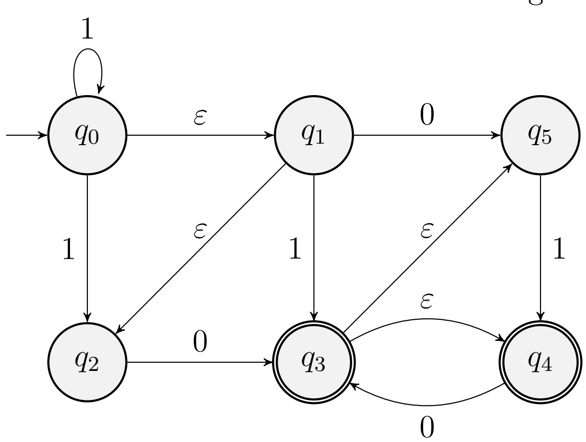
CS 361: Theory of Computation

Assignment 2 (due 09/17/2025)

Instructor: Shikha Singh

LaTeX Source for Solutions: https://www.overleaf.com/read/phmptsqjwyxk#a69fe2

Problem 1. Consider the NFA N in the figure below and fill in the blanks below:



(a) What are the ε-closures of the sets of states $\{q_0\}$ and $\{q_1, q_3\}$, that is, $E(\{q_0\})$ and $E(\{q_1, q_3\})$ as defined in the equivalence proof of DFAs and NFAs.

Solution. $E(\{q_0\}) =$ {Replace With Answer}

$E(\{q_1, q_3\}) =$ {Replace With Answer}

□

(b) What set of states is the output of $\delta(\{q_0\}, 0)$ and $\delta(\{q_2, q_3\}, 1)$.

Solution. $\delta(\{q_0\}, 0) =$ {Replace With Answer}

$\delta(\{q_2, q_3\}, 1) =$ {Replace With Answer}

□

(c) Does the NFA accept the strings $x = 011$ and $y = 101$?

Solution. The NFA accepts/rejects x.

The NFA accepts/rejects y.

□

Problem 2. For the two languages below, give state diagrams of NFAs (with the specified number of states) as well as correspoding regular expressions. Assume $\Sigma = \{0, 1\}$.

(a) The language $\{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$. Give an NFA for this language with six states.

(b) The language that contains a pair of 1s separated by an odd number of symbols (0s or 1s). Give an NFA with 4 states for this language.

Closure Under Operations. Below is a solved example to show that regular languages are closed under the reverse operation. Follow a similar approach to solve **Problem 4**.

Problem 3. (**Solved Example**) For any string $w = w_1 w_2 \ldots w_n$, the reverse of w, written w^R, is the string w in reverse order, that is, $w^R = w_n \ldots w_2 w_1$. For any language L, let $L^R = \{w^R \mid w \in L\}$. Show that regular languages are closed under the reverse operation, that is, show that if L is regular, so is L^R.

Solution. Let M be the DFA recognizing L. We need to construct an NFA that recognizes L^R. We keep all the states in M and reverse the direction of all the δ arrows in N. We set the accept state of N to be the start state in M. Also, we introduce a new state s as the start state for N which goes to every accept state in M be an ε-transition.

Formally, let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts the regular language L. Define $N = (Q \cup \{s\}, \Sigma, \delta', s, \{q_0\})$, where $s \notin Q$, and δ' is defined below:

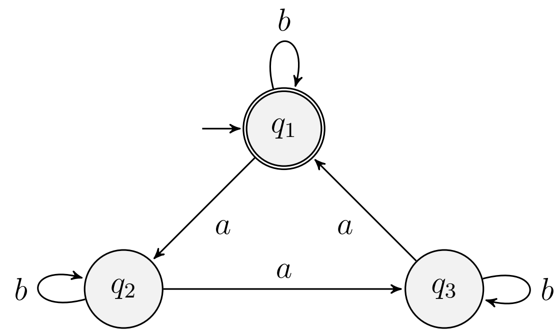
- $\delta'(s, \varepsilon) = F$;
- for all $q \in Q$ and for all $\sigma \in \Sigma, \delta'(q, \sigma) = \{p \in Q \mid \delta(p, \sigma) = q\}$.

To prove correctness, we need to prove $L^R = L(N)$. There are two directions: first, if $w^R \in L^R$ then $w^R \in L(N)$. Since $w^R \in L^R$, there exists a $w \in L$ such that w^R is the reverse of w. Thus, M accepts w: that is, there exists a sequence of states $q_0, q_1, \ldots, q_n$ in M such that $q_n \in F$ and each $q_{i+1} = \delta(q_i, w_{i+1})$. Since there is an ε-transition from s in N to q_n, on input $w^R = w_n \cdots w_1$, one computation branch in N starts at q_n and follows the reverse transitions of M, that is, on input $w_n \cdots w_i$ it goes through states $q_{n-1}, \ldots, q_0$. Since q_0 is an accept state of N, it accepts w^R. The other direction showing $w^R \in L(N) \implies w \in L(M) \implies w^R \in L^R$ is analogous. $\square$

Problem 4. Let $L \subseteq \Sigma^*$ be a regular language. Show that the following two languages are also regular.

$$\text{SUFFIXES}(L) = \{x \in \Sigma^* \mid yx \in L \text{ for some } y \in \Sigma^*\}$$
$$\text{PREFIXES}(L) = \{y \in \Sigma^* \mid yx \in L \text{ for some } x \in \Sigma^*\}$$

Problem 5. Use the state-elimination algorithm on a generalized non-deterministic finite automata (GNFA), (CONVERT(G), described on Page 73 in Sipser (Proof of Lemma 1.60) to convert the following finite automaton to a regular expression. Show your work as you eliminate each state. Refer to similar examples: Example 1.66 and 1.68 in the textbook (*You may attach a clear hand-drawn image of your work.*)



Solution.

□