

実行時エラーにおける自動バグ修正について

新井 理玖[†] 阿部 清彦[†]

[†] 東京電機大学大学院システムデザイン工学研究科

1. はじめに

プログラムのバグを取り除く作業のコストを減らすため自動的にバグ修正するシステムが開発されている^[1]。自動バグ修正では修正に関して修正可能かどうか、修正にかかる時間、修正の精度などの課題があるため、本稿では自動バグ修正の改善を目的としている。

2. GenProg について

GenProg^[1]は遺伝的アルゴリズムに基づく自動バグ修正システムであり、生物の遺伝のように個体集団に対して選択、交叉、突然変異の操作を行い、世代を重ねることで全てのテストを通過する解を求める。GenProg は修正対象プログラムからプログラム文(トークン)を再利用する必要がある、再利用できるトークン有無により修正が困難な場合があり、また処理コストは高い。テストをどの程度通過したかを調べるため、毎回全てのテストを行わなければならない時間がかかる。また、修正結果が人間的な修正でないことがある。これらの点が修正の課題となる。

3. RSRepair について

RSRepair^[2]はランダム探索を使用した自動バグ修正システムであり、単一の変更を行い、世代を重ねることなく繰り返し全てのテストを通過する解を求める。テストの一つでも通過しなかった場合、即座にテストを打ち切るため、処理速度が GenProg よりも速いが、複数個所の修正ができないという弱点がある。

4. プログラムエラーの種類と調査

プログラム中に存在するバグ(エラー)はコンパイルエラー、実行時エラー、論理エラーの三種に分類される。テストができないコンパイルエラーを除き、実行時エラーと論理エラーの修正を検討する。システム改善のため、どのようなエラーが多いかをオープンソースソフトウェアで調査した。

QuixBugs^[3]は修復ツールの多言語間の能力を調べる目的のベンチマークであり、40 のバグを含むプログラムを持つ。QuixBugs 中のテストを行い最初に出てきたエラー40のうち実行時エラーの内訳は表1のようになっている。実行時エラーはエラー箇所の特が容易と考え本稿では実行時エラーに注目した。QuixBugs の実行時エラーは StackOverflowError が最も多く、実際のプログラムでも見られるエラーであり、再帰処理で多く起きる可

能性のあるエラーである。また、遺伝的アルゴリズムに基づいた自動バグ修正システムでは最大世代数 10 世代、制限時間 60 秒の条件では世代内、時間内に全て修正できなかった。構造的に修正が難しいものが多く、時間を確保してもプログラムの正常な動作を破壊してしまうものまであった。

表 1 実行時エラーの数

Runtime errors	Number
ArrayIndexOutOfBoundsException	2
IndexOutOfBoundsException	1
NoSuchElement	1
NullPointerException	2
StackOverflowError	5
Total	11

5. まとめ

今後は実行時エラー、特に StackOverflowError の調査とランダム探索や修正パターンを用意した他の修正方法による修正を試みる。

参考文献

- [1] 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.
- [2] Yuhua Qi, Xiaoguang Mao, Yan Lei, Ziyang Dai, Chengsong Wang, "The strength of random search on automated program repair" ICSE 2014: Proceedings of the 36th International Conference on Software Engineering, 2014, Pages 254-265.
- [3] jkoppel/QuixBugs: A multi-lingual program repair benchmark set based on the Quixey Challenge, <https://github.com/jkoppel/QuixBugs> (2022年1月26閲覧)