

コンパイルエラーの分類に基づき プログラミング実習の指導者を支援する手法

2019SE001 天野妃鞠

指導教員：名倉正剛

1 はじめに

大学での初年度のプログラミング授業では、初学者に対して構文を理解させるため、学習者によるプログラミングを伴うような実習が行われることが多い。この授業では、1回の実習講義で理解させたい内容に対して、構文を理解させることができるよう、様々なバリエーションで構文を記述させるような課題を用意し、難易度のやさしい課題から徐々に難しい課題を順に事前に出題し、受講生はそれぞれの頻度で課題を進めていく。このような授業を行う際、本来は受講者を個別にフォローすべきだが、大学での授業は指導者の不足によりそのような対応ができず、受講者側から自発的に質問する形式がとられることが多い。

2 本研究で対象とする課題

受講者が行き詰ってしまう場合の一つとして、コンパイルエラーの発生がある。コンパイルエラーメッセージを理解することも、初学者が構文を理解するための一助になるが、初学者がエラーメッセージを理解してエラーを解決することは容易でない。受講者が自分の行き詰った状況を適切に認識できれば自発的に質問することもできるが、初学者にとって、自分の状況を適切に認識することは困難である。また、受講者自身では解決できないような問題が複数の受講者に同時に起こった場合、指導者が不足し受講者の状況に対応しきれないことがある。一般的なプログラミング実習では、特定の構文に対して基本的なプログラムを実習させ、段階的に応用的なプログラムを実習させる。その際に受講者が取り組む課題のセットは、全ての受講者で同一であることが多い。したがって、同時に複数の受講者に問題が発生した場合、その問題の原因も類似することが多い。原因が類似するのであればまとめて指導することで指導者の負担を減らすことができる可能性がある。しかし、複数の受講者の問題が類似する原因であることは、受講者に発生したエラーメッセージやソースコードを確認する必要がある。

3 分類に基づく指導者支援手法の提案

3.1 提案手法の概要

本研究では C 言語を用いて行われるプログラミング実習において、受講者のコンパイル実施に伴って発生したエラーメッセージを自動で収集し、エラーの原因によって分類し、原因ごとにまとめて指導者に提示する手法を提案する。先行研究 [1] では、コンパイルに伴って発生する複数のエラーメッセージを、原因ごとに分類し、発生する都度

受講者へ分類された原因を提示する手法を提案している。本研究の提案手法ではこの手法を利用し、演習の開始時点から蓄積した各受講者に発生したエラーメッセージからエラーの原因を特定し、原因ごとにエラーメッセージを分類し、発生時間の比較により解消していそうなエラーを除外し、提示する。指導者はその提示された情報を元に、多くの受講者が同じ原因でエラーを生じている場合に優先的に一斉指導したり、あるいは原因ごとに担当する指導者を割当てて手分けして指導することが可能となる。

3.2 先行研究 [1] でのエラーメッセージの分類

初学者がプログラミング学習を行う際に、ひとつの構文ミスにより複数のコンパイルエラーメッセージが生じると、エラーを容易に除去できず学習を妨げる。そこで先行研究 [1] ではコンパイルエラー発生時にコンパイルエラーメッセージをその原因により分類して提示することにより、初学者によるコンパイルエラーの除去作業を支援する手法が提案されている。この研究では、初学者のプログラミング学習時によく発生する構文ミスに基づき、エラーメッセージの解析や構文ミスの修正に利用できるルール群を定義している。そしてコンパイルエラーの発生時に、それらのルールにしたがって発生したエラーメッセージがどの構文ミスに起因しているのかを特定し、その原因ごとにエラーメッセージを分類して提示する。

3.3 提案手法の流れ

提案手法の処理の流れを図 1 に示す。各受講者に発生したエラーメッセージの収集には、サーバ上にエラーメッセージを蓄積する既存ツールを使用する。次に、指導者が指導する際、それ以前に一斉指導している場合、直近の一斉指導より前に発生したコンパイルエラーのデータは、既に解決している場合が多いため、直近の一斉指導以降のデータに絞らざる。その後、原因特定処理部でエラーの原因を特定し、原因分類処理部で原因ごとにエラーメッセージを分類する。そして、分類結果表示部で分類結果を提示する。処理の手順は、次のとおりである。

- 1) ツールにより各受講者のユーザ ID、ファイル名、ソースコード、エラーメッセージ、記録日時を収集する。
- 2) 収集したエラーメッセージと記録日時から、発生時間の比較により解消していそうなエラーを除外する。
- 3) 原因特定処理部でコンパイルエラーに対して特定したエラー原因が、対象とする構文ミスのどれに該当するかを特定する。特定には、既存研究 [1] でエラーメッセージの解析に利用するルールを拡張し使用する。

