

対話型教授知識構築環境における 知識獲得効率化アルゴリズムの考察

A Study of the Efficient Learning Algorithm to Construct
the Domain Knowledge in the Interactive Environment.

松原行宏 長町三生
Yukihiro MATSUBARA Mitsuo NAGAMACHI
広島大学 工学部
Faculty of Engineering, Hiroshima University

あらまし 知的C A Iを構築する場合、教授知識、教授戦略、学習者モデル、ユーザインターフェースにモジュール分割して各モジュールごとに構成していくが、まず最初の段階として対象領域に関する教授知識をよく分析して知識構造を明らかにしておくことが重要となる。しかしながら、対象領域によりその構造は大きく異なり、明確に分類することは難しく、非常に困難な作業である。そこで教授知識の獲得を容易にする環境の開発が重要となる。本稿では、この立場に立ち従来から提案している対話による教授知識の構成法において、新たに対象領域に関する属性適応型学習機構を付加し、知識獲得の効率化を目指す。

キーワード C A I 知識工学 情報処理システム 機械学習 教授知識獲得手法 対話環境

1. まえがき

近年、システムに問題解決能力を持たせた知的C A Iの研究が盛んになってきている。コンピュータに人間の教師の代りを完全にさせることはできないが、知識の正確な記憶とその取り出し方、個別化等においては、コンピュータは人間よりもすぐれている。例えば、システムが問題解決能力を持つことによって、問題解決の過程を解釈するメタ知識を用意することができ、質問に答えたり、学習者の答えの正誤を判定することができる。さらに、学習者が間違っている場合には、その原因を同定し対処することができるようになる。また、適当な範囲の自然言語や図形を処理して入力形式の制限を緩和することもできる。

このような高度な個別教育を実現するものとして、知的C A Iは教授知識、教授戦略、学習者モデル、対話インターフェースの4つのモジュールから成る枠組みでとらえられている¹⁾。この中で、教授知識は特に対象領域に依存し、明確に分類することが難しく、知的C A Iの構築を困難にしている。現在、教授知識の構成は対象領域の解析や、プロトコル解析を繰り返すことにより構築していくという方法がとられている。これは比較的確に分類しやすい知識(例えば筆算形式における算術演算)でも大変な作業となるのは経験的に明らかである²⁾。

そこで、従来の我々の研究において問題領域の表現に属性を用いた、対話による教授知識の構成法を提案している³⁾。この教授知識の構成法は、実際に人間の教師に問題を解いてもらう形式をとり、知識の使い方に関する知識である問題解決的知識を対象領域としている。本手法では対象領域の属性表現において各属性の記述を同等に扱い、単純化を図ることによりシステムの汎用性を実現しているが、その反対象が複雑かつ大規模になってくると問題領域をうまく表現できなくなると思われる。この場合、構成された教授知識は冗長な属性表現を多く含み、的確な表現とならず、効率的な教授知識獲得がなされなくなる恐れがある。

そこで本稿では、各属性を属性の種類によって上位概念・下位概念に分類し、階層的な構造で表現する。そして属性を階層的に扱うことにより、システムがより効率よく知識獲得し、より適用状況に即した教授知識を構築することを目的とする。

2. 対話形式による教授知識の構成法

本節では、従来我々が提案している対話による形式をとる知的C A Iにおける教授知識の構成法^{3), 4)}について述べる。まず、対象とする問題解決的知識について説明し、その知識獲得におけるシステム構成および基本メカニズムについて述べる。

2.1 対象領域

教授知識は、通常のエキスパートシステムにおける専門知識に対応し、対象領域に関係する知識を表現したものである。知識の表現形式はフレーム型、手続きネットワーク、プロダクションルール、オブジェクト型、述語論理などいろいろ報告されている^{1), 5)}。

2.1.1 知識の分類

知的C A Iで対象とされる知識は以下のものが挙げられる。

① 宣言的知識

事実、概念やルールを表す知識。例えば公式や定理など、「～は～である」のような形式で表される。

② 手続き的知識

具体的なある条件のもとに、一定の行動をすることを表す知識。宣言的知識の解釈過程の結果、生成されたもの。例えば、操作手続きや解法のアルゴリズムや言語文法規則など、一連の操作の列で表される。

③ 問題解決的知識

問題解決は「意図の設定→方略の選択」のサイクルの繰り返しによってなされるが、これをいかにうまく正確に実行させるかに関わる知識である。例えば経験則として得られるノ

ウハウの知識、場面洞察の直観的知識など、IF/THEN 型のプロダクションルールで表される。

④ メタ知識

知識の使い方に関する知識。IF/THEN型のプロダクションルールで表される。

知的CAIは、特に専門家の数が少なく、しかも知識の体系化(整理)が難しい領域で必要であると思われる。また、そのような領域では「ある状況のときにはこのような処理をしなければならない」というような知識が重要である。そこで本稿では③の問題解決的知識を対象領域とした。

2. 1. 2 問題解決的知識

問題解決過程は「意図の設定→方略の選択」のサイクルの繰り返しによって、初期状態から目標状態へ遷移させていく過程であり、次のように置き換えられる。

意図の設定：与えられた問題の特徴を抽出する

方略の選択：適用可能なオペレータの中から最適なものを選択する

オペレータとは与えられた問題にほどこすことのできる問題解決のための操作である。このように問題解決的知識を定義すると、問題解決的知識は状態の特徴とオペレータを強く結びつけた知識といえる。

特徴とオペレータの結びつきという考え方で、問題解決的知識を次の二つとしてとらえることができる。

① 問題分類知識

問題が与えられた時に、うまく特徴を抽出し、問題の型を分類する知識のこと。このような知識は類似した問題を多数解くことにより、共通の特徴を抽出して、適用するオペレータと強く結びつけることによって得られる。

