Assignment 3 (280 points)

Please solve the following problems:

1. (20 points) Consider the function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ defined by:

   - $f(\epsilon) = \epsilon$, $f(0) = 0$, $f(1) = 1$;
   - For $w_i \in \{0,1\}$ and $k \geq 2$, $f(w_1 w_2 \dots w_k) = w_1 w_2 w_1 w_3 \dots w_1 w_k w_1$.

   Determine whether or not f is a DGSM function and prove it.

   Solution sketch: the function f is extensible because if $x = \lambda_1\lambda_2\dots\lambda_k$ and $y = \sigma_1\sigma_2\dots\sigma_m$ then
   $f(xy) = f(\lambda_1\lambda_2\dots\lambda_k\sigma_1\sigma_2\dots\sigma_m)\lambda_1\lambda_1\lambda_1\lambda_3\lambda_1\dots\lambda_1\lambda_k\sigma_1\lambda_1\sigma_2\lambda_1\dots\lambda_1\sigma_m\lambda_1 =$
   $f(\lambda_1\lambda_2\dots\lambda_k)\sigma_1\lambda_1\sigma_2\lambda_1\dots\lambda_1\sigma_m\lambda_1 = f(x)z$

   For the derived function f_0 we have:
   $f(\lambda_1\lambda_2\dots\lambda_k) = f(0)f_0(\lambda_1\lambda_2\dots\lambda_k) = 0f_0(\lambda_1\lambda_2\dots\lambda_k)$,
   and by its definition
   $f(0\lambda_1\lambda_2\dots\lambda_k) = 0\lambda_1 0\lambda_2 \dots 0\lambda_k 0$ so that $f_0(\lambda_1\lambda_2\dots\lambda_k) = \lambda_1 0\lambda_2 0\dots 0\lambda_k 0$.
   Hence $f_0(\lambda_1\lambda_2\dots\lambda_k) = \lambda_1 0\lambda_2 0\dots 0\lambda_k 0$. For any other input $0x$, $f(0x\lambda_1\lambda_2\dots\lambda_k) = f_{0x}(\lambda_1\lambda_2\dots\lambda_k) = f(0)f_0(x)f_{0x}(\lambda_1\lambda_2\dots\lambda_k)$ but by definition
   $f(0x\lambda_1\lambda_2\dots\lambda_k) = f(0)f_0(x)\lambda_1 0\lambda_2 0\dots 0\lambda_k 0$.
   Hence, $f_{0x}(\lambda_1\lambda_2\dots\lambda_k) = \lambda_1 0\lambda_2 0\dots 0\lambda_k 0$ and $f_0 = f_{0x}$. Similarly, $f_1 = f_{1x}$, and thus f has index 3. The three functions f_ϵi, f_0, f_1 give rise to a 3 state DGSM to realize f.

2. (20 points) Let $\Sigma = \{0,1\}$ be an alphabet. Give regular expressions that generate the following languages over Σ:
   $L_1 = \{w \,|\, |w| \leq 5\}$
   Answer: $RE_1 = (\epsilon \cup \Sigma)(\epsilon \cup \Sigma)(\epsilon \cup \Sigma)(\epsilon \cup \Sigma)(\epsilon \cup \Sigma)$
   $L_2 = \{w | w$ starts with `0` & has odd length, or starts with `1` & has even length $\}$
   Answer: $RE_2 = (0\Sigma \cup 1)(\Sigma\Sigma)^*$
   $L_3 = \{w | w$ does not contain the substring `110` $\}$
   Answer: $RE_3 = (0 \cup (10)^*)^* 1^*$

3. (20 points) In certain programming languages, comments appear between delimiters such as $/*$ and $*/$ in C. Let L be the language of all valid delimited comment strings. A member of L must begin with /# and must end with #/ but have no intervening #/. For simplicity, we say that comments themselves are written with only the symbols `a, b`. Hence, the alphabet of L is $\Sigma = \{\texttt{a}, \texttt{b}, /, \#\}$.

