

同期モデルを用いた並行状態グラフの Nスイッチテストについて

高島 和紀[†] 高木 智彦^{††} 古川 善吾^{††}

香川大学工学部信頼性情報システム工学科[†] 香川大学工学部電子・情報工学科^{††}

1. はじめに

並行処理ソフトウェアのテストにおける、複数のオブジェクトが動作するシステム全体の状態を取り扱うテスト技法の一つに並行状態グラフがある。Taylor らが提案した、複数のオブジェクトの状態の組み合わせである並行状態を節点とし、並行状態を遷移させる事象を枝とするグラフである[1]。並行状態グラフは並行処理ソフトウェアを1つの状態遷移図で表現しているため、状態遷移テストのモデルとして取り扱うことができる。並行状態グラフを探索して抽出した事象の系列は、並行処理ソフトウェアのテストケースとなる。

Taylor らが提案する並行状態グラフはインターリーブモデルに基づいて作成されており、オブジェクトの状態数やオブジェクト数が多くなると、生成されるテストケース数が処理可能な数に収まらなくなる。そこで、テストケース数を圧縮するために同時に複数個の遷移を許す同期実行モデルを提案した[2]。結果として、状態網羅テストにおける状態数の圧縮が確認できた。

そこで本稿では、遷移網羅テストにおいても同期実行モデルが有効かどうか検討する。遷移網羅テスト手法の一つにNスイッチテストがある。Nの数を大きくすれば網羅度の高いテストができるが、その分コストがかかることが問題点としてあげられる。このNスイッチテストに、同期実行モデルを適用させ、テスト戦術とその結果を示す。

2. 同期実行モデル

インターリーブモデルにおいて順序付けられた複数処理を同時に実行できる場合、

その始まりに相当する分岐点からその終わりに相当する収束点まで一つの遷移にまとめることで、オブジェクトの状態数を圧縮する。その際、まとめられた複数のメッセージを同時に送ることを可能にする。

また、並行状態グラフを作成する際、一つの遷移で複数のオブジェクトが同時に変化することを許可する。

3. Nスイッチ

Nスイッチは、N+1 推移のすべての有効なシーケンスを実行するテストケースを設計する[3]のものであり、行列式で表現できる。

行側に遷移元の状態、列側に遷移先の状態を対応させ、行列がクロスするセルにその遷移を発生させる事象を記述する。Nスイッチは、行列式の積で求めることができる。

同期実行モデルでも、インターリーブモデルのように、Nスイッチを使用できる。

本稿では、テスト基準については、特に言及しないので、行列式で生成された事象を100%満たすようなテストケースを生成し、同期実行モデルとインターリーブモデルを4節に示す評価関数を使用して比較する。

4. 評価関数

Nスイッチにおける、同期実行モデルとインターリーブモデルの評価関数を式(1)に設定する。この評価関数値は、1を基準とした相対的な比較評価値として扱う。1を下回れば、偏った領域のテストをしているか、もしくは、通過できる事象が少ないということがあげられる。どちらも、より良いテストケースが生成されたとはいえない。この評価関数の中の繰返係数(式(2))は、同じ遷移を複数回使うことによって大きくなる。あらゆる組み合わせを考慮していると言える。

N-switch test using a synchronous execution model

[†] Kazuki Takashima

Dept. of Reliability-based Information Systems Engineering,
Faculty of Engineering, Kagawa University

$$\text{評価関数} = \text{繰返係数} \times \text{補正関数} \quad (1)$$

$$\text{繰返係数} = \frac{\text{のべ通過事象数}}{\text{事象数}} \quad (2)$$

$$\text{補正関数} = \left(\frac{\text{通過事象数}}{\text{事象数}} \right) \quad (3)$$

式(3)は、NスイッチのNが大きくなればなるほど、通過できない遷移が増えてくるために全体の遷移の中でどれだけの領域のテストケースができたかを示すための補正関数として設定する。

5. 検証結果

並行処理ソフトウェアのテストにおける有名例題であるリーダー・ライター問題に適用させた。リーダー2人、ライター2人で検証を行った。

モデル上で異常な状態とそれ以外に分け、それ以外の部分にのみ適用させる。異常な部分については、別のテストを試みることを推奨する。今回はその部分については言及しない。

戦術1とは、並行状態グラフ上の1つの節点内で変化するオブジェクトの数が1以上の同期実行モデルであり、戦術2とは、変化数を複数に限定したものである。

以下の表に、評価関数の結果を、少数以下有効数字2桁で表示する。

表1 0スイッチ

0スイッチ	インターリーブモデル	同期実行モデル(戦術1)	同期実行モデル(戦術2)
評価関数	1.00	3.27	2.27
事象数	164	164	164
通過事象数	164	536	372
不通過事象数	0	0	0

表2 1スイッチ

1スイッチ	インターリーブモデル	同期実行モデル(戦術1)	同期実行モデル(戦術2)
評価関数	6.27	61.71	28.27
事象数	164	164	164
通過事象数	1028	10120	4636
不通過事象数	0	0	0

6. おわりに

検証の結果、同期実行モデルは、インターリーブモデルに比べ、評価関数の値が大きくなることが分かった。しかし、単純に通過事象数だけを比較すると、同期実行モデルは遷

移先候補数が多くなるため、テストケース数も多くなるのが改めて確認できた。

従来、状態数や遷移数が多く、Nスイッチテストが十分に行われていなかった。0~Nスイッチを全て行おうとすると、式(5)から得られる数のテストをしなければならなかった。同期実行モデルを適用すると、式(4)の数で済む。

5節の検証結果を比較すると、同期実行モデル(戦術1)の通過事象数は536であり、インターリーブモデルの1スイッチの1028を下回る。つまり、同期実行モデルを用いることで、インターリーブモデルの0~Nスイッチの総事象数を下回る結果が得られた。

同期実行モデルを適用したことにより、0~Nスイッチをより少ないテストケース数でより広範囲のテストをすることができ、1/2程度に圧縮されたことによって、従来諦めていた部分のテストができる可能性が高くなったといえる。これに伴い、従来よりも信頼感が増すことが期待できる。

同期実行モデルの最大通過事象数

$$= \sum_{i=1}^{object} object C_i \times \text{状態数} \quad (4)$$

インターリーブモデルの0~Nスイッチの総事象数

$$= \sum_{i=0}^N (i+1) \text{乗の行列式} \quad (5)$$

参考文献

[1] R.N. Taylor, D.L. Levine, and C.D. Kelly, "Structural testing of concurrent programs," IEEE Trans. Softw. Eng. vol18 no.3, pp.206-215, 1992.

[2] 高島和紀, 古川善吾, 高木智彦, "並行処理ソフトウェアテストのための同期実行モデルの提案" 平成24年度電気関係学会四国支部連合大会, p.365, 2012

[3] Eric van Veenendaal, ソフトウェアテスト標準用語集(日本語版) <http://www.jstqb.jp/dl/JSTQB-glossary.V2.0.pdf>, 2012/11/29時点