COMPUTER SCIENCE 322 (Winter Term 2004)
Compiler Construction
Prof. Levy

Problem Set 9

Due Friday 09 April

Reading Assignment: Dragon Book 9.1-9.7

Programming Assignment

Using your intermediate-code generator from the previous assignment, complete your compiler by adding a back-end for MIPS code generation. Implement as much as you can, and feel free to complete parts of the previous assignment(s) that you weren't able to finish in time. You will get the most credit for target code that successfully runs in SPIM, so I suggest once more that you start small; *e.g.*, with `let val a = 3 in a + 1 end`.

Your code generator can just emit (`System.out.println()`) one MIPS instruction at a time. You will need a data structure to represent three-address instructions, because you will be using them to generate MIPS assembly code. I suggest building a class hierarchy for your three-address representation (`ThreeAddressPlus extends ThreeAddressArithLogic extends ThreeAddress`) , and having your intermediate-code generator call *e.g.* the `target()` method of each instruction in sequence. This method can be declared using an abstract class or interface, so you can easily re-target for other architectures.