② オペレータの合成に関する知識

オペレータ間の依存関係が強いものを一組の問題解決的知識と考えることができる。これは問題解決過程上で連続するオペレータの組のことを言う。しかし、連続するオペレータが必ずしもすべて依存関係が強いとは言えないので、完全因果性のあるものだけを抽出する。

2. 2 システム構成

図1にシステムの構成を示す。本システムは、教師がコンピュータに例題とその解答を与えるサイクルの繰り返しを行うことにより、教授知識の構築を行う形態をとっている。問題解決的知識を対象領域としているので、公式等の知識は最初から基本オペレータとしてシステム内に与えてある。問題が入力されるとその問題の属性が抽出され、各基本オペレータの条件部とマッチングすることにより適用可能なオペレータをすべて検索する。

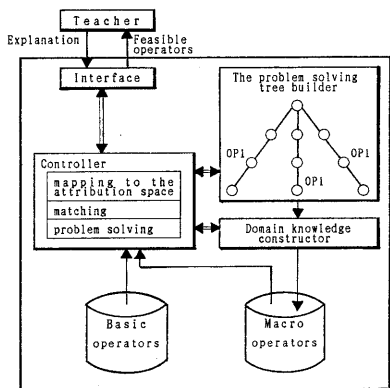


図1 システムの基本構成

次に教師が最適な解法であるオペレータを指定し問題に対してオペレータを実行する。状態が目標状態になるまでこの手順を繰り返し、その過程を問題解決トリー上に書き込む。このような操作をN回実行し、N回分の問題解決過程を問題解決トリー上に表現する。その後、問題解決トリーより教授知識の抽出を行う。このサイクルの繰り返しにより、教授知識を順次抽出していく。

2. 3 知識作成の基本メカニズム

問題解決において、人間が行う問題解決過程とかけ離れたものにならないように注意する必要がある。認知科学の分野における学習・知識獲得のメカニズムを用いた。AnderssonのACT*理論では、人間の手続きの学習を宣言的段階と手続きの段階の2つの段階としてとらえている。

宣言的段階は学習の初期の段階で、宣言的知識をその領域に特有な仕事を行う一連のオペレータに変換する段階である。このような機能を実現するために、手続き化と合成というメカニズムが提案されている^{6),7)}。手続き化は、汎用のプロダクションルールの中の変数項を特定の値に置き換えて新しい知識を作成することである。合成は、一連のプロダクションルールが一定の順序に適用される場合に一段階でその一連の効果を達成する新しいプロダクションルールを作成するメカニズムである。

また、手続きの段階は宣言的段階で得られた知識を、より問題状況に適した知識に調整していく段階である。学習が進むにつれて学習者が問題に適応するようになっていくような段階であり、これを実現するメカニズムとして一般化と弁別がある^{6),7)}。一般化は特定の問題でしか適用できない知識を、類似した問題にも適応できるようにプロダクションルールの条件部を調整することである。弁別は、知識が一般化され過ぎて誤った行動を起こす場合に、正しい実行をするようにプロダクションルールの条件部を厳しくしていくことである。

本手法では、知識表現としてプロダクションルールを用い、以下に示すメカニズム^{6),4)}を基に問題解決的知識を抽出を実現する。ルールの条件部に問題の特徴を属性表現を用いて表現し、実行部に適用オペレータを記述する。

① 手続き化

これは宣言的段階の手続き化に相当し、汎用プロダクションルールの変数部分を特定の値と置換することである。手続き化で生成するプロダクションルールの条件部は、問題の特徴を表す全ての属性の連言表現となる。

② 合成

合成は問題解決過程で連続して適用される2つの知識を1つにまとめる。例えば、知識Aを適用した後知識Bを適用した場合、知識Aの条件部が作成される知識の条件部となり、実行部はそれぞれの知識で実行されるオペレータのシーケンスとなる。

③ 一般化<A>、

一般化のメカニズムを2つ用意する。<A>は効率の良い一般化の方法である。N個の知識(実行部は全て同じ内容)が与えられると、全ての知識の条件部を満たす属性を抽出し、それを、作成する1つの知識の条件部とする方法である。例えば図2.8のような知識が与えられた場合、図2.9に示す知識を生成する。この方法は、ノイズが入った場合など、過度の一般化をしてしまう危険性がある。はこれを防ぐための一般化の方法である。は常に2つの知識だけを対象として一般化する。2つの知識の条件部を比較し、1箇所だけ属性が異なる場合のみ、その属性を削除し、残りの属性を条件部として新しい知識を作成する。<A>との方法を使い分けることにより、効率の良い一般化を実現することができる。

対話による教授知識の構成法では、教師により生の例題(常に最適なオペレータの指定)だけが与えられることになるので、弁別の機能は用意していない。

2.4 教授知識の構成

問題解決トリートリが生成された時点で教授知識の構成を行う。ここで生成される知識が知的CAI上で学習者に教える知識の単位となる。問題解決トリートリとは、設定された意図を満足する知識の生成を学習の1サイクルとし、このサイクルを繰り返すことにより教授知識を次第に構成していくが、この1サイクル中の状態を記述するものを言う(図2参照)。ここでは問題解決トリートリ上に、教師が意図した知識を教えるために用意したN問分の問題と、各問題の問題解決過程を記述する。各ステップでは、問題の属性値表現と各状態に適用可能なオペレータのリスト、さらに実際に適用されたオペレータを記述する。

