

初等アセンブラプログラミングにおけるニアミスプログラムに対するアドバイス文生成法  
 LN-5 A Method of Generating Advice Sentences for Novice Programs  
 Written in Assembly Language

渡辺 博芳† 高井 久美子† 荒井 正之† 武井 恵雄†  
 Hiroyoshi Watanabe Kumiko Takai Masayuki Arai Shigeo Takei

## 1. まえがき

初等プログラミング教育を担当する教員スタッフの作業負荷は大きく、学生に有効な指導を行うためには、このような負荷を少しでも小さくする必要がある。これまでに、我々は、初等アセンブラプログラミングを対象として、提示した課題に対して学生が作成したプログラムが課題の題意を満たしているかどうかの評価作業を支援する事例ベースシステムを開発し、実際の授業で用いることで教員の作業負荷の軽減に大きな効果があることを示した[1,2]。

次のステップとして、学生に対するアドバイスを支援するシステムの開発を検討している。まず、動作は正しくないが完成に近いプログラム(ニアミスプログラム)に対するアドバイスの支援を対象とし、アドバイス支援のための全体的なアプローチについて既に検討した[3]。本稿では、正しく動作するプログラムとニアミスプログラムの差異を解析してアドバイス文を生成する方法を提案する。

## 2. アドバイス支援のアプローチ

### 2.1 全体的な方針

学生にアドバイスを提示する際には、アドバイスの詳しくさとタイミングをどうするかという問題がある。例えば、正しく動作しないプログラムについて、教員は最初、「この辺がいけないんじゃない?」といった簡単なヒントを提示し、それでも分からない場合には、徐々に詳しいアドバイスを提示することが多い。そこで、アドバイスを提示する際に、その詳しくさとタイミングの問題を扱えるような提示方法をとる[3]。そのため、詳しくさのレベル別に、複数通りのアドバイス文を生成する必要がある。

全体としては、アドバイス文を生成する方法として「類似性に基づく方法」と「差異に基づく方法」の2つを検討する。類似性に基づく方法は、過去に行ったアドバイスの事例や正しく動作しないプログラムについてのアドバイスを事例形式で表現した知識とアドバイス対象プログラムの類似性に基づいてアドバイス文を生成する。一方、差異に基づく方法は、正しく動作するプログラムとアドバイス対象プログラムの差異を解析してアドバイス文を生成する。本稿では、後者について述べる。

### 2.2 差異に基づくアドバイス文生成の方針

アドバイス文生成においてベースとする「正しく動作するプログラム」は、我々が既に開発した事例に基づく評価支援システムの評価事例ベースに存在するプログラムが利用可能である。すなわち、評価事例ベースから、アドバイス対象となるプログラムに最も類似するプログラムを含む事例を検索し、検索された事例との差異が「採用条件」を満たせばニアミスと判断し、その差異を解析してアドバイ

ス文を生成する。

初等アセンブラプログラミングにおけるプログラム間の差異として、正しく動作する方のプログラムを基準とすると、以下のようなものが考えられる。

- (a) 誤命令：対応すると思われる命令の命令コードやオペランドに誤りがある。
- (b) 命令の不足：必要と考えられる命令が不足している。
- (c) 過剰な命令：不必要な命令が用いられている。
- (d) 順序の違い：命令を記述する順序が異なる。
- (e) ラベル位置の違い：分岐先に用いられるラベルが付与されている命令が異なる。

以上、5種類の差異を扱うものとし、差異の種類に応じたアドバイス文を生成する。

## 3. アドバイス文生成の流れと処理の詳細

### 3.1 処理の流れ

全体としては、事例の検索を行い、選択された事例を用いてアドバイス文を生成するといった流れになる。事例の検索では、アドバイス対象のプログラムリストと個々の事例の持つプログラムリストとの照合処理を行い、最も類似する事例を選択する。選択された事例のプログラムとの差異が「採用条件」を満たさなければ、アドバイス文の生成はできない。条件を満たす場合は、アドバイスの詳細さのレベルを指定してアドバイス文を生成する。

以後、アドバイス対象となるプログラムを「対象プログラム」、事例に含まれる動作の正しいプログラムを「ベースプログラム」と呼ぶ。

### 3.2 プログラム照合処理

プログラム照合処理は、2つのプログラムリストを入力として比較を行い、それらのプログラムリストの「対応関係情報」と「差異情報」を出力する処理である。「対応関係情報」はベースプログラムと対象プログラム間の命令の対応、レジスタの対応、ラベルの対応を示す対応表である。一方、「差異情報」は、2.2で述べた差異の種類ごとに以下のような情報を持つ。

- (a) 誤命令：「対応するベースプログラムの行番号」と「対象プログラムの行番号」、及び「誤り位置(命令コード、第1~3オペランドなど)」。
- (b) 不足命令：対応先の無いベースプログラムの行番号。
- (c) 過剰命令：対応先の無い対象プログラムの行番号。
- (d) 順序の違い：順序が異なる範囲の対象プログラムにおける行番号リスト。
- (e) ラベル位置の違い：「ベースプログラムの行番号」と「ラベルの違いの種類(どちらか一方にラベルが存在、または、ラベルが不一致)」。

