

形式仕様記述を用いた系統的なプロパティベーステストの活用

馬場 勇輔¹ 荒木 啓二郎² 日下部 茂³ 大森 洋一²

概要：形式的な仕様は、仕様の曖昧さに起因する欠陥を予防し、手戻りによるコストを減らすことを期待されている。我々は、形式的な仕様記述に含まれるプロパティを、自動テストの一種であるプロパティベーステストで系統的に用いる手法を研究している。形式仕様記述をプロパティベーステストの情報源とし、系統的に用いることで、プロパティの妥当性を保証でき、手戻りによるコストだけでなく、実装コードのテストや仕様の検証のコストを減らすことができると期待される。

1. はじめに

ソフトウェア開発において仕様が曖昧なまま開発を進めると、しばしばその曖昧さに起因する欠陥を作りこむことも多い。そのため、仕様を厳密に記述することにより曖昧さを減らす形式手法には、仕様の曖昧さに起因する欠陥の予防が期待される。

我々は、形式手法VDMを用いた仕様をテストや検証に自動活用する手法を研究している。具体的には、VDM仕様に含まれる開発対象のプロパティを、プロパティベーステスト(PBT)に系統的に活用する手法を研究している。PBTとはテスト対象の性質(プロパティ)が満たされるか否かの観点から、テストデータを自動生成しテストする手法である。

図1に提案手法を適用したV字モデルを示す。本研究では、テスト対象の機能の記述を機能設計では陰操作(事前・事後条件のみが記述された実行不可な操作)、詳細設計では陽操作(アルゴリズムが記述された実行可能な操作)とし、陰操作からPBTに用いるプロパティを抽出する。

しかしながら、PBTの問題点として以下の2つが考え

られる。

(1) プロパティの妥当性が保証されないため、欠陥の原因がプロパティか他の要因か判断がしにくい

(2) ランダムテストであり、カバレッジが100%にならない

問題(1)の解決策として陰操作の情報をPBTのプロパティとすることで、陽操作の陰操作に対する妥当性をテストによって確認でき、欠陥の原因の特定が容易になる。問題(2)の解決策として完全なランダムテストではなく入出力変数の各同値クラス[3]からデータを生成することで、短時間でカバレッジを高められる。

本稿では、図1の範囲で提案手法を適用し、自動生成したデータでどの程度テストすべき範囲をカバーできるかを検証する。

2. テストデータ自動生成ツール

作成したツールの構成を図2に示す。まず、ツールの入出力について述べる。入力には陰操作を含むVDM仕様、出力は陽操作に対するテストスイート(TS)と、assertionsファイル(検証ファイル)である。TSは、操作の入出力の同値クラス(テストケース)群を指す。検証ファイルは、操作の入出力が正しいか判定する条件群ファイルを指す。このファイルを用いることで、操作の入出力が正しいかを確認できる。TSをPBTのプロパティとし、各同値クラスからデータを生成することで、テスト未実行部分を減らすことができる。このテストデータを用いて検証ファイルを生成する。テストケースと検証ファイルの対応から、各テストケースにおける入出力の正しさを確認できる。

次にツール内の各コンポーネントの入出力と機能の簡単な説明を以下に列挙する。

Parser 陰操作から、抽象構文木(AST)を生成

AST2DNF ASTから、操作の事前条件・事後条件を選

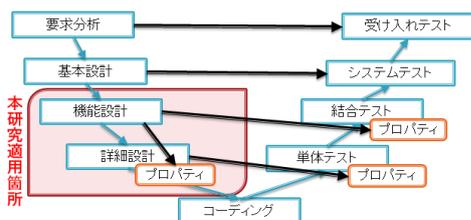


図1 提案手法を適用したV字モデル

¹ 九州大学大学院 システム情報科学府 情報知能工学専攻
² 九州大学大学院 システム情報科学研究院 情報知能工学部門
³ 長崎県立大学 情報システム学科

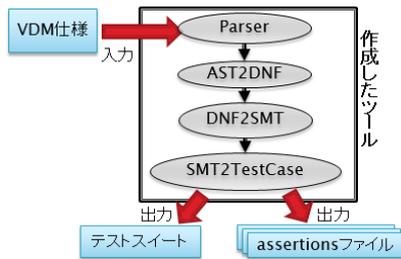


図 2 テストデータ自動生成ツールの構成

言標準形 (DNF) に変換することで、条件中の変数に関する同値分割を行い [3], DNF を生成

DNF2SMT DNF から, DNF の各節に対し SMT Solver により充足不能な節を検出する. その後, 他の節に内包される節を除去し充足可能な節のみの DNF を生成
SMT2TestCase 充足可能な節のみの DNF から, TS と検証ファイルを生成

3. 現状のツールの評価

Myers の三角形問題の VDM 仕様 [3] をツールに入力し, 出力の検証ファイル中の条件が各テストケースを満たすデータで構築されることを確認した. 現状のツールでは再帰呼出に対応していないため, 再帰性のない VDM 仕様に変更した. また, VDM の整数値はインタープリタでの実行時に 32bit 整数値として扱われる. 三角形の成立条件判定時に加算する必要があり, オーバー (アンダー) フローを防ぐため, 本評価では L の要素の値の上下限を定める.

