

手続き的問題解決における学習者モデリング (2)

松田 昇 岡本 敏雄

電気通信大学大学院情報システム学研究科
〒182 東京都調布市調布ヶ丘 1-5-1

あらまし

本研究は、手続き的な問題解決の領域における学習者モデル診断技法の探究が目的とされる。この学習世界では、与えられた課題を解く能力の習得だけではなく、課題を解くために利用された知識そのものに対する理解が重要である。本稿では、定式化された問題解決モデルにおける探索制御知識を、後者の立場での教授を支援するための知識として利用できることを示す。そして、システム自身の問題解決の経験を通して、経験則に基づいた探索制御知識を獲得する枠組を提案する。

和文キーワード 知的 CAI, 学習者モデル診断, 説明に基づく特殊化, 機械学習

Student Modelling for Procedural Problem Solving

Noboru MATSUDA[†] Toshio OKAMOTO[‡]

Graduate School of Information Systems,
University of Electro-Communications
1-5-1, Chofugaoka, Chofu-shi, Tokyo, 182 JAPAN
[†]mazda@ai.is.uec.ac.jp, [‡]okamoto@ai.is.uec.ac.jp

Abstract

This study is intended to investigate the role of a student model for Intelligent Tutoring Systems (ITSs) and develop an effective student model in the domain of procedural problem solving. In this domain, students have to learn the concepts of which the domain consists, as well as the procedural operations to solve given problems. We have developed a student model diagnosing system which deals with this problem. The system requires three types of domain knowledge; the basic operators (BO) which solve the given problem by exhaustive search, the macro operators (MO) which can solve the given problem more effectively, and the search control knowledge which can explain why a particular macro operator is useful in a specific situation. The knowledge of MO is used to infer students' understandings for procedural knowledge, while the knowledge of BO should be used to infer students' understandings for conceptual knowledge.

英文 key words Intelligent Tutoring System, Student Modelling, Explanation-Based Specialization,
Machine Learning

1 はじめに

本研究は、手続き的な問題解決の領域における学習者モデル診断技法の探究が目的とされる。手続き的な問題解決の学習世界では、与えられた課題を解く能力の習得が先ず期待される。しかしながら、単に課題を解く能力を習得させるだけではなく、課題を解くために利用された知識そのものに対する理解が重要であると思われる。例えば、特定の課題を解決することができても、類似した異なる課題を解くことができない学習者は、課題を解決する際に適用された知識に対する理解が十分ではないと考えられる。

筆者らはこれまでに、手続き的な問題解決に必要な知識を次のように分類し、それらの関連を考慮した学習者モデル診断システムを研究・開発してきた [2]。

(K1) 課題を効率的に解決するための定理や公式などと呼ばれる知識。

(K2) K1 の知識を導出するために利用される公理や定義などと呼ばれる知識。

一般に、効率化知識は問題解決の過程において直接観測することが可能であり、他方、原理的知識はそれが困難である。そこで、効率化知識の習得状態を表現した学習者モデルに基づいて原理的知識の理解状態を診断する機構を提案している。

上述した枠組の学習者モデル診断システムでは、効率化知識と原理的知識の関連に関する知識が必要である。その際、効率化知識に含まれる特定の知識が、なぜ正しいのか、もしくは、なぜ有益なのか、などという事柄を説明しなければならない。本稿では、そのために、次の知識を考慮した学習者モデル診断システムを提案する。

(K3) 問題解決過程における特定の局面で適用すべき知識を示唆する知識。

本稿で提案する学習者モデル診断システムは、システム自身の問題解決の経験を通して、経験則に基づいた探索制御知識を獲得する。そこで獲得される探索制御知識は、問題解決のためのプリミティブオペレータとマクロオペレータの関連を説明することができる。これを上述した効率化知識と原理的知識の言葉で表せば、システムは、効率化知識の正当性や有効性などを原理的知識を用いて説明することができることになる。

