

4K-1

テスト自動化ツールの構築

手島 文彰

株式会社 東芝 システム・ソフトウェア技術研究所

1. はじめに

ソフトウェアの品質を保障するためにさまざまな試みがなされているが、最も効果的な手段はプログラムのテストであろう。従来、プログラムのテスト手法に関しては、動的テスト手法が一般的に用いられてきた。つまり、多くのテストデータを用いて、実際にプログラムを実行して、その結果を調べる方法である。これまでは、テスト実施者の経験に基づくアドホック的なテストデータを選択するが、テスト範囲を計測することによって、ソフトウェアの信頼性を定量的に評価する方法が提案され、実施されてきた。ところが、テスト実施者によりテストの内容、量、質および工数が大きく異なる。これでは、ソフトウェア全体としての信頼性が十分であるといえないばかりか、ソフトウェアの生産性の面からも問題である。

本稿では、プログラム仕様に基づいて客観的なテストデータを生成し、プログラムが仕様を満たしていることを検証するテスト支援システムについて述べる。

2. テスト支援システムの機能構成

テスト支援システムは、帰還型テスト自動化システムである。すなわち、仕様記述されたアサーションによる実行検証を行うと同時に、それらの検証情報により表現されるプログラムのふるまいに基づいてテストケースを自動的に生成・追加するフィードバック機構をもつ。また、テスト支援システムは、プログラム開発を想定してサブルーチンからプログラムまでのテストを支援するものである。

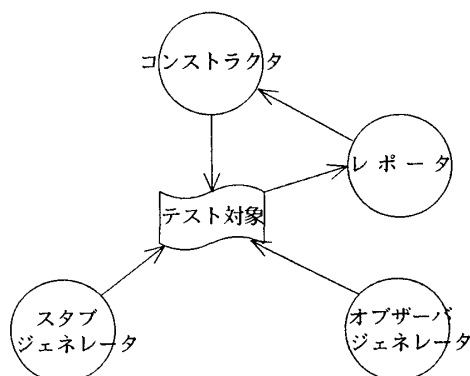


図1. テスト支援システムの機能構成

Construct of Automated Testing Tool.

Fumiaki TESHIMA

TOSHIBA Corp.

つまり、テスト・サイクルにおいては、単体テストおよび統合テストを支援するものであり、図1に示されるような機能構成である。各機能は、以下の通りである。

[1] コンストラクタ

コンストラクタは、テスト対象の入力変数の定義域から入力値を選び、入力変数にその値をアサインする。すなわち、テストケースを生成するのである。このとき、コンストラクタはそれまでの選択された入力値に対するプログラムのふるまいにより、あらたな入力値を選んでいく。そして、テスト対象の全ての入力空間が網羅されるまで、テストケースを生成し、テストを繰返すのである。図1では、コンストラクタおよびレポートによるフィードバックが表現されているが、これは次のような処理を行う。

(1) コンストラクタ

①テストケースを生成する。

もし、入力変数の初期値集合が与えられているならば、そこからテストケースを生成していくが、それ以降はレポートの報告に基づいてテストケースを生成していく。

②テスト対象を起動する。

(2) レポート

テスト報告をまとめて、コンストラクタに報告する。

(3) (1) へ戻る。

[2] スタブ・ジェネレータ

スタブ・ジェネレータは、テスト対象プログラムが実行できるように、必要なスタブを生成する。このとき、生成されるスタブは、仕様書に基づいて入力値に応じた出力変数の値域の中の値を返す。

[3] オブザーバ・ジェネレータ

オブザーバ・ジェネレータは、オブザーバと呼ばれるモニタをプログラム仕様書に基づいて生成する。オブザーバは、図2に示されるように、呼ぶ側と呼ばれる側のサブルーチン間に挿入され、実行検証を行うとともにレポートのための情報を保存する。テスト対象を起動するオブザーバは親オブザーバと呼ばれ、そのテスト対象に従属しているものは子オブザーバと呼ばれる。そして、全てのオブザーバは、親オブザーバになりうる。つまり、図2で親オブザーバを指定することによって、それぞれのサブルーチンをテストすることが可能となるのである。オブザーバの機能は、以下のようなものである。

(1) 子オブザーバの場合

- * 呼ぶ側のサブルーチンの機能的正当性を検証する。
- * 呼ぶ側と呼ばれる側のサブルーチン間の正当性を検証する。
- * 入出力値を保存する。
- * サブルーチンの履歴を保存する。

