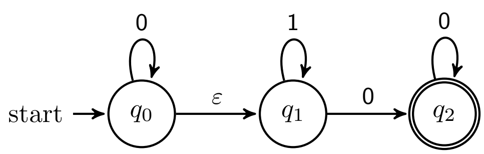
Homework 3

Aleck Zhao

February 20, 2018

1. Give an NFA (both a state diagram and a formal description) recognizing the language $0^*1^*0^+$ with three states. The alphabet is $\{0, 1\}$.

   Solution. The state diagram is given below:



The states are $Q = \{q_0, q_1, q_2\}$, the start state is q_0, the accept states are $F = \{q_2\}$, the alphabet is $\Sigma = \{0, 1\}$, and the transition function is given by

δ	0	1	ε
q_0	$\{q_0\}$	$\varnothing$	$\{q_1\}$
q_1	$\{q_2\}$	$\{q_1\}$	$\varnothing$
q_2	$\{q_2\}$	$\varnothing$	$\varnothing$

$\square$

2. This question studies the number of states in a DFA equivalent to an NFA. Recall that in class we showed an NFA with 4 states that recognizes the language which consists of all binary strings that have a 1 in the third position from the end. For any integer k, it is easy to generalize this construction to an NFA with $k+1$ states that recognizes the language which consists of all binary strings that have a 1 in the kth position from the end. The general transformation from an NFA too a DFA will give us a DFA with at most 2^{k+1} states recognizing the same language.

   Show that, any DFA that recognizes the same language must have at least 2^k states.

   Hint: start by looking at the following two strings: 10^{k-1} and 0^k. Observe that when a DFA takes them as inputs, it must end up at different states, since one string is accepted and the other is rejected.

   Proof. Fix k and consider two strings $a = a_1 a_2 \cdots a_k$ and $b = b_1 b_2 \cdots b_k$ of length k that are different. Then there must exist an index i where they differ. WLOG $a_i = 1$ and $b_i = 0$. Now let $c = 0^{i-1}$. Then ac is a string where the kth position from the end is a 1, and bc is a string where the kth position from the end is a 0. Thus, ac would be accepted while bc is not, so these strings must be different states.

   Thus, for any two strings of length k, there must be two distinct states in the DFA to account for the above process. Since there are 2^k strings of length k, there must be at least 2^k distinct states in the DFA. $\square$

3. Say that string x is a prefix of string y if a string z exists where $xz = y$ and that x is a proper prefix of y if in addition $x \neq y$. Let A be a regular language. Show that the class of regular languages is closed under the following operation.

$$\text{NOEXTEND}(A) = \{w \in A : w \text{ is not the proper prefix of any string in } A\}$$

Hint: Think about when a string $w \in A$ can be the proper prefix of another string in A, then modify the states of the machine to avoid this.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA recognizing A, since A is regular. We wish to construct an DFA N that accepts NOEXTEND(A).

Consider a string w that reaches an accept state $q \in F$. Then $w \in A$. If there exists a string x that reaches $p \in F$ from q, we know that w is the proper prefix of a string in A since wx is accepted. This path can be detected using a DFA. Now, let $F' \subset F$ be the subset of F such that there are no strings from F' to F. Then $N = (Q, \Sigma, \delta, q_0, F')$ is a DFA.

If $w \in$ NOEXTEND(A), then $w \in A$ so w is a path to some state in F, and this state is in F' because w is not the proper prefix of any other accepted string.

Conversely, if w is accepted by N, it by construction it must have been accepted by M, so $w \in A$, and w is only accepted if there is no path to another state in F, so if w is not the proper prefix of a string in A. Thus, N recognizes NOEXTEND(A), so it is regular. $\square$

4. Let $\Sigma = \{0, 1\}$.

   (a) Write a regular expression for the language L consisting of all strings in Σ^* with exactly one occurrence of the substring 000.

   Solution. The string 000 must be immediately surrounded by arbitrary(1^+0) and (1^+00) on the right, and (01^+) and (001^+) on the left, if there is anything to the left or right, respectively. Then since the string could start or end with arbitrary 1s, the regular expression is

$$1^* \left[(001^+) \cup (01^+)\right]^* 000 \left[(1^+0) \cup (1^+00)\right]^* 1^*$$

□

   (b) Write a regular expression for the language L consisting of all strings in Σ^* that do not end with 00.

   Solution. The string can have length 0 or 1. Otherwise, the string can only end in 01, 10, or 11, and the rest can be anything else, so the regular expression is

$$(0 \cup 1)^* (01 \cup 11 \cup 10) \cup (0 \cup 1 \cup \varepsilon)$$

□