

プログラミング学習におけるソースコードの文に対する フィードバックメッセージ生成方法の提案

2017SE031 河本菜々

指導教員：蜂巢吉成

1 はじめに

大学などのプログラミング教育で、学習者はプログラムの作成に行き詰った時、教員にフィードバックを求める。教員は学習者の理解状況に応じてフィードバックを行い、学習者はフィードバックを受けてプログラム作成を進める。しかし、多数の学習者に教員が対応するので学習者は十分なフィードバックを受けられず、プログラム作成が進まなくなり学習意欲が削がれる原因ともなりうる。

自動フィードバックを行うシステムはいくつか提案されているが次の2つの課題がある。

課題1 学習者に直接解答となるソースコードを提示せずに、学習者の理解状況に応じて考えさせるようなフィードバックメッセージを明らかにする必要がある。

課題2 教員の負担が大きくなるような、学習者の理解状況に応じたフィードバックメッセージの生成方法が考えられていない。

本研究では、学習者のソースコードと模範解答を文単位で比較し、学習者に適したフィードバックメッセージの生成方法を提案する。学習者に適したフィードバックメッセージとは、学習者に直接解答となるソースコードを提示するのではなく、学習者の理解状況に応じて考えさせるようなメッセージである。

課題1の解決策として、学習者に適したフィードバックメッセージを、模範解答と学習者のソースコードを文単位で比較して、各編集操作やC言語のプログラムの特徴を分析しその情報の組み合わせを用いて生成する。

課題2の解決策として、フィードバックメッセージを編集スクリプトから自動で判断できる文に関しては自動で生成し、制御式や式文の説明など自動で判断できない部分のみを教員がコメントで記述することで負担を減らす。

これらから、教員の負担を大幅に減少し、学習者は自動フィードバックを受けて、考えながら取り組むことができるのでプログラム作成の行き詰りを改善できる。

2 関連研究

Rishabh[1]らは、オープンオンラインコースを想定した自動フィードバックツールで、プログラム合成を利用して、教員側が作成した模範解答、テストケース、ピアフィードバックを用いて最小限の修正が必要な修正を決定する。

Fangohr[2]らは、テストセットを用意し、学習者が電子メールで作成したソースコードを提出し、教員側が設定したフィードバックを付与し電子メールで返すことで自動フィードバックを行う。

既存のフィードバックは、コンパイル可能なほぼ完成しているソースコードが対象であり、フィードバック内容がよくある間違いから作成された一般化の修正方法である。

石元[3]の研究では、補完、正規化をし、プログラム作成中のソースコードに対応するよう、学習者のソースコードと模範解答の変数名を対応付けて模範解答の変数名を書き換える。その後、差分理解支援ツールで学習者のソースコードと模範解答の差分を検出し、得られたASTと差分の情報から学習用に改良した編集スクリプトを生成する。

3 学習者に適した文に対するフィードバック メッセージ生成方法の提案

3.1 自動フィードバックシステムの概要

フィードバックの対象は、演習で問題の解き方や誤り箇所がわからなくなりプログラム作成に行き詰る学習者である。自動フィードバックを基に理解していない文法を把握し、学習者が調べたり教員に質問しながら正解に辿り着くことを想定する。

提案するシステムは、石元の研究[3]を利用して学習用に改良した編集スクリプトを生成し、これに基づいてフィードバックを行う。まず木鎌の研究[4]を利用してフィードバックブロックでどのような処理に対するフィードバックかを提示し、その後、文単位で1つの文に対して複数のフィードバックで学習者に適したフィードバックを提示する。本研究ではメッセージの生成を考えるが、学習者の理解状況の把握に関しては対象外とする。

3.2 フィードバックメッセージの分析

学習者の理解状況に応じて解法のヒントとなるフィードバックメッセージを生成するには、「必要な操作」「支援対象となる文の行番号」「文の種類」「文の誤り」「文の説明」の5つの情報が必要となると考えた。

操作は「挿入」「更新」「移動」「削除」に分類し、模範解答に近づけさせるのにどうすべきかを明確にするのに必要となる。行番号は、文単位でフィードバックを行うので学習者に混乱させないように、どの文に対してかのフィードバックかを明確にするのに必要となる。種類は文の意味を大まかに説明する粗い文と具体的に予約語などを示す細かい文に分類する。条件分岐の「if」「else」繰り返しの「for」「while」など、どちらを用いるのかを明確にするのに必要な情報である。誤りは、文の種類は書いているが制御式や返り値、式文に誤りがある場合に誤り箇所を指摘するのに必要な情報となる。解法のヒントとなるメッセージを生成するには、何を行う文かの説明が必要である。文の説明

