

TM-0425

解釈型知識獲得システム

瀧 寛和

December, 1987

©1987, ICOT

ICOT

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

(03) 456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

解釈型知識獲得システム

— 知識解釈の問題を中心に —

滝 寛和

(財) 新世代コンピュータ技術開発機構

まえがき

本論文では、解釈に基づく知識獲得システムのシステムイメージ（構成と機能）及びその未解決課題と解決提案について述べる。知識システムの構築上の大きな問題の一つに知識獲得ボトルネックがある。知識獲得支援システム（以下、KASと略す）は、この知識獲得ボトルネックの解消を目的としている。知識獲得は知識源とKASのインタラクションの形態により、Active KAS（AKAS：能動的知識獲得支援システム）とPassive KAS（PKAS：受動的知識獲得支援システム）に分けられる。AKASはインタビュー機能を持つ対話型のシステムである。PKASは、プロトコル解析やテキスト解析の様な知識源に問い合わせの確認が出来ない場合の知識獲得を支援するシステムである。本文ではPKASの技術課題と実現イメージについて述べる。

INTERPRETATION-BASED KNOWLEDGE ACQUISITION

<Problems and An Image of Interpretation-based Knowledge Acquisition>

Hirokazu Taki

ICOT Research Center, Institute for New Generation Computer Technology

4-28 Mita 1-Chome, Minato-ku, Tokyo 108, JAPAN

E-Mail: htaki@icot.jp

abstract

This paper described problems and an image of interpretation-based knowledge acquisition. There is a bottle-neck problem in knowledge acquisition. Knowledge acquisition systems are developed to solve this bottle-neck problem. There are two type systems in knowledge acquisition. One is an active knowledge acquisition system, another is a passive knowledge acquisition system. The active system is an interactive system with an interview mechanism. The passive one deals with protocol analysis and text analysis for knowledge acquisition. This paper introduces the passive knowledge acquisition system's problems and its image.

1. はじめに

知識システムの構築上の大きな問題の一つに知識獲得ボトルネックがある。知識獲得支援システム（以下、KASと略す）は、この知識獲得ボトルネックの解消を目的としている。よく、学習システムと知識獲得（支援）システムが混同されるが、知識獲得（支援）システムでは、人間の専門家の持つ知識を計算機システム上に計算機システムが利用できる形式にして、（専門家の知的作業の実現の意味でその能力を無理に拡張せず、そつくりそのまま移すことをその機能としている（出来るだけ忠実に専門家を計算機上に再現する）。一方、学習システムでは、部分情報や不足している情報から、一般的に言える情報（知識）を導くことを目的としている（特殊な情報を導くこともある）。さて、知識獲得支援システムは知識源とKASのインタラクションの形態により、Active KAS（AKAS：能動的知識獲得支援システム）とPassive KAS（PKAS：受動的知識獲得支援システム）に分けられる。AKASはCTAS [山崎 87] やEPSILON/EM [滝 87] などのインタビュー機能を持つ対話型のシステムである。PKASは、プロトコル解析やテキスト解析 [空閑 87] の様な知識源に問い合わせや確認が出来ない場合の知識獲得を支援するシステムである。

AKASでは、インタビュー技術（知識を連想させる心理学的方法、知識の不足・冗長・矛盾についてのリファイン方法、知識を表現/確認させるための知識表現インターフェイス）が重要な技術である。PKASは、AKASと比較するとユーザに問い合わせできないので、ほとんどの知識獲得過程を自動化しなければならない。PKASの機能イメージとその機能実現のための技術課題、（1）知識の解釈レベルの管理、（2）解釈の多義性の管理、（3）解釈能力の向上について述べる。

