAST:** 乱視かどうかの入力操作, **iTEA:** 涙の性質の入力操作

**nsft, nsft1, nsft3:** ソフトコンタクトレンズ不適合メッセージの生成 (値入換え操作)

**fsft:** ソフトコンタクト適合メッセージの生成

**nhard, nhard7:** ハードコンタクトレンズ不適合メッセージの生成

**fhhard:** ハードコンタクトレンズ適合メッセージの生成

**disp1, disp2, disp3:** 判定結果の表示 (出力操作)

**ind1, ..., ind10:** 分類操作

ただし、要素はコンタクトレンズを選ばれる患者である。

図 B.1 は獲得された操作とプリポスト関係である。

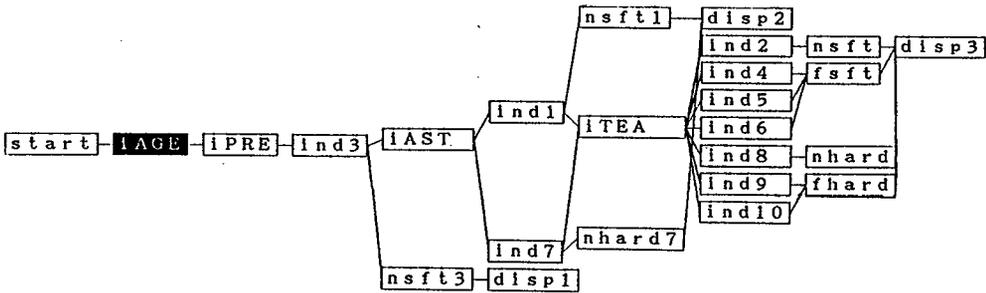


図 B.1 獲得された操作とプリポスト関係画面

図 B.2 は操作の評価基準の一例である。

EPSILON One (Version 1.0)  
イpsilonワン・ゼロ・ワン?

← → ↑ ↓ end

<p>入出力ウィンドウ</p> <p style="text-align: center;">iPRE</p> <p>エバリュエータ :</p> <p>エバリュエータ名 : ind3</p> <p>分類評価式 :</p> <p>r3-resに分類するときの評価式 :</p> <p>age @== presbyopic</p> <p>pres @== myope</p> <p>-----</p> <p>r3-fallに分類するときの評価式 :</p> <p>pres @\== myope</p> <p>-----</p> <p>age @\== presbyopic</p> <p>-----</p>	<p>フロントメニュー</p> <ul style="list-style-type: none"> <li>抽出</li> <li>修正</li> <li>リファイン</li> <li>評価-実行</li> <li>シュミレータ</li> <li>セーブ</li> <li>ロード</li> <li>終了</li> </ul>
参照	

図 B.2 操作の評価基準の例画面

## 付録 C

### ベアリング選定知識の推定

この知識は工作機械の設計 [Inoue 88] [Terasaki 88] においてベアリングを選定する際の操作知識を選定例から推定したものである。

図 C.1 はこの例題の専門家モデルである。ベアリングのカタログから必要なベアリングを選び、選んだベアリングを順序付けする問題である。さらに、順序付けされたベアリングの知識の一部の値を置き換える操作を付加している。

