Reading 3: Daikon

17-654/17-754: Analysis of Software Artifacts
Jonathan Aldrich (jonathan.aldrich@cs.cmu.edu)

Due: Monday, February 27, 2006 (5:00 pm)

10 points

Turn in a file named <username>-17654-R3.{txt,doc,pdf}, where username is your Andrew id. At the top of the document, state your name and Andrew id.

Readings:

- Michael D. Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically Discovering Likely Program Invariants to Support Program Evolution. Proceedings of the International Conference on Software Engineering, 1999.

Note: When answering these reading assignment questions be concise. It is expected that each question set can be answered adequately in a page of text or less. Rambling answers with irrelevant detail will not be received warmly. On the other hand, answers should contain enough detail to understand clearly what you are saying. Good English grammar and syntax is important, as always.

Reading Objectives:

- To understand the invariant inference techniques used in the Daikon tool, together with their application to software engineering problems.

Questions:

- Unlike other practical systems that we have studied in this course, Daikon does not directly find and report errors. Explain at least one way in which the information computed by Daikon could help a software engineer (a) identify existing code errors and (b) avoid introducing new ones when evolving a software system.