COMP481 Automata, Formal Languages, and Computability
Midterm, Spring 2006
March 1, 2006

Name:

- This is a two-hour exam.
- Answer all questions. Please, give clear and rigorous answers.
- This is an open-books open-notes exam.
- You can use any results that were covered in the lecture, on the homework assignments, or in the course textbook examples without proving them. Please, state clearly the result you are using and the source.
- Write your answers in the spaces provided. Use extra paper to determine your solutions, if needed, then neatly transcribe them onto these sheets.
- Make sure you clearly write your name.
- There are 11 pages, including this one. Make sure you have all of them.

GOOD LUCK

<table>
<thead>
<tr>
<th>Problem</th>
<th>Subproblem</th>
<th>Score</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

Total:
Problem 1

For each of the following four languages, determine whether the language is regular or not, and prove your answer.

1. $L_1 = \{ x\#y : |x| \cdot |y| \text{ is divisible by 5} \}.$
   
   **SOLUTION:** $L_1$ is regular. A regular expression for the language is
   $$(\Sigma^5)^*\#\Sigma^* \cup \Sigma^*\#(\Sigma^5)^*.$$

2. $L_2 = \{ a^nba^mba^{n+m} : n, m \geq 1 \}.$
   
   **SOLUTION:** $L_2$ is not regular. Assume $L_2$ were regular and $k$ is the constant. Choose $w = a^kba^kb\alpha^{2k}.$ Clearly, $w \in L_2$ and $|w| \geq k.$ For every possible way of writing $w = xyz$ such that $|xy| \leq k$ and $|y| \geq 1,$ we have $x = a^r, y = a^s,$ and $z = a^{k-r-s}ba^kba^{2k},$ where $r + s \leq k$ and $s \geq 1.$ Take $i = 0;$ then, $xy^iz = a^{k-s}ba^kba^{2k}.$ Since, $s \geq 1,$ it follows that $w \notin L_2,$ a contradiction. Therefore, $L_2$ is not regular.

3. $L_3 = \{ w \in \{ 1, \# \}^* : w = x_1\#x_2\# \cdots \#x_k \text{ for some } k \geq 0, \text{ each } x_i \in 1^*, \text{ and } x_i \neq x_j \text{ for } i \neq j \}.$
   
   **SOLUTION:** $L_3$ is not regular. Assume $L_3$ were regular, and let $L_3' = \overline{T_3} \cap 1^*1^*.$ Then, $L_3' = \{ 1^n\#1^n : n \geq 0 \}.$ Assume $L_3'$ were regular, and let $k$ be the constant. Choose $w = 1^k\#1^k,$ which is clearly in $L_3$ and satisfies $|w| \geq k.$ For every possible way of writing $w = xyz$ such that $|xy| \leq k$ and $|y| \geq 1,$ we have $x = 1^r, y = 1^s,$ and $z = 1^{k-r-s}\#1^k,$ where $r + s \leq k$ and $k \geq 1.$ Take $i = 0;$ then, $xy^iz = 1^{k-s}\#1^k.$ Since, $s \geq 1,$ it follows that $w \notin L_3,$ a contradiction. Therefore, $L_3$ is not regular.

4. $L_4 = \{ w \in \{ a, b \}^* : w \notin a^* \cup b^* \}.$
   
   **SOLUTION:** $L_4$ is regular, since $L_4$ is the complement of $a^* \cup b^*$ which is regular (and regular languages are closed under complement).

Problem 2

An all-NFSA $M$ is a 5-tuple $(K, \Sigma, \Delta, s, A)$ that accepts $w \in \Sigma^*$ if every state that $M$ could be in after reading input $w$ is a state in $A$.

1. Prove that a language $L$ is regular iff there exists an all-NFSA $M$ such that $L(M) = L$.
   
   **SOLUTION:** There are two directions to prove.
   Assume $L$ is regular. Then there exists an FSM, $M$, such that $L(M) = L.$ However, every FSM is also an all-NFSA, since for every string $w$, there is only a single path in $M$.
   Assume there exists an all-NFSA $M$ such that $L(M) = L.$ We can convert $M$ into an FSM $M'$ using exactly the same algorithm we learned in class with one difference: we make a state $q$ in $M'$ accepting iff every element of it is an accepting state in $M.$ Formally, we build $M' = (K', \Sigma, \delta', s', A')$ as follows:
Consider the language

$$L = \{a^k b^i c^j : i, j, k \geq 0 \text{ and if } k = 1 \text{ then } j = i^2\}.$$ 

1. Prove that $L$ satisfies the conditions of the Pumping Theorem.

**SOLUTION:** We need to show that there exists a constant $k$ such that for every string $w \in L$, $|w| \geq k$, there exists a way of writing $w$ as $w = xyz$ such that $|xy| \leq k$, $|y| \geq 1$, and $xy^iz \in L$ for every $i \geq 0$.

Notice that $L = (a^+ \cup \varepsilon)b^*c^* \cup \{ab^ic^j : i \geq 0\}$. The first part is a regular expression, hence corresponds to a regular language, which satisfies the Pumping Theorem. We now prove that any string $w$ in the second part also satisfies the conditions of the Pumping Theorem. Let $k$ be the constant and let $w = ab^\ell c^{\ell_2}$, where $\ell \geq 1$ and $|w| \geq k$. Then take $x = \varepsilon$, $y = a$, and $z = b^\ell c^{\ell_2}$. For $i = 1$, we have $w = ab^\ell c^{\ell_2}$ which is in $L$. For $i \neq 1$, we have $w = b^\ell c^{\ell_2}$ or $w = a^p b^\ell c^{\ell_2}$, where $p \geq 2$, and in both cases, $w \in (a^+ \cup \varepsilon)b^*c^*$, and hence in $L$.

2. Prove that $L$ is not regular.

**SOLUTION:** Let $L' = L \cap ab^*c^*$; then $L' = \{ab^ic^j : i \geq 0\}$. You can prove $L'$ is not regular by using the Pumping Theorem directly; some also proved that $L'^R$ is not regular, and hence $L'$ is not regular. Either way, you get $L'$ to be not regular, and by closure under intersection, we conclude that $L$ is not regular. You are expected to give the full proof, although I am giving a sketch of a solution here.
3. Do parts (1) and (2) contradict the Pumping Theorem? Explain your answer.

**SOLUTION:** No. The Pumping Theorem states what conditions hold if a language is regular. It does not say anything about these conditions if the language is not regular. So, there is no contradiction.

**Problem 4**

Given the following automaton $M$, describe an automaton that accepts $L(M)$.

![Automaton Diagram]

**SOLUTION:** we first convert that NFSM into an FSM, and then swap the accepting and non-accepting states. An automaton for the problem is the following: