

An Introduction to Maude

Neal Harman

This is a short course of about 3-5 lectures introducing the fundamental concepts underlying the Maude system and its use, particularly as this applies to modelling and verifying microprocessors.

- What is term *rewriting*? Maude uses a method of evaluation called term rewriting, which uses a set of term rewriting rules that look suspiciously like equations. However, they are not (at least not in general), and we occasionally need to worry about this if we are to end up with *confluent* and *terminating* computations.
- Basic Maude. This will largely be revision of, with some expansion, the material used to teach Maude in CS_213 System Specification. We will however introduce the use of *sub-sorts* and *membership axioms* to make our models cleaner and easier to read.
- Using Maude to represent microprocessors. Again, largely revision material - we will revisit the concepts of (i) state sets; (ii) next-state functions; and (iii) iterated maps in Maude, together with the necessary ancilliary tools for representing bit strings, memory etc.
- The Maude Meta-Level. In addition to being able to describe systems external to itself, Maude is *reflective* (that is, it can model itself), and can direct its own rewriting via a *meta-level*. Using this meta-level is essential if we are going to successfully verify examples.
- Leading to verification. This short course leads into another short course - on the theory underlying the process of verifying microprocessors, and its implementation in Maude.
- The web pages: in case you're not reading this online the URL is <http://www-compsci.swan.ac.uk/~csneal/MaudeCourse/> Notes will be available in HTML and PDF form.