

LISP-CAIシステムにおける 診断と計画機能

加藤 等

田中淳志

渡邊豊英

吉田雄二

名古屋大学 工学部

プログラム言語LISPを題材にCAIシステムを試作した。知的CAIシステムにおける専門知識モジュールや個人指導モジュールの構造・機能は、十分に解明されていない。本稿では、専門知識の構造について検討し、各学習単位を関係リンクでつなぐ教材構造を提案する。また、個人指導モジュールのタスクである解答診断の方法として、LISPインタプリタ関数EVALを用いる方法を提案する。この方法により、S式の評価問題に関してはシステムが自分で正解を得ることができ、関数定義（プログラム作成）問題においても学生の多様な答案プログラムを診断できる。さらに、個人指導モジュールのタスクである指導スケジューリングの一方法を提案する。

Diagnosis and scheduling functions on LISP-CAI system

Hitoshi KATO, Atsushi TANAKA, Toyohide WATANABE and Yuuji YOSHIDA

Faculty of Engineering, Nagoya University

Furo-Cho, Chikusa-Ku, Nagoya 464-01, Japan

We developed a prototype CAI system which teaches the programming language LISP. Construction methods or functional abilities for domain knowledges and pedagogical knowledges are not clear so far. In this paper, first we study a representation structure of the domain knowledges composed of knowledge units and relationship links among them. Second, we propose a diagnosis method using the LISP interpreter function EVAL. By this method, our system can check automatically whether students' answers are correct and can diagnose the answers by simulating the S-expression step by step. Additionally, we investigate an experimental method of scheduling tutoring strategies.

1. まえがき

知的C A Iの研究目標は、学生個人の理解状況に合わせて的確に教授できる機構を実現することであるが、現在必ずしも十分に対処できているわけではなく、多くの課題があり、また明確でない機構が多い。学生個人々の理解状況を表すために学生モデルを用いることが知的C A Iの特徴であるが、学生モデルの枠組みは明確でない。また、個人指導モジュール、専門知識モジュールも必ずしも機能・構造が明らかになっていない。教授内容の教材に応じて、また教授方法に応じて研究されている。

そこで、以下の問題について検討し、LISPの基本概念を教授するC A Iシステムを開発した。

- 1) 専門知識モジュールはどのような構造を持つべきか
- 2) 学生の解答をどのように診断すべきか
- 3) どのように学生を指導していくべきか

本稿では、1)に関して、いくつかの学習単位とその間を関係リンクでつないだ構造を提案する。2)に関しては、LISPがデータとプログラムを区別しないという性質を利用する。この特徴は、知的C A Iシステムを構築する上で有用な手段となる。すなわち、学生の解答を診断するために、LISPシステムが自分で学生の解答プログラムを実行することによって、正答か誤答かを判断できる。3)に関しては、システム内に指導スケジュールを保持し、それによって教授する機能を実現する。

本稿では、LISPを教授するシステムについて、第2章でまずその概要を、第3章で教材の構造化法を述べる。さらに、第4章で診断法について、第5章で指導計画の立案機能について述べる。

2. LISP-C A Iシステムの概要

試作したLISP-C A Iシステムは、LISPの基本操作を教授する目的で開発された。本章ではシステムの設計目標と構成について概観する。

2. 1 システムの機能

本システムは、LISP以外のプログラム言語を使用したことのある学生を対象に、LISPの基本的な知識を教授する。すなわち、具体的なプログラミング技法を教授するのではなく、LISPにおける基本概念、および基本関数の機能概念を説明し、学生に問題を解かせることにより個人指導を実現する。

試作したシステムの主な機能は、以下のようである。

- 1) 学習項目、例題、問題についてさらに詳しく説明（複数の説明が可能）
- 2) マウスによるメニュー選択、エディタによる答案の入力、複数のウィンドウを用いたシステム出力（操作インタフェース）
- 3) LISPインタプリタ・システムを用いて、学生の答案（プログラム）を診断
- 4) システムは、教材の構造や学生モデルに基づいて指導計画を立て、それによって教授（学生が自分で自由に学習項目を選ぶことも可能）
- 5) 学生の質問（メニュー形式）に対し、専門知識を参照して応答