2. 解釈型知識獲得システムの機能と構造

2.1 解釈型知識獲得システムの機能イメージ

解釈型知識獲得システムは、入力された情報とその入力時の状況から適切な表現にそれらの知識を変換・管理する。

（入力）状況：Sに処理：Pを専門家が行った。

（出力）（1） $S \rightarrow P$ ；Sの時にはPを行うと良い。

（2） $S, P \rightarrow S_2$ ；状況：SはPにより S_2 に変化。

（入力）状況：Sに観測：Oが行われた。

（出力） $S, O \rightarrow S_2$ ；状況：Sには情報Oが加わったので S_2 に変化。

（出力の変換）（ $S \rightarrow P$ ）の変換： $P=Q$ または、 $P \rightarrow Q$ ならば、

（ $S \rightarrow Q$ ）を生成。

同様に、 $S=A, B, C$ で $A=D$ （または $A \rightarrow D$ ）のときに

$S \rightarrow P$ から $D, B, C \rightarrow P$ を生成。

（注：上記は、プロトコル解析の場合の例である。テキスト解析では、ファクトや状況を持たない規則が直接入力される場合がある。）

2.2 解釈型知識獲得システムの構造イメージ

[解釈型知識獲得システムの基本モデル]

解釈型知識獲得システムの基本モデルは第1図に示すように、4つの要素から構成される。知識獲得システムの本体は解釈器(Interpreter)と解釈の為に先行知識(Advance Knowledge for Interpretation)である。この解釈器に与えられるのが入力情報（未整理知識）(Input Knowledge)であり、解釈器の出力が解釈された知識（獲得された知識）(Interpreted Knowledge)である。さて、プロトコル解析などでは、入力情報は一度には与えられず、少しずつ入力される。この先に入力された知識（獲得された知識）を後で

入力される知識の解釈に利用することで、解釈の効率の向上を行うことが考えられる。

[解釈型知識獲得システムの解釈器の機能と構造]

解釈型知識獲得システムの中心要素は解釈器である。この解釈器の機能と構成を考えるのに、技能獲得システム (Skill Acquisition System) の場合について、その機能を参考にしてみることにする。人間は触覚や力覚を用いて、実にうまく感覚技能を必要とする作業を行うことができる、しかしながら、人間はこの感覚技能・知識を言葉や数値・数式で表現出来ない。つまり、人間にたいして、感覚技能・知識についてその内容を問い合わせ出来ないのである。感覚技能・知識の獲得システムは、一種のPKASである。さて、感覚知識 (技能) の獲得システム [滝 84] [滝 85] は第2図に示すような構成を取っている。このシステムはバイラテラルマスタースレーブマニピュレータの構成をとっており、人間は触覚や力覚を用いて、感覚を必要とする作業を行うことができる。この技能獲得システムにおける学習コンピュータの構成を見ると第3図のようになる。この中で、フィルターリングとは、このシステムが獲得するのに不必要な情報を除く働きをする。たとえば、「もし、力覚センサーの入力に対して、人間がまったく無反応ならば、そのセンサー入力、この作業の技能・知識の獲得に不要である。」などのフィルターリング規則で、必要な知識のみを取り込む。解釈変換器は、人間の指令値と力覚センサー入力のパターンから力覚作業パターンの指令 (ロボット制御命令) を生成する。この例では、解釈変換の多義性や変換レベル (ロボット制御命令のレベル) の階層化は行われていない。獲得される技能の形態は、特定の作業に対する (1) 制御命令の並び、(2) 制御命令のパラメータである。さて、この学習コンピュータの機能を拡張し、解釈型知識獲得システムの解釈器の構成と機能を検討すると次の第4図の様になる。

3. 知識獲得における解釈の問題

[知識の解釈レベル問題]

知識を獲得する場合、人間は獲得した知識に対して、「リンゴというものは知らない、赤いということもわからない、しかし、リンゴは赤いと教えられた。だから、リンゴは赤いか?と質問されたら、はいと答えることにしよう。」「電圧測定用テスターやロジックアナライザは電子回路基板の検査に使用すると便利だと教えられた、しかし、いつ、どの様にして使用するのかは知らない。」などのように、獲得した知識がどの程度自分の知的作業に利用出来るか (獲得レベル) に関するメタな情報を持っている。しかしながら、種々の知識獲得や学習の研究において、このような「知識をどの程度システムのものとしたか」については、ほとんど研究が行われていない。言い替えれば、この知識獲得のレベルは知識をどの程度解釈 (理解) 出来ているかのレベルと言える。

