Computer Science Comprehensive Exam—Fall 2011
Compiler Construction

Instructions: Do *not* put your name on the exam, please answer all the questions directly on the exam itself. You may write on the back of the pages. You may need scratch paper to work out the answers before writing them on the exam. Answer **all** the questions. You have 90 minutes. Explain answers as fully as possible, give examples or define terms, if appropriate.

1. How many basic blocks does this following code have?

   Draw the control-flow graph for the code:

```
l0: x = y + x
    w = 2 * x
    if s==u goto L1 else goto L4
L1: s = w + 1
L2: y = s + x
    if y > 1000 goto L0 else goto L3
L4: x = w + u
    u = u - 1
    goto L2
L3:
```

2. Translate each of the following regular expressions into a context-free grammar.

   (a) $((xy^*x) \mid (yx^*y))?$ where x and y are terminals.

   (b) $((0 \mid 1)^+ \cdot (0 \mid 1)^*)|((0 \mid 1)^* \cdot (0 \mid 1)^+)$ where 0, 1, and $\cdot$ are terminals.

3. Consider *your* grammar for question 2(a) above. For each statement below, circle either "true" or "false" as appropriate.

   (a) true / false The grammar is left-recursive.

   (b) true / false The grammar is right-recursive.

   (c) true / false The grammar is LL(1).

   (d) true / false The language is LL(1).

   Please explain your true/false resposes at a general or intuitive level (no detailed proof is needed).

4. Consider *your* grammar for question 2(b) above. For each statement below, circle either "true" or "false" as appropriate.

   (a) true / false The grammar is left-recursive.

   (b) true / false The grammar is right-recursive.

   (c) true / false The grammar is LL(1).

   (d) true / false The language is LL(1).

   Please explain your true/false resposes at a general or intuitive level (no detailed proof is needed).

5. Consider the following grammar:

$$
\begin{array}{lcl}
S & \rightarrow & u\,B\,D\,z \\
B & \rightarrow & B\,v \\
B & \rightarrow & w \\
D & \rightarrow & E\,F \\
E & \rightarrow & y \\
E & \rightarrow & \epsilon \\
F & \rightarrow & x \\
F & \rightarrow & \epsilon
\end{array}
$$

(a) Compute nullable, FIRST and FOLLOW for all nonterminals.

	nullable	FIRST	FOLLOW
S			
B			
D			
E			
F			

(b) Compute the FIRST of the right-hand side of all productions.

	α	FIRST(α)
1	$S \rightarrow u\,B\,D\,z$	
2	$B \rightarrow B\,v$	
3	$B \rightarrow w$	
4	$D \rightarrow E\,F$	
5	$E \rightarrow y$	
6	$E \rightarrow \epsilon$	
7	$F \rightarrow x$	
8	$F \rightarrow \epsilon$	

(c) Fill in the LL(1) parse table for the grammar. Explain clearly why the grammar is *not* LL(1).

	w	u	v	x	y	z
S						
B						
D						
E						
F						

6. Consider the algorithm to compute CLOSE[I] for the set I of LR(1) items for some grammar. Suppose the grammar contains the production $X \rightarrow \gamma$ where X is some non-terminal and γ is some string of terminals and non-terminals. Answer the following questions assuming A is some non-terminal, α and β are strings of terminals and non-terminals, and y and z are terminal symbols.

   (a) If $A \rightarrow \alpha \bullet X$, z is in I, which item or items (if any) would be added to CLOSE[I]?

   (b) If $A \rightarrow \alpha \bullet Xy$, z is in I, which item or items (if any) would be added to CLOSE[I]?

   (c) If $A \rightarrow \alpha \bullet X\beta$, z is in I, which item or items (if any) would be added to CLOSE[I]?

7. For the following augmented grammar:

$$
\begin{array}{rlcl}
0 & S' & \rightarrow & S\,\$ \\
1 & S & \rightarrow & a\,A \\
2 & S & \rightarrow & b\,B \\
3 & A & \rightarrow & C\,a \\
4 & A & \rightarrow & D\,b \\
5 & B & \rightarrow & C\,b \\
6 & B & \rightarrow & D\,a \\
7 & C & \rightarrow & E \\
8 & D & \rightarrow & E \\
9 & E & \rightarrow &
\end{array}
$$

   (a) Give a diagram of the LR(1) states and transitions.

   (b) Give the LR(1) parsing tables.

   (c) Is the grammar LR(1)? Explain.

   (d) Is the grammar LALR(1)? Explain.