SPIN を用いたバイナリモデル検査

土肥　雅俊 (610060)

北陸先端科学技術大学院大学 情報科学研究科

2008 年 2 月 29 日

キーワード: spin, バイナリモデル検査, モデル検査、形式検証, 形式的手法.

1　はじめに

近年、ソフトウェアの信頼性確保のために形式的検証手法が注目されている。しかし、プログラムの振舞いを形式検証することは困難であり、プログラミング言語から仕様記述言語への変換なども必要であった。しかも、これらの作業において、状態爆発や言語間の記述能力の相違が問題としてよく挙げられる。たとえ、経験豊富な技術者であってもプログラムの振舞いや性質を適切に捉え、これらの問題を解決することは容易ではない。そこで、本研究では、SPIN の C 言語埋込み機能を利用した C プログラムのバイナリ検査手法を提案する。SPIN は PROMELA(PROcess MEtamodel LAnguage) 言語を採用しているが、C 言語を直接、検査コード内に埋込む機能も有している。その機能を利用し、検査対象プログラムに大きな変更を加えずに、適切な検査モデルを作り出す手法を提案する。

2　バイナリモデル検査とは

バイナリモデル検査とは本研究で新たに定義した検査手法のことである。まず、検査対象プログラムをコンパイルしてバイナリ実行し、同時に検査プログラムを開始する。そして検査プログラムは、バイナリが使用しているメモリ空間を抽象化して状態と捉え、状態空間を探索していく。このような手順を本手法では取る。よって、バイナリモデル検査と名付けることにした。また、実際にバイナリが動いているメモリ空間を検査対象とすることによって、プログラム運用環境により近い状態で検査が可能となる。メリットとして仕様記述言語とプログラミング言語間の記述能力の違いを解消すること、自動的に検査モデルが作成されること、プロルグラムの動作環境を直接利用できる事などがあげられる。

3　バイナリモデル検査による検査

本研究では、バイナリモデル検査を C プログラムに適用する前段階として検査法概要を定義している。バイナリ検査手法では、まず監視したい C 変数を指定する。そして、C 関数の呼び出し列（C プログラム）に入力変数の集合を与え、その結果得られた出力変数の集合の状態をチェックする。以上の手順で C プログラムの正しさについて保証する。バイナリモデル検査は全てこの概要に従って検査が行われている。

次に、この概要をもとに sort プログラムにバイナリモデル検査を適用し検査実験を行った。その結果、sort 関数の記述を全く変更することなく、任意の値が格納された配列 (入力変数の集合) が正しく操作され、ソートされた配列（出力変数の集合）が得られることを保証できた。

sort プログラムに対するバイナリモデル検査の適用は、成功したが sort プログラムは非常に規模の小さいプログラムである。そこで、システム上のファイル情報を表示する ls プログラムへバイナリモデル検査を適用した。ls プログラムはある程度の規模があり、ソート、リスト構造など代表的なアルゴリズムを含んでいたため、検査対象とした。この実験では、代表的な例として、ls プログラムは内部でシステムコールを呼び出すので検査の結果がシステムの状態に左右されてしまうという問題があった。そこで、プログラムを検査可能にするための検査環境の設定手順について定義し、検査環境の場合分けをした。保証したい性質によって、検査環境を選択できるようにしたのである。その他にも監視するメモリの指定やポインタ変数の扱いに関してプログラムの変更が必要であったので、プログラムを検査可能にするためにはどのようにすれば良いかを示し、その手順を定義した。しかし、検査作業自体は複雑なものではなく、実際に ls プログラムのバグを見つけることもできた。

- 実験結果まとめ
  - 簡単な C プログラムならば、変更を加える事なく、直接プログラムが検査出来きた。
  - ある程度規模のある C プログラムの検証は、いくつかの変更が必要であるが実験作業自体は難しいものではなかった。
  - 無駄な C 変数を状態として指定しないことで、状態数をある程度押さえることが可能だという事を示した。
  - 動的メモリ空間の取り扱いなど、メモリ空間の効率的な指定方法を提案していく必要性があることが判明した。

4　まとめ

本研究では、バイナリモデル検査という独自手法を提案し、様々な問題の解決と整理を行った。まず、バイナリモデル検査手法の概要を整理し、どのような検査手法であるのか

を明確化させた。そして、本検査手法を行うことによって得られる利点について考察した。バイナリモデル検査を行うにあたり、検査対象 C プログラムの内容変更が必要な場合があるので、そのときの検査環境設定に関して場合分けをし、場合に応じてどのような手順を取って、変更しなければならないかを示した。次に、このように整理・一般化した検査方法を検査実験として sort プログラムや ls プログラムなど実際のプログラムに適用し、その効果について考察した。結果として、ある程度の規模の C プログラムであっても、プログラム構造をあまり変更せずに Spin から直接呼び出して検査可能であることが分かった。以上が検査手法に関しての成果である。またそれに関連して、技術面に関しての成果もいくつかあった。監視する C 変数指定方法とその扱い方に対する考察や、効率的にメモリ空間を監視するための工夫、C プログラム中で使用される動的メモリの取り扱い方に対する問題提起と解決策の提案などが代表的な技術的成果である。

これらの研究成果からバイナリモデル検査を利用すれば、検査コストが少なく、動作環境により近い状態での C プログラム検査が可能であることが示せた。