2 手続き的な問題解決のための知識

ここでは、課題を解くための知識を問題解決知識と呼ぶ。問題解決知識は、次の3つに大別することができる。

(K1) 課題を効率的に解決するための定理や公式などと呼ばれる知識 — 効率化知識 —。

(K2) K1 の知識を導出するために利用される公理や定義などと呼ばれる知識 — 原理的知識 —。

(K3) 問題解決過程における特定の局面で適用すべき知識を示唆する知識 — 経験則 —。

本研究で対象とする手続き的な問題解決とは、上記の問題解決知識を繰り返し適用することにより、与えられた課題の解を求めるタスクである。

さて、手続き的な問題解決の学習において、課題を解くことだけが目的であれば、(K1) の知識を習得すれば良いことになる。これらの知識は、一般に、適用条件と実際の適用方法 (手順) が明瞭であり、知識の正しさや知識が成立する理由を知らなくても、課題を解決することができる。ここでは、これらの知識を効率化知識と呼ぶ。

しかしながら、問題解決に対する理解を深めるためには、(K2) の知識に対する理解が重要となる。これらの知識は、効率化知識の中で用いられている概念の説明であったり、効率化知識の正しさや成立理由を説明するための手段となる知識である。したがって、単に課題を解決するだけの能力ではなく、問題解決の理解を深めることが教育目標として設定されている場合には、(K2) の知識を理解させることが重要である。以後、(K2) の知識を原理的知識と呼ぶ。

(K3) の知識は、問題解決を効率的に行なわせるための知識であり、経験則と呼ばれることもある (ここでは以後、これらの知識を経験則と呼ぶ)。例えば、問題解決のある局面において、適用可能な解法規則が複数ある場合に、適切な基準によって解法規則を選択させる知識などがこれに該当する。(K2) で考察した問題解決の深い理解のために重要となる知識の一つである。

3 手続き的な問題解決ドメインの定式化

本章では、本研究で対象とする手続き的な問題解決ドメインの定式化を行なう。後述する学習者モデル診断システムの適用範囲を明らかにすることが、ここでの定式化の目的である。したがって、ここでは、学習者モデル診断という側面から手続き的な問題解決の世界を定式化する。

ここで整理する手続き的な問題解決のモデルは、特定のドメインに依存していない。したがって、具体的なドメインは、実際の教育対象として多数考えられるが、本稿ではドメインの例として、数学における方程式の解法を想定している。

