トップエスイー修了制作

複数のソースコード解析手法の組み合わせによる オープンソースソフトウェアの品質確認

富士通株式会社

出内 将夫

ideuchi.masao@jp.fujitsu.com

開発における問題点

最新技術への追従や、技術標準への準拠をタイムリーに行うために、オープンソースソフトウェア(OSS)を活用したソフトウェア開発が、様々な分野で一般的となっている。ただし、OSSを製品やシステムに組み込み、継続的なサポートを提供するには、OSSの品質、特に信頼性について適切に評価する必要がある。



手法・ツールの提案による解決

様々なソースコード解析技術を組み合わせることにより、OSSの活用範囲における問題点を抽出し、活用範囲を対象とした強化テストの作成、および障害検知の自動化について、低コストで実現する手法を提案する.

解析技術適用の流れ

OSS選定

OSS活用範囲に絞った問題点の明確化

強化テスト作成/実施



ソースコード 利用するI/F

静的解析

全ソースコードを対象に問題点(候補)を抽出

Control Flow Graph(CFG)作成 関数の呼び出し関係 から活用範囲を算出

既存テストの カバレッジ算出 OSSのテストセットで 実行済の範囲を算出

解析結果 組合せ

活用範囲に限定した 問題点の分析, 既存テストケースの カバー範囲を明確化

パス解析

動的解析

活用範囲を対象に, 網羅的なテストケースを作成 実際に動作した際に, 問題が発生するかを確認

解析技術ごとの役割

解析技術	特徴·役割
静的解析	入力したソースコードから問題点を抽出する 技術. 入力外の要因による誤判定も存在.
CFG作成	関数の呼び出し関係をグラフ化する技術. 呼び出すI/Fからのグラフ構造をOSS活用範囲として抽出.
カバレッジ 算出	プログラムを実行した履歴をソースコードと対応づけて管理する技術. ソースコード動作確認の指標として利用.
パス解析	引数をシンボル化し、分岐網羅の実行パス を解析する技術、テストケース作成に利用.
動的解析	プログラム実行時にエラーを検知する技術.

実験結果と課題

> 実験内容

提案する手法のうち、問題点を明確化する部分を OSSの統計解析ソフトウェアであるRに適用した。 C言語のソースコード約106Kstepを対象に、静的解 析にはClang Static Analyzer、CFG作成にはClang、カ バレッジ算出にはgcovを用いた。

> 実験結果

静的解析の指摘数が全体で370件, 利用範囲として算出されたstep数は約3Kstepだった. 利用範囲内の指摘数は5件で, 既存テストケースでうち2件がカバーされていた. 指摘された問題の種別や, 実行回数を考慮した上で, 3件まで問題を絞り込めた.

▶ 課題

強化テストの作成/実施の部分の実施,機能面で仕様を満たしているかどうかの確認を行いたい.