

C 言語関数呼び出しの間違い分析による行き詰まり原因の特定

Analyzing Mistakes of Procedure Calls to Identify Dead End in Programming Learning

谷川 紘平[†]
Kohei Tanigawa原田 史子[‡]
Fumiko Harada島川 博光[‡]
Hiromitsu Shimakawa

1. はじめに

プログラミングを初めて学習する学生にプログラミング技術を習得させるためには、実際にソースコードを記述させる実践的な演習が必要である。そのため多くの情報教育機関では、プログラミングの演習授業を設けている。一般的なプログラミング演習では、少数の教員で多数の学生を指導することが多い。そのため、演習時間内に学生全員に均質な指導をするためには、効率のよい指導が求められる。しかし現状では、演習中に理解に行き詰まった学生が教員に質問をしたさいに、教員は一人の学生への指導に多くの時間を費やしてしまい、効率よく対応できていない。この原因は、教員がソースコードや実行結果から、学生の演習過程に現れるであろう理解に行き詰まる原因を推測する必要があるためである。

本論文では、学生の演習過程に現れる関数呼び出しの履歴に着目し、多くの学生が演習過程に陥りやすい間違いをパターン化する。そして学生の演習を通した間違いの遷移状況をパターン化することで、学生が理解に行き詰まる原因を特定する手法を提案する。本手法により、教員は演習過程に含まれる学生の行き詰まりを推測する手間を削減でき効率よく指導できる。

2. 試行錯誤の過程の見逃し

2.1 理解に行き詰まる原因の把握

プログラミング演習では、学生は教員から与えられた課題に対して、ソースコードの作成とプログラムの実行を繰り返しながら試行錯誤し、課題を達成していく。この試行錯誤の後、課題に行き詰まり自己解決できなくなった学生は教員に質問をする。このような行き詰まりを起こした学生に対して、教員は間違いの内容を指摘し課題の正しい解答となるソースコードを書かせるだけでは不十分である。この場合、教員は学生の理解の妨げとなる原因を把握し解決の指針を示すことが望ましい。そのため質問を受けた教員は、行き詰まりを起こした学生が示すソースコードと実行結果をもとに、その学生が理解に行き詰まる原因を特定する必要がある。

学生の理解に行き詰まる内容は試行錯誤の過程に現れやすい。そのため教員が学生の試行錯誤の過程を把握できれば、効率よく理解に行き詰まる原因を特定できると考えられる。しかしソースコードや実行結果は試行錯誤の結果として得られるものであるため、これらから学生が理解に行き詰まる原因を推測することは困難であり、教員は多くの時間を費やしてしまう。

2.2 試行錯誤の過程の分析

演習中の学生の学習状況を分析できる手法として、文献 [1], [2] が挙げられる。文献 [1] で用いられる手法では、

学生がプログラムを完成させるまでの途中段階におけるプログラムの行数、実行回数、変数や構文の数の遷移を記録することで演習中の学生の進捗を管理できる。しかし文献 [1] の手法では、学生の進捗がなくなったことから、行き詰まりを起こしていることは検知できるが、行き詰まった内容までは判断できない。文献 [2] で用いられる手法は、学生の試行錯誤の過程から得られるコンパイルエラー時のエラーメッセージを解析し、学生の間違いの原因を特定できる。しかし文献 [2] の手法では、試行錯誤の過程におけるアルゴリズム上のエラーを検知できない。また課題に行き詰まった時点での学生の間違いを指摘できるが、試行錯誤の流れを踏まえて学生が理解に行き詰まる原因を把握できない。

試行錯誤の過程を詳細に把握できる手法として文献 [3] が挙げられる。文献 [3] はソフトウェア開発分野におけるプログラム理解のための手法である。この手法を用いると、ソースコードに影響を与えるすべての操作履歴を記録し、プログラム開発過程を詳細に再現できる。文献 [3] の手法をプログラミング演習に適応した場合、ある一人の学生の試行錯誤の過程を追跡して理解に行き詰まる原因を特定できる可能性がある。しかしこの手法では、多くの学生に共通する理解に行き詰まる原因をパターン化し、効率のよい指導に活用することは困難である。