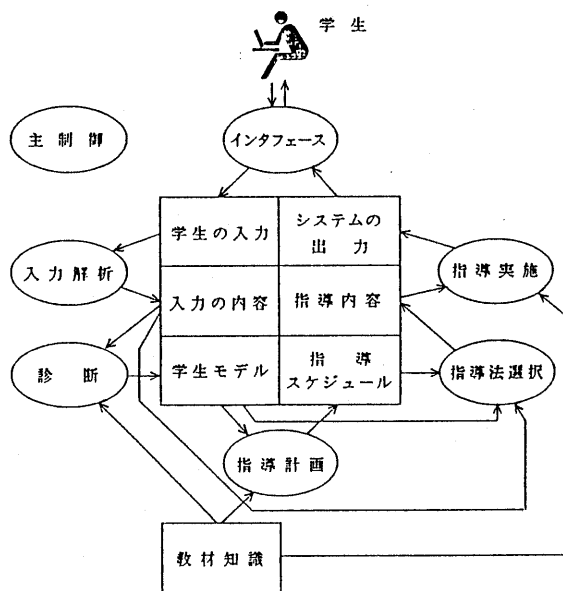


図1 システムの概要

2.2 各手続き部のタスク

図1にシステムの概要を示した。データ領域は現在の指導状況を表し、各手続き部は特定のデータ領域を参照して自分のタスクを実行する。

なお、本システムでは、学生モデルを学習項目に対する知識の有無のフラグ、および理解度からなるオーバーレイ・モデルとして管理する。

以下には、各手続き部のタスクについて述べる。

1) インタフェース

学生との直接的なデータ授受を行う。

2) 入力解析

入力の内容を解析し、学生の意図を認識する。

3) 診断

学生の解答を診断する。

4) 指導計画

システムの指導スケジュールをたてる。

5) 指導法選択

次に行う指導法を選択する。

6) 指導実施

選択された指導案に従って、指導を実施する。

3. 教材の構造化

知的CAIシステムにおいては、教材は学生に教授すべき対象教科に関する知識を保持しているモジュールであり、システムはこの教材を用いて学生に知識を教授する。

本稿では、CAIにおける一般的な教科に依存しない教材の構造を考察し、その下にLISP教材を構築した。

3.1 教材構造

どの内容をどのような順序で教えるのかということは、教育効果に重大な影響を与える。よって、教授順序に従って教材をある単位に分割し、各教材単位にはその適用条件が示されるべきである。

図2に我々が検討・設計した教材の構造を示す。教材は、いくつかの教材項目、例題、図、表、問題からなり、教材項目が一つの概念知識を表す教材単位である。教材項目、例題、問題、図、表の

それぞれが、指導上の単位と考えられ、これらを学習単位と呼ぶ。各学習単位には、以下の関係リンクが張られる。

1) 目的知識

例題、問題、図、表に設定し、目的とする知識（教材項目）を示す。

2) 参照知識

教材項目に設定し、内容の似た知識を示す。

[例] 関数EQと関数EQUAL

3) 関連知識

教材項目に設定し、関連する知識を示す。

[例] 教材項目「関数定義」と関数DEFUN

4) 必要知識

すべての学習単位に設定され、あらかじめ学習すべき知識（前提知識）を示す。

教材の階層構造は、主に章と節によって表される。一つの節はいくつかの学習単位から構成される。さらに、節の集まりが章をなし、章の集まりが教材全体を構成する。本システムでは、フレームの階層構造によって教材の階層構造を実現している。

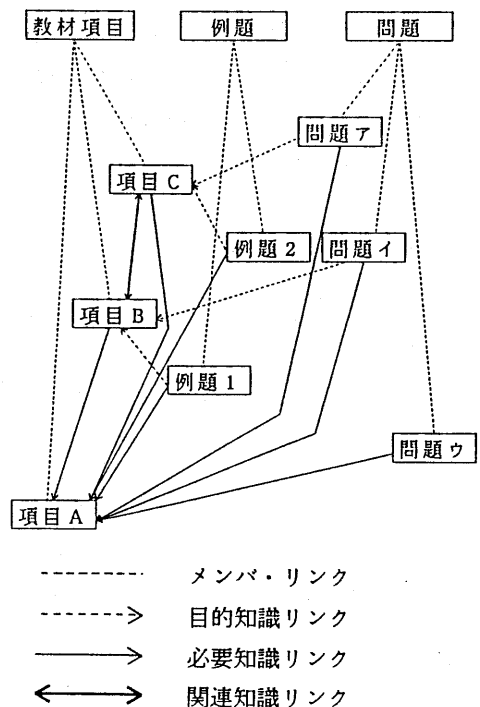


図2 教材構造 (図、表は省いた)