   (a) Give a DFA that recognizes L (10 points).

   (b) Give a regular expression that generates L (10 points).

   Solution sketch: (a) obvious from work done so far, (b) $/\#(/^*(a \cup b) \cup \#)^*/^+\#$.

4. (20 points) Let F be the language of all strings over $\{0,1\}$ that do not contain a pair of 1s that are separated by an odd number of symbols. Give the state diagram of the DFA with 5 states that recognize F.

   Hint: you may find it helpful first to find a 4-state NFA for the complement of F.

   Solution sketch: The formal definition of languages F and $\bar{F}$ are:
   $F = \{w \in \{0,1\}^* | w \neq x1y1z \ for \ z, y, z \in \{0,1\}^* \wedge |y| \neq 2k+1, k \geq 0\}$
   $\bar{F} = \{w \in \{0,1\}^* | w = x1y1z \ for z, y, z \in \{0,1\}^* \wedge |y| = 2k+1, k \geq 0\}$
   An NFA that recognizes $\bar{F}$ is in Figure1.

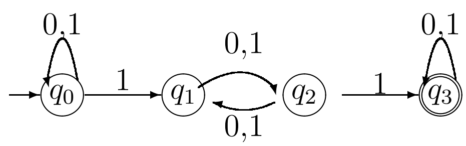
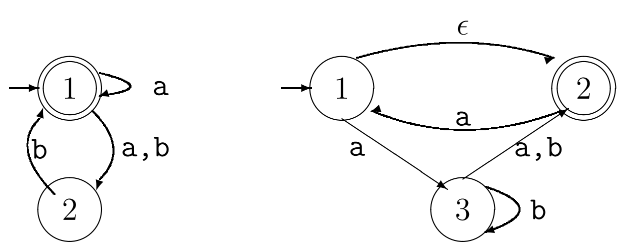


Figure 1: An NFA recognizing $\bar{F}$

   To construct a DFA for F one should proceed as follows:

   (a) Transform the NFA in Figure 1 into the DFA D_1
   (b) Map the DFA D_1 into a DFA D_2 with a minimum number of states
   (c) Swap the accept and unaccept states of D_2 thus obtaining the DFA D_3 that recognizes the language F.

5. (20 points) Use the construction which show that every NFA has an equivalent DFA to convert the two NFAs whose transition diagrams are in the figure below in to equivalent DFA.



   Solution: obvious, following the construction indicated in the problem.

6. (20 points) Let $\Sigma_2 = \{\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\}$. Here Σ_2 contains all columns of 0s and 1s of height two. A string of symbols in Σ_2 gives two rows of 0s and 1s. Consider each row to be a binary number and let $C = \{w \in \Sigma_2^* |$ the bottom row

of w is three times the top wow }. For example $\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \in C$ but $\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} \notin C$. Show that C is regular.

Hint: you may assume that for any language A if A is regular then the language $A^{\mathcal{R}} = \{w^{\mathcal{R}} | w \in A\}$ is also regular. Thus, you may design an automaton which read the input backwards. The symbols it reads are columns of the form $\begin{bmatrix}d_1\\d_2\end{bmatrix}$, $d_1, d_2 \in \{0, 1\}$. The states must be such that if binary sum $d_1 + d_1 + d_1$ is not d_2 (having in view the possible overflow) than the automaton reject, otherwise it performs further the addition of the next digits.

7. (20 points) Let $\Sigma_2 = \{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\}$. Here Σ_2 contains all columns of 0s and 1s of height two. A string of symbols in Σ_2 gives two rows of 0s and 1s. Consider the top and bottom rows to be strings of 0s and 1s and let $E = \{w \in \Sigma_2^* |$ the bottom row of w is the reverse of top row of w }. Show that E is not a regular language.

