要 旨

PDS モデル検査器を用いた
履歴ベースアクセス制御プログラムの検証

勝山 雅彦

Java 仮想機械などのプログラム実行系にはスタック検査と呼ばれる動的アクセス制御機構が導入されており，その拡張もいくつか提案されている．高田らは，Abadi らのアクセス制御機構を形式的に定義した HBAC（History-Based Access Control）プログラムを提案している．この HBAC プログラムを検証する場合は，呼び出し制御スタックを考慮したモデルに基づいて行うことが妥当である．そこで，高田らは文脈自由文法 (CFG) に基づく CFG モデル検査器を HBAC プログラム向きに最適化し実装した．しかし，このモデル検査器はあくまでもプロトタイプであり性能は十分とはいえない．一方，現在主流である有限状態モデル検査器を Pushdown System に拡張した PDS 検査器が近年開発され，改良が重ねられている．

本研究では，PDS モデル検査器 Moped を用い，HBAC プログラムの検証における性能を測定してその実用性を検証する．検証方法として，CFG モデル検査器と PDS モデル検査器で同一の検証対象を用い，プログラムサイズを変化させた場合の検証時間を比較した．実験の結果，単純な検証対象システムなら HBAC プログラム向きの最適化をしなくても PDS モデル検査器の性能は十分であることがわかったが，複雑なシステムになると最適化が必要であるということがわかった．

キーワード　モデル検査，実行履歴に基づくアクセス制御，プッシュダウンシステム

Abstract

Verification of History-Based Access Control program using pushdown system model checker

Masahiko Katsuyama

A dynamic access control infrastructure called stack inspection has been introduced into runtime systems such as the Java virtual machine, and some extensions have been proposed. Takata proposed the HBAC (History-Based Access Control) program that formally models an access control infrastructure proposed by Abadi. When we verify an HBAC program, it is appropriate to use a model that involves the control stack. Therefore Takata implemented a CFG model checker based on context-free grammar (CFG) and optimized it to HBAC programs. However, this model checker is a prototype and its perfomance is not enough. On the other hand, PDS model checkers, which are extensions of finite-state model checkers and are based on the pushdown system, have been developed and improved recently.

In this research, we use PDS model checker "Moped" and measure its performance in the HBAC program verification. We compare the verification time of the CFG model checker and the PDS model checker for several program sizes. The results of the experiment indicate that the performance of the PDS model checker without the optimization for HBAC programs is enough when the verified system is simple; however, when the vefied system is complicated, the optimization is necessary.

key words model checking, history-based access control, pushdown system