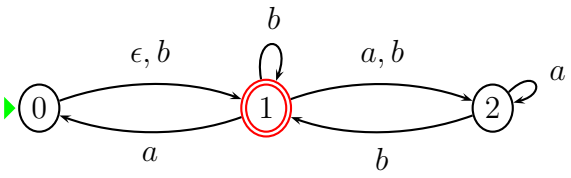
Computer Science Comprehensive Exam—Spring 2010
Compiler Construction

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

1. In general terms, describe a just-in-time compiler. How is it constructed, how does it compare to "normal" compilers?

2. Give a regular expression for strings over the alphabet $\{a, b, c\}$ where the first a precedes the first b.

3. Convert the following NFA over the alphabet $\{a, b\}$ to a DFA using the subset construction. Note tha some edges represent more than one transition. The start state of the NFA, marked by a triangle, is 0; the only final state, marked by double lines, is 1. Be sure to label the states of your DFA with *sets* of the NFA's state labels, so that the correspondence is clear. Do not simplify.



4. Consider the following grammar. (The terminals are lower-case letters; the nonterminals are upper-case.)

$$
\begin{array}{rlcl}
0 & S' & \rightarrow & A\,\$ \\
1 & A & \rightarrow & ABCD \\
2 & A & \rightarrow & \\
3 & B & \rightarrow & x \\
4 & B & \rightarrow & y \\
5 & B & \rightarrow & \\
6 & C & \rightarrow & z \\
7 & C & \rightarrow & h \\
8 & C & \rightarrow & \\
9 & D & \rightarrow & k \\
10 & D & \rightarrow & \\
11 & E & \rightarrow & m \\
12 & E & \rightarrow & n
\end{array}
$$

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

(b) Create the LL(1) parse table.

(c) Is the grammar LL(1)?

5. Construct the LR(1) parsing automaton and table for the following grammar. (The terminals are lower-case letters; the nonterminals are upper-case.) Is the grammar LR(1)? Is the grammar SLR? Explain.

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