   Solution sketch: Assume that E is regular. Use pumping lemma to get a pumping length p that satisfies the conditions of the pumping lemma. Set $s = \begin{bmatrix}0\\1\end{bmatrix}^p \begin{bmatrix}1\\0\end{bmatrix}^p$. Obviously $s \in E$ and $|s| \geq p$. Thus the pumping lemma implies that the string s can be written as xyz with $x = \begin{bmatrix}0\\1\end{bmatrix}^a$, $y = \begin{bmatrix}0\\1\end{bmatrix}^b$, $z = \begin{bmatrix}0\\1\end{bmatrix}^c \begin{bmatrix}1\\0\end{bmatrix}^p$, where $b \geq 1$, and $a + b + c = p$. However, the string $s' = xy^0z = \begin{bmatrix}0\\1\end{bmatrix}^{a+c} \begin{bmatrix}1\\0\end{bmatrix}^p \notin E$ since $a + c < p$. This contradicts the pumping lemma and thus E is not regular.

8. (20 points) Let $B_n = \{a^k |$ where k is a multiple of n }. Show that for each $n \geq 1$, the language B_n is regular.

   Solution sketch: For each $n > 1$ we build a DFA M with the states $q_0, q_1, \ldots, q_{n-1}$ to count the number of consecutive a-s modulo n read so far. For each symbol a that is input, the counter increments by 1 and jumps to the next state in M. It accepts the string iff the machine stops at q_0. That means that the length of the string is a multiple of n. More formally, $M = (Q, \{a\}, \delta, q_0, \{q_0\})$ where $Q = \{q_0, q_1, \ldots, q_{n-1}\}$ and $\delta(q_i, a) = q_j$ where $j = i + 1 \ mod \ n$

9. (20 points) Let $C_n = \{x | x$ is a binary number that is a multiple of $n\}$. Show that for each $n \geq 1$ the language C_n is regular.

   Solution sketch: We can create a DFA M with n states that recognizes C_n by simulating binary division. M has n states to keep track of the n possible reminders of the division process. The start state is the only accept state and corresponds to the reminder 0. The input string is fed into M starting from the most significant bit. For each input bit, M doubles the reminder that the current state records, and then adds the input bit. Its new state is the sum modulo n. We double the reminder because that corresponds to the left shift of the computed reminder in the long division algorithm. If the input string ends at the accept state (corresponding to the reminder 0), the binary number has no reminder on division by n and therefore it is a member of C_n. The formal definition of M is: $M = (\{q_0, q_1, \ldots, q_{n-1}\}, \{0,1\}, \delta, q_0, \{q_0\})$ where for each $q_i \in Q$ and $b \in \{0,1\}$, $\delta(q_i, b) = q_j$ where $j = (2i + b) \; mod \; n$.

10. (30 points) Prove that the following languages are not regular. You may use the pumping lemma and the closure of the class of regular languages under union, intersection, and complement.

    (a) $L_1 = \{0^n 1^m 0^n | m, n \geq 0\}$ (15 points).

    Solution sketch: Assume that $L = \{0^n 1^m 0^n | m, n \geq 0\}$ is regular. Let p be the pumping length given by pumping lemma for L. The string $s = 0^p 1 0^p \in L$ and $|s| \geq p$. Thus, pumping lemma implies that s can be pumped, i.e., $s = xyz$ such that $xy^i z \in L_1$ for $i \geq 0$, $|y| > 0$, $|xy| \leq p$. Consider the division $x = 0^a$, $y = 0^b$, $z = 0^c 1 0^p$, where $b \geq 1$ and $a + b + c = p$. Then by pumping s down we get $s' = xy^0 z = 0^{a+c} 1 0^p \notin L_1$ because $a + c < p$. This contradicts the pumping lemma, and therefore L_1 is not regular.

    (b) $L_2 = \{w | w \in \{0,1\}^*$ is not a palindrome $\}$ (15 points).
    Note: a palindrome is a string that reads the same forward and backward. For example `anna` is a palindrome.

