

平成 30 年 6 月 12 日現在

機関番号：62615

研究種目：若手研究(B)

研究期間：2016～2017

課題番号：16K16039

研究課題名(和文) Webアプリケーションの自動プログラム修正

研究課題名(英文) Automated Program Repair for Web Applications

研究代表者

前澤 悠太 (Maetzawa, Yuta)

国立情報学研究所・先端ソフトウェア工学・国際研究センター・特任助教

研究者番号：90759771

交付決定額(研究期間全体)：(直接経費) 3,100,000円

研究成果の概要(和文)：生成・検証による自動プログラム修正(生成・検証手法)は、ソフトウェアの保守コストを削減するために有望な手法である。本研究では、Webアプリケーション開発への生成・検証手法の適用を提案し、提案手法をRevAjaxMutatorと呼ぶツールに実装した。効果的かつ効率的な自動修正のために、我々はWebアプリケーションの特徴を元にしたプログラム修正操作及び修正空間探索を開発した。20個の実Webアプリケーションを用いた適用事例実験では、RevAjaxMutatorは内14個に対して正しく欠陥を除去するパッチを生成できたことを確認した。

研究成果の概要(英文)：Generate-and-validate program-repair is a promising technique for reducing the software maintenance cost. In this study, we propose the first application of the technique to a domain of Web applications. For effectively and efficiently repairing faulty Web applications, we developed repair operators and space search based on the features of Web applications. We conducted case studies on 20 real-world Web applications and found that RevAjaxMutator suggested the correct patches for 14 of the applications.

研究分野：ソフトウェア工学

キーワード：テスト・デバッグ パッチ生成 Webアプリケーション

### 1. 研究開始当初の背景

生活基盤として利用される Web アプリケーションの開発において、ソフトウェア開発者(以下、開発者)が欠陥を十分に検出・除去(テスト・デバッグ)することが重要である。先行研究では、人工的に欠陥を作りテストの検出能力を測定する変異解析に取り組んできた。変異解析は欠陥を十分に検出するテストの実施に役立つが、開発者は検出された欠陥を正しく除去する必要があり難しい。

### 2. 研究の目的

本研究では、Web アプリケーションの欠陥を自動的に除去する自動プログラム修正手法を提案する。提案手法は「変異解析における欠陥作成の逆操作(デバッグ操作として定義)は欠陥を除去できる」という着想に基づき、

- (1) 正しく欠陥を除去できるデバッグ操作を
- (2) 効率的に探索する。

開発コストの支配項であるテスト・デバッグ作業の自動化は開発工数削減につながるという点で本研究の意義は大きい。

### 3. 研究の方法

(1) 本研究では、Web アプリケーションに典型的な欠陥を挿入する変異操作を精査し、それらが埋め込む欠陥を除去できるデバッグ操作を定義する。現実の Web アプリケーションを用いた予備実験を行い、提案手法を実装したツール(RevAjaxMutator と呼ぶ)が開発者の経験則に基づく手動修正を再現できるか確認する。

(2) 本研究ではさらに、Web アプリケーションの特徴を利用してプログラム修正空間を効率的に探索する手法を開発し、RevAjaxMutator に実装する。現実の Web アプリケーションを用いた適用事例実験を通して、RevAjaxMutator が実用的な時間で Web アプリケーションの欠陥を除去できるか確認する。

### 4. 研究成果

RevAjaxMutator の有用性を評価するために、次の 2 つの研究課題に回答する。

- (1) 欠陥修正能力: RevAjaxMutator は Web アプリケーションの欠陥を正しく修正するためのパッチを生成できるか?
- (2) 実用性: RevAjaxMutator は実用的な時間内で正しいパッチを提示できるか?

#### 4.1 実験設定

適用事例実験で利用する実欠陥を収集するため、我々は GitHub レポジトリから Web アプリケーションにおけるプログラム修正をマイニングする機能を実装した。さらに、修正前後で障害が再現・解消されるか検証することで、実欠陥に対するプログラム修正であることを

確認した。結果として、我々は 20 個の実欠陥を収集した。

(1) また、マイニング結果から、我々は修正対象の実欠陥に対する開発者の実修正(ground truth)を獲得した。RevAjaxMutator が提示したパッチが ground truth と一致した場合のみ、我々は RevAjaxMutator が正しく欠陥を修正できたと判定した。

