人工知能学会誌 Vol. 5 No. 6 (Dec. 1990)〔別刷〕

論 文

Technical Paper

学習された戦略知識による教材知識の生成

Generation of Instructional Knowledge with Learned Strategy Knowledge

山田 誠二* 辻 三郎*
Seiji Yamada Saburo Tsuji

* 大阪大学基礎工学部制御工学科
Dept. of Control Engineering, Faculty of Engineering Science, Osaka University, Toyonaka, Osaka 560, Japan.

1989年10月30日 受理

Keywords: ICAI, instructional knowledge, generation of a text, learning, macro-operator.

Summary

We suggest the method to generate the instructional knowledge with the instructional knowledge acquired by the learning system. Our method can classify the given problems, decide the order in which the problem groups should be taught, and generate the easily understandable text.

At first, a teacher gives the learning system included in the ICAI system the input consisting of basic operators, problems, worked examples. The basic operators are primitive operators for solving problems, and the worked examples are sequences consisting of applied operators in the problem solving. From these input, the leaning system can obtain the strategy knowledge for controlling the applications of operators. The strategy knowledge are macro-operators and heuristic operators.

The classification of problems are done by the analysis of worked examples. We assume that the problems classified into the same group should need the same strategy knowledge. The strategy knowledge are represented by the set of operators applied in the worked example. Therefore, we can classify the problems by the applied operators. Next, the order for teaching the problem groups is determined. Since we consider that the easy problems should be taught at first, the teaching sequence is determined in the order of the smaller strategy knowledge. Furthermore, the layered worked examples are generated by the expansion of applied macro-operators. Finally, from all gained instructional knowledge, the ICAI system can generate the text with the worked examples which students can easily understand.

1. はじめに

現在，知的CAIについてさまざまな研究が行われている[(1)-(4)]が，それらの多くは，学習者モデルの獲得・更新または教授知識に関するものである．しかし，実際には，学習対象教科に関する知識である「教材知識」もCAIの重要な要素である．ほとんどの知的CAI研究において，教材知識は教師により的確に与えられると仮定しているが，教材知識の内容は演習問題，基本オペレータ，オペレータの適用条件，問題のグループ分けとその教授順序など多岐にわたっており[(1)]，これらを教師がすべて適切に入力することは，かなりの労力を要する．さらに，この場合，教師は単に対象領域の専門家であるだけでなく，CAIシステム自身にも精通していることを要求される．よって，教師にとってできるだけ記述が容易な知識を最小限与えれば，あとはその入力を分析し，そこから種々の教材知識をシステム自身が自動的に生成できるような知的CAIシステムが望まれる．つまり，教師はまずCAIシステムを教育し，学習したシステムが今度は人間の学生を教育するということが実現する．

本論文では，学習システム：PiLが教師から学習した戦略知識を用いて，教材知識を自動生成する手法について述べる[(5)]．本手法では，教師にとって記述が容易と考えられる，基本オペレータ，問題，解法例から，教材知識が生成される．

本論文で提案される新しい手法を以下にあげる．

(1) 問題解決時に適用されたオペレータによる類似問題のグループ化．

(2) 難易度による問題グループの教授順序の決定.
(3) マクロオペレータの展開による解法過程の階層化.
(4) 学習者の熟達度に応じた解法例を含むテキストの生成.

次節以降で処理の詳細について実際の処理結果を交えて説明していく. なお, 本システムの対象領域は一次方程式, 二次方程式, 簡単な分数・指数・対数方程式であるが, 説明の簡略化のため本論文では, 一次方程式に限定して述べていく.

2. 学習システム：PiL

教材知識の生成手法を組み込んだ知的 CAI システムの構成は, Fig. 1 のようになる. 従来の知的 CAI システムの構成と異なる点は, 教材知識と教師の間に学習システムが介在していることである. 本論文で提案される手法は, Fig. 1 の破線矢印の部分であるが, まず本節で学習システムについて簡単に説明する.

今回用いた学習システムは, われわれが構築してきた問題解法における戦略知識学習システム：PiL [(6)(7)] で, その特徴は, ①説明に基づく一般化：EBG [(8)], ②選択的マクロオペレータ学習 [(9)], である. PiL の入力は, 基本オペレータ, 問題の集合, そして教師が実際に問題を解いた過程を表す解法例である. Fig. 1 からもわかるように, これらの入力が, CAI システム全体の入力になっている. 入力の具体例は, 文献 (6) (7) を参照. 入力である基本オペレータは, $A=B \rightarrow A+C=B+C$ (A, B, C は, 任意の整式) などの公式の集合である. しかし, この基本オペレータの条件部は, 任意の等式に対して成り立ち, 適用可能である. さらに, 結論部の C は一意に決まらず, 無限個の候補が考えられる. よって, この基本オペレータをそのまま使って問題解決したのでは, $2x=3 \rightarrow 2x+5=3+5$ のようなむだな展開が数多くされ, その結果, 探索空間が爆発的に広がり解には至らない. なぜわれわれが問題を解けるかというと, 基本オペレータをより洗練 (特殊化) したものを使うことにより, 無意味な探索を避け, さらに複数の基本オペレータを一つにまとめたマクロオペレータを使って探索空間を縮小し, 効率を上げているからである.

