

中間言語を用いた自動バグ修正システム

田畑 優吾[†]阿部 清彦[†][†] 東京電機大学システムデザイン工学部情報システム工学科

1. はじめに

現在、システム開発においてデバッグ作業が占めるコストの割合は大きい。そのデバッグ作業を自動化するために、自動バグ修正技術が研究されてきた[1]。自動バグ修正のツールとして、C 言語のプログラムを修正する GenProg [2][などが挙げられる。本研究では、中間言語を用いて、多種の言語によるプログラムの修正を行うことができるシステムを開発することを目的とした。

2. 開発した自動バグ修正システムについて

本システムは、筆者らの研究室で開発している C 言語のバグ修正システム[1]を多種の言語に対応させたものである。修正には、エラーの無い外部ソースコードを利用している。修正対象プログラムに対して、外部ソースコードを利用した挿入処理・削除処理・置換処理をすることで、プログラムを修正する。

3. 中間言語の変換について

中間言語には、python の抽象構文木を用いた。変換前の Java プログラムを図1に、変換後の中間言語例を図2に示す。

```
boolean flag = false
if (year % 400 == 0){
    flag = true
}else if{
    . . .
}
```

図1. 変換前の Java のプログラム

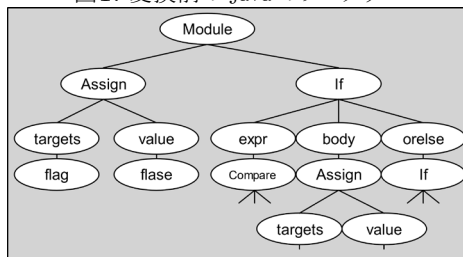


図2. 図1を変換した中間言語

4. 実験

うるう年を求めるプログラムの修正を行った。データとして、AtCoderARC002-A[3]から、以下の4種類を取得した。

- (1) エラーのある python プログラム (339 個)
- (2) エラーのある java プログラム (76 個)
- (3) エラーのない python プログラム (140 個)
- (4) エラーのない java プログラム (280 個)

(1)(2)のプログラムを修正対象のソースコードとし、(3)(4)のプログラムを外部ソースコードとして修正に利用する。ただし、外部ソースコードはランダムで選ばれた3個のコードを使用する。以下の3パターンで実験を行った。外部ソースコードを設定しない場合は、修正対象コードを外部ソースコードとして使用する。

- A) 外部ソースコードを使用した python プログラムの修正
- B) 外部ソースコードを使用しない python プログラムの修正
- C) 外部ソースコードを使用した java プログラムの修正

5. 実験結果

表1. 修正方法と修正成功数

	(A)	(B)	(C)
成功 (置換処理)	405 件	51 件	60 件
失敗 (変換失敗)	306 件	660 件	87 件
失敗 (打ち切り)	306 件	306 件	45 件
修正不可	0 件	0 件	36 件
合計	1017 件	1017 件	228 件

実験結果を表1に示す。置換処理による修正はできたが、挿入処理や削除処理による修正はできなかった。失敗(変換失敗)は中間言語に変換できなかったものの、修正不可は中間言語に変換した際に標準出力の関係でテストケースを通過したものの件数を示している。

中間言語を用いて複数言語の修正ができること、外部ソースコードを使うと修正の幅が広がることが確認できた。

6. 今後の課題

今後は、対応する言語の種類を増やし、修正精度を高めていく予定である。

参考文献

- [1] 長倉雄大, 阿部清彦 “外部ソースコードを利用した自動バグ修正プログラム”, 2022 年電気学会電子・情報・システム部門大会講演論文集, pp.1151-1152, (Sep.2022)
- [2] Claire Le Goues, ThanhVu Nguyen, Stephanie Forrest, and Westley Weimer, “GenProg: A Generic Method for Automatic Software Repair” IEEE Transactions on Software Engineering, Volume 38, pp. 54-72, 2012
- [3] 競技プログラミングコンテスト Atcoder(うるう年を求めるプログラム), (https://atcoder.jp/contests/arc002/tasks/arc002_1) [閲覧日:2022/12/30]