**Compiler Report – Yizhou Hu – Fall 2014**

**Usage:** java JRR < in.txt > out.txt

All the informative outputs are printed to standard error, while the actual assembly code are printed to standard output.

The program does still try to output some assembly code to stderr, even if parsing is not successful. However in this case it does not output to stdout.

**List of files:**

JRR.jj – The main JJ file.

40603.txt – The original test file we made in class.

error1.txt – Demonstrates unary negation: must have presences around each layer.

error2.txt – Demonstrates “warning: reference before assignment”. Does not stop compiling.

error3.txt – Demonstrates missing semicolons, presences and brackets.

error4.txt – Demonstrates Lexical Error – compiler stops working at this error (intended not to reinit() as per the JavaCC “Error Reporting and Recovery” manual).

**Problem:**

1. I disabled all semantic actions after parsing errors, meaning the program can only detect grammar issues from then on. It can no longer produce warnings, nor can it produce a comprehensive symbol table.
2. Integers are sometimes treated as permanent variables.

**Definition of Quad:**

Operators:

mov, +, -, \*, /, %, LT, GT, cpr, &, |, NEG, NOT

~~~~~~~ have symbol ID’s as arguments

jeqz, jmp, ret, exit

~~~~~~~ have no or has quad numbers as arguments

Arguments: integers. -10 denotes null.

Arguments has corresponding but not linked “nextuse” field. -1 denotes no next use or argument non-existent.

Leader: indicates if the quad is a block leader.

Address: the line number of the starting assembly line for this quad

**Definition of Symbol:**

name: String. Symbol name or representation of an integer.

nextuse, regloc, memloc: integers. -1 denotes null or non-existent.

type: String. “int”, “id” or “temp”