コンパイルエラーの原因分析に基づくプログラム修正支援
Support for Correct Program Based on Cause Analysis of Compile Error

金澤　勇輝*1, 酒井　三四郎*2
Yuki KANAZAWA*1, Sanshiro SAKAI*2
*1,2 静岡大学情報学部情報情報科学科
*1,2Faculty for Informatics of Informatics, Shizuoka University
Email: kanazawa@sakailab.info

あらまし：プログラミングを行う際に, コンパイラはエラー文だけでは原因がわかりにくいエラーを出すことがある. プログラミング初学者は知識, 経験不足から解決に時間が掛かってしまう. 本研究ではプログラムのソースコードを分析し, どんなエラーが実際に起きているか診断するツールを開発した. 高い精度で診断が行えたという結果が得られたが, 対象となる範囲が狭いという問題点が見つかった.
キーワード：プログラミング支援，Java，コンパイル, エラー修正

1. はじめに

プログラミングを行う際に,コンパイラはエラー文だけでは原因がわかりにくいエラーを出すことがある.プログラミング初学者は知識,経験不足から解決に時間が掛かってしまう.原因は正確な原因がわからない場合本来エラーでない場所の修正を繰り返してしまう,どこが間違っているのかわかっていても修正の仕方がわからないからである.本研究では上記の問題を解決するために初学者に対して,コンパイルエラーが発生した際にコンパイラとは違う観点からソースコードを分析し,どんなエラーが実際に起きているか診断,修正の支援するツールの開発を行った.

2. 関連研究

Amjad Altadmri ら[1],[2]は,プログラミング教育者へのアンケートから初学者に多い 18 種類のミスを定め,BlueJ の BlackBox テストを用いて 1700 万件以上のコンパイルエラーデータからそれぞれのミスについてどのエラーの発生頻度が高いのか,また,エラー解決までの平均時間の調査を行った.大西淳ら[3]は,Fortran において学習者がプログラミングに行き詰った際に,どこが間違っているかを教えてくれる自動相談システム(CONSULT/C)を開発した.本研究では Java プログラムにおいて,18 種類のミスそれぞれについて解析器を作成し,学習者がコンパイルエラーの修正が困難である場合にエラー原因,解決案を提示しプログラムの修正支援を行うことを目的とする.CONSULT/C との違いは一つのエラーメッセージからエラーを推測,診断を行うのに対して,筆者が提案するツールは複数のメッセージから推測,診断を行う事が出来る点,また,コンパイルが成功した場合でも学習者にとって予期せぬエラーが発生する恐れがあるときに警告を知らせる点である.

3. 提案ツール

本研究ではコンパイルエラーが発生した際に[1],[2]で示された 18 種類のミスの内コンパイルエラーの原因となる 14 種類のミスが発生しているかを AST(Abstract Syntax Tree)を用いて解析を行い,エラー原因を診断するツールを作成した.本システムの特徴を以下に挙げる.

- コンパイラからのエラー文,ソースコードの両方からプログラムのエラーの原因を特定する.
- プログラムに複数のエラーがある場合,既存のコンパイラが出す最初のエラー箇所を修正支援の対象とする.
- 使用者にエラー原因,解決案を示しプログラム修正の支援を行う.
- コンパイルエラーが発生しなくても実行時に予期せぬエラーが発生する恐れのある 4 種類のミスに対して警告を行う.
- コンパイルエラーが発生した際に,対象のソースコード,エラー情報の保存を行う.

4. 評価実験

4.1 実験環境

作成したツールの有効性を示すために以下のような条件で実験を行った.

- 2016 年度静岡大学情報社会学科の 1 年次の講義「プログラミング」の「課題 10」を遂行した学生からランダムに 4 人を選び,それらの学生が課題遂行中に作成したプログラムから 1 つのソースファイルから成るプログラムを対象とする.
- 上記のプログラムのコンパイルを行いツールの診断,また筆者による確認から結果を表 1 のように分類した.