PiL は, 解法例において適用された基本オペレータの条件部を伝播させていくという EBG の手法を用いて, 基本オペレータを精錬したヒューリスティックオペレータ [(6)] を学習することができる. このヒューリスティックオペレータにより, 無意味な探索がほとんどなくなる. さらに, ヒューリスティックオペレータを複数個まとめたマクロオペレータを獲得し, 効率化を図る. マクロオペレータは解法例のサブシーケンスから生成されるが, その候補は非常に多く, PiL では完全因果性 [(9)] というヒューリスティックを用いて, 有効なサブシーケンスを選択している. その結果, 人間が獲得するものと類似したマクロオペレータが学習可能であることが実験的に確かめられている [(6)(9)]. 以降, マクロオペレータとヒューリスティックオペレータを「戦略知識」と呼ぶ. これらの戦略知識は, そのまま教材知識の一部となる (Fig. 1).

一次方程式において PiL が学習したすべての戦略知識を Fig. 2 に示す. 図中の (DS) の付いたマクロは, ワンステップで問題を解けるもので DS マクロと呼ぶ. 他の記号については, 文献 (6) 参照. また, マクロオペレータはそれぞれが, どのヒューリスティックオペレータで構成されているのかが, 記録されている. 学習後の PiL は, 一次方程式のほとんどのパターンの問題を解けることから, Fig. 2 のオペレータが一次方程式で必要な戦略知識全体をカバーしていると見

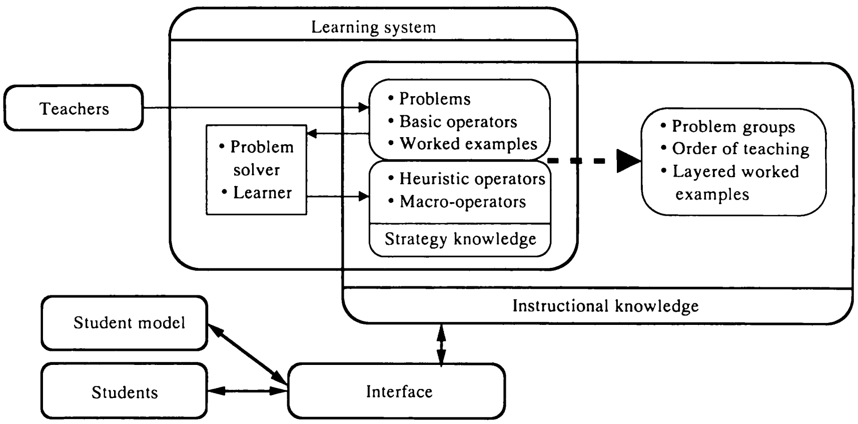


Fig. 1 The structure of ICAI system with a learner.

<Macro-operators>
(R1*AL+R2)/NR1 → R3*AL+R4
(R1*AL)/NR1 → R2*AL
A*R1 − A*R1 → 0
NR1*AL+R1=R2*AL+R3 → 1*AL=R4 (DS)
R1=NR1*AL+R2 → 1*AL=R3 (DS)
R1=NR1*AL → 1*AL=R2 (DS)
R1*AL+R2=R3*AL → 1*AL=R4 (DS)
(NA*AL)/NA → 1*AL
(R1*AL+R2)*R3 → NR1*AL+R4
AL*R1+AL*R2 → NR1*AL
R1*AL=R2*AL+R3 → 1*AL=R4 (DS)
NR1*AL+R1=R2 → 1*AL=R3 (DS)
NR1*AL=R1 → 1*AL=R2 (DS)

<Heuristic operators>
0*A → 0, R1*R2 → R3, R1 − R1 → 0, R1+R2 → R3, NA/NA → 1, R1/NR1 → R2
(R1*AL+R2)/NR1 → (R1*AL)/NR1+R2/NR1
R2*AL+R3=R1*AL → R1*AL=R2*AL+R3
(R1*AL+R2)*R3 → R1*AL*R3+R2*R3
B*R1+B*A → (R1+A)*B, R2=NR1*AL+R1 → NR1*AL+R1=R2
R1*AL=R2*AL+R3 → R1*AL − R2*AL=R2*AL+R3 − R2*AL
NR1*AL+R1=R2*AL+R3 → NR1*AL+R1 − R1=R2*AL+R3 − R1
0+NR1*AL → NR1*AL, NR1*AL+R1=R2 → NR1*AL+R1 − R1=R2 − R1
R1=NR1*AL → NR1*AL=R1, (A*AL)/A → (A/A)*AL

Fig. 2 Learned strategy knowledge for equations of the first degree.

