Advances in Programming Languages

APL7: ESC/Java 2

Ian Stark

School of Informatics
The University of Edinburgh

Thursday 31 January 2008
Semester 2 Week 4

Topic: Some Formal Verification

This is the last of three lectures about some techniques and tools for formal verification, specifically:

- Hoare logic
- JML: The Java Modeling Language
- ESC/Java 2: The Extended Static Checker for Java

JML Review

The *Java Modeling Language*, JML, combines model-based and contract approaches to specification.

Some design features:

The specification lives close to the code

Within the Java source, in *annotation comments* */*@...@*/*

Uses Java syntax and expressions

Rather than a separate specification language.

Common language for many tools and analysis

Tools add their own extensions, and ignore those of others.

Web site: `jmlspecs.org`

ESC/Java 2

> "The Extended Static Checker for Java version 2 (ESC/Java2) is a programming tool that attempts to find common run-time errors in JML-annotated Java programs by static analysis of the program code and its formal annotations."
>
> http://kind.ucd.ie/products/opensource/ESCJava2

It is available both as a command-line tool and a plugin for the *Eclipse* development environment.

ESC/Java performs different kinds of check:

- Checks based on types, flow of data, existing Java declarations;
- JML annotation checking that can be carried out directly;
- Logical assertions that need an external proof tool.

These last ones are passed to the *Simplify* automated theorem prover.

Many Different Checks

ESC/Java 2 checks for very many things. These include:

- Null pointer dereference
- Negative array index
- Array index too large
- Invalid type casts
- Array storage type mismatch
- Divide by zero
- Negative array size
- Unreachable code
- Deadlock in concurrent code
- Race condition
- Unchecked exception
- Object invariant broken
- Loop invariant broken
- Precondition not satisfied
- Postcondition not satisfied
- Assertion not satisfied

JML annotations and assertions can help with all of these.

Soundness and Completeness

As a practical tool ESC/Java makes some compromises: it is not perfect.

- Not complete: it may complain about a correct program.
- Not sound: it may approve an incorrect program.

However, it reliable checks straightforward specifications, and automatically points out many potential bugs.

In particular:

- Distinguishes between *errors* (definitely bad), *warnings* (could be bad) and *cautions* (can't be sure it's good).
- Sources of unsoundness and incompleteness are documented.

Soundness and Completeness

As a practical tool ESC/Java makes some compromises: it is not perfect.

- Not complete: it may complain about a correct program.
- Not sound: it may approve an incorrect program.

However, it reliable checks straightforward specifications, and automatically points out many potential bugs.

In particular:

- Distinguishes between *errors* (definitely bad), *warnings* (could be bad) and *cautions* (can't be sure it's good).
- Sources of unsoundness and incompleteness are documented.

> . . . as we know, there are "known knowns"; there are things we know we know. We also know there are "known unknowns"; that is to say we know there are some things we do not know. But there are also "unknown unknowns" — the ones we don't know we don't know.
>
> (Donald Rumsfeld, 2002)

History

ESC/Modula-3 DEC Systems Research Center (SRC) 1991–1996

ESC/Java Compaq SRC, then Hewlett-Packard 1997–2002

ESC/Java 2 University of Nijmegen, University College Dublin 2004–

K. Rustan M. Leino. Extended Static Checking: A Ten-Year Perspective in *Informatics: 10 Years Back, 10 Years Ahead*. Lecture Notes in Computer Science 2000, Springer.

Notices

Your assignment topic choices are due in tomorrow: send the topic name and three references via submit cs4 apl 1 choice.txt

I shall be away next week. No lectures on Monday and Thursday, but I recommend you take the opportunity to read more on your chosen coursework topic.

Running the ESC/Java 2 plugin for Eclipse requires some careful installation tuning. Details available on the course lecture log.

The next lecture is at 9am on Monday 11 February. The plan is to show a Microsoft webcast on program verification in C# with the Spec# tool.

Demo

ESC/Java 2 in Eclipse