GenProg を用いた Scratch プログラムの自動修正

1255110　高橋 智哉　【 ソフトウェア検証・解析学研究室 】

Automatic Repair of Scratch Programs Using GenProg

1255110　TAKAHASHI, Tomoya　【 Software Verification and Analysis Lab. 】

1　はじめに

Scratch はブロックベースのプログラミング言語であり，コンピュータプログラミングの体験や学習する際の導入として，若い学習者を中心に利用されている．Scratch では，学習者が構文について意識することなくプログラムを作成できるため，構文上のプログラムエラーを排除することが出来る．しかし，機能的なプログラムのバグは依然として存在している．

プログラミング学習において，学習者が作成したバグを含むプログラムに対して，教育者がフィードバックを返すことは教育上良いとされている．しかし，学習者の人数が多くなると教育者の負担が大きくなる．そのため，教育者が学習者のプログラムを理解することを補助するシステムが求められている．

自動プログラム修正（APR）手法によってバグの修正方法を提示することは，バグの位置の特定やバグの原因について理解することを補助する方法として有効と考えられる．本研究では，C 言語用の APR ツールである GenProg[1] に Scratch 用のテストフレームワークである Whisker[2] を適用した Scratch 用 APR ツールを開発し，その有用性を評価する．

2　背景技術

2.1　自動プログラム修正（APR）手法

自動プログラム修正手法とは，プログラムの障害位置を推定し，推定した位置に対して変更を加え，変更によってプログラムが修正されたか検証をする，という一連のプロセスを繰り返すことでプログラムの修正を行う手法である．

2.2　遺伝的プログラミング

遺伝的プログラミングとは，生物の適応進化の仕組みを応用して解の探索を行うアルゴリズムである．以下に遺伝的プログラミングの動作の流れを示す．

1. 第一世代の個体群を生成
2. 各個体を評価し，適合度を計算（終了条件を満たせば終了）
3. 適合度の低い個体を削除
4. 交叉や変異などを行い，次世代の個体群を生成
5. 2 の処理まで戻る

2.3　GenProg

自動プログラム修正手法の 1 つに GenProg がある．GenProg は，バグを含むプログラムを入力とし，遺伝的プログラミングに基づいてプログラムに変更を繰り返し，入力で与えられた全てのテストケースを通過するプログラムを探索する．GenProg が行う変異操作には以下の 3 つがある．

挿入 欠陥箇所の次の行に修正対象プログラム中に存在しているプログラム行を挿入する処理

削除 欠陥箇所を削除する処理

置換 挿入と削除の両方を行う処理

GenProg の終了条件は，全てのテストケースを通過する個体を発見する，または既定の世代数まで到達することである．

3　システム概要

本システムでは，以下の 3 つを入力として受け取り，全てのテストケースに通過する Scratch プロジェクトを出力する．

- バグを含むプログラム（Scratch プロジェクト）
- テストスイート
- テストに関する設定ファイル

提案手法は以下のステップで構成される．

Step1 Scratch のプロジェクトから，そのブロック構造を表した C 言語のソースコードに変換する．

Step2 修正対象プログラムをテストスイートに適用し，カバレッジ情報を取得する．

Step3 初期変異プログラム群を生成する．修正対象プログラムとカバレッジ情報から GenProg が自動的に生成する．

Step4 Step3 及び Step7 で生成される C 言語の変異プログラムを Scratch プロジェクトに変換するための前処理を行う．

Step5 前処理済みの変異プログラムを Scratch プロジェクトに変換する．

Step6 Scratch の自動テストフレームワークである Whisker を用いて自動テストを行う．テスト結果及びカバレッジ情報は GenProg に送る．全てのテストケースを通過するプログラムがあれば，そのプログラムを出力し，システムは終了する．

Step7 Step6 のテスト結果を元に GenProg の終了判定や適応度の評価・優秀な個体の選択を行う．そして，選択された個体に対して交叉や変異の操作を行い，次世代の個体群を生成する．

Step8 Step4 まで戻る．

4　実装

Scratch のプロジェクトファイルを解析し，変数やメッセージなどの定義及びブロック構造を取得し，C 言語に変換する機能を新たに実装した．ほとんどのブロックは関数呼び出しとして表現するが，繰り返しや条件分岐を扱うブロックは，while・if 文を用いて表現する．

得られた C コードに GenProg を適用すると，変異によって上記のブロックの表現方法から外れたプログラムが生成されることがあるため，上記のブロックの表現方法になるように補正してから Scratch のプロジェクトファイルに逆変換する処理を行う．

5　実験

本研究では，以下の 2 つの実験を行なった．

実験 1 Scratch の 3 つの入門者用プロジェクト（Pong Starter, Maze Starter, Hide and Seek）に対して，ランダムな 1 つのブロックに変更を行いバグを生成し，本システムによって生成したバグの修正を試みる．なお，各プロジェクトに対して，変更内容ごとに 2 種類のバグプログラムを生成した計 24 個のプログラムで実験を行う．

実験 2 Scratch Web サイトにユーザが投稿している実験 1 で使用した入門者用プロジェクトを参考に作成されたと思われるプロジェクトの中からランダムに選択し，本システムによってバグの修正を試みる．入門者用プロジェクトを参考に作成されたかどうかは，以下の要素を総合的に判断して決定する．なお，各入門者用プロジェクトに対して 3 つのユーザが作成したプロジェクトを選択し，実験を行う．

- プロジェクト名が酷似している
- 使用されている画像データが同じである
- スプライトの設定が酷似している
- プロジェクトの機能またはブロック構造が酷似している

6　実験結果

表 1 は実験 1 の結果から，変更内容ごとに修正率と実行時間を計算したものである．また，修正対象プログラム中に修正に必要なブロックが存在した場合の修正率は 65.0% であった．

表 1　変更内容ごとの修正率及び修正時間

変更内容	修正率	実行時間（s）
1 ブロック削除	1/6	850.6
1 ブロック追加	6/6	324.6
1 ブロック置換	2/6	858.7
1 ブロック移動	4/6	537.9

表 2 は，実験 2 の結果である．修正率は約 44.4% であり，実行時間は平均 628.89 秒であった．また，修正対象プログラム中に修正に必要なブロックが存在した場合の修正率は約 66.7% であった．

表 2　実験 2 の結果

プロジェクト	修正結果	実行時間（s）	必要ブロック
Pong Starter	O	359.2	O
Pong Starter	X	1226.7	X
Pong Starter	X	1133.3	X
Maze Starter	O	93.4	O
Maze Starter	X	562.6	X
Maze Starter	X	489.7	O
Hide and Seek	O	760.5	O
Hide and Seek	O	503.4	O
Hide and Seek	X	531.1	O

7　まとめ

本研究では，APR 手法の一つである GenProg と Scratch 用自動テストフレームワークの Whisker を用いて，Scratch 用 APR ツールを開発し，評価を行なった．その結果，修正対象プログラム中に修正に必要なブロックが存在している場合において，本システムによる自動プログラム修正はある程度の有用性を示した．

参考文献

[1] Le Goues, C. et al.,“GenProg: A Generic Method for Automatic Software Repair”, IEEE Transactions on Software Engineering, vol.38, no.1, pp.54–72, 2012.

[2] Stahlbauer, A. et al.,“Testing Scratch Programs Automatically”, ESEC/FSE 2019: 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp.165–175, 2019.