なせる．以降で用いる戦略知識は，すべて Fig. 2 で示されているものである．

3．教材知識の自動生成

以下の四つの教材知識の自動生成について説明していく．

• グループ化された問題
• 教材の教授順序
• 階層化された解法例
• テキスト

なお，入力である問題の集合は，市販の問題集 [10] の一次方程式の章の文章題以外のすべての問題である 54 問を抜粋して用い，各節でその処理結果が示される．

3・1 問題のグループ化

数学のテキストの 1 単元は，通常例題とそれを解いた過程である解法例，練習問題で構成されている．例題は，新しい解法が必要な問題で，その解き方を解法例で示すことにより未学習の解法を学習者に理解させる．そして，例題と同様の解法で解ける練習問題が付随しており，学習者はそれらを自分で解くことで，解法の修得を確かなものにする．もし，個々の問題がランダムに与えられていたのでは，学習者は組織的に学習できず，解法パターンの修得は困難である．従来の知的 CAI システムでは，グループ化された問題を事前に与えていたものがほとんどである．そこで，ここでは問題の集合を入力として与え，それらの問題をシステム自身がグループ化する手法について述べる．

グループ化の基準は，「同じ戦略知識の集合で解ける問題は，同一グループとする」というものである．まず，学習システムによる戦略知識の獲得後，それらを用いて問題をすべて解き直す．この際，マクロオペレータ，ヒューリスティックオペレータという優先順位で適用が行われ [6]，すべての問題はこの 2 種類のオペレータで解ける．そして，解法の過程であるオペレータシーケンスをもとに分類を行う．

〔1〕 **問題解決に必要な戦略知識の抽出**

すべての問題を解き直して得られたオペレータシーケンスから，その問題を解くために必要な戦略知識を表す集合：NSK を以下の手順で求める．

(1) オペレータシーケンスから適用されたオペレータのラベルのリスト：OPS を求める．このとき，複数回適用されたオペレータは，重複した要素となる．
(2) OPS の要素のうち，マクロオペレータはヒューリスティックオペレータまで展開して，それを OPS に追加．このとき，もとのマクロは削除せず，残しておく．なお，マクロがマクロを含んでいる場合は，再帰的に展開．
(3) 重複しているラベルを一つ残して他を削除したものを集合と考え，それを NSK とする．

NSK の等しい問題を同一グループとする．具体例として，Fig. 3 に実際にオペレータシーケンスから NSK を求めた過程を示す．図中の Mn, Hn はそれぞれマクロ，ヒューリスティックオペレータのラベルを表し，(n) は上記の処理 (n) の結果である．こうして得られる NSK には，以下のような特徴がある．

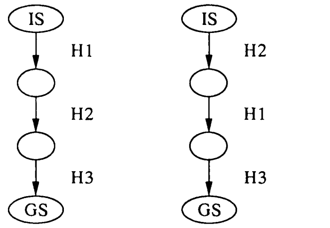
(a) NSK はオペレータの適用順序に依存しない．これは，その問題を解くために必要な戦略知識が重要であり，その適用順序はあまり意味がないことを反映している．Fig. 4 の二つのオペレータ

10x−3x=2+12
▼ M1 :10x−3x → 7x
7x=2+12
▼ H1 :2+12 → 14
7x=14
▼ M2 :7x=14 → x=2
x=2

M1 consists of [H2,H1].
M2 consists of [H3,M3,H4].
M3 consists of [H5,H6].

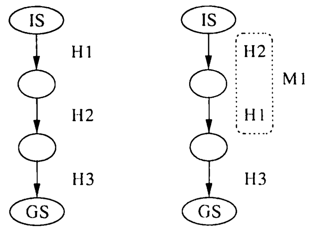
(1) OPS=[M1,H1,M2]
(2) OPS=[M1,H2,H1,H1,M2,H3,M3,H5,H6,H4]
(3) NSK={M1,H2,H1,M2,H3,M3,H5,H6,H4}

Fig. 3 Generation of NSK.



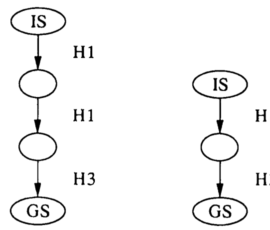
NSK : {H1,H2,H3} = NSK : {H1,H2,H3}

Fig. 4 Application order.