(2) 親オブザーバの場合

親オブザーバの場合、子オブザーバの機能に加えて次のようになる。

- * コンストラクタを起動してテストケースを生成する。
- * テスト対象サブルーチンを起動する。

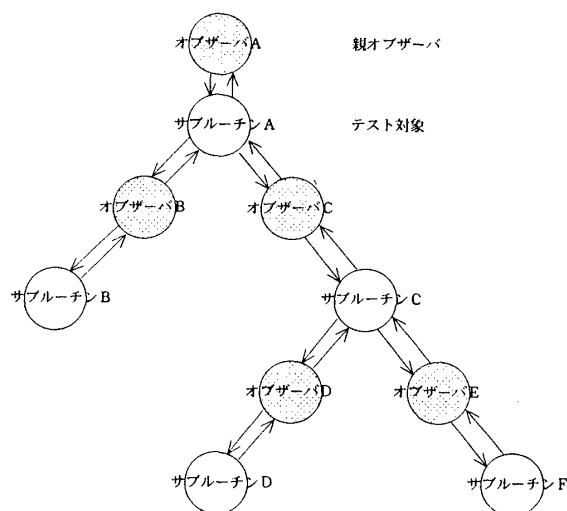


図2. テスト対象プログラム

[4] レポーター

レポーターは、プログラム実行後に、そのテストケースによるプログラムの行為に関する報告書を作成し、コンストラクタに報告する。報告書の内容は、以下の通りである。

- * 入力変数における定義域の選択値
- * 表明違反したサブルーチン名と文番号
- * 表明違反数

ただし、表明違反には次の2種類がある。

- ① テストの実行結果が期待した実行結果（表明）と合致しない。
- ② 表明が実行中に評価されない。

3. プログラムの検証

プログラムの検証は、基本的には呼ぶ側と呼ばれる側のサブルーチン間のデータのやりとりをプログラム仕様書に照らしてテストすることによって行われる。すなわち、従属関係にあるサブルーチン間で、次の3つの条件を満たすことを確認するのである。

- [1] それぞれのサブルーチンの出力が、そのサブルーチンの入力に対応する値域を、それぞれ満たす。
- [2] それぞれのサブルーチンの入出力変数が、それぞれの定義域あるいは値域を満たす。

[3] テスト対象の各入力空間の実行経路集合が互いに独立である。

これらの検証条件は、次のことと等価である。

- * サブルーチンを1つのブラックボックスとみなし、その機能を確認する。
- * サブルーチンのインターフェイスを確認する。
- * テスト対象における定義域が同値分割されていることを確認する。

図2においては、親オブザーバとしてオブザーバAが指定されると、オブザーバAはテストケースを生成し、テスト対象であるサブルーチンAを起動する。これによりテストが開始されるのである。そして、それぞれの子オブザーバは、上記の検証条件のうち条件[1]と[2]が満たされているかどうかを調べる。また、親オブザーバは条件[1],[2],[3]が満たされているかを調べる。このことにより、それぞれのサブルーチンの正しさが検証されるとともに、プログラム全体としての正しさが検証されることになる。

4. エラーの検出

テスト支援システムは、仕様書で表明された入力変数の定義域内外を縦断するようにテストケースを生成する。エラーは、上記の3つの検証条件の真偽によって検出される。例えば、定義域エラーは次のように検出できるだろう。

① 定義域分割の不足

これは、テスト対象の同一の入力空間から選択されたテストケースを実行したとき、上記の条件[1]または[3]を満たさないことに現れる。

② 定義域表明の不適切

これは、テスト対象の入力空間の外側から選択されたテストケースを実行したとき、上記の条件[1]かつ[3]を満たしてしまうことに現れる。

テスト支援システムは、エラーが検出されたならばさらにテストを繰返し、より詳細なエラー情報を得ることができるのである。

5. おわりに

テスト支援システムの概要が示された。本テスト支援システムは、次の特徴をもつ。

- * テスト結果のフィードバック機構をもつため、きめこまかいテストケースを自動的に生成することができる。
- * プログラムの検証を自動的に行える。
- * デバッグのための情報を提供する。
- * テスト労力を省力化することができる。
- * 全体として信頼性の高いソフトウェアを生産することができる。

6. 参考文献

- [1] Dorothy M. Andrews, "Automation of assertion testing: Grid and adaptive techniques.", In 18th Hawai International Conference on System Sciences, page 692-699. (1985)