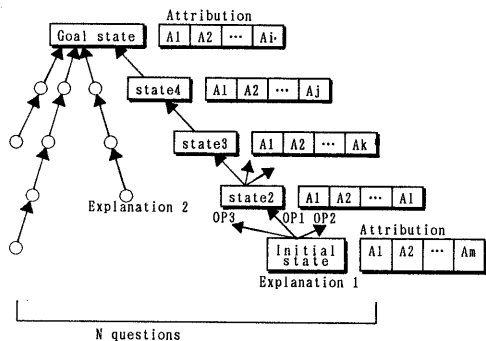


図2 問題解決トリートリ

問題解決トリートリから次の2つの問題解決的知識を抽出し構成する^{37,41}。

①問題分類に関する知識

問題分類に関する知識は、問題を与えられたときまずどのオペレータを適用させたら良いかの指針を与える知識である。従って、問題解決トリートリ上の初期状態(葉の部分)に注目すればよい。各初期状態に適用されるオペレータを抽出し、手続き化により知識を生成し、一般化<A>により条件部を調整する。この知識を教授知識の中に組み込み、教授知識内で一般化により知識の再構成をする(図3参照)。

1つの問題解決トリートリは、ある少数の意図を持った1サイクル学習の実現のために用いられるため、類似問題を多く含むと考えることができる。従って、トリートリ内で生成される知識では一般化<A>を用いても、過度の一般化をする危険性が少ない。しかし、教授知識内で調整を行なう場合、異なる特徴の問題に同一のオペレータを適用することも考えられるので、一般化<A>を用いると過度の一般化をする恐れがある。そこで条件の厳しい一般化を用いる。このようにして構成される教授知識は、それぞれの問題空間の分類を行い、更に同一のオペレータを適用する知識はオペレータ内での問題空間の分類をすることを示す。

②オペレータ合成に関する知識

問題解決トリートリ上に同一のオペレータ使用の系列がM個存在する場合、その2つのオペレータを合成し、問題解決的知識を生成する。合成は生成された知識を知的CAIで活用するという本稿の目的上、オペレータ間の依存関係が強いものでないと意味がないので、完全因果性のあるものに限り実行する。完全因果性とは、 OP_k が適用された後には必ず OP_1 が適用可能であることを意味し、閉じた強い因果関係があることを表わしている。1つのルールのアクションが大きくなりすぎるのを防ぐため、合成可能な組を以下の2つに限定する。

1. 基本オペレータ→基本オペレータ
2. 基本オペレータ→マクロオペレータ

合成の方法は $OP_k \rightarrow OP_1$ であるとする、 OP_k を適用する

問題の属性を条件部に持つルールを生成し(手続き化・合成)、生成されたルール間で一般化により知識の再構成を行なう(図4参照)。また合成の実行順序は、問題解決トリートの根に近く、かつ最初に与えられた問題の方から実行する。ここで一般化<A>、を使い分ける理由は問題分類に関する知識と同様である。

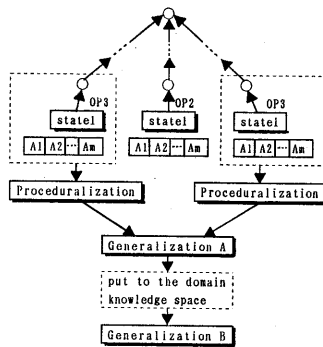


図3 問題分類に関する知識の生成

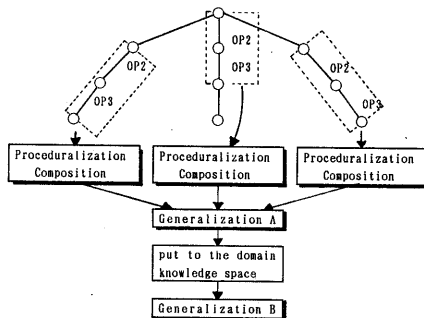


図4 合成に関する知識の生成

以上のように教授知識の構成を行うと、生成された教授知識の構造は、図5のように表現できる。まず、オペレータにより大きく問題空間の分類が行われ、各々の空間の中で階層構造として合成関係が表現される。各問題空間の上位にあるオペレータがマクロオペレータとなる。また、問題空間の各ノードは知識を所有し、その中で、同一オペレータ使用時の問題空間分類がなされている。さらに各々の空間はタイムタグによりリンクされている。

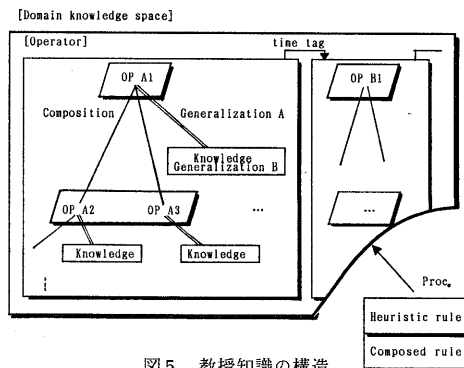


図5 教授知識の構造

2. 5 まとめ

以上に問題解決的知識を対象領域とした、対話による教授知識の構成法について述べた。この方法では教師が生徒に教えるときと同様に、教師がコンピュータに例題を与える過程で教授知識を構成していく方法をとっている。従って、教師の教え方に関する特徴(例題を与える順序、例題の数等)をうまく抽出し、教授知識の構成を単純化したという利点がある。しかし、問題領域の特徴の表現に用いている属性において、各属性の重みを同等に扱っていることにより、いろいろな領域において問題空間をうまく表現することができず、さらに属性の設定方法により大きく問題空間の分類が変化してしまう危険性がある。また、効率のよい教授知識構成ができず、構成された教授知識も的確なものとなっていないなどの問題点が考えられる。

そこで以上の問題点を解決するため、階層的な分類に基づいた属性表現を用いた新しい学習機構を提案し、それにとまう知識獲得メカニズムの拡張について述べる。

3. 属性適応型学習機構

問題空間の特徴表現に属性を用いる。属性とは、物体やその事象を構成するいろいろな要素で、属性名-属性値の組として複数個持たせて対象領域を表現したものである。本節では、属性表現を簡潔にし、階層的に用いることにより、より効率よく的確な知識の構成を行う。