[知識の解釈の多義性の問題]

知識獲得の問題の一つに獲得した知識ベースの管理問題がある。これは、知識ベースの無矛盾性を維持する信念の管理機構 (TMS: Truth Maintenance System) の問題である。しかしながら、知識獲得の途中では、知識の量が充分でない為、無矛盾な知識の信念は複数存在することになる、つまり、知識獲得支援システムにとって、可能な解釈が複数あることになる (多重世界を形成する。)、そして、知識獲得が進むと特定の世界 (ある解釈) において矛盾が生じることがある、その場合にその世界 (解釈) を消去し、無矛盾な世界 (解釈) をのみを残す必要がある。つまり、多重世界の管理と各世界における無矛盾性の維持機構が必要となっている。特に、複数の知識源 (専門家) からの知識獲得では、(知識源ごとにつくられる) 世界間の不整合性の管理も重要である。

[知識の解釈効率の向上]

人間は一般的に、知識が増えるに従って、知識を解釈する能力が向上する。この解釈能力の向上とは、(1) 知識の解釈速度の向上、(2) 知識の解釈の正確さ (解釈多義

性の可能性を絞ること、未解釈な知識を減らすこと）を向上することである。一般に、コンピュータシステムは、処理すべき情報が増加すると全体としての処理速度は遅くなる。いかに、コンピュータシステムにおいて知識の解釈効率を向上させるかが大きな問題である。

4. 知識の解釈レベル（獲得レベル）

人間の知識獲得においては、その知識そのものだけでなく、知識の理解レベルや知識の用法に関するメタ知識も同時に獲得（形成）している。その知識獲得（解釈）レベルの設定を次に示す。

レベル0：外界（知識獲得システムの外部）にある知識、未獲得知識

レベル1：内界にある知識（システム内に取り込んだ知識）だが、未解釈の為
使用不可能な知識

（例）見知らぬ国の言葉を聞いたときのその言葉。日本語しか知らない人が
ロシア語をきいたときのそのロシア語。

レベル2：システム内で使用できる記号に変換された知識ではあるが、他の知識との関係がわからない知識であり、それ単体でしか使用できない知識。

第3章の「リングは赤い」の例のように、「リングは赤いですか？」には「はい」と答えられ、「何が赤いか？」や「リングは何か？」に「リングは赤い」と答えられるが、「赤いくだものは何？」（リングがくだものであることを知らない）には答えられない。いわゆる「まる暗記」であり、まったく応用がきかない。他の知識と相互に関連付けされていない。

レベル3：他の知識と関係付けられた知識、このレベルの関係はフラットなものであり、階層化などの構造化はされていない。

レベル4：階層構造化やモデル化（ドメインモデルなど）に組み込まれた知識ではあるが、問題解決戦略（いつ、どのような対象に、どのような理由から、いかに使用するか（どの知識といっしょに使用するかなど））に組み込まれていない知識。

レベル5：知識の使用される状況（いつ、どのような対象に、使用するか）が分かっている知識ではあるが、知識の使用理由説明やうまい使用方法は分かっていない。専門家モデル〔滝 87〕はこのレベルの知識モデルである。

レベル6：知識の用法（どのような理由から、いかに使用するかなどのメタ知識）も分かっている知識。

種々の知識獲得支援システムでは、ドメインモデル、タスクモデル、専門家モデルなどの枠組みを用意している。これは、その枠組みに知識を入れることで、解釈のレベルを管理しているとも言える。

