Solution sketch 1 - Computational Models - Spring 2011-2012

As always, these solutions might contain mistakes. Please let us know if you found any!

1. Not given.

2. (a) DFA: $A = \langle Q, \{0,1\}, \delta, 0^n, F\rangle$, where: $Q = \{x_1x_2\dots x_n \mid \forall i.x_i \in \{0,1\}\}$, $F = \{1x_2\dots x_n \mid \forall i.x_i \in \{0,1\}\}$, and $\forall x_1x_2\dots x_n \in Q, \sigma \in \Sigma, \delta(x_1x_2\dots x_n, \sigma) = x_2\dots x_n\sigma$.

   (b) NFA: $A = \langle\{q_0, q_1, \dots, q_n\}, \{0,1\}, \delta, q_0, \{q_n\}\rangle$, where: $\delta(q_0, 0) = \{q_0\}$, $\delta(q_0, 1) = \{q_0, q_1\}$, and $\forall 1 \leq i \leq n, \sigma \in \Sigma, \delta(q_i, \sigma) = \{q_{i+1}\}$.

   (c) Regular expression: $(0 \cup 1)^* \circ 1 \circ (0 \cup 1)^{n-1}$, where $(0 \cup 1)^0 = \epsilon$, and $(0 \cup 1)^{n+1} = (0 \cup 1) \circ (0 \cup 1)^n$ for every $n \geq 0$ (Note that $(0 \cup 1)^n$ is not a part of the regular expression! We define and use it here to define a general form of the required regular expressions).

3. $A = \langle Q, \{a,b\}, \delta, q_0, \{q_{ab}, q_{ba}\}\rangle$, where $Q = \{q_0, q_{aa}, q_{bb}, q_{ab}, q_{ba}\}$ and $\forall \sigma \in \{a,b\}.\delta(q_0, \sigma) = q_{\sigma\sigma}$, $\forall \sigma_1, \sigma_2, \sigma \in \{a,b\}.\delta(q_{\sigma_1\sigma_2}, \sigma) = q_{\sigma_1\sigma}$.

   Next, we prove correctness. We describe the sets of words w that cause the DFA to end in each of the 5 states (starting from the initial state q_0):

   - state q_0: the empty word.
   - state q_{aa}: all words that begin in a and end in a.
   - state q_{bb}: all words that begin in b and end in b.
   - state q_{ab}: all words that begin in a and end in b.
   - state q_{ba}: all words that begin in b and end in a.

   Note that these properties are disjoint and cover all possible words, so all we have to do to show the correctness of the DFA is prove that if we run the DFA on w then it will terminate in state $q \in Q$ *if* w satisfies the property ascribed to q above. (If some words were not covered, then we would have to prove the *only if* direction as well, as it might be possible for the DFA to accept more strings than we claim it does.) The proof is by induction on the length of the word w.

   Base case: $|w| = 0$, that is, $w = \epsilon$. The DFA terminates in state q_0, whose property is satisfied by ϵ. The definitions for the other four states are satisfied vacuously.

   Induction step: We make the induction hypothesis that our definitions hold for $w' \in \{a,b\}^n$ for $n \geq 0$. We prove that they also hold for $w = w\sigma$ with $\sigma \in \{a,b\}$. We consider the states in which the DFA might have been in after processing w:

   - state q_0: By the induction hypothesis $w' = \epsilon$ (otherwise, after processing w' the DFA would have been in a state different from q_0). If $\sigma = a$ then the DFA advances to state q_{aa}. $w'a = a$ now begins in a and ends in a, so the property for q_{aa} is met. Similar proof for $\sigma = b$.
   - state q_{aa}: By the induction hypothesis w' begins in a and ends in a (otherwise, after processing w' the DFA would have been in a state different from q_{aa}). If $\sigma = a$ then the DFA remains in state q_{aa}. $w'a$ now begins in a and ends in a as well, so the property for q_{aa} is met. If $\sigma = b$ then the DFA advances to state q_{ab}. $w'a$ now begins in a and ends in b, so the property for q_{ab} is met.