3. 1 属性の分類

はじめに、属性を階層的に表現するための多次元な分類について述べる。例えば、属性は属性値の取り方により、

- ① Yes(1), No(0)の2値でカテゴリ分類されるもの
 - ② 連続的な値を取る選択的なカテゴリ分類されるもの
- の2つの種類があると考えられる。①において、ある属性の値が No(0) である時、与えられた問題の特徴表現として意味をもたせるかどうかによって、生成される知識の条件部が異なってくる。No(0) に意味をもたせた場合、冗長な属性により条件部が大きくなりすぎて知識構成の効率が悪くなる危険性がある。しかし、問題によっては No(0) の値も意味を持つことがあるので考慮する必要がある。また、①と②の関連性なども考慮する必要がある。

そこで、次の3つの点に着目して属性の分類を考える。

- ① 属性値が論理値であるか連続値であるか
属性値がカテゴリ分類されるような計数データ(Yes(1), No(0)のどちらかの値を持つ)であるか、あるいは数値的な計数データ(0, 1, 2, ...のような連続的な値を持つ)であるかにより分類を行なう。

- ② 属性の内容が初期的なものか前向きのものであるか
初期的というのは、事象に面したときに視覚情報(見たままの情報)として得られるようなもの、すなわち初期認知的なものを表現する属性である。また、前向きのというのは、学習者が以前に学習し既に所有している知識(公式や定理等)などを前向きのあてはめようとするような属性のことである。この2つの内容による分類を行う。

- ③ 包含関係にある属性
属性が前提条件を持つことによって、その前提条件となっている属性と包含関係ができる場合や、同じ問題特徴を表現した属性の計数・計量データ間による包含関係を考えた分類を行う。

3. 2 属性の階層化

属性の分類をもとに階層的な属性の表現を行う。各属性間において上位概念-下位概念の関係を考慮し、図6に示す表現方法を用いた。

まず、属性全体を大きく計量データと計数データに分類する。次に計数データの属性において、その下位概念に計数・計量間の関わりや包含関係を考慮し、計量データの属性と強く

関連があるものと、全く関連性をもたない命題を表わす属性とに分類する。さらに計量データの属性と関連がある属性の下位概念に存在の有無を表わす属性と、計量データ中のある1点の値を表わす属性に分類する。また、計量データと関連がない命題を表わす属性の中でも2値の値をもつ属性と、持つ値の数はあらかじめ定められているが2値よりも多いカテゴリ分類されるような属性に分類する。この図の中で、根の部分から左側が初期認知的な属性で、右側に属するものが前向きの公式や定理などをあてはめようとするような属性であると言える。この階層的な属性の表現と包含関係を考慮し、知識獲得の学習アルゴリズムを拡張する。

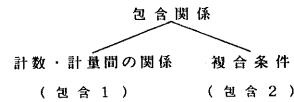
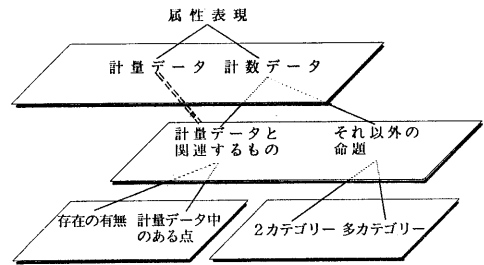


図6 属性の階層的表現

3. 3 学習アルゴリズムの拡張

本稿での知識獲得アルゴリズムは、AndersonのACT*理論をもとに手続き化・一般化・合成のメカニズムを適用して実現しているが、属性を階層的に用いることにより、手続き化、一般化の段階で以下のような問題点がでてくる。

- ・手続き化の段階
手続き化の段階で得られた知識は、用意されたすべての属性を持つようになっており、冗長な属性がたくさん含まれている。
 - ・一般化の段階
一般化<A>は、教えようと意図された知識に関する集合から抽出された知識の一般化に用いられ、冗長な属性が含まれていても効率よく一般化されるので問題はないと思われる。一般化は教授知識内での一般化で、一般化すべき知識どうしすべての属性を比較して属性の内容が1つだけ異なる場合にのみ行われる。しかしながら、属性を階層的に用いると、下位概念の属性をもつその上位概念の属性の内容が異なった場合、その下位概念の属性も明かに異なるので最低2つ以上の属性が異なり、一般化が起こりにくいことがある。そこで、一般化について修正する必要がある。
- そこで、次に述べる新たな拡張学習アルゴリズムを用意した。

3. 3. 1 手続き化の拡張

手続き化の段階で得られる冗長な属性を階層化をもとに削除する。まず、各属性について図7に示すように属性の型、包含関係などをフレームで表現し、属性の分類関係を与える。これをもとに次の3つのメカニズムを提案する。

- ① 同じ種類の属性を表わす属性の中での計数データと計量データの関係
同じ種類を表わす属性の中で計量データは 0, 1, 2... の

内で意味のある1つの値を持つが、それに対する計数データはON (Yes (1)) の状態のものが1つあればその他のものは、必ずすべてOFF (No (0)) の状態になっている。そのOFF (No (0)) 状態の属性は知識表現の中で意味を持たないと考えられる。そこで、計数データの中のOFF (No (0)) の状態の属性は、知識の特徴表現から削除する。