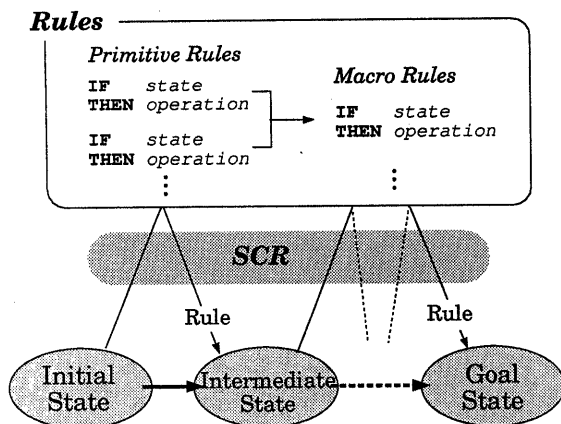


図 1 問題解決過程

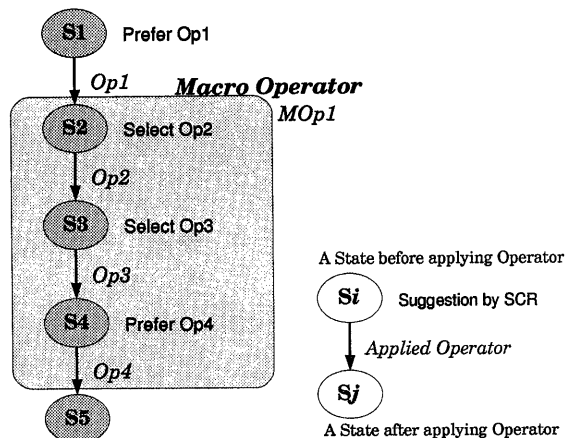


図 2 探索制御知識によるマクロ規則の説明

3.1 対象世界の構成

手続き的問題解決世界の構成要素で、本システムが学習者モデル診断に利用する要素は、次の2つである。

- 状態
- 解法規則
- 探索制御知識

状態は、問題の初期状態、目標状態、中間状態を表現する。初期状態は、必ず与えられる。目標状態は、予め与えられている場合と、そうでない場合がある。前者の例としては定理証明問題などがあり、後者の例としては数式の演算問題などがある。本稿で扱う方程式の解法は、後者に分類される。

解法規則は、状態を遷移させるための手続き (operator) である。各々の解法規則は、適用のための条件と、適用の結果としての作用が状態を用いて記述されている。

一般に、状態に対して盲目的に解法規則を適用している限り、探索の爆発が発生し、解を導くことができない。そのために、探索制御知識は、解法規則の適用を制御する。

3.2 問題解決

本システムで対象とする手続き的問題解決とは、上述した構成要素からなる世界において、初期状態から目標状態へ状態を遷移させる解法規則の系列を求める操作である。そのような解法規則の系列を解と呼ぶ。図1に問題解決過程の模式図を示す。

各状態において、適用可能なプリミティブ規則またはマクロ規則を探索する。現在の状態にマッチする条件部

を持った規則は、全て等しく適応可能であるとみなされる。複数の規則が適応可能である場合には、それらを競合集合に登録する。規則の競合集合が存在する場合、探索制御知識を用いて、最も適切な1つの規則を選択・適用し、状態を遷移させる。

一度、解が求められると、連続したプリミティブ規則の系列からマクロ規則が生成される。システムは、プリミティブ規則を用いて、ここで生成されたマクロ規則に適切な説明を与えることができる。図2では、プリミティブ規則 Op_2, Op_3, Op_4 から1つのマクロ規則 MOp_1 が生成されている。その際、状態 S_2, S_3, S_4 において、Select および Prefer という探索制御知識が適用されている。この結果により、状態 S_2 および S_3 に対して規則 Op_2, Op_3 を適用することが決定的 (Select)、— すなわち、他の競合する規則は全て失敗する — であり、それらの結果遷移した状態 S_4 において規則 Op_4 の適用が推奨される (Prefer) — すなわち、その規則の適用が成功する — ことが明らかになる。そしてそれが、マクロ規則 MOp_1 の成立する理由になっている。

3.3 解法規則

解法規則は、各状態を目標状態へ近付けるための手続きを表現している。ここでは、解法規則を次の2種類に分類する。

- (1) プリミティブ規則
- (2) マクロ規則

マクロ規則は、複数のプリミティブ規則を連続して適用した場合と等価な状態遷移を一つの手続きとして表現

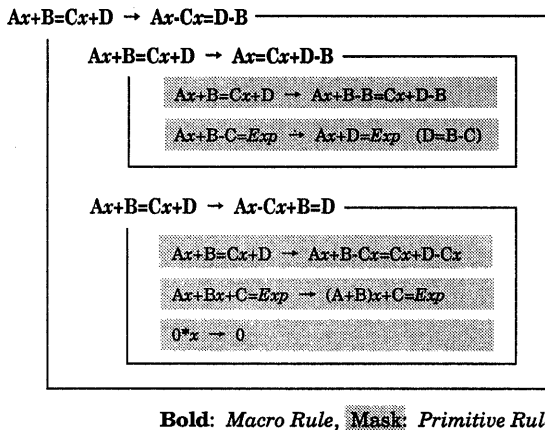


図 3 マクロ規則およびプリミティブ規則

している。マクロ規則は、それを構成するプリミティブ規則の探索を節約する効果を持つので、マクロ規則を用いることにより、一般に探索の効率が向上する。

他方、プリミティブ規則は、それ以上分解することができない規則である。数学の領域では、公理と呼ばれている概念である。プリミティブ規則だけを用いた解法プロセスは、言わば領域の公理に基づいた原理的な説明を学習者に与えることができるが、解法の効率は悪い。

