Assignment 3 (Tool): ESC/Java

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

Due: Thursday, February 9, 2006 (5:00 pm)

100 points total

Turn in a file named <username>-17654-A3.zip, where username is your Andrew id. The file should contain Stack.java, stack-output.txt, Square.java, square-output.txt, and answers.txt (with your answer to question 3). In a comment at the top of Stack.java, state your name, Andrew id, and how long you spent on the assignment.

Assignment Objectives:

- Use ESC/Java to check functional correctness properties of programs

Question 1 (60 points).

Download ESC/Java 2 from:

`http://secure.ucd.ie/products/opensource/ESCJava2/download.html`

Follow the instructions in README.release to install the system. You may have to install Java version 1.4 (Java 5.0 won't work—sorry, this is an annoying incompatibility, since we had to install Java 5.0 for Crystal. Don't uninstall the old Java, we'll use it again later.) When installing Java, DO NOT install it under "Program Files"—install it in some directory with NO space in the name. You can find Java 1.4 at:

`http://java.sun.com/j2se/1.4.2/download.html`

The installation instructions for Windows are not as clear as one might like. If you are working from Windows, you will need to modify escj.bat in two ways:

- Change the line "set ESCJAVA_ROOT=..." so that the right of the equals sign refers to the directory where escjava is installed (the directory where escj.bat is).

- Change the line "set JAVA=..." so that the right of the equals sign refers to the java.exe file. This is typically a path like: `C:\...\Java\jdk1.4.2\bin\java.exe`.

Now take the file Stack.java and StackCheck.java from the class website. Run ESC/Java 2 on both files together. Add pre- and post-conditions and invariants to Stack.java and fix any bugs you find in the code, until ESC/Java runs on both files without producing any warnings. You may not edit StackCheck.java. Nor may you remove the annotations that are already present in Stack.java.

Turn in: (a) your edited version of Stack.java, and (b) a printout of ESC/Java 2's output when run on the two files in stack-output.txt.

Important ESC/Java Note: ESC/Java may not be configured correctly to understand the invariant "theArray.owner == this" in Stack.java. Do not remove this invariant (or take out any other declaration that's in the program already). Instead, if you get a warning about the owner field not being defined, simply call escj with an extra argument: -Specs escjdir/escspecs.jar where escjdir is your escjava directory. Again, you may have trouble on Windows if there is a space in escjdir. This should eliminate the warning about not having an owner field.

If you're curious about what this invariant means, read:

`http://gatekeeper.research.compaq.com/pub/DEC/SRC/technical-notes/SRC-2000-002.html#3.2.17%20owner`

A final hint: sometimes ESC/Java works better if you express invariants directly in terms of fields like topOfStack rather than in terms of calling other functions like isFull(). My guess is that this is just a limitation of the tool.

Question 2 (30 points).

The sum of the odd numbers from 1 through $n*2-1$ is equal to n^2. Write a program that computes n^2 using this series. Check the correctness of your program using ESC/Java, i.e. that it really computes n^2. Your solution should use appropriate @requires, @ensures, and @loop_invariant clauses.

Turn in: (a) your java code Square.java, and (b) a printout of ESC/Java 2's output when run on the Square.java in square-output.txt.

Question 3 (10 points).

In the file answers.txt, criticize the ESC/Java tool (1-2 paragraphs). What did you like about it, and conversely what stands in the way of making this a practical tool?