Automatic Verification and Generation of Dependable Systems

Li Xin, li-xin@jaist.ac.jp
Japan Advanced Institute of Science and Technology

1 Background

The current methods for software and system validation are mostly based on costly simulation and testing. While the fundamental problem for these methods is that they can not cover all possible scenarios of system runs. So it's hard to find subtle errors and scale up with more complex design. An attractive alternative to address these problems is the approach of formal verification. As one approach to formal verification, model checking can automatically verify reactive systems and has been successfully used in hardware and protocol design.

2 Objectives and Methodology

This research aims to extend the application of model checking techniques to automatic verification and generation of dependable software in general. My approach is based on two basic observations:

- program analysis is model checking of abstract interpretation.
- More verification should be done at compile-time instead of run-time.

My research focuses on the following two aspects:

- Model checking infinite state spaces.
- Compile-time verification.

2.1 Current Progress

- Strictness analysis for Haskell (Figure 1)
  Previous approaches to strictness analysis are all based on abstract interpretation and iteration of finding fixed points. These approach becomes costly and unprecise when tackling with higher-order functions and non-flat domain. Our approach try to address these problems by examining strictness semantics of all runs of the function based on pushdown automata. This work is not only a study on model checking infinite state spaces, but a preliminary preparation for compile-time verification.

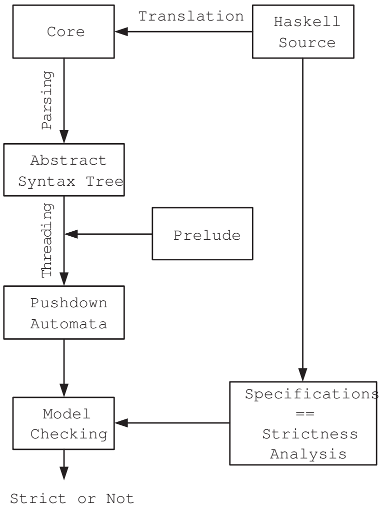

Figure 1: Strictness Analysis for Haskell with Model Checking

- Propose an example for later case study.
  We proposed a mechanism to efficiently record delegation traces for grid computing. It could be a good example for later case study.

2.2 Future Work

- Compile time verification. The first trial is to give proofs to G-machine.
- Development of a model checker based on an algebraic construction of a control flow graph.

A Publications in 2004

- Li Xin, Mizuhito Ogawa. A Lightweight Mutual Authentication Based on Proxy Certificate Trust List.
  Proceedings of the 5th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT04), LNCS 3320, pp.628-632.
- Li Xin, Mizuhito Ogawa. Proxy Certificate Trust List for Grid Computing. Computer Software, *to appear*, 2005

B Development In Progress

Strictness analyzer for Haskell