反例を利用した形式モデル修正支援ツールの開発

佐藤 直人　來間 啓伸
株式会社日立製作所 システム開発研究所

1. 緒言

近年，システムの仕様の誤りを早期発見するための方法としてモデル検査法が注目を集めており，産業界での実用化も現実味を帯びてきた。最近では，SPIN[1]のようなモデル検査の自動化ツールに止まらず，モデルの記述を支援するツールが開発されている[2][3]。

モデル検査によって仕様の誤りを発見した場合，先にモデルの修正を行い，修正したモデルをもとに仕様を改訂する手順が効率的である。しかし，このモデル修正作業は人手で行うため，担当者には対象システム及びモデル記述言語の十分な理解が求められる。

そこで本稿では，モデル記述言語 Promela に対応した，モデル修正支援ツールを提案する。本ツールは，SPIN が出力する反例を利用して Promela モデルの修正方法を出力する。

また，本ツールを実装し，適用実験を行った結果についても述べる。

2. モデル修正支援ツールのアイディア

SPIN を始めとする多くのモデル検査ツールは，モデルの誤りを発見した際に反例を出力する。反例とは誤りが発生するまでのモデルの実行系列で，「モデルに反例の示す誤りがある」ことと「反例(として得た実行系列)はモデルの実行系列である」ことは同値である。さらにそれぞれの否定をとると，「モデルに反例の示す誤りはない」ことと「反例(として得た実行系列)はモデルの実行系列ではない」ことが同値になる。(図 2-1参照)

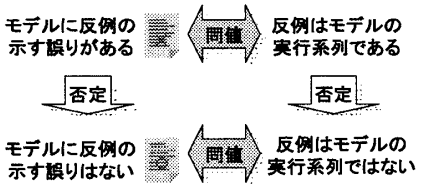

図 2-1：モデルと反例の関係

よって，反例が示す誤りをモデルから除去するためには，反例がモデルの実行系列にならないように，モデルを修正すればよい。

Development of a Model Modification Support Tool based on the Counter-Example Analysis
Naoto SATO, Hironobu KURUMA
Systems Development Laboratory, Hitachi, Ltd.

ただし，前述の通り，反例は“モデルの”実行系列であるため，モデルを変更した場合，その変更を反映した新しい反例が得られる。このモデルの変更を反映した反例は，変更後のモデルにおいて，もとの反例と同等の誤りを示す。つまり，モデルの変更を反映した反例が，変更後のモデルの実行系列にならなければ，モデルを修正できたことになる。(図 2-2参照)

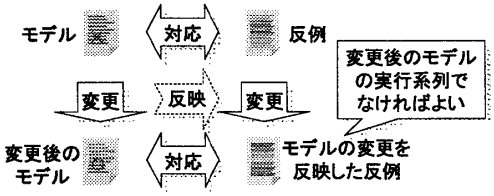

図 2-2：モデルの修正方針

本ツールでは，上記条件を満たすモデルの修正方法を，次の手順で抽出する（図 2-3参照）。

(1) モデル検査ツールが出力した反例を編集し，モデルの実行系列でない，実行不可能な反例を構成する。
(2) 上記反例の編集（変更）方法から，モデル修正方法を逆算する。

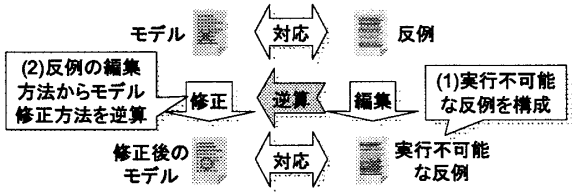

図 2-3：モデル修正方法の抽出手順

このように，実行不可能な反例を構成し，それをもとにモデルの修正方法を抽出する点が本ツールの特徴である。これにより，本ツールの出力するモデル修正方法であれば，反例の示す誤りを除去できるという利点がある。

3. 本ツールが出力するモデル修正方法

次に，本ツールが出力するモデル修正方法について述べる。

Promela によるモデル記述では，システムの動作主体をプロセスで記述する。システム全体の動作は，並列動作するプロセスの交互動作によって表される。よって，検証対象となるシステムを正確にモデル化するためには，交互動作の単位を適切に設定しなければならない。Promela の場合，基本的には１つの文が交互動作の単位