- state q_{bb}: Similar to the previous case.
- state q_{ab}: By the induction hypothesis w' begins in a and ends in b (otherwise, after processing w' the DFA would have been in a state different from q_{ab}). If $\sigma = a$ then the DFA advances to state q_{aa}. $w'a$ now begins in a and ends in a, so the property for q_{aa} is met. If $\sigma = b$ then the DFA remains in state q_{ab}. $w'a$ now begins in a and ends in b, so the property for q_{ab} is met.
- state q_{ba}: Similar to the previous case.

In particular, we proved that the DFA ends in the accepting states q_{ab} or q_{ba} iff its input is a word that starts and ends in different letters. This completes the correctness proof.

4. Not given.

5. (a) $(0 \cup 1)(0 \cup 1)((0 \cup 1)(0 \cup 1)(0 \cup 1)(0 \cup 1))^*$

   (b) $1^*01^*01^*01^*$

   (c) $(0 \cup 1)^*111(0 \cup 1)^*$

6. (a) This language is $L\bar{L} \cup \bar{L}L$. Since L is regular and the regular languages are closed under the complement, union and concatenation, $L\bar{L} \cup \bar{L}L$ is also regular.

   (b) Let the desired language be L'. Let $A = \langle Q, \Sigma, \delta, q_0, F\rangle$ be a DFA accepting L. We construct an NFA: $N = \langle Q, \Sigma, \delta', q_0, F\rangle$, where $\forall q \in Q, \sigma \in \Sigma, \delta'(q, \sigma) = \{\delta(\delta(\delta(q, \sigma), a), b) \mid a, b \in \Sigma\}$ (thus, we simulate three steps in A: one with respect to to the character σ and two that we "guess" - that can use any two characters in Σ). You should prove that $L(N) = L'$. The key observation is that $\delta(q_0, x_1y_1y_2...x_ny_{2n-1}y_{2n}) = q$ (i.e. running w in A terminates in state q) iff there is a computation in N for $x_1x_2...x_n$ that ends in q.

   (c) Let the desired language be $Rot(L)$. Let $A = \langle Q, \Sigma, \delta, q_0, F\rangle$ be a DFA accepting L. For any state $q \in Q$ we build an NFA N_q. N_q will deal with all the rotations occurring at state q. N_q will be composed of two copies of A, the starting state will be q in the first copy and the accepting states will comprise the state q in the second copy. We will add ϵ-transitions from any state in F in the first copy to q_0 in the second copy.

   Formally: $N_q = (Q \times \{1, 2\}, \Sigma, \delta_q, \langle q, 1\rangle, \{\langle q, 2\rangle\})$, where:
   - $\forall p \in Q, \sigma \in \Sigma, i \in \{1, 2\}.\delta_q(\langle p, i\rangle, \sigma) = \{\langle\delta(p, \sigma), i\rangle\}$.
   - $\forall p \in F.\delta_q(\langle p, 1\rangle, \epsilon)) = \{\langle q_0, 2\rangle\}$.
   - All other values of δ_q are $\emptyset$.

   Now, $\bigcup_{q\in Q} L(N_q) = Rot(L)$ (here you have to prove correctness). Since $\bigcup_{q\in Q} L(N_q)$ is a finite union of regular languages, $Rot(L)$ is a regular language.

7. $\forall w \in \Sigma^*, \#_{01}(w) - \#_{10}(w) \in \{-1, 0, 1\}$ (why?). Therefore, we need only finite memory to keep track of that and decide this language in a DFA. It is also possible to note that this language is $L(0(0\cup1)^*0 \cup 1(0\cup1)^*1)$ and is therefore regular.

8. Let $L' = \{xz \mid \exists y \in L_2.xyz \in L_1\}$. Let $A = \langle Q, \Sigma, \delta, q_0, F\rangle$ be some DFA such that $L(A) = L_1$. We construct an NFA N for L':

$$N = \langle Q \times \{1, 2\}, \Sigma, \delta', \langle q_0, 1\rangle, F \times \{2\}\rangle$$

where:

$$\forall q \in Q, \sigma \in \Sigma, i \in \{1, 2\}.\delta'(\langle q, i\rangle, \sigma) = \{\langle\delta(q, \sigma), i\rangle\}$$

$$\forall q \in Q.\delta'(\langle q, 1\rangle, \epsilon) = \{\langle p, 2\rangle \mid \exists w \in L_2.\delta(q, w) = p\}$$

and all other values of δ' are $\emptyset$. We leave the correctness proof for the reader.