テスト対象となる操作 CHARACTERIZATION (操作 C) は整数列 L を入力とし, L が三角形ならばその種類, 非三角形ならば非三角形と返す操作である. 我々は操作 C の陽操作を記述し, 生成された検証ファイルを用いて, C0 カバレッジの確認を行った. このときのカバレッジと呼出数を表 1 に示す. 表 1 の judgeTriangle (関数 J) は L が三角形か判定する関数である.

生成された全ての検証ファイルを用いることで, 操作 C のカバレッジが 100% となった. 仕様に記述していない部分は動作未定義であり, カバレッジが 100% ではない場合, 陰操作に記述していない部分を陽操作として記述している可能性がある. 今回, テストケース数は 7 であり, テストケース毎に 100 個の条件を生成しているため, 操作 C の呼出数が 700 となっている. 関数 J は自然数列型の不変条件として用いているため, L の要素が非自然数であると呼出が行われず, 呼出数が 700 とならない.

生成されたデータの考察を行う. 本研究でのツールの利用目的は, 陰操作に記述されている入力値に対する出力値の条件を陽操作が満たすかを確認することである. よって,

表 1 C0 カバレッジと呼び出し数

Function or operation	Coverage	Calls
CHARACTERIZATION	100.0%	700
judgeTriangle	100.0%	322

入力値の型が違う場合 [5] など陰操作で期待値が明確に定められていないテストデータは, 本提案手法の対応範囲外であるため考慮しない. しかし, 期待値が明確に定められているテストデータに関しては, PBT の仕組みでランダムに多数生成することでほぼ生成できる. ただし, ランダムに生成される可能性の低いデータは, 仕様に明確に記述されていなければ生成されない. 例として, L の要素が全て 0 の場合 [5] に関しては今回の検証では生成されなかった.

4. 関連研究

形式仕様記述をテストに活用する研究は以前から行われている [4]. 特に VDM 仕様を活用するという観点から, 本研究で用いている同値分割手法の研究 [3] がある. 文献 [3] では同値クラスから状態遷移図を生成し, 全状態を遷移する最短パスを生成している. 対して本研究では同値クラスを PBT に活用し, テストデータを自動生成している.

また, VDM 仕様をデジジョンテーブル生成に用いるという研究 [6] がある. 文献 [6] では, if 文などから条件部と動作部に分け, 真偽値を埋めたデジジョンテーブルを自動生成している. 文献 [6] ではテストデータは自動生成せず, 真である条件を満たすデータを手動で作成する. 対して本研究では, 動作定義が明確であるテストケースのみであるが, テストケースを満たすテストデータを自動的に生成できる. また, 文献 [6] では条件が矛盾したテストケースも生成されうが, 本研究では同値分割後に条件が矛盾する節を除去し, 条件が矛盾しないテストケースのみ生成する.

5. おわりに

本稿では, 形式的な仕様に含まれるプロパティを, 自動テストの一種である PBT で系統的に用いる手法を提案した. 上流工程で PBT によりテストデータを自動生成するツールを作成し, 陰操作で示された入出力の条件を満たすテストデータの生成ができていたことを, 陽操作の検証を行うことで示した. 今後は, 実装コードに対するテストデータを自動生成できるようツールの拡張を行う.

参考文献

- [1] 馬場勇輔, 荒木啓二郎, 日下部茂, 大森洋一: 形式仕様記述のプロパティベーステストへの活用, 火の国情報シンポジウム 2016 論文集, 情報処理学会 (2016).
- [2] ディペンダブル・システムのための形式手法の実践ポータル, <http://formal.mri.co.jp/> (2016/12/01 アクセス)
- [3] Dick, J. and Faivre, A.: Automating the Generation and Sequencing of Test Cases from Model-based Specifications, proc. of International Symposium of Formal Methods Europe, vol 670. Springer-Verlag, pp.268-284(1993).
- [4] Hierons, Robert M., et al.: Using Formal Specifications to Support Testing, ACM Computing Surveys, 41(2):9:19:76(2009)
- [5] Myers, Glenford J.: The art of software testing. A Willy-Interscience Pub(1979).
- [6] 西川拳太, 片山徹郎, 喜多義弘, 山場久昭, 岡崎直宣: 形式仕様を用いたデジジョンテーブル生成手法の提案, ソフトウェアエンジニアリングシンポジウム 2014 論文集, 情報処理学会 (2014).