Tutorial 1: An Interpreter for `While`

September 30, 2010

1. The interpreter is written in Java, so if you don't have Java installed, make sure to do that first.

2. Download the interpreter from `http://www.cs.tcd.ie/Edsko.de.Vries/tcs`

3. The interpreter is for a language called `While`, which you will see in more detail in the next few lectures. To get a list of the command line arguments accepted by the interpreter, simply run

   ```
   ./run.sh
   ```

   In this tutorial, we will only use a few of these commands.

   (a) The interpreter allows you to stick bits of code together and pretty-print them. Try running

   ```
   ./run.sh --inline n := 5 --load examples/fac.while --show-while
   ```

   (b) You can also execute `While` programs. Try running

   ```
   ./run.sh --inline n := 5 --load examples/fac.while --small-step-while
   ```

   to run the code showing all intermediate states of the program (small-step semantics).

   (c) Also try

   ```
   ./run.sh --inline n := 5 --load examples/fac.while --big-step-while
   ```

   to show only the state at the end of the program (big-step semantics).
   Can you think of an example program where the big-step semantics would not be very useful (while debugging, say)?

4. The small-step and big-step semantics for arithmetic expressions is implemented as the set of classes in `While/Aexp`; have a look at these files and see how they compare to the rules discussed in the lectures.

5. The Fibonacci sequence is given by

   $$1, 1, 2, 3, 5, 8, 13, 21, \cdots$$

   where every number (except for the first two) is the sum of the two proceeding. Write a small `While` program that computes the nth number in this series, and test your code.