5. 解釈の多義性の管理

解釈型の知識獲得支援システムでは、自然言語の構文解析に見られる「複数の構文解析結果の問題」と同様の解釈の多義性の問題がある。この多義性は、新たな知識の追加によって、更新される必要がある。というのは、無矛盾な解釈と思っていたのが、新しく追加された知識との間に矛盾をもたらす可能性があるからである。単一の世界における無矛盾性の維持としては、TMS〔Doyle 79〕が有名であるが、多義性を管理するには、複数の世界を同時に管理できる必要がある。そのようなシステムとしては、ATMS〔de Kleer 86〕がある。筆者は、このATMSと同等でビジュアルなネット表現で無矛盾性を管理するシステムを考案中である。論理型Complications（LCC）〔滝 88〕は、矛盾の構造をComplications-Netとして表現し、新しく加わった知識をそのネット上にマップすることで、矛盾を検出するシステムである。また、あいまい性を持った知識の管理には、獲得した知識の不整合性を検出する不確実Complications（PCC）〔滝 88〕を利用できる。PCCでは新しく混乱度を導入し、知識の不整合性を評価している。LCCは基本的には、単一世界の矛盾検出機構（Maintenanceは行わない）であるので、多重世界を管理する機構が必要である。多重世界の管理は階層化されたコンテ

キストが利用できる。このコンテキスト機構を実現するために、CONNIVER [Su
ssman 72]、CONPAC [滝 78] のコンテキスト機構を利用する。第5図にL
CCの一使用例を示す。LCCでは知識の依存・導出関係をネット構造で表す。ネット
内にループが形成されるとそれは矛盾を表す。

6. 知識の解釈効率の向上

知識解釈の効率の向上方法には次の2通りの方法が考えられる。

(1) 解釈の可能性を絞り込む。

既に獲得した知識で解釈木の枝を刈ることで速く解を得ることができる。自然
言語の構文解析に、以前の構文解析結果を解析知識として利用するプロダクシ
ョンルールを用いた自然言語理解システムNLIPS [滝 80] がある。

(例) I see a bird on a tree in my house.

可能性1: 私も鳥も私の家の中にいる。

可能性2: 私は家の内。鳥は家の外。

可能性3: 鳥は家の内。私は家の外。

ここですでに、I am in my house. などの文を既に理解していれば、可能性3
は消去できる。(実際にはNLIPSは構文解析の時点で可能性3の生成をゆ
るさない。) ATMSでは、「私は家の内」と「私は家の外」が矛盾すること
を利用して、同様のことができると思われる。

(2) 解釈器を入力される知識に合わせて、部分計算を行う。

第6図は自己適用可能な部分計算器の処理 [藤田 87] を表している。第4図
が解釈型知識獲得システムの解釈器の構成であるが、解釈の一例(解釈器の入
力と出力)をみると非常に興味深いことがわかる。これは、入力と解釈器に部
分計算を適用した場合の結果が「解釈器の出力」と同じになる。つまり、解釈
器自身がある種の部分計算機であり、この解釈器を効率化するのが別の部分計
算機となる。

(解釈の一例)

入力: S1 (AMPの出力が低い。)、端子a-b間を測定器Mで計った。

次に(S2)、「測定器Mの値が低い」が分かった。

次に(S3)、端子a-bを別の抵抗と取り換えた。

解釈器内の知識: 端子a-bはAMP出力抵抗である。

測定器Mは電圧を計る。

出力: S1→AMP出力抵抗の電圧を計る。

S2=S1、かつAMP出力抵抗の電圧を計った。

S3=S2、かつAMP出力抵抗の電圧が低い。

S3→AMP出力抵抗を取り換える。

```
Input: s(1) -> measure(m, a-b)
        ( s(1), measure(m, a-b) -> s(2) )
        ( s(2), measure_show(m, low) -> s(3) )
        s(3) -> change(a-b)
```

Interpreter Knowledge:

```
a-b = amp_output_register (注: これを部分計算で利用するには工夫がいる。)
measure(m, X) -> voltage_measure(X)
measure_show(m, low) -> voltage(low)
```

```

Output: s(1) -> voltage_measue(amp_output_register)
        s(1).voltage_measue(amp_output_register) -> s(2)
        s(2).voltage(low) -> s(3)
        s(3) -> change(amp_output_register)

```

(InputとInterpreter Knowledgeを部分計算するとOutputと同じものがえられる。)