上述したように、マクロ規則はプリミティブ規則から構成されている。したがって、プリミティブ規則を用いることにより、マクロ規則の手続き的な内容を詳細に説明することができる。このように考えると、任意のプリミティブ規則は、それを説明する“さらにプリミティブ”な規則を用意することができると言える。ここでは、ITSにおける学習者モデル診断が目的とされているので、システム設計者が予め定めた基準により、プリミティブ規則のレベルを固定している。

複数のマクロ規則を用いて、さらに多くのプリミティブ規則がまとめられているマクロ規則を生成することができる。図3に、マクロ規則とプリミティブ規則の関連を示す。図においては、“ $Ax+B=Cx+D \rightarrow Ax-Cx=D-B$ ”なるマクロ規則が2つのマクロ規則から構成されていることを示す。さらに、それらのマクロ規則は、5つのプリミティブ規則から構成されている。

3.4 探索制御知識

ここで想定する手続き的問題解決のモデルでは、プリミティブ規則だけを用いて与えられた問題を解決することが可能である。その際、プリミティブ規則を盲目的に適用していると、探索の爆発が発生する。そこで、探

索制御知識を用いて、各状態において適切なプリミティブ規則を選択する。

一般的な問題解決の枠組では、ヒューリスティック知識 (Heuristic Knowledge) の役割が重要となる。これは、問題解決を効率的に行なうための知識である。手続き的問題解決の学習において、ヒューリスティック知識の習得は一つの大きな学習目標であると考えられる。上述した枠組でのマクロ規則は、解法の効率を向上させるので、ヒューリスティック知識の一つであると言える。

一般に、解法規則の条件部に探索制御に関する情報を埋め込むことが可能である。本システムでは、後述するように、探索制御知識をマクロ規則の成立理由として利用する。したがって、解法規則とは別に、明示的に探索制御知識を記述する。すなわち、ここで想定する問題解決モデルにおけるヒューリスティック知識は、プリミティブ規則の適用を制御する知識であり、実際にはマクロ規則として表現されている。

探索制御知識 (SCR: Search Control Rules) は、プリミティブ規則の競合解消のための知識として記述されている。すなわち、ある状態において、競合するプリミティブ規則の集合に対して、(1) reject, (2) select, (3) prefer という言葉で各々の規則に対する適切さを表現する。(1)は、その規則を適用すると失敗する(問題解決を最後まで遂行できない)ことを意味している。(2)は、その規則の選択を意味する。これは、残りの競合する規則の適用は全て失敗することを意味している。(3)は、その規則の適用が競合する他の規則の適用よりも推奨されるということを意味している。そこには、適切さの半順序関係が定義される。

4 学習者モデル診断

本章では、前章までに定式化された手続き的な問題解決ドメインにおける学習者モデル診断システムの構成について述べる。

4.1 診断の課題

学習者の解法過程を観察することにより、学習者の理解している解法規則および理解していない解法規則を同定する必要がある。ITSにおける学習者モデル診断を考えた場合、学習者から直接観測可能な解法規則は、一般的に、効率化知識である。これは、3章で述べたマクロ規則に該当する。ここでは、以後、混乱のない限り“マクロ規則”を用いるが、それは“効率化知識”と同じ意味である。