効率のよい指導のためには、教員がすべての学生の試行錯誤の過程を踏まえたうえで、多くの学生が理解に行き詰まる原因をパターン化して把握できる必要がある。

3. 間違い遷移に着目した行き詰まりの特定

3.1 試行錯誤の過程の取得と分析

本論文では、プログラミング演習中の学生の操作履歴を分析し、試行錯誤の過程において学生が理解に行き詰まる原因を特定する手法を提案する。本手法の全体像を図 1 に示す。本手法は以下に示す 4 つの手順で構成される。

1. 試行錯誤の過程における学生の操作履歴として、関数呼び出しの履歴を記録する
2. 演習後、学生全員から収集された関数呼び出しの履歴を分析し、多くの学生が陥りやすい間違いを示した関数呼び出しの並びを間違いパターンとしてパターン化する
3. 一人の学生の演習を通した間違いパターンの遷移状況を学習パターンとして抽出する
4. 学習パターンに含まれる複数の間違いパターンに共通する間違いの要因をまとめ、学生が理解に行き詰まる原因を特定する

本手法により特定された行き詰まりの原因は、教員が把握できていない試行錯誤の過程を反映している。したがってこれを教員へ通知することで、教員は学生の試行錯誤の過程を推測する手間が省け、効率よく指導できる。

[†]立命館大学大学院 理工学研究科[‡]立命館大学 情報理工学部

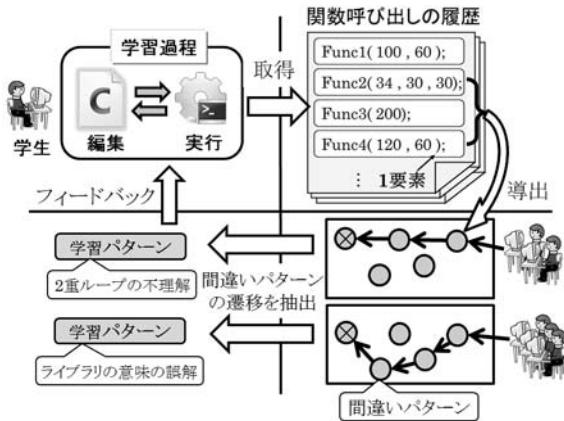


図 1: 手法の全体像

3.2 関数呼び出し履歴の取得

学生には、教員が用意する履歴記録関数を用いた演習をさせる。学生は手続き型言語における順次・反復・選択の処理を組み合わせ、履歴記録関数を呼び出すことで課題を達成していく。履歴記録関数が呼び出されるたびに、関数の種類とその時与えた引数の組が1要素として記録される。学生の試行錯誤の過程から得られる関数呼び出しの要素を関数が呼び出された順に記録したものを関数呼び出し履歴として取得する。これにより、学生の試行錯誤の過程の様子を関数呼び出し履歴として表現する。

3.3 頻出する間違いのパターン化

履歴記録関数を用いた演習の後、学生全員から取得した関数呼び出し履歴を分析し、多くの学生に共通する要素の並びを間違いパターンとして導出する。間違いパターンは、以下の手順で導出される。

1. 多くの学生の関数呼び出し履歴に共通する要素を1つ抽出する
2. 現在までに発見された共通する n 個の要素の並びから始まる $(n+1)$ 個の要素の並びを見つける
3. 手順2を再帰的に繰り返し、学生の関数呼び出しの分岐の様子を木構造を用いて表現する
4. この木構造において、子ノードの分散がもっとも大きいノードに着目し、そこから派生する要素の並びを間違いパターンとして導出する