7. まとめ

これら3つの基本問題はPKASだけでなくAKASにもある。AKASにおいても、獲得した知識が増えてくると、リファインの効率などは当然おちてくるので、「獲得効率の向上」は問題となる。また、「獲得した知識のレベル管理問題」もあるが、AKASでは専門家に知識を問い合わせる際に、システム側で質問を決められるので、答えられる知識のレベルをある程度制御できる。「獲得した知識の多義性」についても、多義性が発生する毎に、専門家にその都度問い合わせができる。言い替えば、AKASでは、知識獲得レベルと多義性の管理はインタビュー問題、特にリファインの問題となる。

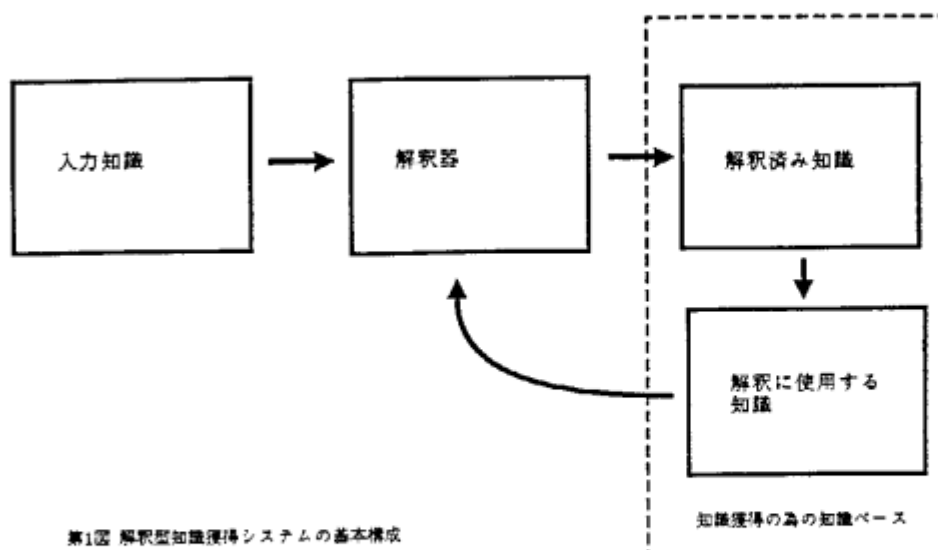
PKAS = 知識の解釈機構 (演繹推論、部分計算、

ただし、構造化やモデル化は知識獲得レベルに含まれる。)

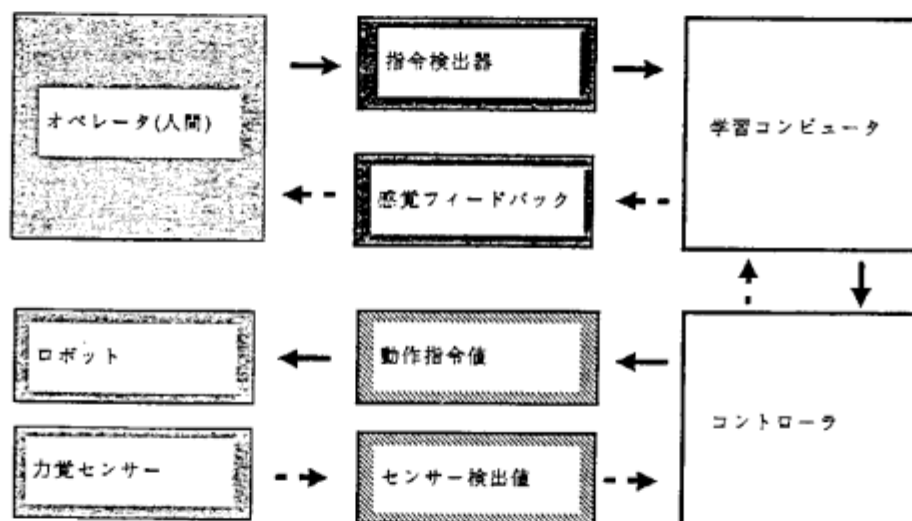
- + 知識獲得レベルの管理 (ドメインモデル、専門家モデルなどの枠組みの利用)
- + 知識の多義性管理 (ATMSまたは、コンテキスト処理+矛盾検出機能)
- + 知識獲得の効率化 (部分計算)

[参考文献]