は具体的に提示できるよう、複数必要となる。制御文は制御式、return 文は返り値、式文は式文の説明を示す。

3.3 フィードバックメッセージの生成方法

分析を踏まえ、学習者に適したフィードバックメッセージを生成する。プログラムの行き詰りを解消するのに必要な操作の「挿入」と「更新」についてフィードバックを行う。「移動」と「削除」は学習者が誤りに気づく可能性が高いので本研究では対象としない。

「挿入」については、支援対象となる文の行番号と文の種類によりフィードバックメッセージが3通りある。文の行番号なし、細かい文の種類は、支援対象の文がどこに挿入すべきか明確にしないまま細かい文の種類をフィードバックしても学習者に混乱を招くだけで有効なフィードバックとならないので組み合わせから外した。

「更新」については文の種類は書けているので、文の誤りと文の説明に分類し、支援対象となる文の行番号の有無で分類する。

これらを基にソースコード1の6行目の制御文がない場合は表1、制御式に誤りがある場合は表2のメッセージ候補となる。メッセージは自動で生成するが、自動で判断できない「文の説明」は教員が模範解答に記述したコメントを用いて生成する。表では「」内が対象部分である。これは、学習者に適したフィードバックとなるよう文の内容によって複数生成する必要がある。よって、最低限1つは教員に記述してもらおうものとし、記述数は教員の任意とする。

表1 for 文：挿入

行番号なし 粗い文の種類	行番号あり 粗い文の種類	行番号あり 細かい文の種類
繰り返しを用いましょう。	6行目に繰り返しを用いましょう。	6行目にfor文を用いましょう。

表2 for 文：更新

フィードバック内容	行番号なし	行番号あり
誤り指摘	制御式に誤りがないか確認しましょう。	6行目の制御式に誤りがないか確認しましょう。
制御式の説明1	「値を1ずつ増やしてデータ数の分」繰り返しをしましょう。	6行目に「値を1ずつ増やしてデータ数の分」繰り返しをしましょう。

3.4 フィードバックメッセージの記述方法

フィードバックブロックはXMLを利用するが[4]、文単位はすべての文にタグ付けを行うとソースコードが見づらく教員の負担が大きいと考える。文単位ではソースコード1のようにコメントで教員が記述し、そのコメントを用いてメッセージを生成する。教員は文末に「//」を用いてコメント付与し、文の説明とそれ以外のコメントと区別できるように「@」を用いて記述する。文の説明を複数生成する場合は1つの説明ごとに「@」で区切り記述する。

ソースコード1 平均の計算 模範解答 コメントの場合

```
1 double average(int arr[],int size){
2     int i;
3     double sum;
```

```
4     double avg;
5     sum=0; // @合計を求める変数
6     for(i=0;i<size;i++){ //
7         @値を1ずつ増やしてデータの数の分
8         sum=sum+arr[i]; //
9         @合計の計算 @配列を累積加算
10    }
11    avg=sum/size; //
12    @平均の計算 @合計をデータの数で割算
13    return avg; // @平均の値
14 }
```

4 考察

学習者に適したフィードバックを出力するには、理解状況を把握する必要があり、フィードバックを求めた回数、編集時間、フィードバック後の修正の有無などの情報から判断できる。情報の取得には、Web上にエディタ、コンパイラ、実行結果とメッセージの出力欄を備えた学習環境を用意する。メッセージの出力は学習者が能動的にプログラムを作成するように、以前のフィードバックから一定時間が経ち、プログラム修正されなければ新たなフィードバックをしないなどの制限も検討する。学習者により適したフィードバックを行うために、模範解答を複数用意し、その中から学習者のソースコードに近い模範解答を選択する方法の検討も必要である。

5 おわりに

本研究では、学習者に考えさせるような解法のヒントをフィードバックすることを目的に、ソースコードを意味のまとまりで分けてフィードバック箇所を特定し、文単位でより詳細なメッセージを返すことで学習者の理解状況に応じた自動フィードバック方法を提案した。今後の課題として、学習者の理解状況の把握、学習者に適したメッセージの出力方法、学習者のソースコードに合わせた模範解答の選択、提案手法全体の自動化が挙げられる。

参考文献

- [1] Singh, R., Gulwani, S. and Solar-Lezama, A.: “Automated Feedback Generation for Introductory Programming Assignments”. ACM SIGPLAN Conference on Programming Language Design and Implementation, pp15-26(2013).
- [2] Fangohr, H., O’Brien, N., Prabhakar, A. and Kashyap, A.: “Teaching Python programming with automatic assessment and feedback”, arxiv:1509.03556[cs.CY](2015).
- [3] 石元慎太郎:”プログラミング学習者の編集途中のソースコードに対するフィードバック方法の提案”, 南山大学理工学部 2019 年度修士論文 (2019).
- [4] 木鎌汐里:”学習者の編集途中のソースコードに対する自動フィードバック方法の提案—処理単位に基づいたフィードバックブロックの定義—”, 南山大学理工学部 2020 年度卒業論文 (2021).