CS 340, Fall 2015 — Oct 1st — Exam 1 Name: \_\_\_\_\_

Note: in all questions, the special symbol  $\epsilon$  (epsilon) is used to indicate the empty string.

**Question 1.** [10 points] Define a regular expression for the language containing all strings of symbols in the alphabet  $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$  where each occurrence of  $\mathbf{b}$  is immediately preceded and followed by at least one occurrence of  $\mathbf{c}$ .

**Question 2.** [10 points] Specify a *deterministic* finite automaton that recognizes the language described in Question 1. Make sure that the start state and final state(s) are labeled, and that each transition indicates the direction and what input symbol is consumed.

**Question 3**. [10 points] Define a regular expression that generates the language over the alphabet {**a**, **b**, **c**} of all strings in which

- every other symbol (first, third, fifth, etc., or second, fourth, sixth, etc.) is a **b**
- there are never two or more adjacent occurrences of **b**

Note that the empty string *is* in this language.

Examples of strings in the language:	Examples of strings not in the language:	
$\epsilon$	abca	
a	abccb	
abc	babbb	
abcbc	ababaa	
abcbcb		
bcb		
bcba		

Question 4. [10 points] Consider the following deterministic finite automaton (DFA):



Show a DFA that recognizes the complement of the language accepted by this DFA (over the same alphabet). In other words, if a string is accepted by the DFA, your DFA should reject it, and if a string is rejected by the DFA, your DFA should accept it.

Question 5. [10 points] Consider the following nondeterministic finite automaton (NFA):



Convert the NFA to a DFA (deterministic finite automaton). Include a table showing, for each reachable set of NFA states, the corresponding DFA state.

**Question 6.** [10 points] Specify a context-free grammar (CFG) that generates all strings of terminal symbols chosen from  $\boxed{\mathbf{a}[]()}$  where the square brackets and parentheses are properly balanced.

Examples of strings in the language: Examples of strings not in the language:

$\epsilon$	$[\mathbf{a})$
a	([a]
[a]	)a[a]
(a)[a]	[(a])
a((a))[a]	a][a

Hint: make sure delimiters (brackets and parentheses) are added in balanced pairs.

Be sure to specify which nonterminal symbol is the start symbol. Each production should have a single nonterminal symbol on the left hand side.

**Question 7.** [20 points] Consider the following context free grammar with the terminal symbols  $() \lambda \mathbf{x} \mathbf{y} \mathbf{f}$  and the nonterminal symbols EV, where E is the start symbol:

$$\begin{split} \mathbf{E} &\rightarrow (\lambda \mathbf{V} \cdot \mathbf{E}) \\ \mathbf{E} &\rightarrow \mathbf{E} \mathbf{E} \\ \mathbf{E} &\rightarrow (\mathbf{E}) \\ \mathbf{E} &\rightarrow \mathbf{V} \\ \mathbf{V} &\rightarrow \mathbf{x} \\ \mathbf{V} &\rightarrow \mathbf{y} \\ \mathbf{V} &\rightarrow \mathbf{f} \end{split}$$

(a) Show a derivation of the input string  $(\lambda \mathbf{x} \cdot \mathbf{f} (\mathbf{x} \mathbf{x}))$ . (Continue on the right-hand side if necessary.)

String	Production	String	Production
Е			

(b) Draw the parse tree for the derivation you found in part (a).

Question 8. [10 points] Consider the following productions in a context-free grammar (CFG):

 $\begin{array}{l} \mathrm{R} \rightarrow \mathrm{S} ~\mathbf{a} ~\mathrm{T} ~\mathrm{R} \\ \mathrm{R} \rightarrow \mathrm{S} \end{array}$ 

Assume that **a** is a terminal symbol, and R, S, and T are nonterminal symbols. Show the pseudocode for a recursive descent parse function for the nonterminal R (*i.e.*, a **parseR** function).

**Important**: Show how to use the lexical analyzer to choose between the two possible productions. Also, show how to raise an error if neither of the productions can be applied.

**Question 9.** [10 points] Create a Turing Machine that takes a string of **a** and **b** symbols as input, and as output produces a reversal of the original string. For example, if the input string is **babbaba**, then the output string (the symbols on the tape when the Turing Machine halts) should be **ababbab**.

Hint: The output string does not need to be in the same place as the original string.

**Hint**: You may find it useful to replace the symbols in the input string with **A** and **B** as part of the reversal process.

Make sure that your start state and final state(s) are indicated, and that each transition specifies an input symbol, output symbol, and direction.