3.2 LISP教材

3.1で述べた教材構造を基に、LISPに関する教材を作成した。具体的なプログラミングを教えるのではなく、LISPの基本概念と、基本関数の機能概念を教材として実現した。よって、教材項目として、例えば関数定義、述語、条件式、再帰的定義、それぞれの基本関数などを対象とする。また、関数を表す教材項目として、COND、EQ、CAR、CDRなどを対象とする。

4. 解答の診断

学生の解答を調べることは、CAIシステムにおいて重要である。本章では、LISP教育システムで用いた学生の答案の診断方法について述べる。

4.1 問題の設定

効果的な指導を実現するには、学生の誤りを同定し、原因を明らかにする必要がある。そのためには、学生の解答を診断する機能が必要である。しかし、一般に診断方法は、問題によって異なる。つまり、問題の型が異なることによって、診断方法も変わる可能性がある。そこで、LISPの問題を作成するにあたって、LISPの入門書に掲載された練習問題などを分析した。その結果、以下の3種類に大別できた。もちろん、これに属さない問題もあるが、数が少ないので当面検討外とした。また、単純な穴埋め式の問題なども対象としない。

1) S式評価問題

S式を示して「評価するとその値はどうか」を問う。

2) 関数定義（プログラム作成）問題

「～する関数を定義しなさい」という問題であり、LISPの入門書の中で最も多かった。

3) 例題の改良問題

例題を示し、例題に出てきた関数の適用範囲を広げるなどの改良を施す。

このうち、3)は関数定義問題として捉えることができ、2)の型と同様に考えることができる。従って、ここでは1)と2)の二つの問題型を設定し、

これらに対して診断法を用意した。

4.2 LISPインタプリタ関数の利用

本LISP教授システムでは、学生の解答の診断のために、LISPのインタプリタ関数EVAL、およびエラー・ハンドラを利用している。以下に簡潔に述べる。

(1) 関数EVALによる評価

よく出題される問題の型の一つに、S式の値を問う問題がある。この問題に対しては、問題の内容をS式で与えておけば、正解をあらかじめ用意する必要はない。なぜならば、本教授システムは、LISPで記述されており、問題をS式データと見て、LISPインタプリタ関数が自ら評価できるからである。つまり、問題となっているS式を引数として関数EVALを呼ぶことにより、S式の値を得ることができる。例えば、

[問題] S式

```
(APPEND (LIST (CAR '(A B C)))  
(CDR '(D E)))
```

を評価するとどうなるか？

に対して、本システムでは、

```
(EVAL (APPEND (LIST (CAR '(A B C)))  
(CDR '(D E))))
```

を実行して正解(A E)を得ることができる。

従って、この型の問題は、システム自身で解くことができ、学生の解答が正解であるかどうかはすぐわかる。誤りであったときは、その原因を調べるために誤り診断過程に入る。

(2) エラー・ハンドラの利用

最もよく出題される問題の型に、簡単な関数を定義する問題がある。この問題に対する学生の解答はプログラムであり、エディタによって入力される。プログラムはそれを書く者によって異なるのが普通であり、その正誤の判断は人間の教師には可能でも、計算機には困難である。

そこで、システムがあらかじめ診断用のデータを用意しておく。学生が定義した関数を、引数としてデータを与えて実行して、正しい結果になれば正しい関数であると判断することにする。診断

データは、学生の誤りを見落とさないように、特殊なデータを含み複数用意し、しかも注意深く選ぶ必要がある。システムは、学生に提示するために正解を持っているので、実行後に得られるはずの正しい結果は、システムが自分でデータを評価して得ることができる。例えば、

[問題] 数だけからなるリストに対してソートを行うプログラムSORTを作りなさい。

[学生の解答] (DEFUN SORT (LIST)

(<プログラムの内容>))

[診断用データ] (3 1 2 4),

(100 -10 200 50 10),

(1)

[システムの正解] (DEFUN SORT-S (LIST)

(<プログラムの内容>))

次の二つを実行して比較する

(EVAL (SORT <診断用データ>))

(EVAL (SORT-S <診断用データ>))

しかし、この方法には問題があり、学生が定義した関数にエラーが存在するとプログラムの実行が停止し、教授システム自身も実行が中断される。本システムでは、LISPシステムのエラー・ハンドラ内の関数を利用して、エラーが起きても教授システムの実行を止めずに教授システムに制御を戻す方法を取る。この際に、発生したエラーの情報(どのようなエラーでどの関数の実行時に起こったかなど)を受け取り、誤りの認識に役立てる。この方法は、マシンなどの機構に依存するが、プログラムを診断するのに有用な手段である。

