

プログラミング学習用ブルーフリーダのフィードバック機能の提案

2014SE019 弘中芳彬 2014SE027 稲生利一

指導教員： 蜂巣吉成

1 はじめに

大学の講義で行うプログラミング演習では、学習者は教育者から出題された課題に対し、実行例と同じ出力をするプログラムの作成によってプログラミング学習を進めていくことが多い。学習者の中には実行結果のみに着目し、課題の教育意図に合わないプログラムを作成して提出する者もいる。教育者が、学習者のプログラムが教育意図を満たしているかを確認し、学習者にフィードバックすれば学習効果はあるが、教育者の負担が大きい。

我々の研究室では、教育意図を満たしていないプログラムを、模範解答のプログラムと比較して発見するプログラミング学習用ブルーフリーダ(以下、ブルーフリーダ)を提案している [1][2][3][4]。ブルーフリーダは、ソースコードから教育意図に関する部分(以下、評価コード)を抽出して比較することで、学習者の書いたプログラムが教育意図を満たしているかを判定する。しかしこれまでのブルーフリーダは、学習者へのフィードバックについては十分に考慮されていなかった。

本研究の目的は、学習者が教育意図を満たすプログラムが書けるように、学習者へのメッセージ表示によるフィードバックを行うことである。本研究では、学習者の書いたプログラムが教育意図を満たしていない場合、プログラムの改善箇所の指摘を行うメッセージでフィードバックし、教育意図を満たしている場合は正しいという旨の結果を返す。メッセージ表示により、学習者は自身のプログラムをなぜ修正しなければいけないのかを考えるようになる。本研究で提案するフィードバック機能により、学習者の学習効率の向上が見込まれる。

2 関連研究

コーディングチェッカや学習者に向けた指摘を行うツールはいくつか提案されている。

proGrep[5]は、プログラミング教育において学習者の学習履歴を収集、分析し、抽出されたパターンに基づいて自動的にアドバイスを行う。コンパイラの出力に対してのアドバイスしか行うことができないので、学習者のコードが教育意図を満たしているかの判定はできない。

C-Helper[6]は、C言語初学者が犯しがちな間違いを、事前に決められたルールに基づき静的解析により発見、指摘し、可能な範囲であれば解決策の提案も行う。C-Helper内で定義されたルールの基で指摘をするので、各課題毎に存在する教育意図に対して

の指摘を行うことはできない。

Truongら[7]は、制御構造の観点から模範解答と判定対象の解答を比較し、差分がある場合にそれぞれのコードの差を埋めるようフィードバックを与えるツールを提案している。このツールでは比較対象の部分が文の構造のみなので、式などの細粒度の教育意図の判定はできない。

CX-Checker[8]は、コーディングルールのカスタマイズ機能を備えたC言語用のコーディングチェッカである。XPathやJavaを用いてルールを記述することができ、課題毎に独自のルールを定義してチェックを行うことができる。しかし、それぞれの課題に対する教育意図に合わせたフィードバックを行おうとする際は、教育者が一つ一つのルールを追加する必要があり負担が大きくなる。

3 ブルーフリーダの概略

ブルーフリーダとは、1章で述べた通り学習者のプログラムが教育意図を満たしているかを、模範解答を用いて判定を行うツールである。ブルーフリーダで扱う学習者のプログラムは、コンパイルに成功し、実行結果と同じ出力を行うものとする。

ここでは、文献[4]の「8個の実数値を配列に入力し、その平均値と最小値を出力するプログラムを作成しなさい」という課題を例にブルーフリーダについて説明する。この課題の教育意図は、「繰り返して計算される変数の繰り返しの前に初期化する」、「繰り返しの繰り返しにおいて不要な計算をしない」、「配列の先頭要素から最後の要素まで走査する」、「配列の先頭要素を処理した後で、次の要素から最後の要素まで走査する」、「for文内で冗長な計算をしない」、「関数を定義し、利用できる」、「マクロを定義し、利用できる」である。なお、「for文内で冗長な計算をしない」に関してはカスタマイズ機能[1]を用いて除算に関する評価コードを抽出し、判定を行う。この課題に対する模範解答のソースコード断片をソースコード1、評価コード断片をソースコード2に、学習者のソースコードをソースコード3、評価コード断片をソースコード4に示す。また、判定結果を図1に示す。

