

2M-4

プログラミング教授を目的としたLisp CAIシステムにおける
プログラム診断法*

西口直樹† 渡邊豊英† 杉江昇†

名古屋大学工学部情報工学科‡

1 はじめに

プログラミング言語の文法を学習しただけのプログラミングの初心者にとって、問題に適した方法で、良いプログラムを簡潔に記述することは、困難である。プログラミングにはある種のテクニックが必要となるからである。プログラミング・テクニックの教授には、熟練者の経験的知識に基づいたノウハウが必要となる。また、学習者によって、その理解度が異なるため、対応の仕方も重要である。

学習者が記述したプログラムの動作と、正しいプログラムの動作には、学習者が初心者であればあるほど大きな隔たりが存在する。この隔たりを埋めるには、多くの例題を繰り返し学習することも重要であるが、プログラムを記述する際の考え方を学ぶことも重要である。そこで、学習者のプログラムの誤りを正すのみでなく、より良い記述法の指導もするプログラムの診断法を本稿では述べる。

2 プログラムの診断

プログラムの誤りは、文法的不一致と意味的相違に分けられる。文法的誤りの発見は、コンパイラにおける技術を用いれば可能である。しかし、意味的誤りの発見は、同じ結果をもたらすアルゴリズムが複数存在したり、同じ動作をするプログラムの記述法が複数存在するために、それらの可能性をすべて考慮に入れて誤りを見つけなければならない。特に、初心者は、簡潔で理解しやすいプログラムを書くとは限らない。

我々のLisp教授システムでの診断部は、文法や意味的誤りを発見する誤り検出、誤りの原因を同定する誤り診断、より簡潔で問題に適したプログラミング・テクニックを教えるテクニック指導から成っている。診断部の処理の流れを図1に示す。まず、誤り検出で、プログラムの状態を次の四つのクラスに分ける。

正 解 学習者のプログラムが正しい値を返し、正常に停止する状態

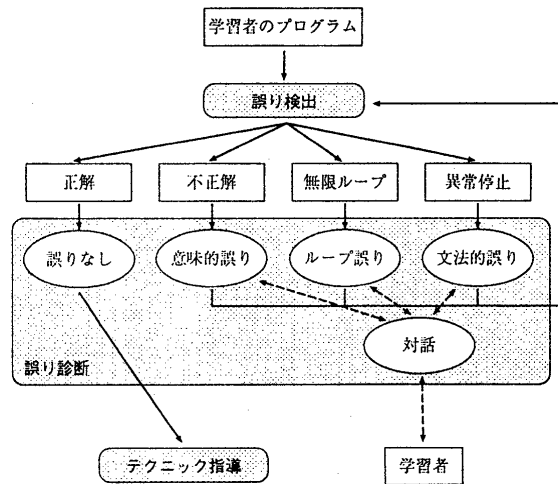


図1: 診断部の概要

不正解 学習者のプログラムは誤った値を返すが、正常に停止する状態
無限ループ 学習者のプログラムが停止しない状態
異常停止 学習者のプログラムが値を返すことなく、途中で停止する状態

このクラス分けは、実際にプログラムを評価することによって行なう。

次に、これらの状態に応じて、誤りの原因を同定する。不正解の時は、学習者のプログラムに意味的誤りが存在すると仮定して、意味的誤りを診断する。無限ループの時は、ループが終了しない原因を推定する。異常停止の時は、文法的誤りが発生したと仮定し、文法の誤りを推定する。誤りは一つとは限らないので、正解以外は、図1に示したように再び誤り検出に制御を移し、正解になるまで誤り検出、誤り診断を繰り返す。正解の時は、テクニック指導へと処理を移す。誤り診断において、誤りの原因が一つに同定できるとは限らない。誤りの原因が同定できない時は、学習者に同定できない原因を問い合わせる。

テクニック指導では、プログラムがより良くなるよう

*A Diagnosis Method of Programs on Lisp-CAI System for Programming Tutoring

†Naoki NISHIGUCHI, Toyohide WATANABE, Noboru SUGIE

‡School of Engineering, Nagoya University

に指導する。このテクニック指導により、ただ単にプログラムを組めるようになるだけでなく、より問題に適したプログラミングができるようになる。

3 意味的誤りの診断

一つの問題に対して多数存在する記述法をあらかじめすべて予想することは不可能である。そのため、学習者のプログラムが理解できない可能性がある。そこで、我々は、学習者との対話を介してこの問題を解決する。

