トップエスイー修了制作

Promelaモデルの自動抽象化の枠組みの提案

東京大学大学院情報理工学系研究科　姜 帆　kyoho@nii.ac.jp

開発における問題点

ソフトウェアの設計で潜在的なバグを早期に発見するために、モデル検査の導入が進んでいる。
Promela/SPINを利用したモデル検査が最も有名であるが、状態爆発の問題が発生する。これをモデル開発者が抽象化によって除去するのは経験と知識を要する。

手法・ツールの提案による解決

モデルの記述と抽象化による状態爆発の抑制の間に、技術レベル上大きなギャップが存在している。そのため、後者のための自動抽象化ツールを開発することは有意義である。
本提案では自動抽象化のための抽象化手法の有効性と、その自動変換可能性についての検討を行った。

状態爆発と抽象化

状態爆発

状態爆発には2種類あり、それぞれ、状態数が非常に多いこと、状態数が理論上無限となる場合である。
前者に対して、モデル検査(状態空間の探索)を行う環境の使用可能リソースを強化することは有効である。
一方、理論上無限となる状態空間に対して、検査環境の強化を行っても、モデル検査は停止しない。そのため、抽象化によって、状態空間を有限の大きさにすることが必要となる。

抽象化

モデル検査における抽象化とは、複数の状態を一つの抽象状態にまとめる操作である。正しいモデル検査結果を保証するための抽象化-精錬のループ構造の手順として、CEGARが提案されているが、CEGARでは抽象化は健全な抽象化であることを必要とする。
健全な抽象化とは、具体状態で成立する性質が抽象状態においても保存される抽象化である。そのため、抽象化によって、検査性質が失われることはないが、一般に偽反例が発生することがある。

2つの抽象化手法

(1)Generic-Specific抽象化
検査対象のモデルが複数のシステムからなる、協調システムであるとき、システムの数が非常に大きい場合に有効である。
無数に存在するシステムのうち、ひとつをマーカーとして任意に選び、残りを環境モデルとして抽象化する。環境のモデルは大きな数のシステムの状態の組合せよりも小さな状態数で表現可能となるため、モデル検査器が探索する状態空間は有限で小さなものになる。
(2) 変数の大小関係に基づく抽象化
モデル検査では、記述された変数の取る値の組み合せによって状態が表現される。そのため、変数の取る値によって、状態数が爆発することがある。

しかし一方で、モデル検査の場合、変数の利用方法が大小比較や、一致判定といったものが主である。そのため、抽象モデル上では、変数同士の大小関係を保存する最小の、抽象的な値とすることが可能である。

実験

本提案の有効性を確認するため、無数にノードが存在する場合の集中排他制御、そしてタイムスタンプを各持つ場合の分散排他制御のシステムの抽象モデルと具体モデルを構築し、状態数について確認した結果、それぞれ状態数が1/800程度となった。

また、それぞれの抽象モデルを、具体モデルを元に、自動で生成するための手順を抽出することができた。ただし、抽象モデルから偽反例の除去が課題として残る。