

進捗状況

2018年11月01日 (木)

Empirically Revisiting the Test Independence Assumption

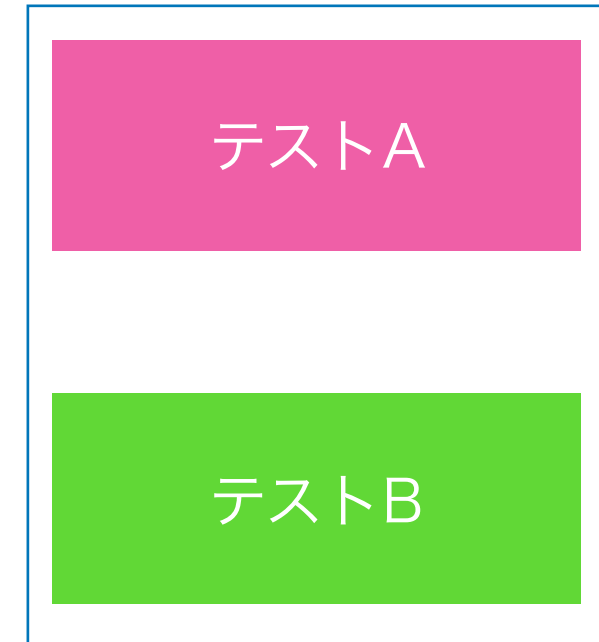
※卒論で既存研究の”テストの依存性”について書きたかったので読みました

依存テスト

順序依存テスト

- 2つのテストAとBを含むテストスイート
 - * テストAを実行してからBを実行：テストはPass
 - * テストBを実行してからAを実行：テストはFail
- テストBを実行した順によってテストAの結果が異なる場合がある
- このようなテストAを、**順序依存テスト**と呼ぶ

テストスイート



Pass



Fail

テストの独立性

- **テストは他のテストに影響を及ぼしてはならない**
 - * テストを任意の順序で実行しても同じ結果が得られるようにしなければならない
- 多くのテスト優先順位付けに関する論文で明示的にテストは独立していると仮定している
 - * 2000年から2013年, ICSE, FSE, ASEなどにテスト優先順位付けに関する31件の論文が掲載された[1].
 - ▶ 27件の論文が明示的にテストの独立性を前提としていた

[1]S. Zhang and K. Muşlu. List of papers on test prioritization techniques from major software engineering conferences and journals, 2000–2013.
http://homes.cs.washington.edu/~szhang/test_prioritization_paper_list_2000-2013.txt.

依存テストの発生原因

- 他のテストとの相互作用に起因
 - * 開発者がコードを記述する際に、**実行環境や使用するリソースを初期化し忘れる**などのミスが原因
 - ▶ 実行環境：グローバル変数の値、ファイルの内容など
- 依存テストが存在すると、別のテストによって実行環境が変更された時に元の実行環境を無効にする可能性がある

貢献

- 開発者にとって有用
 - * 依存テストは**偽陽性や偽陰性の発生原因につながるため**，無視できない問題であると分かる
- 開発者はどのコーディングパターンがテスト依存を引き起こすのかを知ることができる

問題と目的

- 問題点

- * 既存研究では、依存テストによって及ぼされる影響について調査していない

- 研究の目的

- * 依存テストが及ぼす影響を分析する
 - * 依存テストを抽出するためのツールを提案し様々なプロジェクトで評価実験を行う

卒論の リサーチクエスチョン

卒論での目的

- 偽陰性による見逃し欠陥を収集する
 - 再修正されたプロダクトコードでは見逃し欠陥が修正されている可能性があり、偽陰性の発生原因が明示的に分かる可能性が高いと考えられる。
- 偽陰性の発生原因を明らかにする

リサーチクエスション

- RQ1：REOPENEDされたテストはどの程度の頻度で偽陰性を引き起こすのか？
 - * REOPENEDのタグがつけられたテストは、テストコードを修正する必要があった原因とテストコードの修正パッチについての議論があるため、テストコードの欠陥の発生原因を特定することが可能である。よって偽陰性を引き起こしたテストの発見が容易になる
- RQ2：REOPENEDされたテストのうち、偽陰性を引き起こしたテストに不足であったテストケースは何か？
 - * 偽陰性を引き起こしたテストに不足だったテストケースが何かを分析することにより、テストが必要な箇所を特定し、テストを行う範囲を広げることができる

リサーチクエスチョン候補

- RQ：REOPENEDされたテストは他のテストと依存関係にあったのか？
 - * 開発者がコードを記述する際に、**実行環境や使用するリソースを初期化し忘れる**などのミスが原因であるとされているため、依存テストが存在しているかどうかを分析することで偽陰性を引き起こす原因を明らかにできるかもしれない（？）