モデル検査による大規模な計算機環境での検証手法

戸川 博貴

0810042

Togawa Hirotaka

情報科学研究科

2010 年 2 月 9 日

キーワード　モデル検査　形式手法　クラスター分析　Promela　環境モデル

背景

近年、社会のあらゆるところにソフトウェアは使われているため、ソフトウェアの信頼性を保証することが重要になっている。ソフトウェアの信頼性を保証する方法としてモデル検査がある。これまで,OSEK/VDX を対象にモデル検査を用いた検証手法を研究し,複数の検査モデルを生成できるようになっている。しかし、検査モデルが多くなると1 台の計算機ではメモリの容量が不足する問題があり検証することは難しい。さらに、検証結果が膨大になることで問題点を把握することが困難になる。本研究は大規模な計算機環境でSPIN を用いて検証を行い、その結果をどのように解析・表示するか検討する。本研究の特色は、実際に大規模な計算機環境で検証を行うことである。SPIN は誤った結果に至るまでの反例を表示する特徴があるので、大規模な計算機環境に利用すれば複数の検査モデルの検証が可能になるだけではなく、個々の検査モデルを比較・検討することも可能となる。

問題点

大規模の計算機環境を用いてモデル検査を行うためにいくつかの問題点がある。１．検査モデルを計算機環境に最適な方法を使って分配しなければ検査時間にばらつきが生じる問題がある。2 つ目は、数万個の検査モデルを使ってモデル検査を行うので、検査結果が膨大になり問題を把握することが困難になる。そこで、直感で理解できる表を提案する。3つ目は、検出される反例は何種類のエラーを含んでいるかわからない。そこで、エラーのバリエーションを区別する方法を考える。これらの問題を踏まえて小規模な計算機環境から実験を行い、問題を整理し、SPIN を用いて大規模な計算機環境でも適用できるような検証方法を考案する。

提案手法

１．検査モデルを最適に分配する方法

検査モデルを最適に分配するため、検査モデル1つあたりの検査時間を見積もって計算機に分配する方法で行う。

実験は２つの場合で行う。

1つ目は、検査時間を求めるため、各環境モデルの検査モデルから最大200個を抽出し、モデル検査を行う。検査モデル1つあたりの検査時間を求める。次に、各環境で求めた検査時間を用いて全ての検査モデルを使ってモデル検査を行ったときの総合時間を求める。求めた総合時間を使用する計算機の台数で除算し、計算機1台当りの検査時間を求める。検査モデルを計算機に分配する。

2つ目は、検査モデルスクリプトの行数とコンパイル時間の関係から検査モデルの検査時間を求める方法である。最初にモデル検査を行い、行数とコンパイル時間の依存関係を調べ、近似式を求める。検査モデルのスクリプトの行数を階級ごとに区切り、階級値を近似式に代入する。求めた値は各階級の検査時間とする。次に、各行数の階級値で求めた検査時間を用いて全ての検査モデルを使ってモデル検査を行ったときの総合時間を求める。求めた総合時間を使用する計算機の台数で除算し、計算機1台当りの検査時間を求める。最後に、検査モデルを計算機に分配する。

２．問題を把握する方法

問題点を把握する方法、直感で理解できる表を提案する。まず、抽象度が高い表を表示する。例えば、タスクの数または資源の数に焦点を当てた表である。詳細な内容を調べたい場合は、タスクの数を指定してタスクの数における検査結果を表示する。このとき、タスクの数を指定しているので資源の数に対応した表を作成こともできる。より詳細に検査結果を調べたい場合は、タスクや資源の優先度、タスクと資源の参照関係、多重度に注目して表示を行う。

３．エラーのバリエーションを区別する方法

エラーのバリエーションを区別する方法は、クラスター分析を適用する。クラスター分析は、個体がもつデータを用いることで個体を区別することができる方法である。クラスター分析を適用する前処理を行う。最初に、grepコマンドを使い反例結果の中から必要な情報を抽出する。次に、diffコマンドを使い反例間の差分を求める。そして、反例間の差分の数を距離として変換する。求めた距離をクラスター分析に適用する。

実験結果と考察

１．検査モデルを最適に分配する方法

1つ目の実験結果は、検査時間が最長で約6 時間であることから1 台の計算機で検査を行うよりも約18倍の速さで検査を行うことができた。しかし、環境がタスクの数が2、資源の数が4 の検査モデルを割り振った計算機の結果を比較すると約3 時間半の時間差が生じた。原因は、規模の大きい環境の検査モデルをランダムに抽出し、検査モデル1 つの当りの時間を求めたため検査時間の誤差が生じたと考えられる。この結果から全ての検査モデルに

ついてモデル検査を行い、検査時間の傾向を調べる必要がある。2つ目の実験結果は、計算機66台で2 時間～2 時間半の範囲で検査を行うことができた。しかし、残り2 台の計算機では検査に6 時間と9 時間かかった。原因は、行数が700～800 行間で近似式では200 秒ほどで終了すると予測されたが実際は800～1000 秒要するモデルが存在したためである。

２．問題を把握する方法

段階を経て詳細に調べることで理解を深めながら検討を行えることが望める。また、検査結果の境界によりどこに問題があるか検討を行えることがわかった。課題は、多くの検査結果から知りたい項目において検査結果を抽出して比較を行う方法を考えなくてはいけない。

３．エラーのバリエーションを区別する方法

前処理としてエラーの内容を抽出し、エラーのバリエーションごとに重みづけを行ってクラスター分析を適用した結果エラーの内容ごとに分けることができた。また、同一のエラーでは、詳細な区別をするため反例の実行列を抽出して距離を求めた。この結果、実行列に差分が生じるので区別すことが可能であることがわかった。課題は、エラーの種類ごとにグループ化を行う方法と同一エラーでのグループ化を1度に行う方法を考えることである。