情報処理学会論文誌　プログラミング　Vol. 3　No. 3　20 (June 2010)

発表概要

ヒープ中の同期ロックの整合性に関するモデル検査

松田元彦[†1]　前田俊行[†1]　米澤明憲[†1]

カーネル・モジュール等に対して，排他ロック等の呼び出し整合性の検査を行うため，抽象実行に基づくモデル検査器が提案されている．しかし，カーネルではロック等はヒープ中に置かれることが多いのでヒープを扱うことができる検査器が必要である．一方，ヒープに関する性質を記述・検査するのに用いられる Separation Logic のサブセットに対して決定手続きが提案されている．そこで我々は開発中のモデル検査器のソルバにヒープに関する決定手続きを組み込み，ヒープ中にあるロックの整合性検査を試みた．ヒープの決定手続きは，ヒープの制約を等式の制約として生成するので，ソルバに対してフロントエンドとして組込み込むことができる．また，述語セットのリファインメントを行わずに実用的な速度で検査を行うため，関数の入口で行う述語セットの選択を導入している．また，ヒープに関するモデル検査行う場合の一般的な課題について述べる．

Model Checking Consistency of Usage of Synchronization Locks in Heap

Motohiko Matsuda,[†1] Toshiyuki Maeda[†1] and Akinori Yonezawa[†1]

Several model checkers have been proposed for checking consistent usage of kernel routines such as synchronization locks in kernel modules. Such model checkers fail to check locks when they are allocated in heaps. It is because they are typically based on abstract interpretation using integer constraint solvers and cannot handle descriptions of heaps. Recently, some decision procedures are proposed for subsets of Separation Logic, which is popularly used in describing properties of heaps. We present an extension to a model checker, which integrates a decision procedure to an integer constraint solver, where the extension generates heap constraints as a set of integer constraints, and can be integrated to many solvers. And also, the checker implements predicate selection at function call entries, which attains practical speed without using predicate refinement. The paper includes some discussion on general issues of heap related model checking.

（平成 22 年 1 月 27 日発表）

†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