5A-3

モデル検査自動化ツールの開発
～入力支援機能と状態遷移表縮約機能～

森 奈実子†, 高田 沙都子†, 長谷川 保†, 村田 由香里†, 進 博正‡
†株式会社 東芝 ソフトウェア技術センター ‡株式会社 東芝 研究開発センター

1. はじめに

組込みシステムの発展とともに，組込みソフトウェアへの信頼性や安全性の要求が高まる一方で，ソフトウェアの大規模化，複雑化が進み，人手による網羅的な品質検証は困難な状況になっている．このような問題を解決する手段として上流工程で網羅的に自動検査を行えるモデル検査に注目が集まっている．

モデル検査では，モデルとして記述された仕様に対して，システムの取りうる状態を網羅的に探索することで，与えられた制約条件に対して違反がないかを漏れなく検査できる．ただし，モデル検査を実施するためには，特殊なモデル記述言語と時相論理に習熟する必要がある．そこで，筆者らはモデル検査自動化ツール[1]の開発を進めている．本稿では，新たに拡張した入力支援機能，状態遷移表縮約機能および，その効果と妥当性の評価結果について述べる．

2. モデル検査自動化ツール

モデル検査自動化ツールは，状態遷移表と制約条件を入力として，検査用モデル，時相論理式の自動生成を行い，モデル検査器 SPIN[2]を利用してモデル検査を行う．しかしながら，図 1 に示す通り，実際の開発現場では状態遷移表ではなく状態遷移図やフローチャートなど別の記述方法を用いることが多く，直接ツールを利用できないことが多い．また，大規模な状態遷移表の検査では，状態が爆発的に増加し，メモリ不足による検査途中終了や検査がいつまでも完了しないという状況が発生した．そのため，これまでは動作仕様から検査用モデルを手作業で書き起こしたり，人手で状態遷移表の縮約を行うなどして対処してきた．

筆者らはこれらの人手による作業コストの削減と，ツールの適用範囲拡大を目的として，新たに入力支援機能と状態遷移表縮約機能を開発している．本章では各機能について述べる．

Development of Automated Model Checking Tool : Input Support & State Transition Table Contraction
Namiko Mori†, Satoko Takada†, Tamotsu Hasegawa†, Yukari Murata†, Msahiro Shin‡
†Toshiba Corporation, Software Engineering Center
‡Toshiba Corporation, Research & Development Center

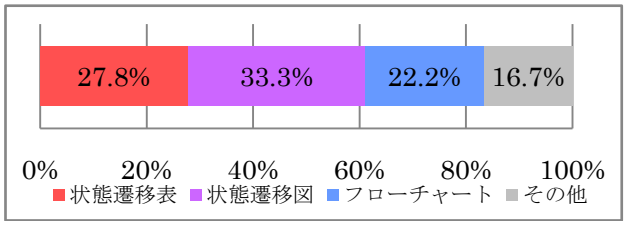

図 1：検査適用事例で使用された記述方法

2.1. 入力支援機能

入力支援機能は，状態遷移図やフローチャートから状態遷移表に自動的に変換する機能である．状態遷移図やフローチャートは，XMI(XML Metadata Interchange)ファイル形式でデータを出力できる UML モデリングツール ArgoUML[3]で記述する．XMI ファイルから状態遷移表を構成するための情報を抽出して，本ツール上に状態遷移表を展開していく．

フローチャートから状態遷移表への変換を図 2 に示す．フローチャート上での「条件分岐」をイベントとして，「処理」をアクションとして定義する．また，イベントに対してのアクションの実行パスを状態と定義する．これにより，フローチャート上の条件分岐のパス毎の値の変化を状態として捉えた状態遷移表を生成できる．

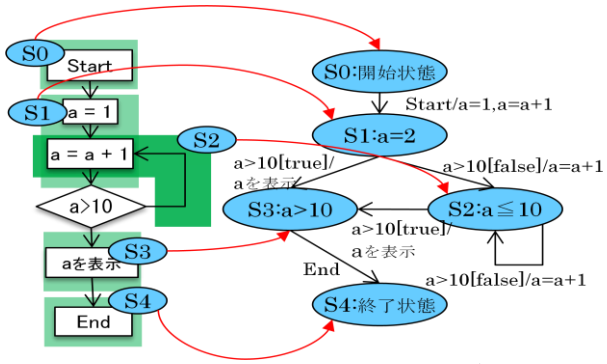

図 2：フローチャートからの変換方法

2.2. 状態遷移表縮約機能

状態遷移表縮約機能は，制約条件に対する検査に直接関係のないイベントや状態を自動的に縮約する機能である．状態の縮約では，同値とみなす状態群を指定することで複数の状態を同一の状態に統合する(図 3)．また，イベントの縮約では遷移元，遷移先が一致する状態遷移がある場合，一致する箇所のみを同一イベントとして統

