

状態遷移仕様からのテストデータ生成方式

4R-2

岡安 二郎 平山 雅之 加地 浩一 三原 幸博
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

現在、我々は家電製品等に組み込まれる小規模制御用ソフトウェアの開発を支援する一貫支援環境の構築を行っている。本環境では状態遷移に基づく形式的仕様記述(状態遷移仕様記述)をもとに、プログラムの自動合成とテストという2つの方向からの支援を目指している。

従来、制御用のソフトウェアに対して仕様全体をテストすることは困難であり、重要な部分や誤りの発生しやすい部分をリストアップし、テストするという形がとられていた。しかし、この方法ではシステム全体として仕様をみたしているかを検証するのは難しい。そこで、本環境では状態遷移仕様からのテストデータ自動生成により、テストの十分性を高めるとともに省力化を図り、ソフトウェア開発の生産性・信頼性の向上を実現するアプローチをとる。

本稿では、テスト支援の中心的技術課題のひとつである、状態遷移仕様からのテストデータ生成方式について述べる。

2. データ生成の方針

2.1. 支援するテストの範囲

本テストデータ生成では、プログラムがシステムの動作仕様をみたすものになっているか(仕様通りのものになっているか)を検証するためのテストデータを生成する。

2.2. データ生成の方法

テストデータ生成方法の手順としては、状態遷移仕様から連続した遷移の系列を選定し、システムに対してその遷移系列を引き起こさせるようなイベント列をテストデータとして生成する方針をとる(fig. 1)。状態遷移仕様における遷移系列を選定するとは、その仕様によって記述されるシステムの一連の振舞い、動作の流れを規定することに相当する。ゆえに、この遷移系列を引き起こすようなデータを生成して用いることで、システムに仕様に記述した動作を実際に起こさせることができ、この動作をモニタすることにより、検証が可能となる。

2.3. テストデータに対する要求

テストデータのもととなる遷移系列に対する要求を以下に列挙する。

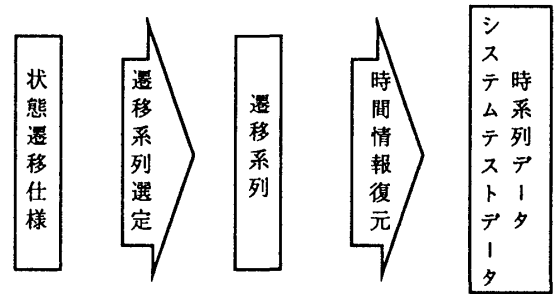


fig. 1 テストデータ生成方法

(1) 状態遷移仕様における状態・遷移を網羅する

(2) システムの実動作に基づく

(1)により、各状態での局所的な振舞いの検証が可能となる。

制御用のソフトウェアに対するテストにおいては、単に状態や遷移を網羅するだけでは不十分であり、システムの実動作に基づく、時間の経過を考慮した(結果として系列長の大きい)テストデータを生成する必要がある。(2)をみたす系列に基づくテストデータにより、長時間経過の後をはじめとられる処理のテストや、制御動作を多数回繰り返した後ではじめて現れる不具合の発見が可能となるのである(例えば、fig. 2のような仕様において、“積算タイムアウト”というイベントをシステムに対して引き起こすには、実際にタイムアウトが起こるまで時間が経過するようなテストデータを生成しなければならない)。次章では、これらの要求をみたすアルゴリズムを説明する。

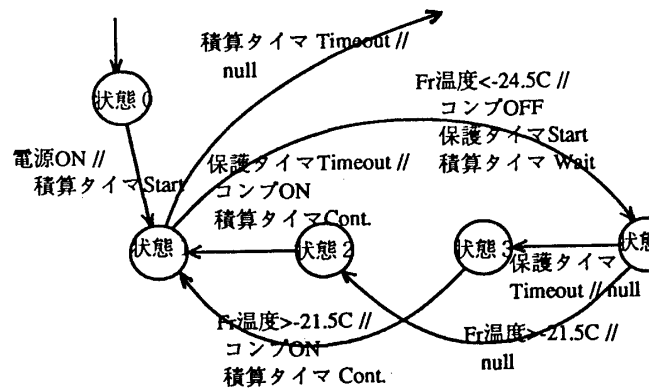


fig. 2 状態遷移仕様例

A Method of Generating Test Data Based on Specifications Modeled by State-Transition Diagrams

Jiro OKAYASU Masayuki HIRAYAMA Kouichi KAJI Yukihiro MIHARA

Systems & Software Engineering Laboratory, TOSHIBA CORPORATION

3. 遷移系列選定アルゴリズム

本アルゴリズムは時間のシミュレートを基本とした遷移系列選定アルゴリズムである。時間に関する事象(時間経過、タイマに関するイベント・アクション、タイマの状態)のシミュレーションをもとに、状態遷移図を有向グラフとみなして探索を行い、実動作に基づく遷移系列の選定を実現する。具体的なアルゴリズムを fig. 3 に示す。

