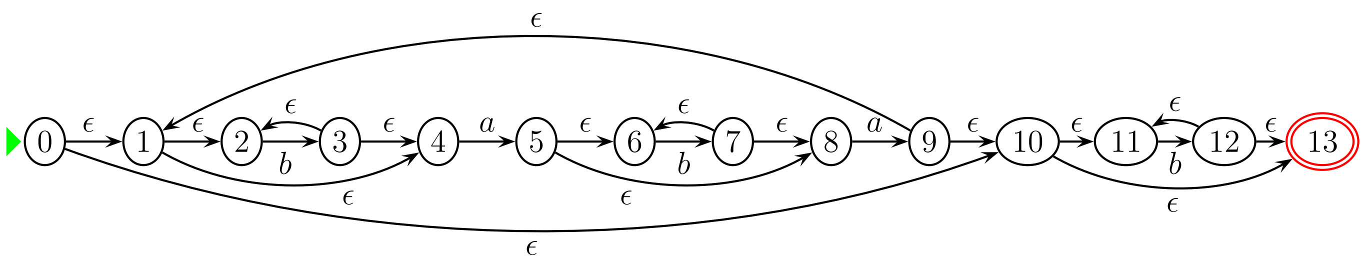
Computer Science Comprehensive Exam—Spring 2007
Compiler Construction

Instructions: Do *not* put your name on the exam, please answer all the questions directly on the exam itself. Answer **all** the questions. Explain answers as fully as possible, give examples or define terms, if appropriate.

1. Do all reasonable programming languages have a LALR(1) grammar? Explain.

2. Convert the following NFA over the $\Sigma = \{a, b\}$ to a DFA using the subset construction. The start state of the NFA, marked by a triangle, is 0; the only final state, marked by double lines, is 13. Your result should have five states. Label your states with capital letters A, B, C, D, and E and fill in the table below so that the correspondence is clear between the states of your DFA and *sets* of the NFA's state labels. Fill in the table with the transition on each state of your DFA. Do not simplify.



DFA	NFA	a	b
A			
B			
C			
D			
E			

3. Rewrite the entire grammar below eliminating left recursion.

$$
\begin{array}{rlcl}
1 & S & \rightarrow & A \\
2 & A & \rightarrow & A\,a\,B\,d \\
3 & A & \rightarrow & A\,b\,B \\
3 & A & \rightarrow & A\,c \\
4 & A & \rightarrow & a\,A \\
5 & A & \rightarrow & B\,d \\
6 & B & \rightarrow & b \\
7 & B & \rightarrow & B\,b
\end{array}
$$

4. Consider the following augmented grammar over the alphabet $\{a, b, c\}$.

$$
\begin{array}{lll}
0 & S' & \to S\,\$ \\
1 & S & \to A \\
2 & S & \to c\,b \\
3 & A & \to a\,A\,b \\
4 & A & \to B \\
5 & B & \to c
\end{array}
$$

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

	nullable	FIRST	FOLLOW
S'			
S			
A			
B			

(b) For all productions, compute the FIRST of the right-hand side, or the FOLLOW of the left-hand side, as appropriate for computing the LL(1) parsing table.

$N \to \alpha$	$null(N)$?	FIRST(α)	FOLLOW(N)
0 $S' \to S\,\$$			
1 $S \to A$			
2 $S \to c\,b$			
3 $A \to a\,A\,b$			
4 $A \to B$			
5 $B \to c$			

(c) Complete the *entire* LL(1) parse table below.

	a	b	c	$\$$
S'				
S				
A				
B				

(d) Is the grammar LL(1)? Please circle the correct answer: yes / no. Explain.

5. Consider the algorithm to compute CLOSE[I] for the set of LR(1) items I 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]?

6. Consider the following grammar.

$$
\begin{array}{llll}
1 & S & \rightarrow & \texttt{id} \\
2 & S & \rightarrow & V := E \\
3 & V & \rightarrow & \texttt{id} \\
4 & E & \rightarrow & V \\
5 & E & \rightarrow & \texttt{n}
\end{array}
$$

Add the augmenting production. Build the state transition diagram and parsing table for LR(1) parsing. Is the grammar LR(1)? Please circle the correct answer: yes / no.