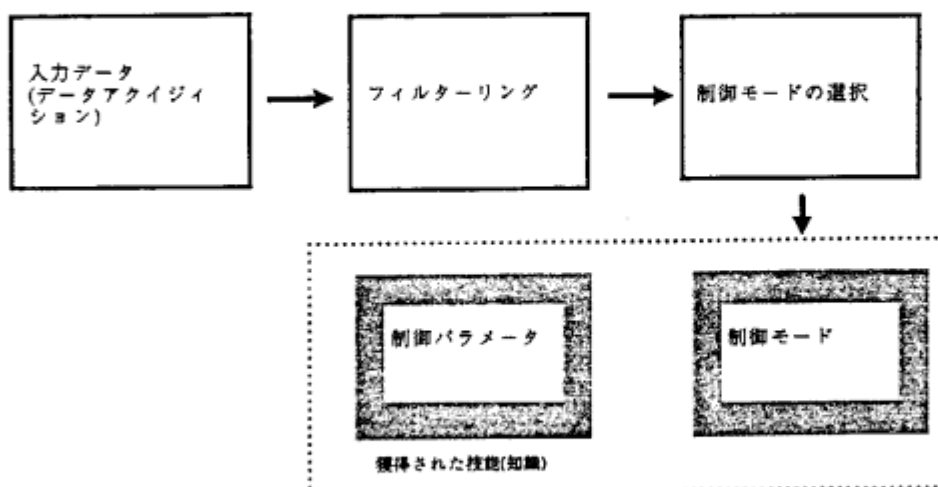
- [空閑 87] 空閑 茂起, 他: 文書知識プリミティブを用いた知識獲得の方法
情報処理学会 第35回全国大会 5P-10, 1987年9月
- [滝 78] 滝 寛和, 人工知能用言語 CONPAC, 大阪大学 基礎工学部
制御工学科, 卒業論文, 1978年
- [滝 80] 滝 寛和, 安部 意広, 辻 三郎: 知識ベースを利用した構文解析と
拡張されたプロダクションシステムを持つ自然言語理解システム,
電子通信学会研究会資料 AL79-119 (1980)
- [滝 84] 滝 寛和, 他: 力覚制御ロボット向き自動プログラミングシステム,
第28回システムと制御研究発表講演会 G2, 1984
- [滝 85] H. Taki, Y. Sakaue: A Programming System for Intelligent Robots with
force feedback, Robots 9 Conference Proceedings(SME), Vol1, 1985
- [滝 87] 滝 寛和, 他: 知識獲得支援システム(EPSILON/EM)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52-4, 25/31, 1987年5月
- [滝 88] 滝 寛和: 不確定Complications, ICOT-TR (to appear)
- [滝 88] (2) 滝 寛和, 井上 克己: 論理型Complications, ICOT-TR (to appear)
- [藤田 87] 藤田 博, 古川 康一: 自己適用可能な部分計算プログラムの実現と応用,
情報処理学会 ソフトウェア基礎論研究会資料 87-22
- [山崎 87] 山崎 毅文, 滝 寛和, 他: 類型タスク構造に基づく知識獲得
— 分類型問題向け知識獲得支援システムCTAS —,
日本ソフトウェア科学会, 第4回大会D-4-2, 1987年11月
- [de Kleer 86] de Kleer, J., An Assumption-based TMS, Artificial Intelligence
28 (1986) 127-162
- [Doyle 79] Doyle, J., A Truth Maintenance System, Artificial Intelligence
12 (1979) 231-272
- [Sussman 72] Sussman, G. J., and McDermott, D., From PLANNER to CONNIVER -
A Generic Approach, Proc. AFIPS FJCC (1972) 1171-1179