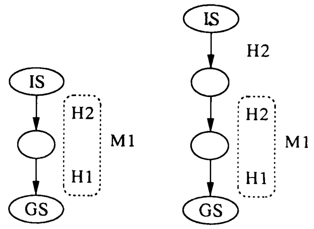
NSK : {H1,H2,H3} ≠ NSK : {H1,H2,H3,M1}

Fig. 5 A macro operator.



NSK : {H1,H3} = NSK : {H1,H3}

Fig. 6 Plural applications.



NSK : {H1,H2} = NSK : {H1,H2}

Fig. 7 Application of operators expanded from a macro operator.

PG0 [16]: 5+x=12, 12+x=5, x+0.5=2.4, 2/3+x=−1/3, x−8=5, x−2/3=1/3, 4x+5=17, 6x−10=−4, −4x+5=13, −3x+7=−14, 3 −2x=7, 4−5x= −11, 1/3x+2=5, 1/4x −1/2=2/3, 2/3x+4=10, −3/4x+2/3=17/12
PG1 [2] : 11=x −3.5, −3/5=x −2/5
PG2 [6] : 4x=24, −3x=12, −4x=8, 1/3x=6, 1/5x= −2, −2/3x=24
PG3 [2] : 12=6x, 12=1/2x
PG4 [2] : 2x=x −3, 4m=21−3m
PG5 [1] : 15x−3x=36
PG6 [5] : 4x−10=2x, 5−8y=2y, t+3=4t, x+2=3/4x, 2/3y −1=2y
PG7 [10] : 5x −9=8x+3, 2x −17= −7−3x, 7+4x=5x−12, 19 −4x=5x+9, 4x−10=3x−6, x−11=−6−4x, 22−x=x+34, 5y−4=7y+3, −5+6y=25−3y, 4x+3=2x+3
PG8 [5] : 2x−(x−3)=5, 4x−(3−x)=7, 3x+2(x−5)=−20, 6x−2(x−5)=16, 3x−2(4x−7)+21=0
PG9 [3] : 3(x−4)=5x+6, 4(2x −1)=5(x+4), 2(x−3)+5=3(x+2)
PG10 [1] : 5−2(7x − 2)=2
PG11 [1] : 4(t−1)+3(3t+5)=3t

Fig. 8 Classified problems.

シーケンスは，同一グループになる（図中の Mn, Hn, IS, GS はそれぞれ適用されたマクロ，ヒューリスティックオペレータ，初期状態，目標状態を示す．Fig. 5〜7, 9, 10 でも同様）．

(b) マクロも NSK の要素にすることによって，NSK 中のヒューリスティックオペレータが同じでも，マクロオペレータが異なると NSK が一致しないため別グループとなる．つまり，すでに学習済みのヒューリスティックオペレータの組合せであるマクロオペレータの学習も，一段階進んだ学習として区別する．Fig. 5 ではヒューリスティックオペレータは同じであるが，マクロが異なるので，別グループとなる．

(c) (3)により，同じオペレータが複数回適用されても，NSK には影響がない (Fig. 6)．これは，同じオペレータが何回適用されようとも，新たな戦略知識の学習にはなっていないと考えることによる．

(d) (2)により，マクロオペレータ中のヒューリスティックオペレータが再度適用されても，NSK に影響がない (Fig. 7)．学習済みのマクロオペレータを構成するヒューリスティックオペレータは，すでに学習されていると考えるからである．

実際に 54 問の一次方程式の問題についてグループ化を行った結果を Fig. 8 に示す．図中では PGn がグループ名で，[ ] 内がそのグループに属する問題数，そのあとが問題である．ここでは入力された 54 問の問題が，12 のグループに分類された．

〔2〕 例題の抽出

得られた各問題グループは，同一の解法を修得する

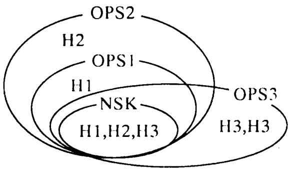


Fig. 9 Example problems.

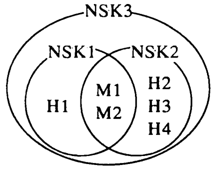


Fig. 10 Review problems.

< The order for teaching >
{PG2, PG3} : 4x=24
PG5 : 15x−3x=36
{PG0, PG1} : 5+x=12
PG8 : 2x−(x−3)=5
{PG4, PG6} : 2x=x−3
PG7 : 5x−9=8x+3

< Review problems >
PG9 : 3(x−4)=5x+6
PG10 : 5−2(7x−2)=2
PG11 : 4(t−1)+3(3t+5)=3t

Fig. 11 The order of teaching and reviews.