4.3 誤り診断の方法

ここでは、問題の型に対し、実際に学生の誤りをどのように診断するかについて述べる。

(1) S式評価の問題における診断

前節で述べたように、S式の値を問う問題に対しては、システムが学生の解答の正誤を簡単に決定できる。誤答の場合は、どうして誤ったかを同定するために、誤りの診断過程に入る。

学生の入力に誤りの場合、システムは問題のレベルが最も深いリストを左から順に一つずつ提示

し、評価するとどうなるかを学生に尋ねる。誤答の原因を求めるために、教材内に混乱しやすい関数に関する知識を保持している。それを各ステップにおいて問題の関数と入れ換えて評価し、学生の解答と一致すれば、学生がその関数と混乱していると見なす。これをレベル順にあげて繰り返すようにシステムが制御する。この方法により、関数の混同などの誤りを同定できる³⁾。

(2) プログラム作成問題における診断

簡単な関数を定義する問題においては、解答は(DEFUN <関数名> <引数> <フォーム>)の形式をしている。従って、まず解答がDEFUN式を満たしているか否かを調べる。ここで、DEFUN式を満たしていなければエラーが発生するから、教材項目「関数定義」および「DEFUN」における誤りとして処理する。形式を満たしていれば、解答診断用のデータを、システムが保持している正解と学生の定義した関数の両方に引数として与えて、結果を比較する。その結果が一致すれば、学生の解答は正解であると見なす。学生の関数を実行しているときに、エラーが発生すれば、エラー情報を抽出して原因となった関数と誤りの種類(引数の型、数など)を得る。そして、該当する関数項目に関する誤りとして記録する。例えば、

[学生の解答] (DEFUN 2ND (LIST)

(CDR (CAR LIST)))

[診断用データ] (A B C), ((1 2) 3 (4)), ()

この場合、最初のデータを与えたときにエラーが発生する。このエラーは関数CDRの引数の型に関するエラーで、学生は関数CDRの引数の型がわかっていないと診断される。

この例からもわかるように、この方法だけでは学生の誤りを的確に診断できたとはいえない。例では、学生は単に関数CARと関数CDRを混同して用いただけである。エラーの内容から的確に誤りを同定できるように、今後検討が必要である。

4.4 再教育のための指導案

前節では、学生の解答の診断法について述べた。解答を診断して誤りの原因を見つけたら、その誤

りを直すように指導しなければならない。

一般に、理解度が異なる学生に対する指導の内容は異なっているべきである。例えば、理解度のよい学生に対しては、誤りをそれとなく示唆すればよく、懇切丁寧に解説する必要はない。理解度のよくない学生に対しては、正解を示して解説するなどの指導が必要である。

試作システムでは、問題の目的知識リンクが指す教材項目の理解度に着目して以下のように指導案を決める。

- 1)教材項目の理解度がよい学生に対しては、簡単なメッセージによって誤りを示唆する。
- 2)理解度が普通、あるいは不明の場合は、誤りを直接指摘する。
- 3)理解度がよくない場合、教材項目を解説する。
- 4)ほとんど理解されていない場合、正解を提示する。

なお、最初は理解度がよい学生でも、問題を誤るたびに理解度は下げられるので、指導案が変わる。

5. 指導法のスケジューリングと選択

知的CAIシステムにおいて、個人指導モジュールは、次にどのように指導するかを決定し、その指導を実行するモジュールである。

人間の教師は、次に何をするかを決定するだけでなく、引き続きどのように指導していくかを指導計画として立案していると考えられる。本稿では、簡単な指導の計画機能を実現したので報告する。

5.1 指導過程におけるスケジューリング

指導スケジュールは、システムがどのように指導していくかという予定を表す。指導スケジュールの作成には、次の機能が要求される。

- 1)学生を指導し始めるときに計画をたてる。
- 2)原則として教材項目のやさしいものから始め、順次むずかしいものを選ぶ。
- 3)学生の質問などには、すぐ対応する。

4)学生に自由な教材項目の選択を許す。

5)各教材項目の知識を学ぶのに適当な例題、問題などを選ぶ。

6)スケジュールは現在の指導状況によって随時更新される。

7)学生の理解度に合った指導法を選ぶ。

8)スケジュールが空になったら、再度計画をたてる。

5.2 指導スケジュールの立案

前節2)、5)から、指導計画部は教材項目・例題・問題の必要知識リンク、目的知識リンクを参照する必要がある。すなわち、必要知識を先に学習するように教材項目を選び、その項目を目的知識とする例題や問題を選ぶ。