プログラム照合処理は次のような流れで行う。

- (1) 対応候補の生成：命令コードが等しい命令は対応する可能性があるとして、命令の対応表の候補として設

†帝京大学理工学部, School of Science and Engineering,  
Teikyo University

定する。またその対応が成立する場合に、対応付けられるレジスタやラベルの対応を、それぞれレジスタの対応表とラベルの対応表の候補とする。

- (2) 候補との絞込みと対応関係の決定：ベースプログラムと対象プログラムは完全には照合しないはずなので、対応する個所を最大化するような照合戦略をとらなければならない。そこで、最も多くの命令の対応を成立させる条件となるレジスタ、またはラベルの対応を優先して決定する。
- (3) 差異の同定：まず、ベースプログラムと対象プログラムで対応先を持たない命令を、それぞれ「不足命令」、「過剰命令」として拾い上げる。次にそれらの比較を行い、一部修正により対応すると判断できるものを「誤命令」と判定し、「不足命令」と「過剰命令」からは削除する。最後に、命令の対応情報を用いて、「命令の順序の違い」と「ラベル位置の違い」を同定する。

### 3.3 アドバイス文生成処理

プログラム照合で生成した「対応関係情報」と「差異情報」を用いて、詳細さのレベルに応じたアドバイス文を生成する。最も簡単なヒント(レベル 1)では、差異の種類に関わらず、誤りのあると思われる行番号を提示し、レベル 2 以上では差異の種類に応じた文章を生成する。複数の種類の差異が存在する場合は、それら全てに対するアドバイス文を提示する。

最も簡単なヒント(レベル 1 のアドバイス)を提示するためには、まず、差異の存在する対象プログラムにおける行番号を求める。不足命令については、1つ前の命令が対応する対象プログラム内の行番号を用いる。次に、求めた行番号の列に連続する部分が含まれている場合は、その部分を「x行目～y行目」という表現に変換する。

レベル 2 以降は差異の種類に応じた文章を生成する。生成方法の詳細の説明は省略するが、「誤命令」については、おおよそ以下のようなアドバイス文が生成される。

- (レベル 2) x行目の命令に誤りがあるのでは？  
 (レベル 3) x行目の「命令コード」が誤っているのでは？  
 (レベル 4) x行目の「命令コード」は yy では？  
 他の種類では、レベル 3 までのアドバイス文を提示する。

## 4. 実験

### 4.1 実験内容

本学では CASL II を教材として初等アセンブラプログラミング演習授業を行っている。2001 年度の授業において、学生のプログラム提出時に収集した動作の正しくないプログラムは、提示した 13 課題に対して全部で 801 であった。これを事例検索のための実験システムを利用して、同様なプログラムをまとめた結果、バリエーションは 395 となった。一方、授業において収集した正しく動作するプログラムを用いて事例ベースを構築した。事例数は全部で 427 となった。

これまで述べた方法に基づいて、アドバイス対象プログラムを入力して、評価事例ベースを検索し、類似事例を得た場合に、全てのレベルのアドバイス文を出力するような実験システムを作成した。今回の実験では、事例の「採用条件」を以下のように設定した。

- ・差異同定前の段階で、不足命令と過剰命令の合計が 4 命令以下であること。
- ・命令の順序が異なる部分が 3 命令以下であること。
- ・レジスタの対応が全てとれていること。

昨年度、収集した動作の正しくない 395 プログラムに対してアドバイス生成を試みた。また、生成されたアドバイス文の一部について教員スタッフが A～D の 4 段階評価(A,B は実場面で利用可能)を行った。

### 4.2 実験結果

採用条件を満たす事例を得てアドバイス文を生成できたケースは、395 プログラムのうち 211(53.4%)であった。これは、1 度の課題提出で構築された評価事例ベースを用いることで、ニアミスプログラムへのアドバイスの約半分程度を差異に基づくアドバイス支援システムがカバーできる可能性を示唆している。

13 問のうち比較的簡単な 3 問について、アドバイス文の評価を行った。これらの 3 問に限ると、対象となった動作の正しくないプログラムは 71 であり、これらのうち、45(63.4%)でアドバイスが生成された。アドバイス文の評価結果は表 1 の通りである。教員が利用可能と判断したアドバイス文は 60%から 75%程度であった。

表 1 アドバイス文の評価結果

	レベル 1	レベル 2	レベル 3	レベル 4
A	25	21	21	10
B	9	9	9	2
C	4	7	5	3
D	5	7	8	5
合計	45	45	45	20
A+B	34	30	30	12
A+B の割合	75.6%	66.7%	66.7%	60.0%

## 5. むすび

初等アセンブラプログラミングにおける差異に基づくニアミスプログラムへのアドバイス方法を提案し、予備的な評価を行った。今後、教員が利用可能と判断するケースを増加させるために、プログラム照合処理、事例の採用条件、アドバイス文生成法に関して、更に調整を行い、再度、生成したアドバイス文の品質を評価したい。

本研究の一部は科学研究費補助金 No.14580428, 人工知能研究振興財団 研究助成 No. 12AI320-1 の補助による。

### 参考文献

- [1] 渡辺博芳, 荒井正之, 武井恵雄: 事例に基づく初等アセンブラプログラミング評価支援システム, 情報処理学会論文誌, Vol.42, No.1, pp.99~109, 2001.
- [2] Watanabe,H., Arai,M. and Takei,S.: Case-Based Evaluation of Novice Programs, Proc. of AI-ED2001, pp.55~64, 2001.
- [3] 渡辺博芳, 荒井正之, 武井恵雄: CPU とアセンブラ授業におけるプログラム評価とアドバイスの支援, 情報処理学会研究報告 2001-CE-59, Vol.2001, No.34, pp.69~74, 2001.