ための問題の集合と考えられる．よって，グループ中の一つの問題を例題とし，同じグループの残りの問題を練習問題とすることができる．ただしこのとき，どの問題を例題とするかが重要である．

例題は，解法ステップ数の多い難しい問題を解かせて計算力をつけさせるためのものではなく，解き方の理解が目的である．よって，理解すべき戦略知識が含まれている問題であれば，解決ステップ数は少ないほうが，つまり簡単な問題のほうが例題として好ましい．いま，同一グループの各問題の NSK は同じなので，教授すべき新たな戦略知識は，同一グループ中のすべての問題の NSK に含まれている．よって，同一グループの問題のうち，OPS の最小のもの，つまり最も簡単な問題を例題とする．例えば，Fig. 9 は何れも NSK = {H1, H2, H3} の同一グループの三つの問題を表しているが，その中で OPS の要素数最小の OPS1 が例題として選ばれる．また，前述の Fig. 8 では，各グループの問題は簡単な順になっており，先頭の問題が例題である．

そして，同一グループの例題以外の残りの問題を練習問題とする．このとき，練習問題も OPS の小さい順に並べる．これにより，練習問題においても簡単なものから徐々に複雑な問題へと，段階的な復習が可能になり，解法の修得が円滑に行われる．

3・2 問題グループ間の教授順序

グループ化の後に，それらをどのような順序で教授すべきかが問題になる．水道方式[11]などの特殊な例を除いて，通常の教育では，最も基本的な問題を最初に教え，徐々に複雑な問題を教授していく．これは，われわれの手法においては，学習者の修得している戦略知識（学習者モデル）と NSK との差異が最小の問題グループから，順次教えていくことに対応する．本手法では，学習者モデルとして，すでに教示した戦略知識の集合というオーバレイモデル[1]を用いる．よって，学習者モデルの初期状態は空集合であり，問題を教授するごとに，その問題を解くのに必要であった新たな戦略知識（=NSK）を学習者モデルに追加していく．学習者モデルと問題グループとの差異は，具体的には問題グループの NSK と学習者モデルの差集合の要素数で評価する．ただし，差異が 2 以下である問題グループどうしは，一つのグループにまとめて教授する．これは，わずかな差異によって必要以上に細かい「単元」が生成されるのを防ぐためである．

順序付けを行う具体的なアルゴリズムを以下に示す．

＜教授順序決定アルゴリズム＞

学習者モデル：S，教授順序リスト：T，復習問題リスト：Rのすべての初期値を空リストに設定する．また，問題グループのリスト：Pの初期値をすべての問題グループのリストで設定する．

(1) Pが空なら終了．T, R が出力．
(2) P中の問題グループそれぞれについて，その NSK とSの差集合：$\mathrm{NSK} \cap \overline{\mathrm{S}}$ を求め，その要素数をDとする．
(3) D＝0 の問題グループをPから削除し，Rに追加．
(4) Dが 2 以下の問題をPから削除し，Tの最後の問題グループと一つにまとめる．
(5) P中の問題グループでDが最小のもののうち一つ：MPG をPから削除し，Tの最後に追加．また，MPG の NSK をSに追加し，(1)に戻る．

(3)で新しい戦略知識を含まない問題グループ：Rが生成されることに注目してほしい．このような問題グループは，例えば Fig. 10 のような場合に生成される．この図は，三つの問題グループ：PG1〜3 の各 NSK の包含関係を表しているが，まず PG1 が教授されたとする．すると，Sには {H1, M1, M2} が含まれるので，そのSと残る PG2，PG3 との差異は両方とも {H2, H3, H4} で等しい．つまり，差異最小のグルー

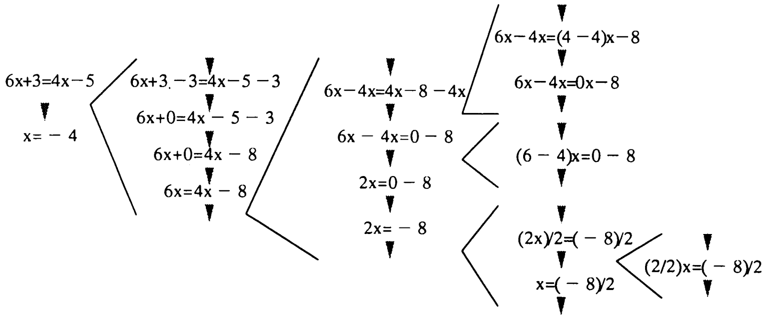


Fig. 12 A layered worked example (▼ : operators).