指導計画部は、以下の場合に呼び出されてスケジュールをたてる。

- 1)一般に学生を指導し始めるとき
- 2)スケジュールが空になったとき

指導スケジュールは一般に次の形をしている。

(指導案1 指導案2 ... 指導案n)

学生が自分で学習項目を選んで学習を始めるか、システムに指導を任せて学習を始めるかによって、最初にたてられる指導スケジュールは異なる。前者を学生モード、後者をシステム・モードと呼ぶ。いずれも指導の途中では、スケジュールを同じように更新する。それぞれにおいてスケジュールは次のように作成する。

●学生モード

- ①学生が学習したい教材項目を選ぶ。まずこれを指導案とする。
- ②同項目を目的知識とする図や表を指導案として前に、例題、問題を指導案として同項目の後に加える。この際、使用知識に未習のものがあれば加えない。
- ③同項目の必要知識があって、その中に学習されていないものがあるときは、そのうちの一つを選び、指導案として先頭につけ加える。必要知識がすべて学習されているときや、必要知識がないときは、立案は停止される。

④③で選ばれた項目について②③を行う。

[例] 図3のような教材において、学生が教材項目「再帰呼び出し」を選んだ場合は、以下のようにスケジュールをたてる。ただし、教材項目「述語」はすでに学習されているとする。

((学習 条件式) (学習 例題2-2)
(出題 問題2-2) (学習 再帰呼び出し)
(学習 例題2-3) (出題 問題2-3))

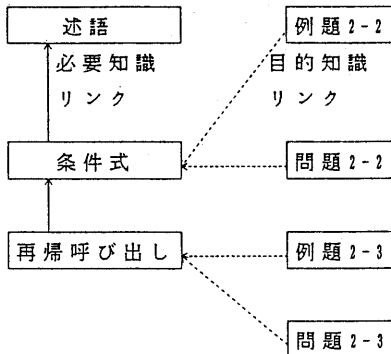


図3 LISP教材の例(指導計画説明用)

●システム・モード

- ①未習の教材項目を選ぶ。
- ②そのうちの、必要知識がないか、または必要知識が学習済みのものを選び、指導案とする。
- ③②において選ばれたものが二つ以上あるときは、難易度の低いもの、必要知識の理解度の高いものを選ぶ。
- ④③で選ばれた項目について、同項目を目的知識とする図や表を指導案として前に、例題、問題を同項目の指導案として後に加える。この際、使用知識に未習のものがあれば加えない。

[例] 図3のような教材において、システムは以下のようにスケジュールをたてる。ただし、教材項目「述語」はすでに学習されているとする。

((学習 条件式) (学習 例題2-2) (出題 問題2-2))

5.3 指導法の選択

指導法選択モジュールは、次の指導を選ぶ。選ぶ対象として、前節で述べたシステムの指導スケジュールの外、学生の解答の診断結果、および学生の要求(質問、参照、解説、問題の解答・ヒント

など)がある。これらのうち、どれを選ぶかは、今回単純に以下のようにしているが、他の選び方も可能である。

- 1)学生の質問、参照、解説、問題の解答・ヒントなどの要求には、すぐ答える。
- 2)学生の誤った解答を受け取ったら、診断結果を基に診断手続き部によって示された指導案を選ぶ。
- 3)学生からの要求、誤った解答の入力がなければ、スケジュールの先頭にある指導案を選ぶ。

6. LISP教授システムの実現法と実行例

本章では、LISP-CAIシステムの実現法とその実行例を示す。

6.1 システムの実現法

前章までに述べた機能に従って、試作LISP-CAIシステムを実現した。実現には、ワークステーションUNISYS/KS301上のCommonLISPと、エキスパート・システム構築シェルであるKEEを用いた。特に教材の階層構造と、学習単位の実現には、フレーム(KEEではユニットという)とその属性継承機能を利用した。

6.2 LISP指導の実行例

この節では、試作LISP教授システムの実行例を示す。

(1)システムの診断機能の例を図4に示す。この例では、システムは学生の誤った(再帰関数の停止条件がない)解答に、診断用データを与えて実行し、スタック・オーバフロー・エラーをエラー・ハンドラから受け取っている。そのため、学生に指導のためのメッセージを表示している。

