

プログラミング初学者のためのエラー読解支援ツールを用いた学習支援

Learning Support by Using Error Reading Assistance Tool for Programming Learners in Elementary Grades

瀧本 恭之, 戸川 聡

Yasuyuki TAKIMOTO and Satoshi TOGAWA

四国大学

Faculty of Management and Information Science,
Shikoku University

Email: doors@shikoku-u.ac.jp

あらまし: プログラミング初学者はエラーメッセージの読解に苦戦するため、デバッグ作業遂行が困難である。言語処理系から出力されるエラーメッセージは初学者にとって理解することが難しい。そこで初学者を対象とした読解支援の枠組みを提案する。初学者の学習レベルに応じて、適応的なエラーメッセージを段階的に提示する。エラー原因を示唆するメッセージを提示することで、自身で考えさせ、気づきを与える。本研究では読解支援ツールのプロトタイプツールを実装し、評価実験を行った。

キーワード: プログラミング学習、初学者支援、適応的デバッグ環境

1. はじめに

近年、日本では義務教育課程においてプログラミング教育を必修化する取り組みが行われている⁽¹⁾。プログラミング教育の必修化に伴い、プログラミング学習者が増加傾向にある。プログラミング教育を行うねらいには論理的な思考能力と問題解決能力の習得がある。それらの能力習得が期待されるプログラミング教育だが、プログラミング初学者にとってプログラミングスキルを習得することは困難である。

初学者がプログラミングを学習する際、陥る問題がいくつかある。その問題のひとつに、デバッグ作業遂行の困難さが挙げられる。初学者においてデバッグ作業が苦手な理由として、言語処理系から出力されるエラーメッセージを読解できないことが挙げられる。結果、デバッグ作業が停滞する。エラーメッセージはデバッグ実行時における情報を示唆する。しかし、標準的に出力されるエラーメッセージはデバッグに関する事前知識が不足していると読解が難しい。そのため、初学者はエラーメッセージの読解に苦戦する。エラーメッセージの読解を支援できれば、デバッグ作業に掛かる時間を短縮できる。これによりプログラミング初学者の学習が促進される。

本研究では、初学者を対象としたエラーメッセージ読解支援の枠組みを提案する。エラーメッセージ読解支援ツールのプロトタイプを開発し、評価実験を行う。

2. プログラミングにおけるエラー読解支援

2.1 一般的なエラーメッセージにおける特徴

エラーメッセージはソースコードのコンパイル後、なんらかの問題が発生した時に表示される。エラーメッセージはエラーが発生したファイル名、発生している行番号、エラー番号、エラーに関する簡潔なメッセージなどを表示する。エラーメッセージは英

語で出力されることが多い。

2.2 エラーメッセージ読解における問題点

初学者がエラーメッセージを読解する場合において、本研究ではエラー番号とメッセージの可読性に着目した。

エラー番号は発生したエラーの種類について、数字やアルファベットで記述されている。これは、発生したエラーに対する対処法を知るために有益である。しかし、エラー番号単体ではエラーの内容を理解できない。エラーの原因を知るためには、エラー番号で示されるエラー内容を調べる必要がある。初学者はエラー番号が示すエラーの内容の意味を判読できない。そのためエラー番号で示されるデバッグ情報を得ることは難しい。

エラーの内容は簡潔なメッセージで説明される。しかし、メッセージは英語で記述されており、学習者は、まず英語を読解しなければならない。このため、初学者のプログラミングに対するモチベーションが低下する。

これらの原因により、初学者はエラーメッセージが示すエラー情報を理解することができない。仮に理解できたとしても、エラーを引き起こす原因とエラーメッセージが示す内容を結びつけることができない。

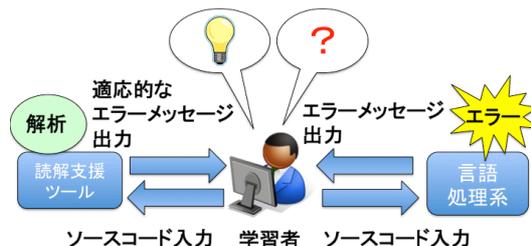


図 1 エラーメッセージ読解支援の枠組み

2.3 支援の枠組み

図1にエラーコード読解支援の枠組みを示す。初学者は言語処理系にソースコードを入力する。なんらかの誤りが検知された場合、エラーメッセージが出力される。本来、エラーメッセージを読解し、デバッグ作業を始めるが、初学者は提示されたエラーメッセージを理解できない。