筆者らはこれまでに、対象領域における汎用的な問題解決器(解法エキスパート)を用いて学習者の解法過程

をトレースすることにより、適用された解法規則を同定する手法を主に利用してきた [2]。しかしこの方法では、学習者の誤ったマクロ規則の適用を正しく認識できない場合がある。すなわち、システムの持っていないマクロ知識を学習者が適用した場合である。マクロ規則に対する振動オペレータやバグ規則など (以後、これらの知識をバグ知識と呼ぶ) を用意すれば、ある種の誤りを認識することが可能であり、上述した問題に対する一つの解決策である。しかし、そのようなバグ知識は強くドメイン依存であり、それらを記述することは一般的に困難である。

このような事柄を踏まえると、バグ知識に依存しない診断手法を構築することは意義深いと思われる。ここで提案する基本的な考え方は、次の通りである。すなわち、学習者に出題した課題の探索空間 (初期状態から連続して解法規則を適用した時に、到達可能な状態の集合) の部分空間をグラフで表現する。このグラフを解法グラフと呼ぶ。学習者の解法過程と解法グラフを比較することによって、マクロ規則の理解状態を診断する。この場合システムは、(1) 学習者の理解しているマクロ規則と (2) 適切な状況で適用することのできないマクロ規則を同定することができる。

次にシステムは、適切な状況において適用することのできなかったマクロ規則に関して、学習者がその規則を適用することができなかった理由を診断する。すなわち、効率化知識が理解できていない根本的な原因を診断する。筆者らは、その診断の根拠を原理的知識 (すなわち、プリミティブ規則) の理解に求めている。すなわち、効率化知識が理解できていない原因は、原理的知識の理解が不十分だからであるという立場である。例えば、特定の公式を理解したつもりでいる学習者が、特定の状況ではその公式を適用できるが、類似の異なった課題に適用することができないのは、その公式を原理的知識に関連付けて理解していないからであると考えことは妥当である。

ここで、マクロ規則とプリミティブ規則の関連を表現した知識に基づいた診断機構が必要とされる。次に、その機構について説明する。

4.2 プリミティブ規則の診断

前述したように、解法規則は、マクロ規則とプリミティブ規則から構成されている。システムは、プリミティブ規則と探索制御知識だけを用いて課題を解決することができる。そして、探索制御知識がプリミティブ規則の系列を説明する。一連のプリミティブ規則の系列は、マクロ規則として参照される。

マクロ規則とプリミティブ規則の関連をこのように

とらえると、学習者があるマクロ規則を適用することができなかった時に、その理由をプリミティブ規則に対する理解状態に求めることが可能な場合があると考えられる。そして、あるマクロ規則が理解できていない学習者を教育する場合に、直接その規則を教授するのではなく、そのマクロ規則を構成するプリミティブ規則に着目した教授が望まれる。すなわち、解法グラフによって、理解されていないと判断されたマクロ規則に関連したプリミティブ規則を求めることにより、学習者に理解させなければならない規則 (概念) を同定することが可能となる。

上述した機構を実現するために、システムは、マクロ規則とプリミティブ規則の関連に関する知識を有する。これらの知識は、システム自身の問題解決の経験を通して獲得される。次の章では、システムがこれらの知識を獲得するメカニズムについて説明する。

5 経験に基づく探索制御知識の獲得

ここで提案するシステムは、システム自身が問題解決の経験を通して、探索制御知識およびマクロ規則を学習する解法知識の獲得システムである。本章では、その構成について説明する。

5.1 システムの基本構成

前提知識として、システムは、極めて汎化された形での探索制御知識とドメインの一般的な課題を解決するのに十分なプリミティブ規則を有する。

前者は例えば、“あるノードに対して規則 (プリミティブ規則；以下、混乱のない限り、規則といった場合にはプリミティブ規則を意味する) の適用が成功するならば、その規則の適用が推奨 (prefer) される” という知識である。このように過度に汎化された知識は、実際の問題解決では利用することができない。そこで、システムは、このように過度に汎化された知識を適度に特殊化して、探索に利用できる探索制御知識を学習する。