まず、システムの解答プログラム群の中から、学習者がプログラムに用いたアルゴリズムと同じアルゴリズムのものを取り出してくる。このとき、複数のプログラム候補が存在する可能性があるが、候補を絞りきれない場合には学習者に問い合わせることにより解決する。次に、学習者のプログラムのどの部分に誤りがあるのかを調べる。ある処理を実行するプログラム(関数)は、その中でまた別の処理をするプログラムに分割できる。分割される処理ごとにプログラムを評価し、システムの解答プログラムと比較することにより、誤りの部位を同定する。その後、さらに処理を細かくしていき、誤りを完全に特定する。分割する場所が特定できない時は、学習者に問い合わせる。つまり、分割する際に生じた矛盾点を学習者に問い合わせて矛盾を解消し、分割を進める。分割の単位は、あらかじめシステムに用意された解答プログラムに依存して決定される。

4 テクニック指導

ここでは、学習者のプログラムがより簡潔で、問題に適したプログラムになるような指導を行なう。我々は、プログラムが、より簡潔で、問題に適しているということを以下のように考える。

簡潔とは 現在は、再利用も考慮してプログラミングをする必要がある。そのため、理解し易いプログラムを組むことが望ましい。現在、好ましいとされている記述法は構造化することである。プログラムを構造化すると、見やすくすっきりする。構造化されたプログラムを簡潔なプログラムと定義する。

問題に適しているとは 同じ処理を実行するプログラムでも、アルゴリズムの違い等によって、制御構造が大幅に異なるプログラムとなる。それらのプログラムの中で、処理速度、簡潔さ等の点から最も望ましいと思われるものを問題に適しているとする。

テクニック指導における処理の流れを図2に示す。はじめに、プログラムで用いられたアルゴリズムが何であるかを推定する。次に、そのプログラムに含まれる冗長な記述、より簡潔になる記述を探し、それらが存在すれば、学習者に提示する。最後に、問題に適したプログラムであるかを判定し、そうでなければ、より適したプ

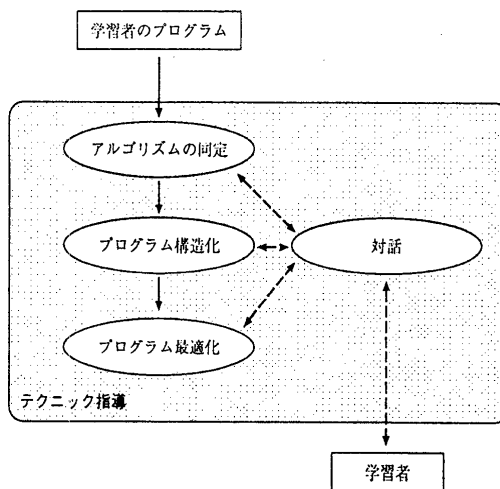


図 2: テクニック指導

プログラムをすべて、説明を加えながら提示する。

ただ単に、より良いプログラムを提示するだけでは、学習者には理解しづらいために、学習者の理解度に応じて、対応を変える。

5 おわりに

プログラミングの初心者を対象として Lisp-CAI システムのプログラム診断について述べた。学習者の犯す誤りを直すだけでなく、より簡潔で、問題に適したプログラムになるような指導を加えることにより、プログラミング・テクニックの学習を可能にする。

現在、我々は、ワークステーション UNISYS/KS301 上に Common Lisp で実装を進めている。実装が途中であるため、まだこの方法の有効性や問題点等は、評価できない。

参考文献

- [1] D.Sleeman & J.S.Brown: "Intelligent Tutoring Systems", Academic Press, Inc. (1982).
- [2] E.Wenger: "Artificial Intelligence and Tutoring Systems", P.432, Morgan Kaufmann Publishers, Inc. (1987).
- [3] 加藤, 田中, 渡邊, 吉田: "LISP-CAI システムにおける診断と計画機能", 情処研報, Vol.90, No.39, pp.9-16 (1990).