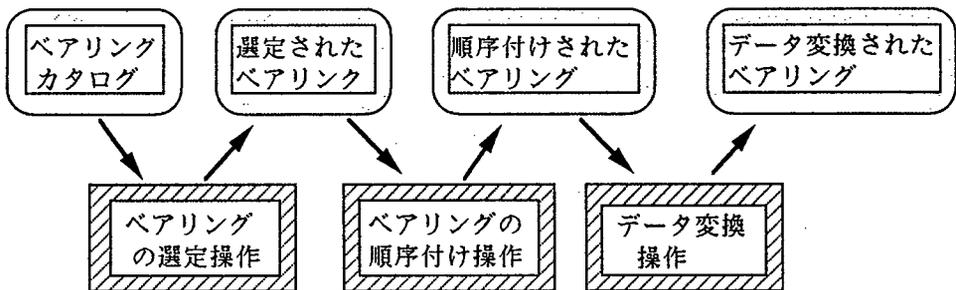


図 C.1 ベアリング選定の専門家モデル

ベアリングの性質を表す知識の例 [Nippon-seikou 86] を次に示す。

[深溝玉軸受, [ラジアル荷重, 可能], [アキシシャル荷重, 可能],  
 [アキシシャル荷重方向, 両方向], [合成荷重, 可能],  
 [高速回転, 特に可能], [高精度, 特に可能],  
 [低雑音と低トルク, 特に可能], [剛性, undefined],  
 [内輪と外輪の許容傾き, 十分に可能], [調心作用, undefined],  
 [内輪と外輪の分離, undefined], [固定側用, 適用可],  
 [自由側用, 軸伸縮逃がせば可], [内輪テーバ穴, undefined] ]

```

Pseudo771PSI/V2 1989年12月12日 Version 1554/0158/0231/011204/015200  CSPA起動時刻 01-Mar-90 15:08:09
PE00(00) 0 wcsi 00001E9D 00007100000700
PE00(00) 0 08006300000700
PE00(00) 0 Induction
PE00(00) 0
PE00(00) 0 selection criterion [高精度] = [ [特に可能] ]
PE00(00) 0
PE00(00) 0 wcsi 00001F0C 10000000000000
PE00(00) 0 0001F400301E99
PE00(00) 0 wcsi 00003F1F 00030112010690
PE00(00) 0 00040112012290
$!pl
PE00(00) 0 Induction
PE00(00) 0
$version
Version :
$go/pimos 20
Throw_goal
(Number of
[

selection criterion [低雑音と低トルク, 高精度] = [ [特に可能, 特に可能] ]
selection criterion [ラジアル荷重, 高精度] = [ [可能, 特に可能], [十分に可能,
特に可能], [特に可能, 特に可能] ]
selection criterion [アキシシャル荷重, 高速回転] = [ [可能, 特に可能], [十分に可!
能, 特に可能], [十分に可能, 十分に可能], [不可, 特に可能], [不可, 十分に可能] ]
selection criterion [アキシシャル荷重, 高精度] = [ [可能, 特に可能], [十分に可能!
, 特に可能], [不可, 特に可能] ]
Shell for ShellUser

```

図 C.2 選択基準の推定例の画面

図 C.2 は選定知識を推定した PSI-II (並列推論 OS である PIMOS

[Chikayama 88] を搭載したフロントエンドプロセッサ) の画面である。

[深溝玉軸受、マグネット玉軸受、アングュラ玉軸受、複列アングュラ玉軸受、組合せアングュラ玉軸受、自動調心軸受、同筒ころ軸受、複列同筒ころ軸受、片つば付円筒ころ軸受] から [深溝玉軸受、アングュラ玉軸受、組合せアングュラ玉軸受、同筒ころ軸受、複列同筒ころ軸受] を選定した

場合の選択基準が属性：高精度が {特に可能} であることを推定している。また、複数の属性を考慮した結果が下の画面（ウインドウ）に示されている。

〔深溝玉軸受、アンギュラ玉軸受、組合せアンギュラ玉軸受、同筒ころ軸受、複列同筒ころ軸受〕を〔深溝玉軸受、アンギュラ玉軸受、同筒ころ軸受、組合せアンギュラ玉軸受、複列同筒ころ軸受〕に並べ換えた場合の順序付け基準の推定を行った。図 C.3 は順序付け知識を推定した画面である。順序付け基準が2種類推定されている。

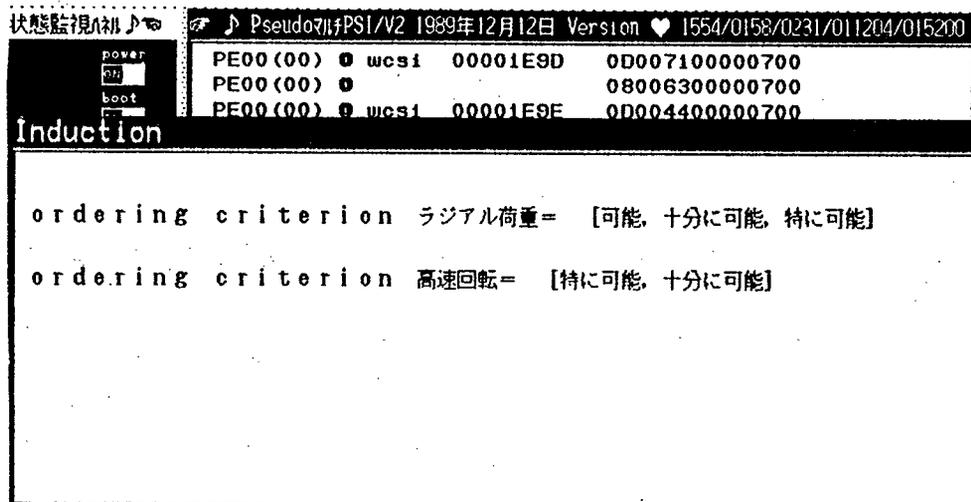


図 C.3 順序付け基準の推定例の画面