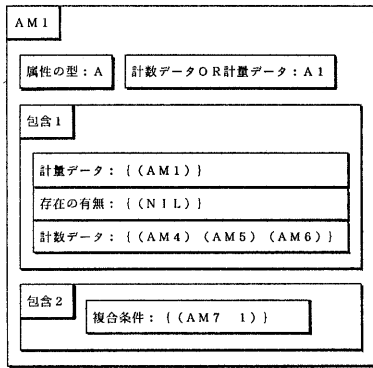


図7 属性のフレーム表現の例

②存在の有無を表わす属性がある場合の計数データの取り扱い

a) 存在の有無を表わす属性がON (Yes (1)) 状態 (その項目について存在することを表わす) の場合

上述の1)の手法に従い、OFF (No (0)) 状態の計数データの属性を削除する。

b) 存在の有無を表わす属性がOFF (No (0)) 状態 (その項目について存在しないことを表わす) の場合

その項目についての内容が存在しないということなので、その項目に関する計数データの属性は全てOFF (No (0)) の状態で、計量データの属性も属性値に0しか持たない。そこで、全てのOFF (No (0)) 状態の計数データの属性は意味を持たないので削除し、計量データの属性も意味を持たないので削除する。結局、存在の有無を表わす属性のみ意味を持つので残すことになる。

③包含関係

a) 包含条件: ON (包含の対象となる属性がONの状態) 包含される属性の値にも意味があるのでそのまま残しておく。

b) 包含条件: OFF (包含の対象となる属性がOFFの状態) 包含される属性もOFFの状態なので意味を持たず、削除してしまおう。

以上の学習アルゴリズムを適用して、手続き化により的確な知識が構成することができる (図8参照)。

3. 3. 2 一般化の拡張

一般化において、各属性に種類別に階層構造を持たせ、各軸 (属性の種類) ごとに上位概念と下位概念を明確にする。それにより、各属性に重みが与えられ、一般化のアルゴリズムを修正する必要がある。そこで、属性を各軸ごとに比較し、できるだけ上位概念の属性で一般化が行われるように、各軸より一般化の候補を取り上げるようにする。これを実現するため、次の4つのメカニズムを用意した。

①存在の有無、計数データ、計量データの属性間の結合がある場合

a) 存在の有無を表わす属性値が等しい場合

・ON (Yes (1)) 状態どうしの場合

計量データの属性値が等しい時は、計数データの属性も等しいのでそのまま残す。計量データの属性値が等

しくない場合、この計量データの属性における次元での一般化に意味があり、計数データの属性は等しくないので削除し、一般化の候補とする。

・OFF (No (0)) 状態どうしの場合

計数データ、計量データの属性は、手続き化の段階で削除され残っていないので、そのまま存在の有無を表わす属性だけを一般化の候補とする。

b) 存在の有無を表わす属性値が等しくない場合

存在の有無を表わす属性値が等しくないということは、この次元で一般化を行う必要がある。そこでこの存在の有無を表わす属性のみを一般化の候補として残す。

②計数データ、計量データの属性間の結合がある場合

a) 計量データの属性値が等しい場合

計数データの属性値も等しいので、そのまま一般化の候補にする。

b) 計量データの属性値が等しくない場合

計数データの属性値も等しくないので、計量データの属性のみを一般化の候補にする。

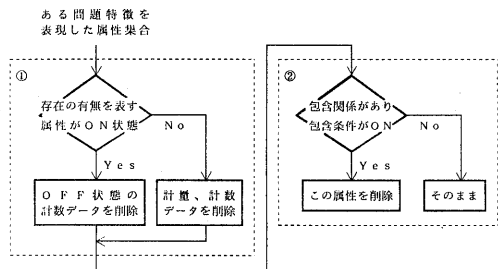


図8 手続き化レベルのアルゴリズム

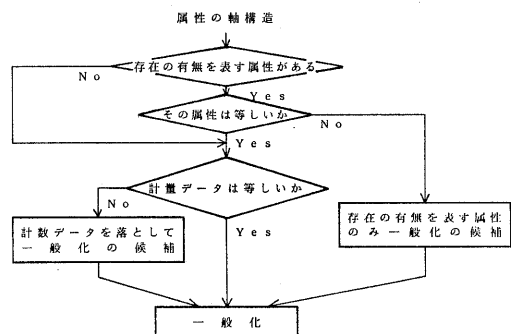


図9 一般化レベルのアルゴリズム

③存在の有無、計量データの属性間の結合がある場合

a) 存在の有無を表わす属性値が等しい場合

・ON (Yes (1)) 状態どうしの場合

計量データを一般化の対象とするので、そのまま一般化の候補とする。

・OFF (No (0)) 状態どうしの場合

計量データの属性は手続き化の段階で削除され残っていないので、そのまま一般化の候補とする。

b) 存在の有無を表わす属性値が等しくない場合

存在の有無を表わす属性を一般化の対象とする必要があるため、存在の有無を表わす属性のみを残し一般化の候補にする。

④ 包含関係がある場合

a) 包含条件が等しい場合

- ・ ON 状態の場合
包含される属性を一般化の対象とするのでそのまま。
- ・ OFF 状態の場合
包含する属性が残っており互いに等しいのでそのまま。

b) 包含条件が異なる (ON 状態と OFF 状態の属性) 場合

属性の数は異なるが、一般化 は数の少ないほうを基準に行い、この包含関係には強い関連がなく、包含する属性を基準に一般化 の対象となるのでそのままよい。

以上の一般化 におけるアルゴリズムは、一般化の対象となる2つの知識に対して行われ、一般化が成功すると一般化された知識をそのまま抽出し、一般化が失敗すると候補を取り出す (アルゴリズム適用) 以前の知識に戻して知識をもたせるようにする。これにより、従来のアルゴリズムに比べて効果的な一般化 が行えるようになる (図9参照)。また一般化 は、過度の一般化による弊害をふせぐために条件をかなり厳しくしている。この程度を調整するパラメータは、一般化 が実行される際の異属性数であるが、アルゴリズムの修正に伴い、パラメータの値に関して考慮する必要があらう。

