

Computer Science Comprehensive Exam—Spring 2015

Compiler Construction

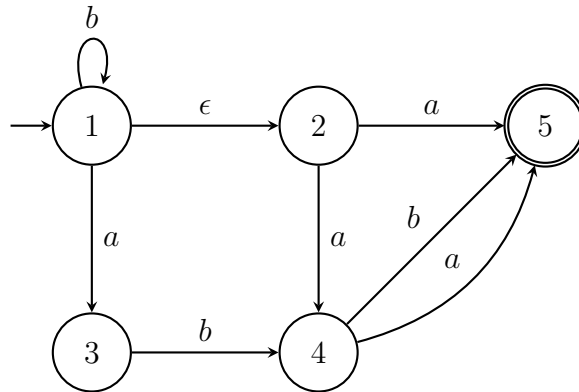
Instructions: Do *not* put your name on the exam, please answer all the questions directly on the exam itself. You may write on the back of the pages. You may need scratch paper to work out the answers before writing them on the exam. Answer **all** the questions. You have 90 minutes. Explain answers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

- For each statement below, circle either “true” or “false” as appropriate.
 - true / false The set of languages recognized by SLR grammars is a proper subset of the languages recognized by context-free grammars.
 - true / false The set of languages recognized by LR(1) grammars is a proper subset of the languages recognized by LL(1) grammars.
 - true / false The set of languages recognized by LALR(1) grammars is a proper subset of the languages recognized by LR(1) grammars.
- Why are there no possible shift/shift conflicts in an LR parsing table?
- Find the shortest string that is not in the language represented by the regular expression $a^*(ab)^*b^*$.
- Find a regular expression corresponding to the language of all strings over the alphabet $\{a, b\}$ that contain exactly two a 's.
- Consider the following grammar.

```
0  $S' \rightarrow S \$$ 
1  $S \rightarrow ( X$ 
2  $S \rightarrow E \mathbf{sq}$  )
3  $S \rightarrow F ($ 
4  $X \rightarrow E )$ 
5  $X \rightarrow F \mathbf{sq}$  )
6  $E \rightarrow A$ 
7  $F \rightarrow A$ 
8  $A \rightarrow$ 
```

- Carefully build the start state (*and only the start state*) of the grammar.
- Given only the start state of the parsing automaton, fill in as much information as you can for the corresponding row in the action and goto table.
- What can you conclude about the grammar knowing only the start state?

6. Convert the following NFA over the alphabet $\{a, b\}$ to a DFA using the subset construction. Note that some edges represent more than one transition. The start state of the NFA is 1; the final state, marked by double lines, is 5. 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.



7. Consider the following grammar (uppercase letters are nonterminals):

- 1 $S \rightarrow I : L$
- 2 $S \rightarrow I$
- 3 $I \rightarrow \mathbf{id}$
- 4 $I \rightarrow L \mathbf{lb}$
- 5 $L \rightarrow \mathbf{id}$
- 6 $L \rightarrow (\mathbf{lb} : L)$

- (a) Give any string/sentence of length seven in the language generated by the grammar.
- (b) Compute nullable, FIRST and FOLLOW for *each* non-terminal.
- (c) Is the grammar LL(1)? yes / no Explain.
- (d) Define what *left factoring* means.
- (e) Is it possible to left factor the grammar? yes / no If so, give the result. If not, explain.

- 1 $S \rightarrow I X$
- 1a $X \rightarrow : L$
- 1b $X \rightarrow$
- 2 $S \rightarrow I$
- 3 $I \rightarrow \mathbf{id}$
- 4 $I \rightarrow L \mathbf{lb}$
- 5 $L \rightarrow \mathbf{id}$
- 6 $L \rightarrow (\mathbf{lb} : L)$

- (f) Is the result LL(1)? yes / no Explain.

8. Consider the following grammar (uppercase letters are nonterminals):

- 0 $S' \rightarrow S \$$
- 1 $S \rightarrow \mathbf{a g d}$
- 2 $S \rightarrow \mathbf{a A c}$
- 3 $S \rightarrow \mathbf{b A d}$
- 4 $S \rightarrow \mathbf{b g c}$
- 5 $A \rightarrow B$
- 6 $B \rightarrow \mathbf{g}$

Answer all the following questions on the remainder of the page and on the back. Circle the word “yes” or “no” as appropriate, and don’t forget to explain.

- (a) Give LR(1) parsing automaton for the grammar.
- (b) Give the entire LR(1) parsing tables.
- (c) Is the grammar LR(1)? yes / no Explain.
- (d) Give the entire LALR(1) parsing tables.
- (e) Is the grammar LALR(1)? yes / no Explain.