プが二つあることになる．このとき，仮に PG2 が次の教授問題グループとして選択されれば，教授後のSと PG3 は，差異がなくなり，教授しなくてもよくなる．このような問題グループは，新しい解法の教授には関与しないが，学習済みの知識を確かなものにするために役立つので，「復習問題」として扱われ，テキスト生成時に，章末にまとめて載せられる．

以上のアルゴリズムにより得られた結果が，Fig. 11 である．図中の問題は各グループの例題を表す．PG2 と PG3, PG0 と PG1, PG4 と PG6 がそれぞれ一つのグループにまとめられている．これは，PG3, PG1, PG6 がそれぞれ「両辺を入れ換える」だけで，PG2, PG0, PG4 と同じ解法で解けるので，差異が 2 以下になったためである．

3・3 解法例の階層化

解法例がマクロオペレータを含んでいる場合は，それを展開することにより解法例の階層化が可能になる．まず，演習問題がシステムに与えられた場合，PiL の問題解決システムがその問題を解く．このとき，マクロオペレータが優先的に適用され問題が解かれ，解法例が構成される．マクロオペレータは，複数のヒューリスティックオペレータあるいはマクロオペレータで構成されているので，いま得られた解法例中で適用されているマクロオペレータをより詳細なレベルのオペレータシーケンスに展開することができる．階層化された解法例の例を Fig. 12 に示す．

3・4 テキストの生成

本節では，以上のようにして得られた問題グループ，問題グループの教授順序，階層化された解法例，を用いてテキストを自動生成することについて述べる．

〔1〕 テキストの構成

一般に，数学のテキストは，以下の要素で構成されている．

章
　単元：例題，例題の解法例，練習問題
　章末：復習問題

ここで「章」とは，方程式，不等式，図形などの基本的に異なる問題分野の単位で，「単元」とは，同じ章の中において基本的には問題の性質は同じだが，解法が若干異なる問題分野の単位と考えられる．

本論文では，入力の問題は，「章」ごとに与えられると仮定して，与えられた問題を「章」に分類することはやらない．また，単元は問題グループに対応し，例題，練習問題，復習問題はそれぞれ 3・2 で得られている．あとは，例題の解法例を生成すれば，テキストが構成できる．

〔2〕 良い解法例とは？

テキスト[10]における例題の解法例を見ると，全体をすべて詳細に記述しているのではなく，すでに学習済みの部分は省略していることがわかる．この省略に加え，未学習の部分を詳細に教示することにより，平板な解法例ではなく，学習すべきポイントの明確な解法例となる．

階層化された解法例を用いて，このようなめりはりのある例題の解法例を生成することができる．つまり，すでに学習されている解法の部分は，マクロオペレータを使っておおまかな解法例を生成し，未学習の部分はヒューリスティックオペレータを用いて，詳細な解法例を作ればよい．

〔3〕 テキストの生成

例題の解法例を基にテキストの生成を考える．例題の解法例生成では，3・2 で用いた学習者モデルを使い，すでに学習されたマクロの部分は詳細化せずに，まだ学習されていないマクロの部分を詳しく表示する．生成アルゴリズムを以下に示す．

<テキスト生成アルゴリズム>

3・2 で得られた教授順序リスト：Tを [G1, G2,…, Gn] とし，i=1 とする（ただし，Gi は 3・1〔2〕の方法で，簡単な順に並んでいるとする．つまり，先頭は例題）．また，学習済みのオペレータのリスト：LSK

【A worked example 1】

4x=24
▼ h
(4x)/4=24/4
▼ h
(4x)/4=6
▼ h
(4/4)x=6
▼ h
x=6

< Exercises >

(1) −3x=12　(2) −4x=8
(3) 1/3x=6　(4) 1/5x=−2
(5) −2/3x=24　(6)12=6x
(7)12=1/2x

【A worked example 2】

−3x+15x=36
▼ h
(−3+15)x=36
▼ h
12x=36　In the same way to WE1
▼ DS
x=3

【A worked example 3】

x+5=12
▼ h
x+5−5=12−5
▼ h
x+0=12−5
▼ h
x+0=7
▼ h
x=7

【A worked example 4】

2x−(x−3)=5
▼ h
2x−1*x−1*(−3)=5
▼ h
2x−x−1*(−3)=5
▼ m
x−1*(−3)=5
▼ h
x+3=5　In the same way to WE1
▼ DS
x=2

【A worked example 5】

2x=x−3
▼ h
2x−x=x−x−3
▼ m
x=x−x−3
▼ h
x=(1−1)x+3
▼ h
x=0x−3
▼ h
x=0−3
▼ h
x=−3

【A worked example 6】

5x−9=8x+3
▼ h
5x−9+9=8x+3+9
▼ h
5x+0=8x+3+9
▼ h
5x+0=8x+12
▼ h
5x=8x+12　In the same way to WE5
▼ DS
x=−4

【Reviews】

(1) 5−2(7x−2)=2　(2) 3(x−4)=5x+6
(3) 4(2x−1)=5(x+4)　(4) 2(x−3)+5=3(x+2)
(5) 4(t−1)+3(3t+5)=3t

Fig. 13　The generated text.

