CS4212 – Compiler Design August 24, 2012
School of Computing
National University of Singapore

Recitation 1

(used in class on August 31, 2012)

The instructor will guide the class towards developing solutions to the exercises given below. The students are strongly encouraged to attempt the exercises on their own before coming to class, and be ready to discuss/challenge the approach suggested by the instructor, as well as contribute their own ideas.

Exercise 1

Consider the operator declarations used in the naïve interpreter given in the file `interp.pl`. These declarations are more permissive than needed, allowing the construction of Prolog terms that would not normally be valid programs. Unfortunately, it is not possible to adjust the operator declarations so as to reject those incorrect programs. The only solution here is to define a validation predicate that would succeed only for terms that constitute valid programs. Develop such a definition of a validation predicate.

Exercise 2

The syntax of the language accepted by the interpreter given in `interp.pl` is also, in a way, too restrictive. We cannot place semicolons in front of closing braces, and we have to enclose nested statements in braces even when they are made up of a single instruction. Moreover, we are not allowed to use if statements without else clauses. Last but not least, the language is restricted to the use of only 2 variables, and allows a very restrictive set of arithmetic and logic operators. Modify the interpreter so that all these restrictions are removed.