CIS 262: Fall 2008: Midterm 1 Solutions

Problem 1

Let $\Sigma = \{0, 1\}$. Consider the language L consisting of words that contain 010 (that is, the language given by $(0+1)^*010(0+1)^*$).

Draw a DFA A for L.

Answer:

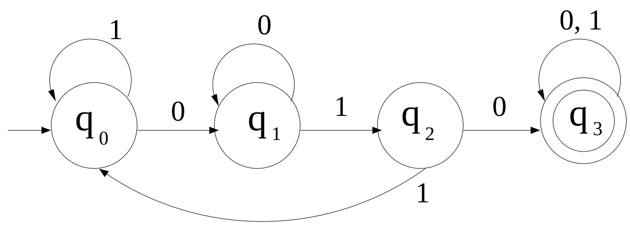

Figure 1: DFA for accepting words that contain 010.

For each state q of the DFA A in part (a), precisely describe the set of words that lead the DFA from its initial state to q.

Answer: Let $W(i)$ be the set of words that lead to the state q_i in the DFA in Figure 1, $0 \le i \le 3$. Note that $W(0) \cup W(1) \cup W(2) \cup W(3) = \Sigma^*$ and $W(i)$'s should be disjoint.

- $W(3) = L$, given a word w, $\hat{\delta}(q_0, w) = q_3$ if and only if w contains the substring 010.
- $W(2) = (0+1)^*01 - L$, all words w that end with 01 but do not contain substring 010.
- $W(1) = (0+1)^*0 - L$, all words w that end with 0 but do not contain substring 010.
- $W(0) = (0+1)^* - (W(3) \cup W(2) \cup W(1))$, everything else; i.e. $\{\varepsilon, 1\} \cup$ $\{w | w$ ends with 11 and w does not contain the substring 010 $\}$.

Problem 2

Let $\Sigma = \{a, b\}$. Let L_1 be the set of words w that contain an even number of a's. Let L_2 be the set of words w that end with b. Let $L_3 = L_1 \cap L_2$.

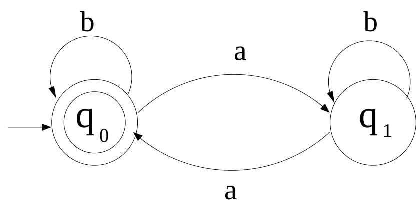

Figure 2: DFA A_1 for accepting words in L_1

For the DFA A_3, which states are equivalent (or indistinguishable) to each other?

Since q_0p_1 is the only accepting state, first we put 'x' in all pairs involving q_0p_1. Then we put 'x' for the pair $\langle q_1p_0, q_0p_0\rangle$, because on symbol b, q_1p_0 goes to q_1p_1 and q_0p_0 goes to q_0p_1, and the pair $\langle q_1p_1, q_0p_1\rangle$ already contains an 'x'. Similarly we put 'x' for the pair $\langle q_1p_1, q_0p_0\rangle$ (they are also distinguishable w.r.t

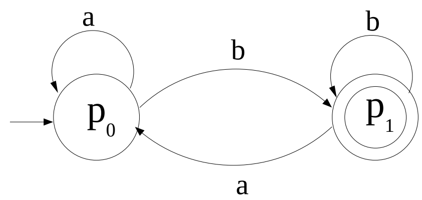

Figure 3: DFA A_2 for accepting words in L_2

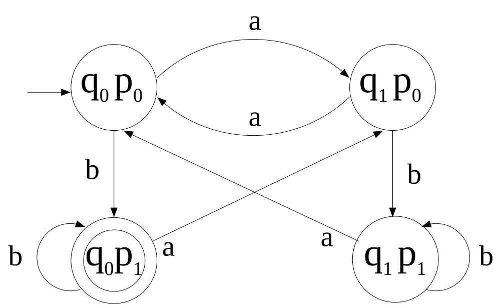

Figure 4: DFA A_3 by product construction for accepting words in $L_3 = L_1 \cap L_2$.

Table 1: Final output of table-filling algorithm on the DFA given in Figure 4

q_0p_0	-	-	-	-
q_0p_1	x	-	-	-
q_1p_0	x	x	-	-
q_1p_1	x	x		-
	q_0p_0	q_0p_1	q_1p_0	q_1p_1

symbol b). But we cannot put an 'x' for the pair $\langle q_1p_0, q_1p_1\rangle$ - on a they both go to state q_0p_0 and on b they both go to q_1p_1. Hence these two states are indistinguishable in the DFA given in Figure 4.

Use the answer in part (d), to produce the minimal DFA for L_3.

See Figure 5.

Problem 3

For each of the statements below, state whether the statement is True or False. In either case, give a short justification (one or two sentences).

1. The language of a DFA is empty if and only if the set of its final states is empty.

   Answer:

   False.

   If the set of the final states of a DFA is empty, then clearly the language accepted by the DFA is empty, but the converse is not true. If all of the final states of a DFA are not reachable from its initial state then the language is empty, even if the DFA has at least one final state (see Figure 6).

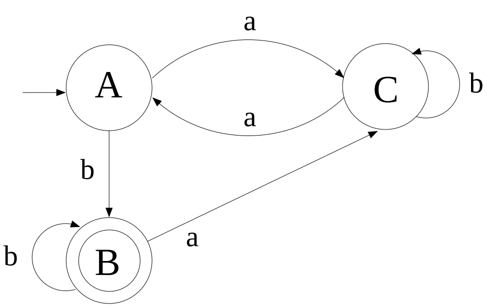

Figure 5: The minimal DFA for L_3: $A \equiv q_0p_0$, $B \equiv q_0p_1$, $C \equiv \{q_1p_0, q_1p_1\}$.