を空リストに設定．また問題グループは，それぞれNSKを持っているとする．

(1)　i＝n なら，**3・2** で得られた復習問題をテキストの最後に付けて終了．

(2)　LSK に Gi の NSK 中のヒューリスティックオペレータを追加．

(3)　Gi 中の先頭の問題（例題）を LSK を用いて問題解決システムで解く．このとき，マクロ，ヒューリスティックオペレータの順に優先的に適用されるとする．また，Gi の残りは練習問題とする．

(4)　問題解決により得られた解法過程を例題の解法例として，テキストを生成する．最後に，DS マクロで解かれている場合は，そのマクロが学習された例題：WEn を参照する文章（In the same way to WEn）を表示する．これは，「以下は例題 n と同様に解けます．」という意味である．

(5)　Gi の NSK 中のすべてのマクロオペレータを LSK に追加．

(6)　i ← i＋1 として，(1)に行く．

このアルゴリズムで得られたテキストの例を Fig. 13 に示す．図中では，例題 1 のみ練習問題を付けており，他は省略している．解法例中の矢印の右に付いている記号は，h：ヒューリスティックオペレータ，m：マクロオペレータ，DS：DS マクロを表す．

まず，例題 1 は最初の問題なので学習者はマクロオペレータをまったく学習していない．よって，その解法例はすべてヒューリスティックオペレータで構成され，全体が詳細に説明される．しかし，次の例題 2 では解法例の 3 行目以降は，例題 1 と同じであるのでその旨を表示して割愛されている．これにより，例題 2 で新しく学習されるべきオペレータシーケンスである 1 〜 3 行目の「変数項をまとめる」部分が強調される．

さらに，例題 4 の 3 〜 4 行，例題 5 の 2 〜 3 行では，例題 2 で学習済みの「変数項をまとめる」というマクロを用いて省略がされている．このように，各例題で学習すべき解法シーケンスを強調した解法例を生成することにより，わかりやすいテキストの生成が実現されている．

また，学習システム：PiL は，問題解決システムを持っているので，学習者の要請により，任意の練習問題や復習問題について，その解法を例題と同様に適度な省略を行って教示できることを強調しておく．

〔4〕 生成されたテキストの評価

生成されたテキストを元の問題集 [10] と比較するこ

<PG1> **2** : 4x=24 **2** : –3x=12 **2** : –4x=8 **2** : 1/3x=6 **2** : 1/5x=–2 **2** : –2/3x=24 **2** : 12=6x **2** : 12=1/2x

<PG2> **4** : –3x+15x=36

<PG3> **1** : x+5=12 **1**: 12+x=5 **1**: x+0.5=2.4 **1** : 2/3+x=–1/3 **1** : x − 8=5 **1**: x–2/3=1/3 **3** : 4x+5=17
3 : 6x–10=–4 **3** : –4x+5=13 **3** : –3x+7=–14 **3** : 3–2x=7 **3**: 4–5x=–11 **3**: 1/3x+2=5
3 : 1/4x–1/2=2/3 **3**: 2/3x+4=10 **3** : –3/4x+2/3=17/12 **1** : 11=x–3.5 **1** : –3/5=x–2/5

<PG4> **6** : 2x–(x–3)=5 **6** : 4x–(3–x)=7 **6** : 3x+2(x–5)=–20 **6** : 6x–2(x–5)=16 **6** : 3x–2(4x–7)+21=0

<PG5> **4** : 2x=x–3 **4** : 4m=21–3m **4** : 4x–10=2x **4** : 5–8y=2y **4** : t+3=4t **4** : x+2=3/4x **4** : 2/3y–1=2y

<PG6> **5** : 5x–9=8x+3 **5** : 2x–17=–7–3x **5** : 7+4x=5x–12 **5** : 19–4x=5x+9 **5** : 4x–10=3x–6
5 : x–11=–6–4x **5** : 22–x=x+34 **5** : 5y–4=7y+3 **5** : –5+6y=25–3y **5** : 4x+3=2x+3

<reviews>
6 : 5–2(7x–2)=2 **6** : 4(t–1)+3(3t+5)=3t **6** : 3(x–4)=5x+6 **6** : 4(2x–1)=5(x+4) **6** : 2(x–3)+5=3(x+2)

Fig. 14 The comparison with a drill.