多くの学生の行動が大きく異なった箇所には、学習障壁が存在する可能性が高いと考えられるため、本手法ではこれをもとに間違いをパターン化する。

3.4 間違い遷移状況の導出

学生が理解に行き詰まる原因を効率よく把握するには、学生の試行錯誤の過程に着目する必要がある。試行錯誤の過程において、学生はいくつかの間違いパターンに陥りながら演習を進めていく。プログラミングに必要な知識への理解度が似た学生同士は、試行錯誤の過程で同じような間違い方をする可能性が高い。また演習中の学生の理解度は、試行錯誤の過程で変化していくと考えられる。そこで、本手法では試行錯誤の過程における学生の間違いパターンの遷移状況を学習パターンとして導出する。導出された学習パターンは、演習中の学生の理解度や思考の変化の様子を表しており、理解に行き詰まる原因の特定のために有益だと考えられる。

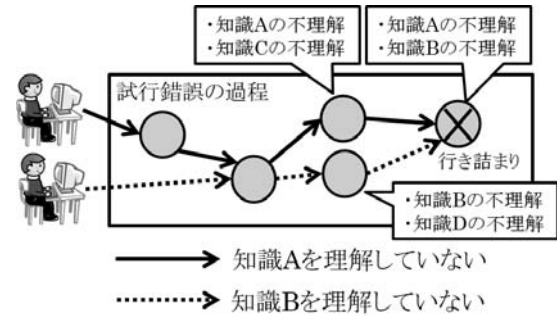


図 2: 行き詰まりの原因特定

3.5 理解に行き詰る原因の特定

導出された学習パターンごとにも共通する間違いの要因をまとめ、学生が理解に行き詰まる原因を特定する。理解に行き詰まる原因の特定方法を図2に示す。各学習パターンには、複数の間違いパターンが含まれている。また各間違いパターンには、いくつかの間違いの要因が存在する。学生が理解に行き詰まる原因を特定するために、まず導出されたすべての間違いパターンについて間違いの要因を調べる。そして課題に行き詰まった学生の学習パターンに着目し、そこに含まれる複数の間違いパターンの間違いの要因として共通するものを理解に行き詰まる原因と特定する。

こうして特定された理解に行き詰まる原因をもとに指導することで、教員は課題に行き詰まった学生から質問を受けたさいに、試行錯誤の過程を推測し理解に行き詰まる原因を考える手間を削減でき、効率よく指導できる。

4. おわりに

本論文では、関数呼び出しの履歴を分析することで、学生が陥る間違いの遷移状況をパターン化し、学生が理解に行き詰まる原因を特定する手法を提案した。

過去に我々は、演習中の学生から関数呼び出し履歴を収集し、多くの学生に共通する間違いパターンを導出した[4]。手法[4]によって導出された間違いパターンは、試行錯誤の過程において学生が陥る間違いを詳細にパターン化できていることが判明した。

今後は、手法[4]で導出された間違いパターンの遷移状況から理解に行き詰まる原因を導出し、その教育的効果を検証する予定である。

参考文献

- [1] Juehua Zhang, 西田 知博, 安倍 広多, 石橋 勇人, 松浦 敏雄, プログラミング授業を支援する学習環境 PEN + , Journal of Informatics, Vol.5 No.1, 2008
- [2] 西 輝之, 劉 渤江, 横田 一正, デバッガとの連携による C 言語学習支援システムの提案, 電子情報通信学会技術研究報告, Vol.106, No.583, pp.173-178, 2007
- [3] 大森 隆行, 丸山 勝久, 開発者による編集操作に基づくソースコード変更抽出, 情報処理学会論文誌, Vol.49, No.7, pp.1234-1244, 2008
- [4] 谷川 紘平, 原田 史子, 島川 博光, 関数呼び出しの分析による頻出学習パターンの導出, 情報科学技術フォーラム講演文集, Vol.9, No.3, pp.111-114, 2010