

UCT 2006 CSC3005 Compilers

Practical Assignment 3: Instruction Selection

Write a program to convert IR trees into machine code, selecting instructions using the **maximal munch** algorithm.

The IR language to support is a subset of that discussed in the text, including:

SEQ, MOVE, MEM, CONST, TEMP, BINOP, CALL_EXT

You may assume that all generated IR code conforms to the source language of Assignment 2, implying that CALL_EXT is only used for print operations.

Your program must read in the IR tree as a single line of text from a file and convert it to machine instructions, following the Jouette architecture provided in the textbook, with one additional instruction, CALL_print, to print the value of a register. For this and all other instructions, ignore the register allocation completely at this stage.

Your main program must accept a single command-line parameter that is the name of a data file containing a single line of IR tree, with no spaces.

Your output must be a list of machine instructions, for example:

```
ADDI r<-r0 + 1
ADDI r<-r0 + 2
ADD r<-r+r
ADDI r<-r0 + 3
MUL r<-rxr
CALL_print r
```

You may use Java or any other programming language that the TA agrees to. You may use any built-in data structures and any parser tools (though those are not necessary).

Test your code with and provide output for the 4 data files attached (testdata.tar.gz). In each case, use output redirection to capture the output and save it to appropriately-named files (e.g., test1.out).

Your assignment will be marked according to the following criteria:

- Correctness (40%) (test programs generate correct output)
- Documentation (20%) (comments within hand-written source – no report necessary)
- Efficiency (10%)
- Stress (30%) (hidden test programs generate correct output)