

Spin モデル検査器を用いた MVC 型 Web アプリケーションの検証支援

1190325 里見 優太 【ソフトウェア検証・解析学研究室】

1 はじめに

一般に、ソフトウェアの不具合を見つけ出す手法としてテストがよく用いられる。しかし、見逃しのないように十分網羅的なテストを実施することはコストが大きく、また見逃しが無いことを保証することも困難である。テストが持つこのような欠点を補う目的で、モデル検査等の形式手法を用いて不具合を検出する方法が多数研究されている。Web アプリケーションにおける上記のような不具合を形式手法を用いて検出する研究として [1] がある。先行研究では MVC アーキテクチャを使用した Web アプリケーションのソースコードから、形式仕様記述言語 Alloy を用いた検証コードの自動生成を行い、検証を行うことで不具合がないか確かめている。検証内容は Alloy 記法でソースコード内にコメントとして記述する。

しかし、[1] では、Alloy の詳細な仕様や [1] で行う変換内容を理解していなければ検証項目の記述を行うことができない問題や、数値をうまく取り扱えず、数値に関する検証ができない問題がある。

そこで、本研究では形式手法の詳細な仕様を理解していなくても容易に検証項目を記述でき、数値に対する処理を検証することができる Web アプリケーション検証支援ツールを開発する。

2 提案手法

提案手法の概略を図 1 に示す。本研究では図 1 に示す、変換器を作成した。また、検証対象は Python の Web アプリケーションフレームワークである Django を使って構築されたアプリケーションとした。本手法では検証対象のモデル内で宣言された整数型のフィールドに対して、データの処理を行う際に不具合がないことを検証する。利用者は、検証に用いる変数と検証したい性質を検証内容記述ファイルに記述し、検証対象ソースコード内にコメントとして検証を行う場所を記述する。それに対して本システムは、検証対象ソースコードと検証内容記述ファイルに基づいて Promela コードを自動生成し、

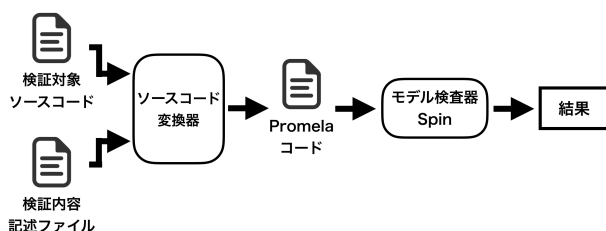


図 1 提案手法の概要

モデル検査器 Spin で検証を行う。

検証内容記述ファイルと検証対象ソースコードの例をプログラム 1 と 2 に示す。プログラム 1 の 5~6 行目が検証内容の記述である。検証対象メソッドの実行前・実行後の変数の値を、変数名に before_・after_ を付けて表している。

プログラム 1 検証記述ファイルの例

```

1 # use_value
2 stock.count = 10
3 cart.count
4
5 # add_cart_assert
6 spin_assert1 assert(after_stock.count ==
   before_stock.count -1)

```

プログラム 2 検証対象ソースコードの例

```

1 def add_cart(request, item_name):
2     . . . . .
3     stock = Stocks.objects.get(name=item_name)
4     cart = Cart.objects.get(name=stock.name)
5     stock.count -= 1
6     cart.count += 1
7     # spin_assert1
8     . . . . .

```

3 実験結果

本研究で作成した変換器の評価を行うために、Django を用いて 2 種類のショッピングサイトアプリケーションの作成を行い、検証を行った。一方は先行研究 [1] で検証対象に用いられているものと同じ仕様で実装を行い、もう一方は外部ライブラリを用いて機能を追加したものを実装した。その結果、双方のアプリケーション共にモデルに記述されている値のみを用いた処理については変換を行うことができた。また、前者のアプリケーションについては作成中に混入した不具合を検証を行うことで見つけ出すことができた。しかし、一部の処理や外部ライブラリを用いた処理に対して、変換を行うことができない問題があると判明した。

4 まとめ

本稿では、MVC 型 Web アプリケーションで用いる整数型フィールドに対するアクションに不具合が含まれていないか検証するためのシステムの作成を行った。その結果、先行研究では見つけ出すことの出来なかった、アプリケーションに含まれている不具合を見つけることが出来た。今後の課題として、現在対応していない構文への対応や外部ライブラリを交えた場合の変換規則の追加が考えられる。

参考文献

- [1] 水谷 浩明, 結縁 祥治, “Alloy を用いた Ruby on Rails アプリケーションの開発支援”, 電子情報通信学会技術研究報告, SS2011-64, pp. 43-48, 2012