ソースコード1 模範解答のソースコード断片

```
1 #define DATASIZE 8
2 double avgDoubleArray(double d[],int size){
3     int i;
4     double sum;
5     sum=0;
6     for(i=0;i<size;i++){
7         sum=sum+d[i];
8     }
9     return sum/size;
```

```

10 }
11
12 double minDoubleArray(double d[],int size){
13     int i;
14     double min;
15     min=d[0];
16     for(i=1;i<size;i++){
17         if(d[i]<min){
18             min=d[i];
19         }
20     }
21     return min;
22 }
23
24 int main(void){
25     double data[DATASIZE];
26     double avg, min;
27     printf("input data:\n");
28     readDoubleArray(data, DATASIZE);
29     avg = avgDoubleArray(data, DATASIZE);
30     min = minDoubleArray(data, DATASIZE);
31     printf("avg=%f,\nmin=%f\n", avg, min);
32     return 0;
33 }

```

ソースコード 2 模範解答の評価コード断片

```

1 $var=0;
2 for(=0;<;++){
3 $var=;
4 }
5 /
6 $var=[];
7 for(=1;<;++){
8 $var=;
9 }
10
11 avgDoubleArray:2
12 minDoubleArray:2
13 DATASIZE:5

```

ソースコード 3 学習者解答のソースコード断片

```

1 #define DATASIZE 8
2 double avgDoubleArray(double d[],int size){
3     int i;
4     double s,ave;
5     s=d[0];
6     for(i=1;i<size;i++){
7         s=s+d[i];
8         ave=s/size;
9     }
10    return ave;
11 }
12
13 double minDoubleArray(double d[],int size){
14     int i;
15     double m;
16     m=200;
17     for(i=0;i<size;i++){
18         if(d[i]<m){
19             m=d[i];
20         }
21     }
22    return m;
23 }
24
25 int main(void){
26     double data[DATASIZE];
27     double avg,min;
28     printf("input data:\n");
29     readDoubleArray(data,DATASIZE);
30     avg=avgDoubleArray(data,8);
31     avg=avgDoubleArray(data,DATASIZE);
32     min=minDoubleArray(data,8);
33     printf("avg=%f,\nmin=%f\n",avg,min);
34     return 0;
35 }

```

ソースコード 4 学習者解答の評価コード断片

```

1 $var=[];
2 for(=1;<;++){
3 $var=;
4 /
5 }
6 $var=200;
7 for(=0;<;++){
8 $var=;
9 }
10
11 avgDoubleArray:3
12 minDoubleArray:2
13 DATASIZE:4

```

<<評価コードの比較>>

| <模範解答> | <学習者の解答> |
|---------------|-----------------|
| \$var=0; | < |
| for(=0;<;++){ | < |
| \$var=; | < |
| } | < |
| / | < |
| \$var=[]; | \$var=[]; |
| for(=1;<;++){ | for(=1;<;++){ |
| \$var=; | \$var=; |
| | > / |
| | > } |
| | > \$var=200; |
| | > for(=0;<;++){ |
| | > \$var=; |
| } | } |

| | | |
|------------------|--|------------------|
| avgDoubleArray:2 | | avgDoubleArray:3 |
| minDoubleArray:2 | | minDoubleArray:2 |
| DATASIZE:2 | | DATASIZE:4 |

図 1 判定結果

ソースコード 3 では、avgDoubleArray 関数内で合計を求める変数の初期値を配列の 0 番目の要素にし、配列の 1 番目から最後の要素まで走査を行っている。minDoubleArray 関数内では最小値の初期値に定数を代入し、繰り返しは先頭の要素から走査している。また、for 文内で繰り返し平均の計算をしている。さらには main 関数内で avgDoubleArray 関数の呼出しを余分に行っており、マクロの利用が適切に行えていない部分も存在する。この時、ソースコード 1, 2, 3, 4 の通り、プルーフリーダでは模範解答、学習者解答のそれぞれのソースコードから教育意図に該当するコードを抽象化した評価コードを抽出し、それらと比較することで図 1 のような結果を返す。評価コードの抽出により、変数の初期値は適切か、繰り返しの走査は適切に行えているかといったことがわかるようになる。しかし、図 1 に示す結果の表示方法では学習者に向けたフィードバックが十分に行えているとは言えない。現在のプルーフリーダでフィードバックを行うには次の 2 つの問題点がある。

1. フィードバック時のメッセージ内容

2. diff では正しく判定できない場合があること

1 は、課題毎に特有の要素があるがそれを評価コードから特定することができないので、課題の内容に応じたフィードバックを行えない。また、評価コードに違いのある行の指摘が行えないので、学習者がなぜプログラムを直さなければいけないのかわからない可能性がある。教育者が学習者に対して口頭で指導を行うことはできるが、負担が大きくなる。

2 は、diff コマンドは二つのファイルから最長共通部分列を基にファイル間の差分をチェックするので、図 1 のように意図しない部分が最長共通部分列に該当してしまうと正しい判定が行えなくなる。