(2) RevAjaxMutator を実行する計算資源として、一般的な開発者の開発環境を想定して、Amazon EC2 の c4.xlarge タイプ(4CPU・7.5GB メモリ)の仮想マシンを利用した。仮想マシン上に欠陥版の Web アプリケーションを構築し、夜間の実行状況を想定して、8 時間をタイムアウトに設定して RevAjaxMutator を起動し、正しいパッチを提示するまでに要した解析時間を測定した。

#### 4.2 比較実験

(1) 我々はまず、RevAjaxMutator に実装したデバッグ操作( $R_{a,jax}$ )の欠陥修正能力を評価した。比較対象として、生成・検証による自動プログラム修正(生成・検証手法)の開門ツールである GenProg のデバッグ操作( $R_{stmt}$ )<sup>引用文献①</sup>を RevAjaxMutator に実装した。加えて、変異解析における欠陥作成操作は欠陥除去にも利用できるという双対性<sup>引用文献②</sup>に関して、我々は Web アプリケーション向けの特化変異操作( $M_{a,jax}$ )<sup>引用文献③</sup>と汎用変異操作( $M_{gnrc}$ )<sup>引用文献④</sup>も RevAjaxMutator に組み込み比較実験を行った。さらに、Web アプリケーションのための構築・是正による自動プログラム修正(構築・是正手法)の最先端ツールである VejoVis<sup>引用文献⑤</sup>を実欠陥に対して適用した。

(2) 我々は次に、RevAjaxMutator に実装した修正空間探索( $S_{a,jax}$ )の実用性を評価した。先行研究<sup>引用文献⑥</sup>により、修正空間において、GenProg に実装されている遺伝的プログラミングに基づく探索よりランダム探索が優位であることが示されている。そこで我々は、RevAjaxMutator にランダム探索( $S_{rand}$ )を実装して比較実験を行った。

#### 4.3 結果・考察

(1) 表 1 に欠陥修正能力に関する実験結果を列挙する。チェックマーク✓は実欠陥それぞれに対して正しいパッチを生成できたことを示す。実験結果から、提案手法  $R_{a,jax}$  は 20 個の実欠陥のうち 14 個に対して動作した。一方、比較対象である既存手法はそれぞれ( $R_{stmt}$ ,  $M_{a,jax}$ ,  $M_{gnrc}$ , VejoVis)=(2, 1, 0, 3)個の実欠陥でしか動作しなかった。この結果から、RevAjaxMutator は既存手法よりも高い欠陥修正能力があると考えられる。

この欠陥修正能力に関する有望な結果は、提案手法  $R_{a,jax}$  が①Web アプリケーションにお

表 1 欠陥修正能力に関する実験結果

Ajax program $\mathcal{P}$	$R_{ajax}$	$R_{stmt}$	$M_{ajax}$	$M_{gnrc}$	Vejovis
multiscroll.js (MSJS)	-	-	-	-	✓
material (MT)	-	-	-	-	-
flatpickr (FP)	✓	-	-	-	-
lazy.js (LJS)	✓	-	-	-	-
handsontable (HOT)	✓	-	-	-	-
jquery-ui (JQUI)	-	-	-	-	-
Leaflet (LL)	✓	-	-	-	-
tabulator (TBLT)	-	-	-	-	-
roundcubemail (RM)	✓	-	-	-	-
moodle (MDL)	✓	-	-	-	-
areweskimyet (AWSY)	-	-	-	-	-
mortar-list-detail (MLD)	-	-	-	-	-
TimedMediaHandler (TMH)	✓	✓	-	-	-
wikihiero (WH)	✓	-	-	-	-
gallaria-press (GP)	✓	-	-	-	-
grand-media (GM)	✓	-	-	-	-
hero-maps-pro (HMP)	✓	-	-	-	-
in-over-your-archives (IOYA)	✓	✓	✓	-	✓
press-this (PT)	✓	-	-	-	-
themes-plus (TP)	✓	-	-	-	✓
Total	14	2	1	0	3

る②欠陥修正に特化した設計に起因すると考えられる。既存手法  $R_{stmt}$  は②欠陥修正操作だが①C プログラムといった汎用的な設定を適用対象としている。また、既存手法  $M_{ajax}$  は① Web アプリケーションに特化しているが②欠陥を挿入するための操作である。既存手法  $M_{gnrc}$  には①汎用的な設定において②欠陥を挿入するために設計されており、適用事例実験においては動作しなかった。既存手法 Vejovis には① Web アプリケーションに特化して②欠陥を修正するためのツールだが、Web アプリケーションの特徴の 1 つである Document Object Model (DOM) における修正対象としており、Web アプリケーションの特徴を包括的に対象とする RevAjaxMutator の方が適用範囲が広いことが示された。

