
類似コード検出ツールを用いたテストコード再利用に向けた調査

Investigation for test code reuse using similar code detection tool

倉地 亮介* 崔 恩潯† 飯田 元‡

あらまし テストコード自動生成ツールによって生成されるテストコードは開発者の保守作業を困難にする課題がある。我々は、解決方法として既存テストの再利用によるテストコード自動生成ツールを提案する。本研究では、提案ツールの実現に向けて類似コードペアの分類を行い、類似コードペア間の類似度とテストコードペア間の類似度の関係を調査した。

1 はじめに

ソフトウェア開発のテスト工程において、テスト作成コストを削減するためにテストコード自動生成ツールの利用が進んでいる。しかし、既存のツールによって生成されるテストコードはテスト対象コードの作成経緯や意図に基づいていないという性質から開発者の保守作業を困難にする課題がある [1]。そこで、解決方法として既存テストの再利用によるテストコード自動生成ツールが必要であると考え。

我々は、テストの再利用手法として、類似コード間でのテスト再利用手法を提案する。この手法は、類似コード検出ツール [2] を用いてテストコードがないコード片 a に対して類似したコード片 a' を検出し、その類似コード片に対応するテストコードをコード片 a に再利用する方法である。本手法で、テスト再利用の候補となるのは「どちらか片方のコード片にテストコードが存在する類似コードペア」である。

既存テストの再利用は命名規則に従った保守性の高いテストコードを利用できる。一方で、類似コード間でのテスト再利用は再利用候補となる類似コードペアが存在しないとできないことや、テスト対象となる類似コードペア間の関係に依存するので困難な作業である。そのため、どのようなテストコードが類似コード間で再利用できるのかを明らかにすることがテストコードの再利用支援において重要である。

本研究では、既存の Java プロジェクト内の類似コードペアをテストコードの有無によって分類し、プロジェクト内に存在する再利用候補となる類似コードペアの割合と、「両方のコード片にテストコードが存在する類似コードペア」を対象に類似コードペア間の類似度とテストコードペア間の類似度の関係を調査した。

2 類似コードペアの分類

調査を行うために、類似コードペアをテストコードの有無によって3種類に分類する。類似コードペアの分類手法は、主に以下の5つのステップから構成される。

Step1: プロジェクト内のテストコードとテスト対象コードを収集

Step2: Step1 で得られたテスト対象コードから類似コードペアの検出

Step3: Step1,2 で得られた類似コード片とテストコードをメソッド単位で対応付け

Step4: Step3 の対応付け表からテストコードの有無により類似コードペアを分類

3 調査概要

調査 1: プロジェクト内にテストの再利用候補となる類似コードのペアはどの程度存在するか

*Ryosuke Kurachi, 奈良先端科学技術大学院大学

†Eunjong Choi, 京都工芸繊維大学

‡Hazimu Iida, 奈良先端科学技術大学院大学

表 1 既存 Java プロジェクト中の類似コードペアの分類結果

プロジェクト名	テストコードが存在しない類似コードペア		どちらか片方のコード片にテストコードが存在する類似コードペア		両方のコード片にテストコードが存在する類似コードペア		合計
	数	割合	数	割合	数	割合	
Apache maven	260	63.9	139	34.2	8	2.0	407
Apache kafka	442	70.8	135	21.6	47	7.5	624
Apache kylin	177	72.2	60	24.5	7	2.9	245
合計	879	68.9	334	26.2	62	4.9	1275

表 2 類似コードペア間の類似度とテストコードペア間の類似度の関係

	テストコードペア間の類似度			
		Not Similar	type2	type3
	類似コードペア間の類似度	Not Similar	21	0
		type2	6	77
		type3	6	28

調査 2: 類似コードペア間の類似度と対応するテストコードペア間の類似度にはどのような関係があるか

4 調査結果

調査 1 の結果を表 1 に示す。3 つの Java プロジェクト中のテスト対象となる類似コードペアの内、26.2%が再利用候補になることが分かった。この結果は、類似コードペアの中で片方のコード片にはテストコードがあるにも関わらず、もう片方のコード片はテストされていないものが全体の 4 分の 1 以上を占めており、提案ツールの実現により多くのコード片にテストコードを再利用できる可能性を示している。

調査 2 では、類似コードペアの分類によって得られた「両方のコード片にテストコードが存在する類似コードペア」62 個とそれに対応するテストコードのペア 153 個の類似度の関係を表 2 に示した。調査の結果、テスト対象となる類似コードペアが類似していない (Not Similar) 場合は、類似するテストコードペアが存在しないことが分かった。また、類似コードペアの類似度がタイプ 2,3 の場合は、テストコードの類似度もタイプ 2,3 が多い結果となった。この結果から、テストコードペアの間の類似度と対象の類似コードペア間の類似度には相関関係があり、類似度が高いほどテストコードを再利用できる可能性を示している。

一方で、類似コードペア間の類似度がタイプ 2 と高いにもかかわらず、テストコードペアが類似しない組み合わせが 6 件検出された。これらの類似コードペアのメソッドは、同じ制御構造を持つが、最後に出力する数値の選択だけが異なっている。検出された例として、Apache kafka では、同一の制御構造で特定のオブジェクトを取得した後、get メソッドでデータを取得する処理と、delete メソッドでデータを削除する処理が類似コードペアとなっていた。このように共通のデータを使用し、互いに関係した処理であっても、異なる処理を実行していること場合は、テストコードを再利用することは難しい。今後は、類似コードペア間の振る舞いに着目して分類を行い更なる調査をする予定である。

参考文献

- [1] S. Shamshiri, J. Rojas, J.Pablo Galeotti, N. Walkinshaw and G. Fraser. How Do Automatically Generated Unit Tests Influence Software Maintenance?. *Proc. of ICST*, pp.239–249, 2018.
- [2] Chanchal, K. R. and James, R. C.: NICAD: Accurate Detection of Near-Miss Intentional

An example of use for `fose2019.cls`

Clones Using Flexible Pretty-Printing and Code Normalization. *Proc. of ICPC*, pp.172–181 (2008).