Midterm Exam

Instructions:

• Please write neatly: if I can’t read it, it isn’t right.
• The exam is closed-book, closed-notes, closed-devices, closed-neighbors. Anyone violating these rules will be fed to a komodo dragon kept just for this purpose.
• If I speak of an “efficient” procedure, I mean this loosely: something that runs in polynomial, and not exponential, time.
• Good luck and kind wishes,

Phillip Rogaway

Name:

<table>
<thead>
<tr>
<th>On page</th>
<th>you got</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Σ</td>
<td></td>
</tr>
</tbody>
</table>
1 Fill in the Blank

Complete the following narrative, following the conventions of lecture and your text.

1. A DFA $M = (Q, \Sigma, \delta, q_0, F)$ has $|Q| = 10$ states and $|\Sigma| = 2$ characters. Then there are ________ points in the domain of $\delta$, and ________ points in the range of $\delta$. 

Answers are numbers.

An NFA $M = (Q, \Sigma, \delta, q_0, F)$ has $|Q| = 10$ states and $|\Sigma| = 2$ characters. Then there are ________ points in the domain of $\delta$, and ________ points in the range of $\delta$. 

Answers are numbers.

2. Every NFA-acceptable language is DFA-acceptable. To prove this, suppose you have an NFA $M = (Q, \Sigma, \delta, q_0, F)$. For simplicity, let’s assume that $M$ has no $\varepsilon$-arrows: $\delta(q, \varepsilon) = \emptyset$ for all $q \in Q$. We must convert the NFA $M = (Q, \Sigma, \delta, q_0, F)$ into a DFA $M' = (Q', \Sigma, \delta', q'_0, F')$ for the same language. As an example, if the NFA $M$ has 10 states, 3 of them final, then the DFA $M'$ we build from it will have $|Q'| =$ ________ states, of which $|F'| =$ ________ will be final. 

Answers are numbers.

3. You are given a first DFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ with $|Q_1| = 10$ states, $|F_1| = 5$ of them final. You are given a second DFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ with $|Q_2| = 10$ states, $|F_2| = 5$ of them final. Suppose you use the product construction to make a DFA $M = (Q, \Sigma, \delta, s, F)$ for $L(M_1) \cup L(M_2)$. It will have $|Q| =$ ________ states and $|F| =$ ________ of them will be final. 

Answers are numbers.

4. Suppose you convert $(00 \cup 11)^*$ into an NFA $M$ using the procedures shown in class and in the book. Then $M$ will have ________ states. The answer is a number.

5. In class we described how you can convert a DFA $M = (Q, \Sigma, \delta, q_0, F)$ into a context-free grammar $G = (V, \Sigma, R, S)$ for the same language. Our grammar will have $|V| =$ ________ variables and $|R| =$ ________ rules. Answers are formulas involving the components of the DFA.
2 Short Answer

1. Draw the smallest (=fewest state) DFA possible for the language \( L = 0 \cup (111)^* \). Be careful; probably no partial credit. Make sure to include all necessary transitions and to mark your start state final state(s).

2. Suppose you want to prove that there is no smaller DFA for \( L = 0 \cup (111)^* \) than the one you just gave. Give the first few sentences for such a proof. If your proof is going to involve examining multiple cases, identify them and then select one case to work out in full. No credit if problem (1), just above, is wrong.

3. Alice grades problem 1 (top of this page). She looks at Prof. Rogaway’s key and marks as wrong any solution with a different number of final states. A student complains: “While my DFA has a different number of final states” he says, “it too is a smallest DFA for \( L \); it just works differently.” Could the student be right? Carefully explain.
4. Carefully state the **Pumping Lemma** for regular languages.

5. Use the pumping lemma to prove that \( L = \{a^n b a^n : n \text{ is even}\} \) is **not** regular.

6. Give a CFG for the language \( L = \{a^n b a^n : n \text{ is even}\} \). Your CFG should use at most one variable and two rules.