合する(図 4)．縮約した状態の遷移先は縮約前と同じであるため，縮約前の状態遷移表が持つ情報は縮約によって損なわれることはない．そのため，制約条件に直接関係のない箇所を縮約しても，縮約前の状態遷移表で検出される違反事例を検出することができると考えられる．ただし，縮約したことで本来発生し得ない違反事例も生じるため、解析時には縮約した状態やイベントによって違反原因が引き起こされるものではないかを吟味する必要がある．

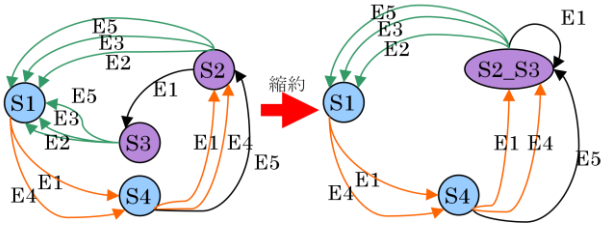

図 3：状態の縮約

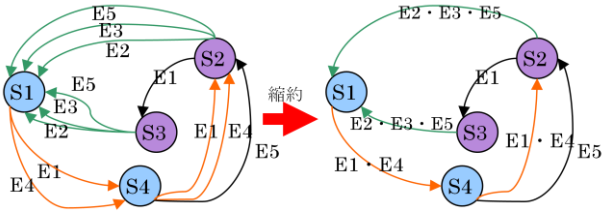

図 4：イベントの縮約

3. ケーススタディ

3.1. ケーススタディ概要

ケーススタディの対象は，二重系の監視制御システムである．対象システムでは二つのシステムが並行に動作しており，一方がエラーでダウンしてしまった場合，他方がダウンを検知して起動するよう冗長化されている．そのため，両方のシステムが同時に起動してはならないという制約条件がある．本ケーススタディでは，対象システムが上記制約条件を違反しないかどうかを検査し，縮約前と縮約後の状態遷移表に対する検査結果を比較することで状態遷移表縮約機能の効果と妥当性を評価した．

3.2. 結果と考察

対象システムは同じ動作仕様の二つのシステムが並行に動作しているため，同じ状態遷移表が二つある．一つの状態遷移表における縮約結果を表 1 に示す．イベント数が 31%減少，状態遷移数が 76%減少という結果となった．イベント縮約により統合された同一イベントが新たに追加されるため，イベント数の削減幅は小さくなったが，複数の遷移を一つの遷移に統合することで多くの状態の遷移を縮約でき，遷移数を大幅に削減できる結果となった．

表 1:状態遷移表上での縮約結果

	イベント数(個)	状態数(個)	遷移数(個)
未縮約	39	7	155
縮約済	27	6	38

表 2:検査実施時の探索範囲とメモリ消費量

	状態数(個)	探索の深さ	メモリ消費量[*1] (Mbyte)
未縮約	8920862	342783	97.392 (127.90)
縮約済	185827	16120	4.05 (34.568)

検査実施時の探索範囲とメモリ消費量を表 2 に示す．検査の結果，状態数は 98%減少，深さは 96%減少，メモリ消費量は 96%減少という結果になった．SPIN では実行系列の順序と変数値との組み合わせによって状態が決まる．そのため，状態遷移表の遷移数が減少したことで，検査用モデルの実行系列が減少し，並行時の実行系列の順序と変数値との組み合わせである状態数が激減してメモリ消費量の削減に繋がったと考えられる．また，縮約した状態遷移表に対して検証を行った結果，未縮約の状態遷移表で検出した違反事例と同様の原因が検出できた．縮約を行った場合でも，制約条件に対する検査が正しく行えると確認できた．

このようにメモリ消費量が大幅に削減でき，検証の妥当性も確認できたことで，以前は適用できなかった大規模な状態遷移表への適用も期待できるようになった．

[*1] メモリ消費量は，総メモリ消費量と深さ探索で消費する一定量の差とする．括弧内が総メモリ消費量を指す．

4. おわりに

本稿では，モデル検査自動化ツールの適用範囲拡大とコスト削減を目的として，モデル検査自動化ツールに入力支援機能と状態遷移表縮約機能の開発を行った．また，二重系の監視制御システムを対象にケーススタディを行い，その効果と妥当性を確認した．今後の課題として，入力支援機能を利用できる UML モデリングツール群の拡大や，検査に必要な変数にのみ着目した状態の縮約の検討が挙げられる．

参考文献

[1] 高田沙都子,森奈実子,村田由香里:モデル検査自動化ツールの開発～検査自動化と反例解析効率化～,情報処理学会 第 74 回全国大会(2012).
[2] SPIN Model Checker, http://SPINroot.com/SPIN/whatiSPIN.html.
[3] Argo UML, http://argouml.tigris.org/.