問題解決過程における規則の適用事例は、極めて特殊化された探索制御知識であると言える。なぜならば、全く同一の状況に遭遇した場合に、システムは成功事例と同一の規則を適用すれば良いからである。しかしながら、このように過度に特殊化された知識もまた、実際の問題解決に貢献することは期待できない。

以上の事柄から、システムは与えられている探索制御知識を、探索に貢献するように適度に特殊化しなければならない。そのために、説明に基づく特殊化 (*Explanation Based Specialization*; EBS) の手法 [1] を用いている。本システムでは、解法過程のトレースが訓練事例であり、過度に汎化された戦略知識が目標概念である。

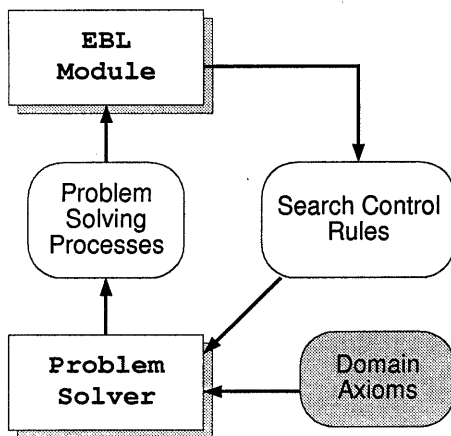


図4 探索制御知識獲得システムの構成

上述した機構を実現するために、システムは、基本的に次のモジュールから構成される。(1) 問題解決器、(2) EBS による知識の特殊化モジュール、(3) 領域知識。システムの基本的な構成を図4に示す。

5.2 問題解決器

システムには、問題解決器として、基本的に前向きのプロダクションシステムが実装されている。これは、現時点において例として想定されているドメインが、数学の代数における方程式の解法であることによる。

問題解決は、与えられた課題(方程式)にマッチするプリミティブ規則を適用して、式を変形させる行為である。式が特定の形(目標)になるまで、規則の適用が繰り返される。プリミティブ規則の形式および例を図5に示す。各プリミティブ規則は、(1) その規則が対象とする方程式のパターン(ターゲット部: Inquiry)、(2) ターゲット部の構成に関する制約条件(制約条件: PreConditions)、(3) 規則の適用により変換された結果の式のパターン(結論部: NextInquiry)、(4) 変換のための手続き(手続き部: PostConditions)から構成されている。

問題解決器は、個々の規則のターゲット部と与えられた課題(方程式)のパターンを比較し、マッチする規則を検索する。ターゲット部が一致する規則は、次に、制約条件をチェックする。

ターゲット部および制約条件がともに満足された場合には、手続き部を実行する。手続き部は、新たな課題の連言である。図5に例示した規則のターゲット部は、`partially_calculable(A,B,C,D)`, `calculate(B,[E])`, `substitute(A,E,F,C,D)` という3つの課題から構成されている。

プリミティブ規則の形式

```
op( ID,
    Inquiry,
    PreConditions,
    PostConditions
) :-
    NextInquiry.
```

プリミティブ規則の例

```
op( 0003,
    equation( e( L, R ), X ),
    [],
    [
        partially_calculable(L,Lp,Tree,Tp),
        calculate( Lp, [Lp1] ),
        substitute(L,Lp1,Li,Tree,Tp)
    ]
) :-
    equation( e( Li, R ), X ).
```

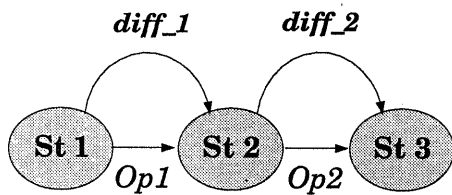
図5 プリミティブ規則

ターゲット部に現れるある種の課題は、それを直接解決するための手順が関数(Black Box)として実装されている。例えば、上記の `partially_calculable(A,B,C,D)` という課題は、多項式Aにおける任意の2つの同類項、もしくは、Aに含まれる計算可能な分数式の項を求める関数を用いて直接解決することができる。直接解決するための関数が存在しない場合、その課題は、副目標(subgoal)として再帰的に解決される。