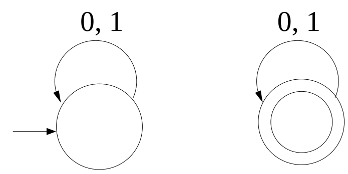

Figure 6: A DFA D on $\Sigma = \{0, 1\}$ with a final state but $L(D) = \phi$.

2. If A and B are DFAs such that $L(A) = L(B)$ then A and B must have the same number of states.

   Answer: False.

   You can have many DFAs with different number of states for a language. For example in Problem 2, the DFAs in Figure 4 and Figure 5 both accept L_3, but they have four and three states respectively. Also given any DFA for a language, you can add as many 'dummy' states as you want which are not reachable from any state of the original DFA, and the DFA will accept the same language.

3. Regular languages are closed under the operation of set difference.

   Answer:

   True.

   If A and B are two sets, then $A - B = \{x | x \in A, x \notin B\}$. Given any two sets A and B, it is easy to see that $A - B = A \cap \bar{B}$, where $\bar{B}$ denotes the complement of the set B.

   Now consider any two regular languages L_1 and L_2. Since regular languages are closed under complementation, $\bar{L_2}$ is regular, and since regular languages are closed under intersection, $L_1 \cap \bar{L_2}$ is regular. Hence $L_1 - L_2$ is also regular, i.e. regular languages are closed under the set difference.

Problem 4

A word w is a *palindrome* if it is its own reverse (that is, $w = w^R$). For example, 010 is a palindrome, but 001 is not. Let $\Sigma = \{0, 1\}$. Prove that the set of palindromes is not regular using the Pumping Lemma.

Answer:

Let us denote the language $\{w | w$ is a palindrome $\}$ as L. Assume the contradiction that L is regular. Then the pumping lemma holds for L and L has a pumping constant, say n.

Now let us cosider the word $w = 0^n10^n$. Clearly $|w| \geq n$ and $w \in L$. Then according to the pumping lemma, we can break $w = xyz$ such that $|xy| \leq n, y \neq \varepsilon$, and for all $k \geq 0$, $xy^kz \in L$.

Now consider any split of w, say $w = xyz$, such that $|xy| \leq n, y \neq \varepsilon$. Since $|xy| \leq n$, the substring y should lie entirely within the first n 1's. Suppose $x = 1^i, y = 1^j$ and $z = 1^{n-i-j}01^n$ ($|xy| = i+j \leq n$, $j \geq 1$). Now consider the word xz, i.e. when $k = 0$. $xz = 1^i 1^{n-i-j}01^n = 1^{n-j}01^n$. Since $j \geq 1$, $n - j \neq n$, i.e. xz is not a palindrome and therefore $xz \notin L$. This is a contradiction to our assumption that L is regular. Hence the set of all palindromes is not regular.

Problem 5

For two words y and w, we say that y is a *subword* of w, if $w = xyz$ for some words x and z. For example, the word 011 has following subwords: ε, 0, 1, 01, 11, and 011. For a language L, the set of its subwords, denoted *subwords*(L), contains all the subwords of the words in L. That is, a word y belongs to *subwords*(L) precisely when there exists a word w in L such that y is a subword of w.

1. If $L = 0^+10^*$, write the regular expression that captures *subwords*(L).

   Answer: Note that any word in *subwords*(L) can be of the following types:

   - the subword entirely belongs to the leading 0-s, in this case the regular expression is 0^*.
   - the subword takes zero or more 0-s from the leading 0-s, the 1, and zero or more trailing 0-s; regular expression: 0^*10^*.
   - the subword entirely belongs to the trailing 0-s, again in this case the regular expression is 0^*.

   So the final regular expression: $0^* + 0^*10^*$.

2. Let L be a language, and let A be a DFA for L with states Q, initial state q_0, final states F, and transition function δ. Define an automaton B that accepts *subwords*(L). The automaton B can be a DFA or an NFA or an ε-NFA.

   Answer: We will construct an ε-NFA B for *subwords*(L) from A. The set of states of B is Q and an additional state s. The initial state is s. A state q is final for B, if there is a path from q to a state in F. More precisely, the set of final states F' for B is all states $q \in Q$ such that $\hat{\delta}(q, u) \in F$ for some word u. Finally, the transitions of B are all transitions of A, and ε-transitions from the initial state s to every state reachable from q_0 in A. That is, the new transition function δ' is defined by $q' \in \delta'(q, a)$ iff either $a \in \Sigma$ and $q \in Q$ and $\delta(q, a) = q'$ or $a = \varepsilon$ and $q = s$ and $q' = \hat{\delta}(q, x)$ for some word x.

3. Prove that B accepts a word if and only if it belongs to *subwords*(L).

   Answer: Consider a word w in *subwords*(L). Then there exist words x and y such that xwy is in L. Let $\hat{\delta}(q_0, x) = q_1$, $\hat{\delta}(q_1, w) = q_2$, and $\hat{\delta}(q_2, y) = q_3$. By definition of δ', there is an ε-transition from s to q_1. Also, $q_2 \in \hat{\delta}'(q_1, w)$. Hence, $q_2 \in \hat{\delta}'(s, w)$. Since $q_3 \in F$, $q_2 \in F'$, and hence B accepts w.

   Consider a word w accepted by B. From definition of B, there exists a state q_1 such that there is an ε-transition from s to q_1, and $\hat{\delta}(q_1, w) = q_2$ with $q_2 \in F'$. By definition of B, there is a word y such that $\hat{\delta}(q_2, y) \in F$, and a word x such that $\hat{\delta}(q_0, x) = q_1$. As a result A accepts xwy, and hence, w is in *subwords*(L).