Department of Computer Science Columbia University

Sample Final Examination – COMS W4115

Programming Languages and Translators

Monday, April 29, 2013 2:40-3:55pm

Closed book, no aids. Do questions1-5. Each question is worth 20 points. Question 6 is extra credit, 10 points.

1. Briefly describe

   a) a C compiler and a Java compiler by drawing a diagram of each
   b) applicative-order evaluation and normal-order evaluation
   c) function abstraction and function application in lambda calculus
   d) all the redexes in the lambda expression `(λx.1)((λx.x x)(λx.x x))`

2. True or false? Briefly justify your answer.

   a) The set of viable prefixes of an SLR(1) grammar is always a regular language.

   b) Removing left recursion from an SLR(1) grammar always produces an LL(1) grammar.

3. Consider the following C while-statement assuming `s` and `t` are strings, which in C are pointers to arrays of characters terminated by the null character '\0'.

```
while (*s++ = *t++) ;
```

   Translate this statement into three-address code. Explain what your code does and how it works. State all your assumptions.

4. Consider the function

```
int fact(int n) {
   if (n < 1) return 1;
   else return (n * fact(n - 1));
}
```

   Explain in high-level terms how a machine-language program for this function would execute `fact(2)`. Show the contents of the run-time stack before, during, and after each procedure call. Explain all your assumptions.

(over)

5. Live-variable analysis

   a) Define what it means for a variable to be live at a program point.
   b) State two possible ways a compiler might use live-variable information.
   c) For the basic block below determine the sets of variables that are live at each program point in the basic block. You can assume that no variables are live on exit from the basic block.

```
a = 1
b = 2
c = 3
a = a + b
c = c - a
print a
print c
```

6. [Extra credit, 10 pts] Can every regular language be generated by an LL(1) grammar? Prove your answer.