表 1 ツールの診断結果の表示

コンパイル成功		コンパイル失敗	
1	問題なく成功	1	ツールの対象外エラー
2	ツールが正しく警告を表示	2	エラーの原因と診断結果が一致
3	誤って警告を表示	3	対象のエラーを別の対象エラーと判断
4	警告を出すべきところで表示な	4	対象外エラーを対象エラーと判断
		5	対象エラーだが表示なし

4.2 実験結果

実験に使用したファイルは合計 767 件で,その内コンパイルに成功したものが 339 件,失敗したものが 428 件であった.分類の内訳を表 2 に示す.また対象外エラーに含まれる 335 件のファイルについて,さらに「a.コンパイラのエラー文から初学者が原因がわかる」,「b.コンパイラのエラー文から初学者が原因がわからない」と筆者が判断し 2 つに分類した.

表 2 実験結果内訳

コンパイル成功		コンパイル失敗			
1	327	1	335	a	199
				b	136
2	12	2	86		
3	0	3	0		
4	0	4	0		
		5	7		
計	339	計	428		

5. 考察と結論

実験から作成したツールが対象と出来るエラーの割合は全体の中だと 21.8%,コンパイラのエラー文から原因がわからないエラー内でも 40.6%と高いとは言えない結果になった.このエラーの内,AST を用いた静的解析で診断が行えるエラーについては解析器を作成し対象範囲を増やしたいと考えている.静的解析を行う事ができないエラーについては別の形で静的解析が行えるようになる仕組みを考える必要がある.また精度自体は 9 割を超えていたが本当に支援を必要としているのは複数のエラーが発生している場合であるためこの点でも構文解析を行えるようにする仕組みが必要である.

ツールが一部対象のエラーの診断を行えなかった原因はプログラム中に「不正な文字の使用」,「中括弧の過不足」等の AST による構文解析が出来なくなるエラーが同時に発生していたことが考えられる.これらのエラーによりプログラムが静的解析できない状態になっており正確なエラーの解析が行えなかった.

観察の結果,初学者がコンパイルエラーを引き起こした際に,一度でプログラムの修正が完璧に行えたというケースは少なく,何度も同じミスを発生させているケースが多いことがわかった.ツールがわかりやすい原因・解決案の提示を行うことでプログラムの修正に掛かる時間を減らす事が期待できる.

今回の実験では「エラー原因の診断が行えるのか」しか測定をしていないため初学者にとってわかりやすい原因,解決案の提示,UI の作成,また学習効果があるのかも検討する必要がある.

今後の課題は以下の通りである.

- 本研究の分析方法で可能な未対応エラーについて,解析器を作成しツールの対象範囲を大きくする.
- 静的解析が不可能なエラーが発生した際に,本研究の解析方法が行える形にするために別の方法での支援を検討する.
- 実際にツールを初学者に使用してもらい,解決案を読むことによって正しく修正を行うことが出来るか,原因,解決方法が理解することが出来るのか評価する実験を行う.

6. おわりに

本研究で作成したツールは対象となるエラーのみが発生している場合には正しい判断を行うことが出来たが,エラーが複数発生していた際に診断が正しく出来なくなってしまうことやまた対象となるエラーの範囲が狭いという問題点が明らかになった.これからはさらにツールの対象範囲を広げ,精度を上げると共に,初学者にとって理解しやすい支援を目指し続けたい.

参考文献

(1) N. C. C. Brown and A. Altadmri, “Investigating novice programming mistakes: educator beliefs vs. student data”,In Proceedings of the tenth annual conference on International computing education research, 2014,pp. 43–50.

(2) Altadmri, A, & Brown, N. C, “ 37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data”,In Proceedings of the 46th ACM Technical Symposium on Computer Science Education,2015,pp. 522–527.

(3) 大西淳,島崎真昭, "CONSULT/C: コンパイル時のエラーに関するプログラム相談手法/システム," 情報処理学会研究報告ソフトウェア工学 *(SE)* 1989.11 (1988-SE-064),1989, pp.185-192.