4 フィードバック機能の提案

4.1 概略

学習者へ向けたフィードバックを行うには 3 章で述べた 2 つの問題点があり、それらを解決するために本研究ではメッセージ表示によるフィードバック機能を備えたブルーフリーダを提案する。ブルーフリーダは、模範解答、学習者解答の評価コードをそれぞれ抽出する。得られた評価コードを一行ずつ比較してプログラムの改善箇所を特定し、その内容に応じたメッセージを学習者にフィードバックする。

4.2 設計

本研究で提案するフィードバック機能を備えたブルーフリーダは、プログラミング演習などの場面で使用されることを想定して設計を行った。ブルーフリーダの全体像を図 2 に示す。

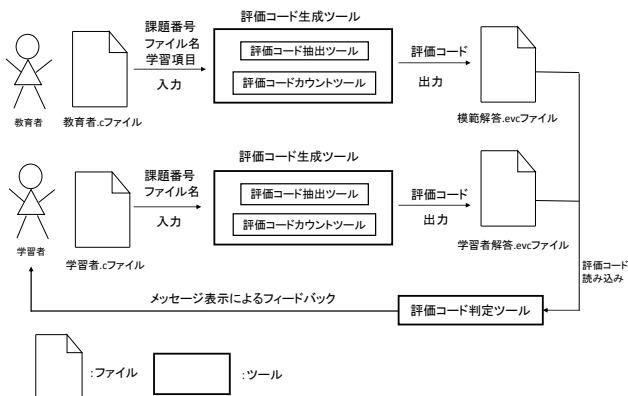


図 2 ブルーフリーダ全体像

4.3 評価コードの比較方法

評価コード判定ツールでは、図 3 で示す流れでプログラムの改善箇所を特定し、それに対応したメッセージ表示によってフィードバックを行う。

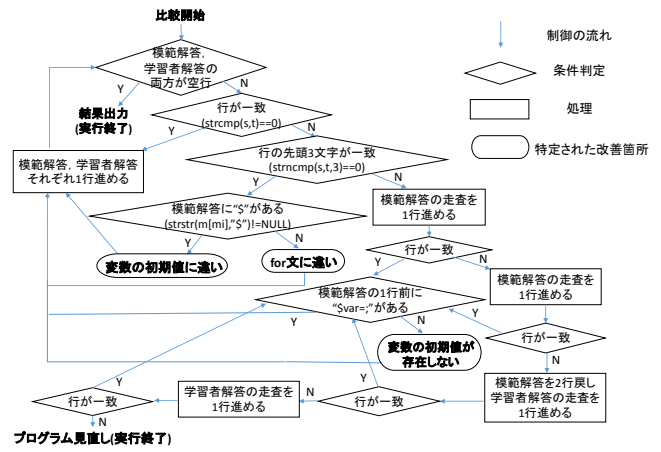


図 3 プログラム改善箇所特定

評価コード生成ツールで抽出される評価コードは、for, if, while の丸括弧内は制御式の演算子等が入り、その後はブロックの開始と終了の {} が続く。また、文字列 \$var= で始まる代入文は、= の後はリテラルか配列を表す [] が抽出される。そこで提案方法では、strcmp を用いて行の先頭 3 文字が等しいか否かを確認する。

図 3 中の「for 文に違い」に関しては、for 文に改善箇所があることを特定した後に for の初期値、条件式、変化式のどこを改善すべきかを確認するために、セミコロン毎に文字列を分割し、それらと比較することで改善箇所を特定する。それにより、for 文の構文のみならずより細部の指摘も行える。

本研究で扱うプログラムは実行例と同じ出力をするものであり、学習者解答の評価コードは模範解答のものから大きく逸れていることは考えにくいので、模範解答、学習者解答のみの走査を行う最大の行数は 2 行とし、それ以上走査しても行が一致しない場合はプログラムを見直す旨のメッセージを表示させる。

また、関数やマクロの判定方法は、教育者と学習者の評価コードから関数名、マクロ名と使用回数を読み込む。そして対応する関数名、マクロ名に対して使用回数を比較することで判定ができる。メッセージ内容は関数名、マクロ名の使用回数の違いを指摘するものとする。これにより学習者は関数を適切に呼び出しているか、マジックナンバーを使用せずマクロを使用しているかを確認できる。

4.4 評価

実際にフィードバックを行った結果について述べる。図 4, 5 は、3 章で説明したプログラムに対しての判定結果である。メッセージ表示によるフィードバックが行えているが、「for 文内で冗長な計算をしない」に関しては指摘が行えていない。これに関しては 5.1 節で説明する。

```
--- 判定結果 ---
関数readDoubleArrayの使用回数は適切です
関数avgDoubleArrayの使用回数が多いです
関数minDoubleArrayの使用回数は適切です
マクロDATASIZEの使用回数が少ないです
```