(2)システムの計画機能の例を図5に示す。この例において、学生は教材項目「関数定義法」を選択したが、必要知識である「関数」、「関数の呼び出し」をまだ学習していない。そのために、システムはスケジュール

((学習 関数)

(学習 関数の呼び出し)

(学習 関数定義法))

をたて、これに従って指導を進めている(これらの教材項目には、例題や問題は設定されていない)。

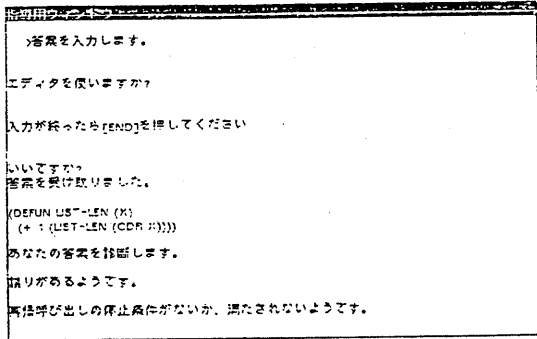


図4 診断機能の例

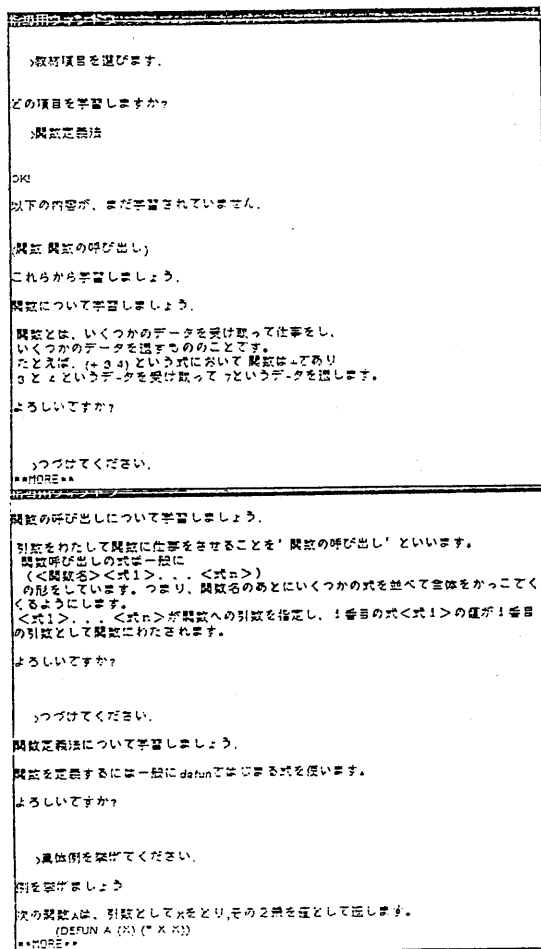


図5 計画機能の例

7. むすび

本稿では、LISPを題材にC A Iシステムを報告した。各学習単位を関係リンクでつなぐ教材構造を開発した。また、解答診断の方法として、LISPインタプリタ関数EVALを用いる方法を提案した。この方法により、S式の評価問題に関してはシステムが自分で正解を得ることができ、関数定義(プログラム作成)問題においても学生の多様な答案プログラムを診断できる。さらに、教材の関係リンクを用いた指導スケジューリングの方法を提案した。

今後の課題として、以下のものがあげられる。

- 1) エラー情報からの誤りの同定法の改良
現在の方法では、エラー・メッセージの情報しか利用していない。学生の誤りとプログラムに含まれるエラーとの関連を調査し、適切に誤りを同定するように改良する必要がある。
- 2) 用語集など教材の充実
用語に対する質問に答えられないため、用語集を用意する必要がある。
- 3) 自然言語による対話機能の充実
現在では、学生の入力ほとんどマウスによるメニュー選択である。自然言語による入力を許容する対話インタフェースが必要である。

謝辞 日頃より熱心にご指導いただいた中京大学 福村晃夫教授、名古屋大学稲垣康善教授、鳥脇純一郎教授、並びに研究室の皆様へ感謝します。

参考文献

- 1) 河合他：論理プログラミングと帰納推論による汎用知的C A Iシステム、情処学論、Vol. 26, No. 6, pp. 1089-1096(1985).
- 2) 竹内他：摂動法による学習者モデル形成と教授知識について、情処学論、Vol. 28, No. 1, pp. 54-63(1987).
- 3) 加藤他：LISP関数教授における誤り同定の方法、平成元年度電気関係学会東海支部連合大会講演論文集、pp. 512.