本アルゴリズムの技術的特徴は、以下に挙げる点である。

- (A) 時間経過のシミュレート
- (B) タイマの状態、タイマに対するイベント・アクションのシミュレート
- (C) イベント優先度、遷移優先度
- (D) 一時的枝刈り

(A)(B)は時間のシミュレートの基本機能に対応する。

(C)は遷移選定の際、設計者の意図した、システムの実動作に最も対応している遷移を選定するための中心的な機能である。優先度は設計者により与えられ、複数の遷移が選定可能な状況では、優先度の最も高い遷移が選定される。ここで時間に関するイベントを最優先とし、さらにイベントがシミュレーションの上で生起している場合に限りその遷移を選定するというルールによって、実動作に基づく遷移系列の選定が実現できる。また、時間以外のイベントについても、実動作において生起可能性の高いイベントの優先度を高く定めることで、設計者の意図する遷移系列を選定できる。

(D)は遷移系列の系列長の爆発を防ぐとともに、遷移の網羅率を向上させる働きをする。

```

初期化：探索開始状態をスタックにプッシュ
主処理：while (状態スタックが空でない)
    { if (現在の状態から可能な遷移が存在する)
      { 優先順位が一番高い遷移を行う;
        時間に関するアクション;
        次状態をスタックにプッシュ;
        時間を進める;}
      else
        { if (遷移直後) 遷移履歴を遷移系列として選定;
          遷移履歴を一つ後戻り;
          スタックをポップ;
          時間の復帰;
          アクションの取消;
          一時的枝刈り;}}
    
```

fig.3 遷移系列選定アルゴリズム

4. テスト(遷移系列)の評価方法

制御用のシステムは外部環境に対応した特定の振舞いの系列を繰り返すため、実動作と同じ状況を十分テストしたかを測る尺度が必要となる。そこで、以下に2つの網羅率を定義する。(fig. 4)

(1) パス網羅率(Cp)

設計者の指定した遷移系列(パス、システムの実動作に対応させる)をどの程度忠実に通っているかの尺度。

(2) 重みつき網羅率(Cw)

一状態からの遷移が複数存在するときに、特定の遷移

(システムの実動作に対応させる)に重点をおいてテストをしているかの尺度。

(1) パス網羅率

設計者の指定した遷移列をPiとする。選定された遷移系列(Xj)の含むPiの最大部分列をSPIjとすると、パス網羅率は

$$C_p = \frac{\sum_i \max_j(\text{length}(S_{Pij}))}{\sum_i \text{length}(P_i)} \times 100\%$$

(2) 重みつき網羅率

状態(Si)からの遷移(Tij)に対する重みをWijとする。遷移系列(Xk)によってTijを実際に通過した回数をNijkとすると、重みつき網羅率は

$$C_w = \frac{\sum_i \min_k(\sum_j N_{ijk} / (\frac{W_{ij}}{\min_j(W_{ij})}), 1)}{\sum_i \sum_j 1} \times 100\%$$

fig.4 網羅率の定義

例として、tab. 1 に冷蔵庫の冷凍庫部分の状態遷移仕様に対し、本アルゴリズムを適用した結果の網羅率を示す(比較対象は深さ優先探索による系列)。特にパスの網羅率に対して、本アルゴリズムが有効であることが分かる。

tab.1 網羅率の比較

	深さ優先探索	本アルゴリズム
Cp	50%	100%
Cw	48.6%	49.8%

5. おわりに

状態遷移仕様からのテストデータ生成方式、特に遷移系列選定アルゴリズムに重点をおいて述べた。テストデータ生成によるテスト自動化の効果としては、

- (1) テスト作業の省力化による生産性の向上
 - (2) テスト支援によるソフトウェア信頼性の向上
 - (3) 勘にたよった狙い打ちテストを不要とする
- などが期待される。

実際に本アルゴリズムによるテストデータ生成ツールは実現済みであり、シミュレータによる仕様検証に用いられている。今後、遷移系列から時系列データを生成する方法を確立し、システムテストの自動化を目指していきたい。

なお、本稿の内容は、IMAP開発の一貫として行っているものである(IMAP: Integrated software Management and Production support system)。

[参考文献]

Tsun S. Chaw, "Testing Software Design Modeled by Finite-State Machines", IEEE Trans. Software Eng. Vol. SE-4, No. 3, May 1978.

F. Teshima, et al., "System Design Verification based on Formal Specifications", Proc. of 2nd European Conf. on Software Quality Assurance, 1990.