図 4 判定結果 (関数・マクロ)

```
教育者の指摘:
2つ目の繰り返しで使用する変数の初期値には, より適切なものがあります
2つ目のfor文の初期値には, より適切なものがあります
3つ目の繰り返しで使用する変数の初期値には, より適切なものがあります
3つ目のfor文の初期値には, より適切なものがあります
判定結果: プログラムを修正しましょう
```

図 5 判定結果 (繰り返し)

5 考察

5.1 カスタマイズ機能を用いた際の判定

本研究で提案したフィードバック機能は, カスタマイズ機能を利用した際の判定には対応していない. カスタマイズ機能を用いた際に正しく判定を行うためには, 評価コード生成ツールから得られる評価コードを基にして, 教育者が改善箇所を特定させるプログラムを評価コード判定ツールに追加する必要がある.

5.2 メッセージ内容

現在は, 予め設定したメッセージの表示でフィードバックを行っている. しかし, 受け取ったメッセージの内容を学習者が理解できないことも考えられる. これを解決するには, メッセージを容易に変更できるように, メッセージを別のファイルから読み込む形にすれば良いと考える. そうすることで学習者の理解度やファイルの提出回数など, 様々な場面に応じたメッセージ表示が可能になる.

5.3 模範解答が複数存在する場合

同じ教育意図の中でも模範解答が複数存在する課題がある. そういった課題の時は, 教育者が模範解答を複数用意し, その中で最も合致するものを判定対象とすれば学習者に向けたフィードバックを行える. しかし, 教育者は模範解答を複数用意しなければならないと, 負担が大きくなる. これを解決する方法として, 文献 [3] で提案された模範解答派生機能を利用することが考えられる. 派生機能を用いることで, 教育者の負担をかけることなく複数の模範解答を用意することができる.

5.4 ブルーフリーダの使用性

現在のブルーフリーダでフィードバックを行うためには, 図 2 の通り 2 つのツールを端末上で実行する必要がある. また, 実際の講義での使用を考慮すると, 実行環境は誰がどのように整えるのか, 模範解答や学習者解答はどのように入手してどこに保存するのかといった問題が挙げられる.

これらの問題は, Web 上でツールを動かせるようになれば解決できる. 学習者が Web 上でファイル提出すると結果が返されるようになればその場でフィードバックを受けることができ, 判定結果を得るまでにかかる時間や手間が省ける.

6 おわりに

本研究では, 学習者のプログラムに対して改善箇所に応じたフィードバックを行える機能を提案した. 手法としては, 模範解答と学習者解答の評価コードを一行ずつ比較していき, 改善箇所を特定してそれに応じたメッセージを表示させることで学習者にフィードバックする. これにより, for 文と関数, マクロに関しては学習者に対してフィードバックをすることができた. 今後の課題としては, 5 章で述べた 4 つのことが挙げられる.

参考文献

- [1] 清水祐輔, 加賀弘晃, 松井敦紀: “カスタマイズ可能なプログラミング学習用ブルーフリーダの提案”, 南山大学情報理工学部 2016 年度卒業論文, 2017.
- [2] 後藤悠太, 長谷優磨, 田原寛隆: “教育意図を利用したプログラムのブルーフリーダの提案”, 南山大学情報理工学部 2015 年度卒業論文, 2016.
- [3] 堀尾美貴, 金崎真奈美, 佐藤成: “プログラミング演習における模範解答派生機能を備えた学習用校正ツールの提案”, 南山大学情報理工学部 2014 年度卒業論文, 2015.
- [4] 蜂巢吉成, 吉田敦, 阿草清滋: “学習項目を利用したプログラミング学習用ブルーフリーダの試作”, 日本ソフトウェア科学会第 33 回大会, pp.1-6, 2016.
- [5] 長慎也, 寛捷彦: “proGrep-プログラミング学習履歴検索システム”, 情報処理学会研究報告, コンピュータと教育研究会報告, Vol.2005, No.15, pp.29-36, 2005.
- [6] 内田公太, 権藤克彦: “C-Helper: C 言語初学習者向けツール C-Helper の予備評価”, ソフトウェア工学の基礎 XIX 日本ソフトウェア科学会 FOSE2012, 近代科学社, pp.231-232, 2012.
- [7] N.Truong, P.Roe, P.Bancroft: “Static Analysis of Student’s Java Programs”, Proceedings of the Sixth Australasian Conference on Computing Education - Vol.30, ACE’04, pp.317-325, 2004.
- [8] 大須賀俊憲, 小林隆志, ほか: “CX-Checker: 柔軟なカスタマイズが可能な C 言語コーディングルールチェッカー”, 情報処理学会論文誌, Vol.53, No.2, pp.590-600, 2012.