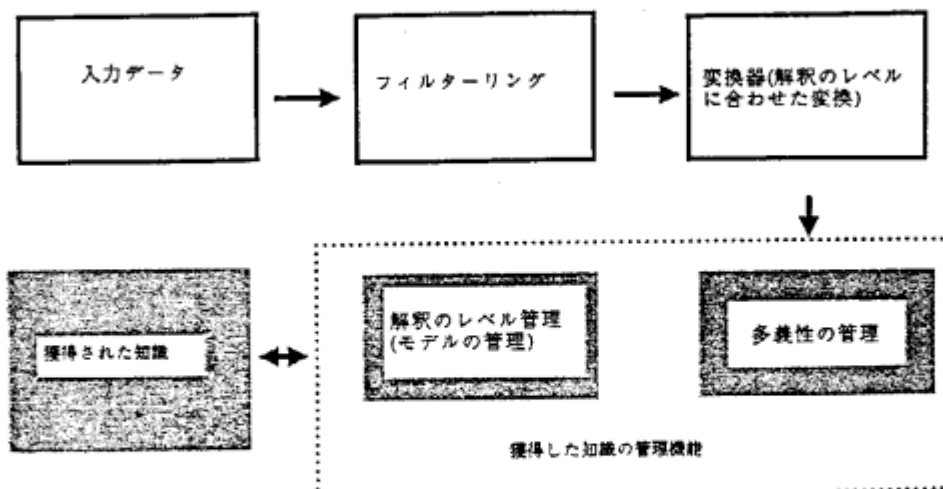
第1図 解釈型知識獲得システムの基本構成



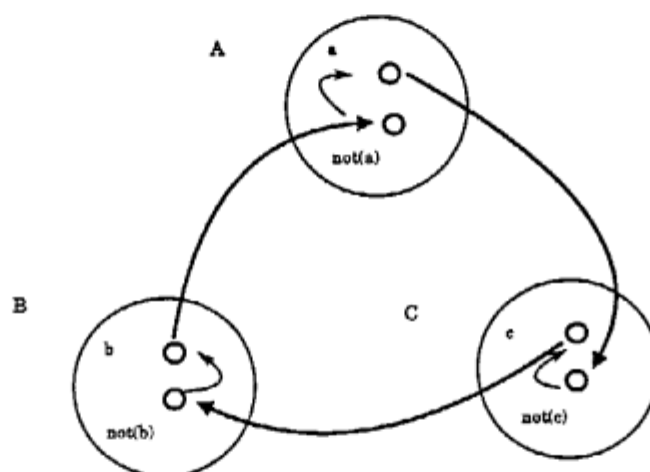
第2図 技能獲得システムの構成(例)



第3図 解釈器の構成(例)

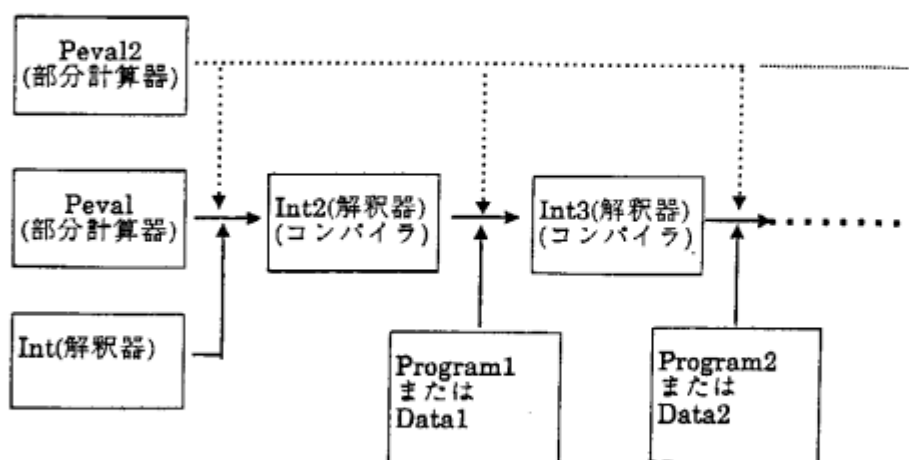


第4図 解釈器の構成(概要)



A&B&Cならば矛盾を表すネットを示す。
Aの存在は $\text{not}(a) \rightarrow a$ で表す。
AとBとCが皆存在するときは図のようになる。
LCCでは矛盾はネットのループで表される。

第5図 論理型COMPLICATIONSの例



第6図 反帰的な部分計算プログラムの部分計算