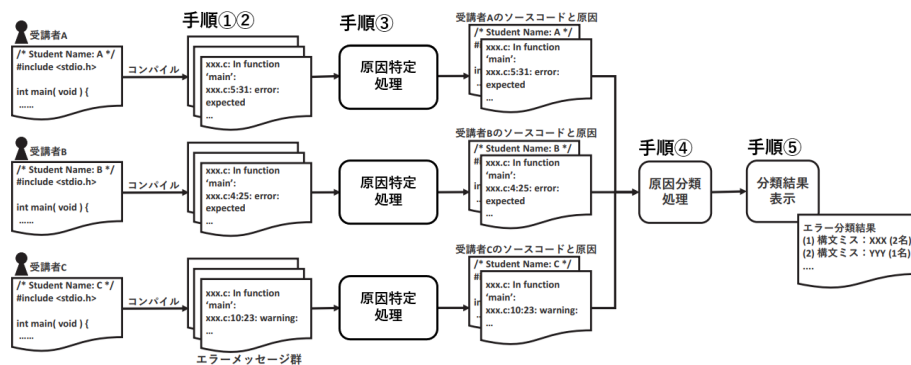


図1 提案手法の処理の流れ

- 4) 原因分類処理部で特定した構文ミスごとにエラーメッセージを分類する。手順3の結果として、ある受講者のコンパイルエラーに複数の原因が含まれ、複数の構文ミスが該当する場合は、その受講者のエラーメッセージを複数の分類に含める。
- 5) 分類結果表示部で分類結果を提示する。

3.4 対象とする構文ミス

提案手法では初学者のプログラミング学習時に発生する典型的な構文ミスを対象としており、先行研究 [1] でのルール定義の際に対象としている、11種類の構文ミスを対象とする。また、変数宣言時の型キーワードの間違え、組み込み関数に対する #include 文の欠落の2種類の構文ミスを、ルール追加による対応可能性を示す目的として、追加で定義している。

3.5 分類結果の提示方法

まず、3.4に示す対象とする構文ミスごとに発生しているエラー件数を図2に示すよう、各行に構文ミスの略称(Category)、該当する件数(Count)、構文ミスの説明を提示する。これにより、一斉指導すべき構文ミスの発見につながると考える。さらにそれらの分類に対するリンクをたどることで、分類ごとの詳細な情報を指導者が閲覧できるようにする。提示する情報は、その構文ミスが生じている受講者のユーザ名、ソースコード、エラーメッセージ、記録日時である。これにより、ソースコードやエラーメッセージから、原因発生箇所が同一箇所か把握したり、少ない件数の場合、ユーザ名から個人指導もできると考える。

4 評価実験

大学初年度のプログラミング演習で発生した構文ミスとその際のコンパイルエラーメッセージを利用し、提案手法で分類、提示した。対象とした演習は南山大学工学部1年生のC言語の実習形式の授業である。Cプログラミング演習の最初期の1週分の演習での課題に対するものである。分類した結果、図2の提示結果となった。また、今回対象としたデータでは近似曲線による極小値が取れず、データの絞り込みを行うことができなかった。

Category Count		
brktSyn	90	括弧 ('()', '{}', '[]') や引用符 ("", '') の欠落や混同
calTyp	73	メソッド呼び出し時の引数間違え (例えば型の間違え)
noRtnSem	37	非 void 型の返戻値のメソッドにおいて return 文のない制御フローが存在
forSepSyn	26	for のセパレータを ';' ではなく、',' に間違えて記述
curIfSyn	22	if 文の条件文の括弧が '(' ではなく '{' に間違えて記述
other	15	
smiHdSyn	5	メソッド定義 (プロトタイプ宣言ではない) の "(...)" の直後に ';' を記述
parCalSyn	5	メソッド呼び出し時の '(' 部分の記述の欠落
typCalSyn	4	メソッド呼び出し時に '(' 部分に引数の型名を含めて記述
keySyn	3	メソッド名や変数名にキーワードを記述

図2 提示画面

5 考察

評価実験では、C(brktSyn) と I(calTyp) の構文ミスにより多数のコンパイルエラーが発生しており、それぞれ C(brktSyn) が 90 件、I(calTyp) が 73 件発生している。この件数の中には、エラーメッセージを確認し、受講者が自力で構文ミスを修正する場合も含まれる。しかし、初学者にとって、エラーメッセージを理解することは容易ではないため、指導者はエラーメッセージの理解方法 (読み方) を指導することも重要である。また、C(brktSyn) ではコードの整形、I(calTyp) ではデータ型の基礎を指導することにより、修正に効果のある指導を行うことができると考える。このように、多数のコンパイルエラーの原因となっている構文ミスに対し、一斉指導を行うことで、1つの構文ミスにより長時間課題を進めることができない受講者の支援になると考える。

対象とするデータの絞り込みについて、評価実験では近似曲線による極小値が取れず、絞り込むことができなかった。評価実験で対象とした実際の演習のコンパイルエラーの発生が想定と違い、一斉指導によりコンパイルエラーが減ることがなかったことが原因であると考えられる。

参考文献

[1] 近藤亮太, 名倉正剛: コンパイルエラーメッセージの分類に基づく初学者へのプログラミング学習支援手法, 第9回 実践的IT教育シンポジウム rePiT2023 in 函館, 2023, pp.63-74.