以上の修正アルゴリズムを図10に示すような対話による教授知識の構成法に適用することにより、効率がよく、的確な知識獲得ができるようになる。

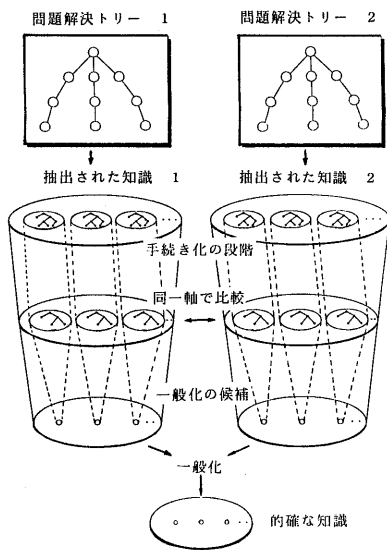


図10 属性適応型学習機構

3. 4 まとめ

本節では属性を階層的に表現し、従来の知識獲得のメカニズムを拡張することにより、効率の良い教授知識の構築を実現する方法を提案した。更に、構成された教授知識もよりの確なものとなるように、一般化についても拡張を行った。

4. 従来法との比較

本節では、前節で示した拡張アルゴリズムと従来のアルゴリズムによる教授知識の構成法を因数分解における問題解決の知識に適用し、構成された教授知識を比較評価する。

4. 1 因数分解の属性および基本オペレータ

本稿では、因数分解の知識を対象領域としており、37種類の属性を用いて因数分解の問題空間を表現した⁴⁾。属性は意味的内容により大きく以下の4つに分類し、用意した。

- a. 項数に関するもの
- b. 変数に関係するもの (次数も関連)
- c. 各項の特徴
- d. 項間の特徴

これらの属性を前述の分類方法に適用する。

また、システムにあらかじめ与える基本オペレータ (基本的な式変形を行うルール) を、特徴により5つに分類し、30個用意した⁴⁾。

- ① 公式で、直接因数分解されるもの 11個
- ② 公式で、直接因数分解されないもの 2個
- ③ 式変形に関するもので、いつでも実行可能なもの 8個
- ④ 式変形に関するもので、条件部を持つもの 8個
- ⑤ 終了判定条件 1個

これらの属性と基本オペレータをあらかじめシステムに与えておき、実際に例題を与える。例題は、まず市販の問題集から8セット計64問を与えて因数分解に関するある程度の知識を獲得した後、更に学習させた知識構造を安定させるために類似した問題を4セット計30問追加する (表1参照)。例題をMON1からMON12まで順に与え、1つのセットを1つの問題解決トリー上に表現する。以上のものをあらかじめ用意してから因数分解の領域に関する教授知識を構成する。

表1 システムに与える問題セット

セット数	特徴 (セットの意図)	例	数
MON 1	共通因数のくくりだし	$X(X+2)-X-2$	8問
MON 2	$[A^2-B^2]$ 公式	$4X^2-(2Y-Z)^2$	9問
MON 3	$[A^2-2AB+B^2]$ 公式	$9X^2-6XY+Y^2$	6問
MON 4	$[X^2+(A+B)X+AB]$ 公式	$X^2+8XY+15Y^2$	8問
MON 5	$[ACX^2+(AD+BC)X+BD]$ 公	$5X^2+7X-6$	8問
MON 6	$[A^3-B^3]$ 公式	$64A^3-8$	7問
MON 7	複2次式	X^4+4X^2-5	8問
MON 8	項数の多い問題	$X^2+2XY+Y^2-X-Y-2$	10問
MON 9	共通因数のくくりだし	$(A+B)^2+C(A+B)$	8問
MON10	$[A^2-B^2]$ 公式	$4X^2-1$	8問
MON11	$[A^2+2AB+B^2]$ 公式	X^2+6X+9	7問
MON12	$[A^3+B^3]$ 公式	$8X^3+1$	7問

4. 2 属性適応型学習機構の適用

以上に基づき、属性適応型学習機構の学習アルゴリズムを因数分解の領域に適用した。図11に構成された教授知識を示す。図中の棒はオペレータにより大きく問題空間が分類されたことを示しており、オペレータ番号1(OP1)からオペレータ番号19(OP19)まで順にリンクしている。棒の右肩の数字はオペレータの重要度 (適用回数) と、そのオペレータに関するルールの重要度を表わす。各棒のトップレベルのものが、該当するオペレータの問題分類に関する知識を表現しており、*印はその知識がないことを示している。また、階層表現になっているものは、オペレータの合成を表わす。それが複数あるものは、上下関係が生成順序を示している。各知識はリスト表現され、リストの要素が複数個あるものは、オペレータ内で問題空間の分類がなされている。知識のリストの最初の数字が知識番号で後の数字が重要度を表わしている。

4. 3 考察

① 効率性の比較

図12に生成ルール数の変化のグラフを示す。縦軸は生成されたルールの数で、横軸は教示問題セット (MON1からMON12) である。また、図13に知識当りの属性数の変化のグラ