とにより，評価を行う．主に問題のグループ化とその教授順序を元の問題集と比較していく．

元の問題集では，54 問の問題が六つに分類されていた．問題集の分類と生成されたテキストの分類の関係を Fig. 14 に示す．この図は生成されたテキストの問題グループ：PGn と復習問題：reviews を表しているが，個々の問題の前に付いている太字の数字 n が，その問題が問題集中では n 番目のグループに属することを表す．

・問題グループについて

全体的には，グループ分けは，問題集と近いものが得られているといえる．まず，グループ数が六つ（復習問題を除く）と，元の問題集と全く一致している．これは，本手法のグループ分けの詳細度が適切なことを示している．

問題集では別グループに分類されていた問題が混在しているのは，PG3 だけである．このグループでは，x の係数が 1 である問題と 1 でない問題が同一グループになっているが，もとの問題集では区別されていた．この違いは，われわれのシステムでは，両者とも同一のマクロオペレータ（Fig. 2 の NR1 ∗ AL+R1=R2 →1 ∗ AL=R3）で解かれるため，全く同じ NSK をもってしまうことによる．また，PG2 は 1 問だけであるとか，問題集中の 6 番目のグループが，PG4 と復習問題に分かれているとかの若干の違いがある．

・教授順序について

問題集中の 2, 4, 1, 3, 6, 4, 5, 6 の順で PG の教授順序が構成されている．これは，大まかに問題集と一致していると考える．

4．ま と め

知的 CAI に機械学習システムを組み込み，その学習システムによって得られた戦略知識を基に種々の教材知識を生成する手法について報告した．その結果，本システムでは，教師にとって入力が容易な形式のデータから教材知識を CAI システム自身が獲得することが可能となった．さらに，われわれの手法は，オペレータシーケンスの解析をその基盤にしているので，オペレータが具体的にどのような操作を行うか，またどのように表現されるのかに依存しない．

また，本論文で提案された手法は，教師の教材知識生成モデルと見ることもできる．そして，このモデルは「教師は，その問題を解くために必要な戦略知識集合の包含関係に基づき，問題の依存関係を把握する」という仮説に基づいている．本手法により妥当な実験結果が得られたことは，われわれの仮説を裏付けるものであると考える．

謝　　辞

本研究の契機を与えてくださった大阪大学産業科学研究所の安部憲広助教授，東京工業大学工学部の沼尾正行助手，また，原稿を読んで有益な指摘をしてくれた石川智浩氏と辻研 FAI グループの皆さんに感謝します．

◇ 参 考 文 献 ◇

（1） 大槻，山本：知的 CAI のパラダイムと実現環境，情報処理学会誌，Vol. 29, No. 11, pp. 1255-1274 (1988).

（2） 河合，他：論理プログラミングと帰納推論による汎用知的 CAI システム，情報処理学会論文誌，Vol. 26, No. 6, pp. 1089-1096 (1985).

（3） 中村，他：類推ドメイン原理を用いた学生モデルの構築手法について，電子情報通信学会研究会報告，Vol. 88, No. 7, TE-88-1, pp. 65-72 (1988).

（4） 岡本，他：知的 CAI のための教授世界知識の表現とその推論の方法，電子情報通信学会論文誌，Vol. J 70-D, No. 12, pp. 2658-2667 (1987).

（5） 山田，辻：EBL とマクロオペレータ学習を用いた知的 CAI，電子情報通信学会研究会報告，AI-98-11, pp. 69-76 (1989).

(6) 山田，安部，辻：問題解決における戦略知識学習システム：PiL——一次方程式・不等式でのケーススタディー，人工知能学会誌，Vol. 3, No. 2, pp. 206-215 (1988).

(7) 山田：問題解決における戦略知識の学習，大阪大学大学院基礎工学研究科博士論文 (1989).

(8) Mitchell, T. M., Keller, R. M. and Kedar-Cabelli, S. T.: Explanation-Based Generalization: A Unifying View, *Machine Learning,* Vol. 1, No. 1, pp. 47-80 (1986).

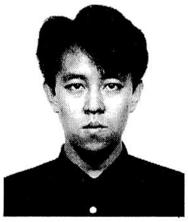
(9) Yamada, S. and Tsuji, S.: Selective Learning of Macro-Operators with Perfect Causality, *Proc. of IJCAI-89,* pp. 603-608 (1989).

(10) アタック 2001「数学中１」，文研出版，pp. 43-48 (1988).

(11) 遠山，長妻：水道方式の授業展開，国土社 (1964).

〔担当編集委員：中村順一，査読者：竹内　章〕

著者紹介



山田　誠二（正会員）

1984 年大阪大学基礎工学部卒業．1989 年同大学院博士課程修了．同年より基礎工学部制御工学科助手．工学博士．人工知能，特に，説明に基づく学習，マクロオペレータ学習，制約プログラミング，時制推論に関する研究に従事．情報処理学会，日本認知科学会会員．

辻　　三郎（正会員）

1953 年大阪大学工学部卒業．1955 年同大学大学院修士課程修了．同年，電子技術総合研究所入所．1970 年より大阪大学基礎工学部制御工学科教授．工学博士．現在，人工知能，ロボティクス，コンピュータビジョンなどの研究に従事．情報処理学会，電子情報通信学会，計測自動制御学会，IEEE 各会員．