前向き探索の場合、探索が目標に向かわずに発散する場合がある。そこで、本システムでは、次の制限を設けることにより、探索の発散を抑制している。(1) 探索時間の制限、(2) 状態の繰り返しの禁止、(3) 状態の差異の繰り返しの禁止。(1)は、初期状態からの規則の適用回数の制限である。(2)は、規則の適用の結果、問題解決の履歴の中で同一の状態(方程式の例では、同一の方程式)に遷移する場合には規則の適用を失敗とする。(3)は、規則の適用前と適用後の差異を定義し、直前の差異と現在の差異に変化がない場合には、規則の適用を失敗とする(図6参照)。

5.3 目標概念の特殊化

システムは、問題解決の事例に基づいて、探索制御知識(目標概念)を特殊化する。一般に、EBLの枠組では、訓練事例を一般化することにより、新たな知識を獲得する。その際、どの程度、知識を汎化するかという問題を考える必要がある。最も汎化された概念は目標概念であり、最も特化された概念は訓練事例である。一般化の程度を制御するためにバイアスと呼ばれる知識を利用することが一般的である。



If *diff_1* is equal to *diff_2*
Then *Op2* is failed.

図 6 差分に基づく規則適用の失敗

本システムでは、特殊化に対するバイアスとして、証明スキーマ (Proof Schema) と呼ばれる知識を適用する。証明スキーマは、解法事例に合わせて、目標概念をどの程度特化させるかという事柄を規定する。図 7 に規則適用の成功を示唆する探索制御知識 (op_succeeds) の証明スキーマを示す。

各目標概念は、次の形式である。ps(*ID*, *TC*, *Body*)。ここで、*TC* は目標概念を表し、*Body* は目標概念を特殊化させるための規範を表している。システムは、成功した問題解決の事例を証明スキーマを用いて説明する。説明とは、証明スキーマの *Body* を証明する行為に該当する。

図 7 では、op_succeeds に対する特殊化の規範として、2つの例が示されている。1つは、規則の適用により直接解が求まる場合であり、他方は、規則を適用した結果の状態で、特定の規則適用が成功することが分かっている場合である。

applicable(*Node*, *Inq*, *Op*) は、あるノード (*Node*) における課題 (*Inq*) に対して、規則 *Op* の条件部および制約条件が満足されることを表す。システムは、プリミティブ規則を解釈することにより、applicable 知識を予め生成する。solves(*Inq*, *Op*) は、*Inq* に対して *Op* を適用した結果遷移したノードが、解としての条件を満たすことを表す。

child_node_after_applying_op(*ChildNode*, *ChildInq*, *Node*, *Inq*, *Op*) は、ノード *Node* における課題 *Inq* に対して規則 *Op* を適用した結果遷移したノードが *ChildNode* であり、そこでの新たな課題が *ChildInq* であることを示す。これは、EBL で最も基本的な変数束縛の逆伝搬 (regression) を行なわせるための手続きである。

5.4 EBL による探索制御知識の学習例

ここでは、EBS に基づいた探索制御知識の学習例を示す。問題解決過程の例として、次の状況を想定する。

Node; Problem	[Op]
n3; equation(e([m(1,x)], [2,1]), A)	[op4]
n2; equation(e([m(1,x)], [3]), A)	[op7]
n1; equation(e([m(1,x)], [3]), 3)	[finish]

この時、ノード n2 において、次の探索制御知識が学習されている (詳細は省略)。

```
op_succeeds(N, equation(e([m(1,D)], [B]), C), op7)
[
  match_preconds([realp(B)]),
  known(N, [B=C]),
  solves(equation(e([m(1,D)], [B]), C), op7)
]
```