7. In a few clear sentences, describe the main idea behind the proof of the **Pumping Lemma** for Context-Free Languages. Draw a picture that helps to explain the main idea.
8. Prove or disprove: the CFLs are closed under intersection.

9. Let $L = \{x \in \{a,b,c\}^* : x \text{ contains exactly one } a \text{ and exactly one } b\}$. Write a regular expression for this language. Make it as short as possible.

10. For $a \in \Sigma$ and $x \in \Sigma^*$, let $d_a(x)$ be the string obtained by deleting each $a$ from $x$ (eg, $d_a(abbca) = bbc$ and $d_a(aaa) = \epsilon$). Let $D_a(L) = \{d_a(x) : x \in L\}$. The CFLs are closed under $D_a$. Explain why.

11. In class we described the **CYK algorithm** to decide if a string $x$ is in the language of a CFG $G$. In a cogent and well-written paragraph, sketch how this procedure works. You should not write out detailed pseudocode; write out the sort of explanation you’d find in a well-written book describing the ideas underlying the algorithm.
3 No-Justification True-False

Darken (completely fill in) the correct answer. If you don’t know an answer, please guess.

1. True  False  If \( M = (Q, \Sigma, \delta, q_0, F) \) is a DFA and \( F = Q \) then \( L(M) = \Sigma^* \).
2. True  False  If \( M = (Q, \Sigma, \delta, q_0, F) \) is an NFA and \( F = Q \) then \( L(M) = \Sigma^* \).
3. True  False  Complementing the final state set of an NFA \( M \) gives an NFA for \( \overline{L(M)} \).
4. True  False  For a DFA \( M = (Q, \Sigma, \delta, q_0, F) \), we let \( \delta^* \) be the Kleene-closure of \( \delta \), \( \delta^* = \bigcup_{i \geq 0} \delta^i \).
5. True  False  If \( L \) is regular then \( L^* \) is regular.
6. True  False  If \( L \subseteq \Sigma^* \) for an alphabet \( \Sigma \), then \( L \) is regular iff \( L \cup \Sigma \) is regular.
7. True  False  If \( L \subseteq \Sigma^* \) for an alphabet \( \Sigma \), then \( (L \cup \Sigma)^* \) is regular.
8. True  False  If \( L \) is cofinite, meaning that its complement is finite, then \( L \) is regular.
9. True  False  The intersection of a finite language and an arbitrary language is regular.
10. True  False  The union of an infinite number of regular languages is regular.
11. True  False  Every subset of a regular language is regular.
12. True  False  A regular expression \( \alpha \) is a string.
13. True  False  The pumping lemma is a useful tool for proving languages regular.
14. True  False  \( \emptyset^* = \emptyset \).
15. True  False  \( L = \{ w \in \{0,1\}^* : w \text{ contains an equal number of 01's and 10's} \} \) is regular.
16. True  False  The intersection of a CFL and a regular language is context free.
17. True  False  \( \{ a^{2^n} : n \geq 0 \} \) is regular.
18. True  False  If a language \( L \) is not regular, this can always be demonstrated by using the pumping lemma.
19. True  False  If \( A \) and \( B \) agree on all but a finite number of strings, then one is context free if the other is.
20. True  False  An efficient algorithm is known that takes a regular expression \( \alpha \) and a word \( w \) and decides if \( w \in L(\alpha) \).
21. True  False  An efficient algorithm is known that takes in two DFAs and decides if they accept the same language.
22. True  False  Every regular language can be accepted by an NFA with only one final state.
23. True  False  There is a language \( L \) for which \( L = L^* \).
24. True  False  If \( G = (V, \Sigma, R, S) \) is a CFG, \( L(G) \neq \emptyset \), and \( S \rightarrow S \in R \), then \( G \) is ambiguous.
25. True  False  There is a CFL \( L \) such that \( G \) is ambiguous whenever \( L = L(G) \).