Solutions to Quiz 1

Problem 1 [3 points]

Consider the NFA depicted below. Circle the correct choice of YES or NO in each line of the table below. No explanations or computations are required. You will get 1 point for each correct answer, and 0 point for missing or wrong answer.

<table>
<thead>
<tr>
<th>$w$</th>
<th>Does NFA $M$ accept $w$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>YES</td>
</tr>
<tr>
<td>0101</td>
<td>YES</td>
</tr>
<tr>
<td>000111</td>
<td>YES</td>
</tr>
<tr>
<td>001</td>
<td>YES</td>
</tr>
<tr>
<td>0111</td>
<td>YES</td>
</tr>
<tr>
<td>001100</td>
<td>YES</td>
</tr>
</tbody>
</table>

Figure 1: NFA $M$
Problem 2 [5 points]

Give a DFA for the language of all strings $w \in \{a, b\}^*$ such that $w$ ends with an $a$ and does not contain the substring $aa$. For example, $abab$ is in $L$, but $abaabb$ and $aba$ is not.

![DFA Diagram]

Problem 3 [5 points]

Give a regular expression for the language $L$ of all strings $w \in \{0, 1\}^*$ such that string $w$ begins or ends with 00 or 11. For example, 00101 and 011 are in $L$, but 010 is not in $L$.

$$(00 \cup 11)(0 \cup 1)^* \cup (0 \cup 1)^*(00 \cup 11)$$

Problem 4 [5 points]

Transform the following regular expression into an equivalent NFA using the procedure studied in class. (You may omit some of the redundant $\epsilon$ transitions, but your automaton should have a structure that closely resembles the regular expression.)

$$((a \cup ba)(ab)^*ba)^*$$

![NFA Diagram]
Problem 5 [6 points]

Transform the following NFA into an equivalent DFA using the procedure studied in class. States of the DFA should be labeled by sets of states of the original NFA. You only need to give the state diagram of the DFA.

The corresponding DFA is:
Problem 6 [5 points]

Transform the following automaton into an equivalent regular expression using the procedure studied in class. You should remove the states in alphabetical order A, B, C.

The corresponding regular expression is: \(0(10)^*(1 \cup 10 \cup 11)\).

Problem 7 [6 points]

Prove that the language \(L = \{a^m b^n : m \geq 2n \geq 0\}\) is not regular using the pumping lemma.

**Proof:** First, notice this is the language of all words form by concatenating \(a\)'s followed by \(b\)'s where the number of \(a\)'s is at least twice as many the number of \(b\)'s. Assume by contradiction that \(L\) is regular. Then, by PL, we know that there exists a pumping length \(p > 0\) such that any word \(w \in L, |w| \geq p\) can be partitioned as \(xyz = w\) (with \(|xy| \leq p\) and \(|y| > 0\)) in such a way that, for any \(i \geq 0\), the word \(xy^i z\) also belongs to \(L\). We consider the word \(a^{2p} b^p\) which belongs to \(L\).

Clearly \(|w| \geq p\). Moreover, any partition of \(w\) into \(xyz\) must be such that \(y\) comprises only \(a\)'s (since \(|xy| \leq p\)). Then, it must be the case that \(y = a^k\) for some \(k\) \(0 < k \leq p\) \((k > 0\) since \(|y| > 0\)). Now, consider the word \(w' = xy^i z\), for \(i = 0\). Clearly, \(w' = a^{2p-k} b^p\), and since \(2p - k < 2p\), \(w\) is a word that a number of \(a\)'s that is less than twice as many \(b\)'s. Therefore, \(w'\) is not in \(L\). Nevertheless, by PL, \(w' \in L\). We've got a contradiction. In consequence, it must be that our assumption that \(L\) is regular is false.