となるが，atomic 節等を使用することで，複数の文の纏まりを交互動作の単位にすることができる(図 3-1参照)。

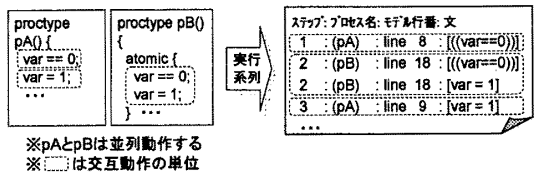

図 3-1：atomic 節の使用例

このように，atomic 節はモデルの実行系列を決定する重要な要素であるが，通常の文のように，モデルの動作内容を定義する記述ではない。よって，通常の文に比べると，その記述漏れの可能性は高い。また，記述漏れがあってもコンパイルエラー等が発生することはないため，記述漏れを検知することも困難である。以上の理由から，本ツールでは，atomic 節の挿入によるモデル修正方法を採用することにした。

4. モデル修正方法の抽出

では，実際にモデルの修正方法を抽出する具体的な手順を説明する。

(1) 実行不可能な反例の構成（図 4-1）

反例において，隣接はしていないが，特定プロセスの文に着目した場合に隣接する文のペアを選択する。次に，選択したペアの後の文を前の文の直後に移動する。さらに，得られた反例を実行してその実行可能性を判定する。

(2) 反例に対応するモデル修正方法の抽出

(1)で構成した反例が実行不可能なら，選択した文のペアを atomic 節で纏めることがモデル修正方法となる。□

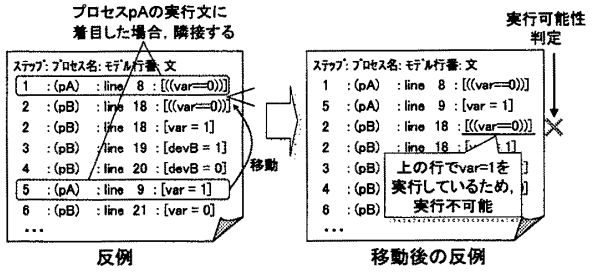

図 4-1：実行不可能な反例の構成方法

上記の手順で抽出した修正方法によりモデルを修正した場合，修正後のモデルに対応する反例は(1)で構成した反例になる。この反例は実行不可能であるため，抽出したモデル修正方法は，反例が示す誤りを除去する修正方法と言える。

5. 本ツールの実装と適用実験

本ツールを Java で実装した。実装したツールは，修正方法を出力するだけでなく，修正方法の選択入力を受け付けて修正後のモデルを表示する。本ツールを使って行った適用実験の内容を以下に簡単に記す。

(1)約 250 ステップの Promela モデルを SPIN で検証し，SPIN が出力した反例に本ツールを適用した。その結果，本ツールは 1 件の修正方法を出力した。

(2)本ツールが出力した修正方法を選択し，モデルを修正した。

(3)得られたモデルを再度 SPIN で検証し，新たな誤りがないことを確認した。

通常のモデル修正作業では，まず反例を解読し，その結果をもとに任意にモデルを修正する。その後モデルを再検証し，誤りを除去できたかを確認する。今回，本ツールを適用したことにより，ツールが出力する修正方法を選択するだけでモデルを修正できた。

ただし，モデルを修正することにより，除去した誤りとは異なる，新たな誤りを追加してしまう可能性がある。よって，本ツールを適用した場合でも，修正したモデルの再検証は必要になる。

6. 結言

本稿では，反例を利用したモデル修正支援ツールを提案した。また，本ツールを実装し，適用実験を行った結果について述べた。最後に，本ツールの評価と今後の課題について述べ，まとめに代える。

■本ツールの評価

- 反例の示す誤りを除去する修正方法を出力するため，モデル修正作業の効率化が可能。
- ただし現状では，2 つの文を纏める atomic 節の挿入による修正方法のみ出力可能。この方法で修正できない誤りには未対応。

■今後の課題

- 出力可能な修正方法の追加
  - ➢ 3 つ以上の文を纏める atomic 節の挿入
  - ➢ 同期チャネルの挿入
- SPIN とのインターフェイス連携

参考文献

[1] Gerald J. Holzmann : The SPIN Model Checker, Addison-Wesley, 2003 年 9 月.

[2] 文部科学省ソフトウェア推進プロジェクト「e-Society プロジェクト」

[3] モデル検査によるソフトウェアテストの実践研究会, http://www.modelcheck.jp/