しかしながら、6 個の実欠陥に対して RevAjaxMutator は正しいパッチを生成できなかった。これら結果を精査したところ、RevAjaxMutator が実行時パラメータや局所・大域変数、ライブラリ仕様に関わる欠陥の修正に実装上の限界があることがわかった。これら欠陥を修正対象とすると探索すべき修正空間が爆発する恐れがあるため、高度なデータフロー解析を RevAjaxMutator に組み込むことが今後の課題である。

(2) 表 2 に実用性に関する実験結果を列挙する。既存手法  $S_{rand}$  はランダム性があるため、各実欠陥に対して 10 回実行して平均時間を算出した。また、表記  $N/A$  は正しいパッチを提示できなかった場合を示し、タイムアウトの 8 時間を代入して平均時間を算出した。実験結果から、既存手法  $S_{rand}$  は実欠陥の修正に平均 273.8 分の解析時間を要したのに対して、提案手法  $S_{ajax}$  は平均 170.5 分で実欠陥を修正できた。この結果から、RevAjaxMutator は既存手法より効率的にプログラム修正空間を探索でき、実用性が高いと考えられる。

表 2 実用性に関する実験結果

Ajax program $\mathcal{P}$	$S_{ajax}$	$S_{rand}$
multiscroll.js (MSJS)	$N/A$	$N/A$
material (MT)	$N/A$	$N/A$
flatpickr (FP)	3.9m	68.2m
lazy.js (LJS)	1.1m	65.9m
handsontable (HOT)	43.9m	$N/A$
jquery-ui (JQUI)	$N/A$	$N/A$
Leaflet (LL)	0.2m	0.2m
tabulator (TBLT)	$N/A$	$N/A$
roundcubemail (RM)	110.2m	99.3m
moodle (MDL)	10.5m	101.8m
areweskimyet (AWSY)	$N/A$	$N/A$
mortar-list-detail (MLD)	$N/A$	$N/A$
TimedMediaHandler (TMH)	2.0m	107.2m
wikihiero (WH)	14.1m	161.2m
gallaria-press (GP)	11.3m	285.6m
grand-media (GM)	1.1m	270.1m
hero-maps-pro (HMP)	1.0m	21.6m
in-over-your-archives (IOYA)	7.5m	394.0m
press-this (PT)	410.2m	$N/A$
themes-plus (TP)	3.0m	61.0m
Average	170.5	273.8

#### 4. 4 結論

本研究では、Web アプリケーション分野における生成・検証による自動プログラム修正手法の適用を提案した。適用事例実験の結果から、RevAjaxMutator は夜間実行を想定した 8 時間以内に、20 個の実欠陥のうち 14 個に対して正しい修正パッチを提示できた。したがって、RevAjaxMutator は開発者が高信頼の Web アプリケーションを構築するために役立つと結論づける。

#### <引用文献>

- ① W. Weimer, et al., “Automatically Finding Patches Using Genetic Programming”, ICSE’ 09, pp. 364-374
- ② V. Debroy and W.E. Wong, “Using Mutation to Automatically Suggest Fixes for Faulty Programs”, ICST’ 10, pp. 65-74
- ③ K. Nishiura, et al., “Mutation Analysis for JavaScript Web Applications Testing”, SEKE’ 13, pp. 159-165
- ④ S. Mirshokraie, et al., “Efficient JavaScript Mutation Testing”, ICST’ 13, pp. 74-83
- ⑤ F.S. Ocariza, et al., “VejoVis: Suggesting Fixes for JavaScript Faults”, ICSE’ 14, pp. 837-847

- ⑥ Y. Qi, et al., “The Strength of Random Search on Automated Program Repair”, ICSE’ 14, pp.254-265

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 0 件)

[学会発表] (計 0 件)

[図書] (計 0 件)

[産業財産権]

○出願状況 (計 0 件)

○取得状況 (計 0 件)

[その他]

ホームページ等

<https://github.com/mzw/RevAjaxMutator>

6. 研究組織

(1) 研究代表者

前澤 悠太 (MAEZAWA, Yuta)

国立情報学研究所・先端ソフトウェア工学・

国際研究センター・特任助教

研究者番号：9 0 7 5 9 7 7 1

(2) 研究分担者

なし

(3) 連携研究者

なし

(4) 研究協力者

なし