この状況でシステムは次に、ノード n3 における規則 op4 の適用事例に基づいて、図 7 に示した os4 を特殊化する。os4 にノード n3 の状況を代入すれば、次のようになる。大文字ではじまる項は、値の束縛されていない変数を表す。

```
op_succeeds(n3, equation(e([m(1,x)], [2,1]), A),
             op4)
[
  applicable(n3, equation(e([m(1,x)], [2,1]), A),
             op4),
  op_succeeds(CN, CI, NextOp),
  child(CN, CI,
        n3, equation(e([m(1,x)], [2,1]), A), op4)
]
```

ノード n3 の子供のノードで既に特殊化されている探索制御知識を代入すれば、次のようになる。

```
op_succeeds(n3, equation(e([m(1,x)], [2,1]), A),
             op4)
[
  applicable(n3, equation(e([m(1,x)], [2,1]), A),
             op4),
  op_succeeds(n2, equation(e([m(1,D)], [B]), C),
             op7)
  child(n2, equation(e([m(1,D)], [B]), C),
        n3, equation(e([m(1,x)], [2,1]), A),
        op4)
]
```

```

% An operator (Op) succeeds if it directly solves the inquiry (Inq).
ps( os01,
    op_succeeds( Node, Inq, Op ),
    [
        applicable( Node, Inq, Op ),
        solves( Inq, Op )
    ] ).

% An operator succeeds if it succeeds after precursor operator is applied.
ps( os04,
    op_succeeds( Node, Inq, Op ),
    [
        applicable( Node, Inq, Op ),
        op_succeeds( ChildNode, ChildInq, NextOp ),
        child_node_after_applying_op( ChildNode, ChildInq, Node, Inq, Op )
    ] ).

```

図 7 証明スキーマの例

最後に、`applicable` および `n2` における `op_succeeds` を展開することにより、次の探索制御知識が獲得される。

```

op_succeeds(N,equation(e([m(1,x)], [2,1]),A),op4)
[
    known(N,[ partially_calculable(C,E,F,G),
                calculate(E,[H]),
                substitute(C,H,I,F,G)
            ]),
    match_preconds([realp(J)]),
    known(NC,[J=D]),
    solves(equation(e([m(1,B)], [J]),D),op7),
    child(NC,equation(e([m(1,B)], [J]),D),
        N,equation(e([m(1,B)],C),D),
        op4)
]

```

6 おわりに

ここで提案した手法では、システムが自らの問題解決の経験を通して探索制御知識を獲得する。また、獲得された探索制御知識は、システム自身の問題解決に利用可能である。そのために、人間にとって極めて自然な知識が獲得されていると考えられる。

本稿で提案した枠組を学習者モデル診断システムに組み入れることにより、領域の定理や公式という効率化知識を原理的に説明することが可能になる。これは、効率化知識を機械的に暗記している学習者に対して、理解を促進させる場合に極めて有効であると思われる。

本稿では、説明に基づく特殊化 (EBS) による探索制

御知識の獲得を中心に述べた。システムは、問題解決過程において適用されたプリミティブ規則の系列から適切なマクロ規則を生成しなければならない。そのようにして生成されたマクロ規則を教育に利用した場合に、本研究で目的とされている手続き的問題解決のための学習者モデル診断システムが機能する。マクロ規則の生成および教授への適用手法に関しては、別の機会に報告したい。

本稿で述べた探索制御知識獲得システムは、Sparc Station 上の CESP 言語を用いて実装されている。CESP 言語の提供に関しては、AI 言語研究所の方々に多大なるご協力を賜わっていることに感謝致します。

参考文献

- [1] Steven Minton. *Learning Search Control Knowledge: An Explanation-Based Approach*. Kluwer Academic Publishers, Boston, 1988.
- [2] 松田昇, 岡本敏雄. 仮説推論に基づく ITS-学習者モデルとその認識方法について. 信学論 A, Vol. J75-A, No. 2, pp. 266-274, 1992.