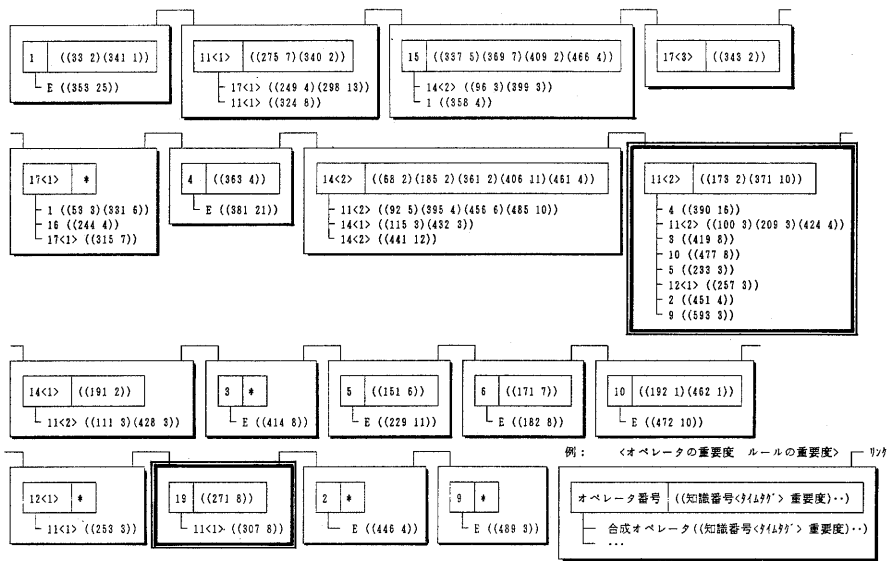


図 1 1 構成された教授知識

フを示す。縦軸は1知識当りに必要な属性の数で、横軸が問題セット (MON1 から MON12) である。

これらの図より、教示が進んでいくにつれて教授知識内で一般化 が従来より効率よく行われていることを示している。

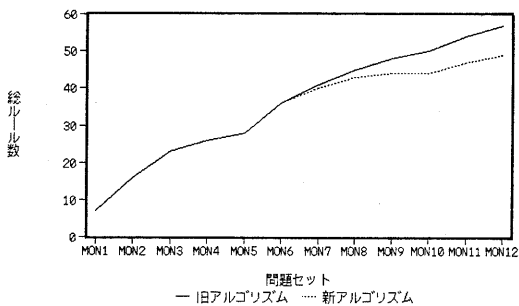


図 1 2 生成ルール数

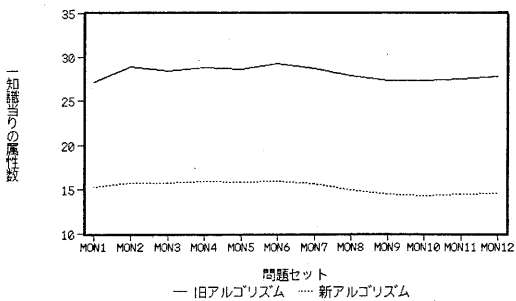


図 1 3 知識当り属性数

る。また図 1 3 より、問題表現に用いる属性の数が明らかに少なくて済むことがわかる。更に、曲線も従来のアルゴリ

ムに比べてなめらかになっているので、各属性の重みのバランスがうまくとれていることを表わしている。以上より効率の良い知識の構成ができていくといえる。

②一般化の強度の変化による比較

一般化 は、属性が1つだけ異なるときにしか成功しない。そこでこの条件を変化させ、一般化による知識獲得の比較を行った。一般化 が生じるための属性の差異の変数を2~4個とし、各条件で実行した結果での知識の生成過程を図 1 4 に示す。この図より教授知識内の一般化 (一般化) は起こりやすくなっているが、新しいアルゴリズムでは属性を階層化し、軸による分類を行っているので、教示を増やすにつれて過度の一般化が行われる危険性が高くなると思われる。従って、このパラメータの決定は対象領域に対応させて、適切に決定することが必要である。

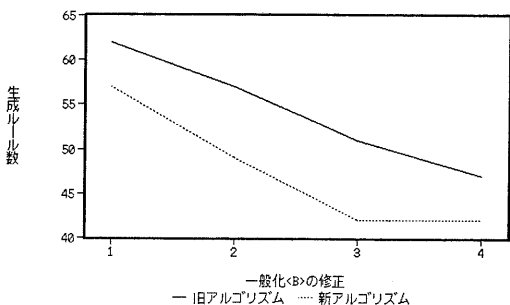


図 1 4 一般化の強度

③生成された知識の比較

次に、知識の内容について比較し評価する。比較に用いる知識として、因数分解において重要と思われる次の2つの知識を取り上げた。

- ・式を整理してから行う (項数が多い場合は、まず降べきの順に整理する)
- 基本オペレータとして「降べきの順に並べる」(OP19)が

与えられており、知識は

((SR27 (R27 ((AM4 0)(AM28 2))) 1))
とされ、任意の文字において降べきの順になっていなければ
(AM28 2)適用できる。これに対して生成された知識(MOP19
)は、次のようになった。

a. 従来のアルゴリズムによる知識
((SR271 (R271 (
(AM37 0)(AM34 0)(AM33 0)(AM32 0)(AM31 0)
(AM29 0)(AM28 2)(AM24 0)(AM18 0)(AM15 0)
(AM14 0)(AM13 1)(AM12 0)(AM11 0)(AM10 0)
(AM8 0)(AM6 0)(AM5 0)(AM4 0))) 8))
b. 修正アルゴリズムによる知識
((SR271 (R271 (
(AM10 0)(AM11 0)(AM13 1)(AM15 0)(AM 24 0)
(AM28 2)(AM29 0) 8))

bにおいて(AM10 0)は共通因子がない、(AM11 0)は係数が
くり出せない、(AM13 1)は項数が4以上、(AM15 0)、(AM24
0)、(AM29 0)は同類項がないことを意味する。従って、この
知識は「項数が多く任意のもじにおいて降べきの順になって
いなければ降べきの順にせよ」という意味を持つ。aにおい
ても(AM37 0)~(AM31 0)は項数が3でない、(AM18 0)~(AM1
4 0)、(AM12 0)は同類項がない、(AM8 0)~(AM4 0)は項数が
3以下でないことを意味し、後は同じ属性値をとり bと同様
の意味となる。明らかに bのほうが的確な知識表現となっ
ていることがわかる。