そこで本研究で開発する読解支援ツールにソースコードを入力する。読解支援ツールは初学者の学習レベルに応じて、適応的にエラーメッセージの内容を変更する。学習レベルの判断は、読解支援ツールに入力されたソースコードに含まれるエラーの頻度で判断する。エラーを含むソースコードが読解支援ツールに連続的に入力されたならば、エラーの原因を解消できていないと判断する。これにより、エラーメッセージの内容を変更する。

初期段階から解答そのものを提示することは初学者にとってよい学習環境とは言えない。いきなり解答を提示するのではなく、解答につながるヒントを提示することで、初学者自身に考えさせ、気づきを得ることが重要である。学習レベルに応じ、エラーメッセージの出力を変更しながら、誤りの原因を示唆する。それにより、初学者に対しエラーの存在を気づかせる。エラーが解消するまで繰り返しデバッグを行うことで、エラーの根本的な原因に気づかせる。この枠組みによってプログラミング学習が促進される。

3. プロトタイプツールの開発と評価実験

3.1 プロトタイプツールの実装と支援対象の言語

提案する支援の枠組みの有効性を検証するためにプロトタイプツールを開発し、評価実験を行った。本ツールの開発には Ruby と Ruby Gem で提供されている Racc を使用した⁽²⁾。Racc は C 言語で言語処理系を実装する際に使用される yacc の Ruby 版パーサジェネレータである。

支援対象のプログラミング言語は Ruby とした。Ruby は初学者向け言語として注目されている。構文がシンプルで記述が容易である。また、型宣言が存在しないため記述量が少ない。その反面、Ruby は文法が柔軟であるため、初学者がプログラムを記述する際に混乱するおそれがある。そこで、エラー検知する対象の構文を制限する。

3.2 エラーメッセージの出力機能

読解支援ツールから出力される機能として、学習者の学習レベルに応じた適応的なエラーメッセージ出力機能を実装する。実装する機能は、コンパイラのフロントエンド部を参考にした。今回、字句解析と構文解析の2つのフェーズとして解析する。複数回入力し、誤りが連続して改善されなかった場合は読解支援ツールのモードが切り替わる。これ以降、出力されるエラーメッセージの内容が変更される。出力されるエラーメッセージ内容が三段階で変更される。モード1では各解析フェーズでの誤りの有無



図2 プロトタイプツールの実行イメージ

を提示する。モード2ではエラーを検出した行番号を提示する。モード3ではソースコード上の誤っている可能性がある場所を提示することとした。

3.3 プロトタイプツールの開発

プロトタイプツールの機能として、出力するメッセージは日本語を用いる。プロトタイプツールが出力するエラーメッセージにはエラーコードを含まない。プロトタイプツールは誤りのあるソースコードが4回連続で入力されるとメッセージ内容が切り替わるように実装した。文法は変数への値の代入と制御構造の if-else 文の検出を実装した。

3.4 実行結果

図2に誤りを含むソースコードをプロトタイプツールに入力した実行イメージを示す。各エラーの種類を検出し、ソースコードの誤りについて適応的なエラー出力を確認できた。また字句解析、構文解析の各フェーズにおいてもエラー検出できることが確認できた。

4. おわりに

本研究では、プログラミング初学者を対象とした学習支援の枠組みの提案した。提案した枠組みを元に読解支援のためのプロトタイプツールを開発した。入力されたソースコードから誤りを検出し、初学者向けメッセージの出力を確認した。今後は、読解支援ツールに意味解析機能を追加し、検出可能な制御構造を増やすことを検討している。

参考文献

- (1) 羽野 三千世, "小学1年生にプログラミング教育--武雄市、DeNA、東洋大", ZD Net Japan, <http://japan.zdnet.com/article/35060367/>, 2015-02-13
- (2) 青木峰郎, Rubyを256倍使うための本 無道編, 東京, 株式会社アスキー, 2001, 226p