    Solution sketch: Assume that $\overline{L_2} = \{w | w \in \{0,1\}^*$ and w is a palindrome $\}$ is regular. Let p be the pumping length given by pumping lemma. The string $s = 0^p 1 0^p \in \overline{L_2}$ and $|s| \geq p$. Following the argument used for language L_1 above we get a contradiction, which ensures that $\overline{L_2}$ is not regular. But $\overline{L_2}$ is the compliment of L_2 which is a closure operator over regular languages. Thus, L_2 is not regular.

    (c) $L_3 = \{wtw | w, t \in \{0,1\}^+$ (15 points).

    Solution sketch: L_3 is similar to L_2.Hence, a similar argument can be used to show that it is not regular.

11. (20 points) Let $\Sigma = \{1, \#\}$ and consider the language $Y = \{w | w = x_1 \# x_2 \# \ldots \# x_k$ for $k \geq 0$, each $x_i \in 1^*$, and $x_i \neq x_j$ for $i \neq j\}$. Prove that Y is not regular.

    Solution sketch:

Sol 1: Observe that $\overline{Y} \cap 1^* \# 1^* = \{1^n \# 1^n \| n \geq 0\}$. The language $\{1^n \# 1^n | n \geq 0\}$ is not regular (as one can shown using the pumping lemma). On the other hand $1^* \# 1^*$ is a regular language and $\cap$ is a closure operator over the regular languages. In addition, if L is regular then $\overline{L}$ is also regular. Therefore Y is not regular.

Sol 2: Assume to the contrary that Y is a regular language and let p be the pumping length provided by pumping lemma for regular languages. Let $s = 1^{p!}\#1^{p!} \in Y$s and $|s| \geq p$. Then the pumping lemma says that s can be split into $s = xyz$ such that $xy^iz \in Y$ for all $i \geq 0$, $|y| > 0$, and $|xy| \leq p$. Because $|1^{p!}\#| \geq p$ the condition $|xy| \leq p$ implies that y is among the `1`-s in the left-side of $\#$. Let $l = |y|$ and let $k = (p!/l)$. Obviously k is an integer because $l \leq p$ and thus it is a divisor of $p!$. Therefore adding $xy^{k+1}z$ adds $p!$ copies of `1` to the left of $\#$ getting the string $1^{2p!}\#1^ip!i \notin Y$ which violate the condition $xy^iz \in Y$ for any $i \geq 0$.

12. (20 points) For each of the following languages give the Minimum Pumping Length (MPL) and justify your answer:

    (a) (5 points) $L_1 = 001 \cup 0^*1^*$.
    Solution sketch: $MPL(L_1) = 1$. MPL cannot be zero because $\epsilon \in L_1$ and ϵ cannot be pumped. Every nonempty string $s \in L_1$ can be pumped by the division: $x = \epsilon$, y is the first symbol of s and z is the rest of s.

    (b) (5 points) $L_2 = 1^*01^*01^*$.
    Solution sketch: $MPL(L_2) = 3$ because $00 \in L_2$ but 00 cannot be pumped. Every $s \in L_2$, $|s| \geq 3$i contains a `1` among its first three symbols, and so it can be pumped by the division xyz where y is the first symbol `1` in s, x the symbols to the left of first `1` of s, and z the symbols to the right of first `1` of s.

    (c) (5 points) $L_3 = 1011$.
    Solution sketch: $MPL(1011) = 5$ because $1011 \in L_3$ and cannot be pumped. Every string in the language of length 5 or more (there are none) can be pumped (vacuously).

    (d) (5 points) $L_4 = \Sigma^*$.
    Solution sketch: $MPL(L_4) = 1$ because $\epsilon \in L_4$ and cannot be pumped. However any string $s \in L_4$, $|s| \geq 1$, can be pumped by the division $s = xyz$, $x = \epsilon$, y first symbol of s and z the rest of the symbols of s.

Due date: 14 October 2009.