・まとまったものは1文字で表わす
「任意の項をかっこでくくって別の文字で置換する」とい
う基本オペレータ(OP11<2>)が与えられており、知識は
((SR13 (R13 ((AM4 0))) 1))
で表現されている。これは問題が目標状態にないときに(AM4
0)いつでも適用可能である。これに対して生成された知識(MOP11<2>
)は、次のようになった。

a. 従来のアルゴリズムによる知識
((SR371 (R371 (
(AM36 0)(AM35 0)(AM29 0)(AM27 0)(AM26 0)
(AM25 0)(AM23 0)(AM21 0)(AM20 0)(AM19 0)
(AM17 0)(AM15 0)(AM14 0)(AM13 0)(AM12 0)
(AM11 0)(AM10 0)(AM4 0)(AM2 0)(AM1 4)
) 10))
((SR173 (R173 (
(AM36 0)(AM35 0)(AM34 0)(AM33 0)(AM32 0)
(AM31 0)(AM29 0)(AM28 2)(AM27 0)(AM26 0)
(AM25 1)(AM24 0)(AM23 0)(AM22 0)(AM21 0)
(AM20 0)(AM19 1)(AM18 0)(AM17 0)(AM16 0)
(AM15 0)(AM14 0)(AM13 0)(AM12 0)(AM11 0)
(AM13 0)(AM12 0)(AM11 0)(AM10 0)(AM9 1)
(AM8 0)(AM7 2)(AM6 1)(AM5 0)(AM4 0)
(AM3 3)(AM2 0)(AM1 4))) 2))
b. 修正アルゴリズムによる知識
((SR380 (R380 (
(AM1 4)(AM10 0)(AM11 0)(AM23 0)(AM25 0)
(AM26 0)(AM27 0)(AM29 0))) 12))

aにおいてSR371に着目すると、4次式で(AM4 1)項数が
3以下(AM13 0)、 $\bigcirc^3 \cdot \bigcirc \Delta \cdot \bigcirc \Delta \times$ の項をもたない(AM19
0)~(AM23 0)のときに適用することを示している。これは問題
が複2次式であるときに適用することを意味する。
bにおいては、一般化が成功しており1つの知識になっ
ている。知識の内容は4次式で(AM1 4)、 $\bigcirc^3 \cdot \bigcirc \Delta \cdot \bigcirc \Delta \times$
の項をもたない(AM23 0)~(AM27 0)のときに適用することを
示している。これは項数に関係なく複2次式に適用できる知

識となっている。

その他の問題分類に関する知識、オペレータの合成に関す
る知識についてもうまく生成されていることがわかる。従っ
て、以上で提案した手続き化、一般化についてのアルゴリ
ズムの修正を行うことで、よりの確な知識を構成できる。

5. むすび

本稿では、知的C A Iを構成する枠組みの1つである教授
知識について述べ、その構成は困難であり、単純化すること
が必要であることを述べた。そこで、従来の知的C A Iにお
ける教授知識の構成法をもとに、新たな拡張アルゴリズムを
適用することにより、効率が高く、よりの確な教授知識の構
成を目的とし、その手法について提案した。

従来の研究は、教授知識の中で最も重要と思われる問題解
決的知識を対象領域とし、問題領域の特徴表現に属性を用い
ている。構成方法は教師とシステムとの対話によって構築し
ていく形式を取っている。また、知識獲得のメカニズムはAC
T理論に基づき、より人間の思考に近いものとなっている。
これらのことにより、教授知識の構成を単純化し、効果的
に行うことができるようになった。しかし、問題領域の特徴表
現に用いた属性の記述をすべて同等に扱っていることにより、
効率良く知識獲得ができず、抽出された知識も的確なもの
となっていないとも言えない。これらの問題点を解決する
ために、属性を階層的に表現した。更に、属性を内容により
分類し階層的に用いることにより各属性に対する重み付け
を実現し、それに伴い知識獲得のメカニズムを拡張した。最
後に、実際に因数分解の領域に関する知識に適用し、その
結果、効率よく教授知識の構成が行われ、また、構成され
た知識も意図していたとおりの確なものとなっていること
を確認した。

しかしながら、因数分解の領域は知識を分類しやすく、問
題空間も小さいので、構成された知識も大きいとはいえない。
知識を明確に分類することが難しく、問題空間の大きい領
域に適用した場合、基本オペレータが多数必要になり、ま
た、多くの知識が生成されるので、知識の構造が複雑にな
るとい問題が生じる。そこで、抽出されたオペレータを機
能により階層的に分類することが必要である。そうすること
により、知識の分類が明確に表され、より適用に即した知
識構造となるであろう。

【参考文献】

- 1) 大槻説乎, 山本米雄: 知的C A Iのパラダイムと実現環境, 情報処理学会誌, Vol. 29, No. 11, 1257/1265(1988).
- 2) 畑本恵子, 長町三生, 伊藤宏司, 辻敏夫: 算術演算の知的C A Iに関する一研究, 人間工学, Vol. 23, No. 4, 257/265(1987).
- 3) Matsubara, Y., Nagamachi, M., Matsushima, K.: "Interactive Learning Mechanism in Domain and Tutoring Knowledge for Intelligent Tutoring System" Human Factors in Organizational Design and Management-III, Elsevier Science Publishing, Inc. (1990)(printing).
- 4) 小野真道, 松原行宏, 長町三生: 知的C A Iにおける教授知識・教授戦略の構成法に関する一考察, 情報処理学会「教育に置けるコンピュータ利用の新しい方法」シンポジウム論文集, 65/72(1989).
- 5) 溝口理一郎, 角所収: 知的インタフェース, 情報処理学会誌, Vol. 29, No. 11, 1257/1282(1988).
- 6) Anderson, J. R.: "Knowledge Compilation: The General Learning Mechanism", An Artificial Intelligence Approach Volume II, 289/389, Morgan Kaufmann Pub. Inc. (1986).
- 7) Wenger, E.: "Artificial Intelligence and Tutoring Systems", Morgan Kaufmann Pub. Inc., 289/306(1987).