発表概要

ヒープの決定手続きを組込んだSMTソルバーを使うC言語ソースコード静的検査器

松田 元彦[1,a)]　前田 俊行[1]

2011年11月2日発表

抽象実行に基づくソースコード検査は，少ないアノテーション等の追加で検査が可能であり，利用しやすいツールを提供できる．実際，デバイスドライバ等に対して，排他ロックに対する呼出しの整合性検査などが行われている．対象となるコードではロックがヒープ中に置かれることががあるが，その場合ヒープが一要素のみであるとして簡略化して扱うことがある．一方，ヒープについてはSeparation Logicの部分言語に対する決定手続が既知である．抽象実行には，等式や整数に関する決定手続を使うSMTソルバーを用いるが，そのSMTソルバーにヒープの決定手続を組込む．それによって，ヒープの状態表現を簡略化せずに抽象実行が可能になる．現状できる検査内容は同じであるが，ヒープ状態の表現を明示することで検査の条件がより明確になると考えている．ヒープ中のデータのシェイプ構造についてはリストだけを扱い，あらかじめアノテーション等の形で与えるものとする．本発表では，SMTソルバーへの決定手続き導入の問題点の議論と，そのSMTソルバーを使う静的検査器の実装概要について述べる．

Static Checker for C Source Code Using SMT Solver with Decision Procedure on Heap

Motohiko Matsuda[1,a)]　Toshiyuki Maeda[1]

Presented: November 2, 2011

Checkers based on abstract interpretation are promising for the ease of their use. Several checkers have been proposed for checking consistency of usage of library routines such as synchronisation locks in device drivers. Such checkers are typically based on abstract interpretation using an SMT solver for integer constraints, but they become imprecise when locks are placed in a heap area, because they may make a simplifying assumption that the heap consists of a single element. For a subset of Separation Logic, which is popularly used in describing properties of the heap, a decision procedure is known. We discuss issues of adding the heap decision procedure to the existing SMT solver, and present an implementation overview of the static checker, which integrates heap property constraints into integer constraints to check consistency of usage of locks in a heap.

[1] 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo 113–0033, Japan
[a)] matu@acm.org