Experiments in Connection IOA and Daikon: Examples in Banking

Gustavo Santos, Toh Ne Win

30 June, 2001

1 Abstract

This document describes a set of experiments done on examples presented in Stephen Garland and Nancy Lynch's paper on "Using I/O Automata for Developing Distributed Systems". The purpose of the experiments was to see if Daikon, a dynamic invariant detector, could be useful in the design, implementation or proving of distributed algorithms.

We:

- Rewrote an automaton presented in the paper in a format usable by the IOA simulator.
- Ran the automaton in the simulator on a prescribed set of randomized transactions.
- Fed the program trace generated by the simulator to Daikon and recorded the invariants that were suggested.
- Sent the suggested invariants to the Larch Prover, where some of them were proved easily and others with difficulty.

Ultimately, we had hoped to use this analysis method in a feedback loop, whereby the proved invariants (through LP) would be sent back to the IOA simulator for a second run, in the hope of discovering deeper invariants. Although we still hope to do this, this paper presents only one cycle of the loop.

Notes on invariant proving. Invariants discovered by Daikon are easy to prove in LP when:

- They verify function of program data structures, such as whether all members of a designated data structure really do fulfill a simple predicate.
- Are subcomponents of more complex invariants that have until now been proved by hand.

Invariants discovered by Daikon are difficult to prove in LP when:

- They involve sums over sets, but this might be due to a lack of good libraries of set theorems in LP.

Daikon is good at detecting invariants that are simple and do not involve multiple levels of referencing between data structures.

Daikon is bad (currently) at detecting invariants that involve references from inside one data structure to another. For example, the following invariant would not be detected by Daikon:

If the ith member of an array of booleans is true, then there exists a record in set Foo such that Foo.j = i.

Our conclusion, ultimately, is that more experiments are necessary on different examples to see the success of this method.