

SMT ソルバを用いた Simulink モデルのテストケース生成

八田 竜起[†]
福井大学[†]石井 大輔[‡]
北陸先端科学技術大学院大学[‡]

1 はじめに

組み込みシステムの開発において、Simulink モデルのカバレッジテストが重要である。テストにはカバレッジ要件を満たす入力信号 (テストケース) が必要である。しかしモデルが大規模や非線形になると、要件を満たす入力信号を手動生成することは難しいという問題がある。そこで本研究では、Simulink のテストケースを自動生成することを目的とする。手法として、SMT ソルバによる静的解析を提案する。提案手法を実装し、中規模モデルで高カバレッジが達成できることを確認する。

2 Simulink モデルのテスト

MATLAB/Simulink^{*1}はモデリング、シミュレーション、解析のためのツールである。様々な演算機能を持つ箱 (ブロック) 同士を結線してモデルを設計する。図1に Simulink モデルの例を示す。

Simulink モデルのテストの流れを説明する。

1. テストケースとなる入力信号を用意する。
2. 用意した入力信号でシミュレーションする。
3. 各ブロックの出力信号をステップごとに得る。
4. 得られた出力信号からカバレッジを測定する。
5. 1~4を繰り返しシミュレーション結果を累積しフルカバレッジ (カバレッジ 100%) を目指す。

モデル内の論理的振る舞いをシミュレーションによりどれだけ網羅したかを示す割合をカバレッジという。高カバレッジなテストの実施は、Simulink モデルの品質保証につながる。

Simulink モデルのテストではモデルが大規模、非

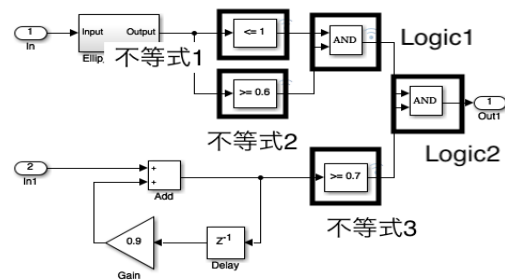


図1 Simulink モデルの例

```
(declare-const _vr_170_1_10 Real)
(declare-const __v1_95_1_10 Real)
(declare-const __v1_96_1_10 Real)
; [Logic] Logical Operator
(assert (= __v1_96_1_10
  (ite (and (> _vr_95_1_10 0)
    (> _vr_170_1_10 0)) 1 0) ))
(check-sat)
```

図2 モデルの制約充足式

線形になると、手動でテストケースを用意することが難しくなるという問題がある。自動生成ツールが存在するがモデルが大規模、非線形になると生成がうまくいかないことが報告されている [1].

3 SMT ソルバによるテスト手法

SMT ソルバ (Satisfiability Modulo Theories) とは制約充足問題を解くツールであり、複数の制約を同時に満たす変数値が解として得られる。SMT ソルバで制約充足問題を解くには SMT-LIB 形式^{*2}で制約を記述する。SMT ソルバでの充足可能性判定の結果が sat であれば充足解が存在することがわかり、その充足解を得る。結果が unsat であれば充足解が存在しないことがわかる。

Simulink を SMT ソルバで静的解析するために、モデルを等価な SMT-LIB 形式の制約充足式にエンコードする [2]. モデルをエンコードした制約充足式には、各ブロックの入出力の関係が記述される。図

Test case generation of Simulink models using SMT solvers

[†] Ryuki Hatta, University of Fukui

[‡] Daisuke Ishii, Japan Advanced Institute of Science and Technology

^{*1} <https://jp.mathworks.com/products/simulink.html>

^{*2} <http://smtlib.cs.uiowa.edu/index.shtml>

2に図1のモデルを制約充足式に変換した例の一部を示す。この例では、ステップ10におけるLogic1ブロックの入出力間の制約を記述している。

以下に本研究で提案するSMTソルバを用いたテストケース生成手法を示す。

入力: モデル, ステップ数, 信号値の上限下限, デッドロジック数

出力: カバレッジ, デッドロジック

1. モデル内の全ブロックの情報を格納する構造体配列 *bt* を作成する。
2. モデルを SMT-LIB 形式にエンコードする。
3. (1回目) 構造体型配列 *cov* を初期化する。
(2回目以降) カバレッジを測定し結果を構造体配列 *cov* に格納する。
4. *bt* のカバレッジ情報を更新をする。
5. *bt*, *cov* を元に満たしていないカバレッジ条件のリストを作成する。満たしていないカバレッジ条件がなければ終了し、カバレッジを表示する。
6. 手順5のリストからランダムに1つ選び、制約としてエンコード結果に追加する。
7. SMTソルバで充足可能性を判定する。
 - (a) 結果が *sat*:充足解として入力信号(テストケース)を得る
得られたテストケースで Simulink でシミュレーションする。
 - (b) 結果が *unsat*:デッドロジックとし、以後、手順5のリスト作成時に削除される。
8. ステップ3へ

デッドロジックとはシミュレーション中に絶対に満たすことが出来ないカバレッジ条件である。*bt*にはブロックごとに、ブロックの名前や種類、カバレッジなどが記録される。手順6で、カバレッジ条件の制約をモデル全体のエンコード結果に追加することで、モデルの制約を満たす振る舞いが存在し、かつその振る舞いがカバレッジ条件を満たすかを判定する制約充足問題を作る。

4 実験結果

本手法の有効性の確認として、図1のモデルにおいて不等式2の判定結果が真になるテストケースを生成できるまでの求解時間を既存手法であるモンテカルロ法を用いたテストケース生成手法[1]と比較

手法 \ 条件	≥ 0.6	≥ 0.9
本手法	120s	760s
モンテカルロ法	90s	生成不可 (800s)
本手法 + パス制約	100s	570s

表1 実験結果

する。不等式2の条件は、 ≥ 0.6 と ≥ 0.9 の2つ用意する。表1に実験結果として各条件のもとでの求解時間を示す。不等式2の条件が ≥ 0.6 のときは、本手法よりもモンテカルロ法を用いた手法の方が求解時間が短い。しかし、不等式2の条件が ≥ 0.9 で厳しくなるとモンテカルロ法を用いた手法では、テストケースを生成することができなかった。一方、本手法では約760sでテストケースを生成することができたため、解の存在範囲が厳しい問題において、本手法が有効であるといえる。

静的解析においてはパス制約[3]が有用である。パス制約を用いた場合の検証実験も行った。前述の実験において、不等式3に対してパス制約を用いて論理的振る舞いを固定する。表1に結果を示す。どちらの条件の時もパス制約を用いることで求解時間を短縮できることが確認できた。

5 まとめ

本稿では大規模、非線形な Simulink モデルへの有効的なテストケース生成手法として、SMTソルバを用いたテストケース生成手法を提案した。またその有効性を中規模モデルを用いて既存手法と比較し、解の存在範囲が厳しい問題において、既存手法よりも有効であることを確認した。また、パス制約を用いることで求解効率が向上することを確認した。

今後の課題として、大規模モデルにおいても効率良く扱えるようにすることと、パス制約をより有効的に活用する方法の検討が挙げられる。

参考文献

- [1] T.Tomita, 他. Template-Based Monte-Carlo Test Generation for Simulink Models, CyPhy ワークショップ, pp. 63-78, 2017.
- [2] 武仲紘輝, 石井大輔. Simulink モデルの SMT-LIB エンコード方法に関する実験. 電子情報通信学会 MSS 研究会, 2019.
- [3] P.Godefroid, 他. DART: Directed Automated Random Testing. PLDI, pp. 213-223, 2005.