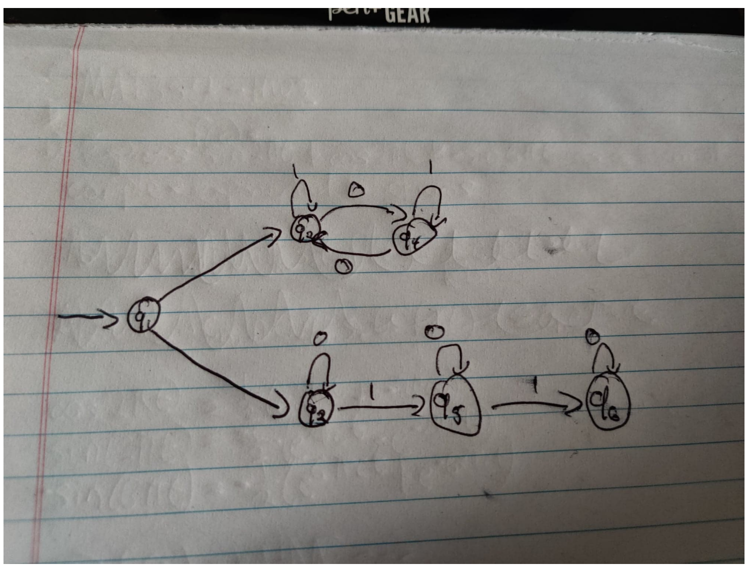
Computer Science 452 - Homework Assignment #2

Hari Amoor, NetID: hra25

February 4, 2020

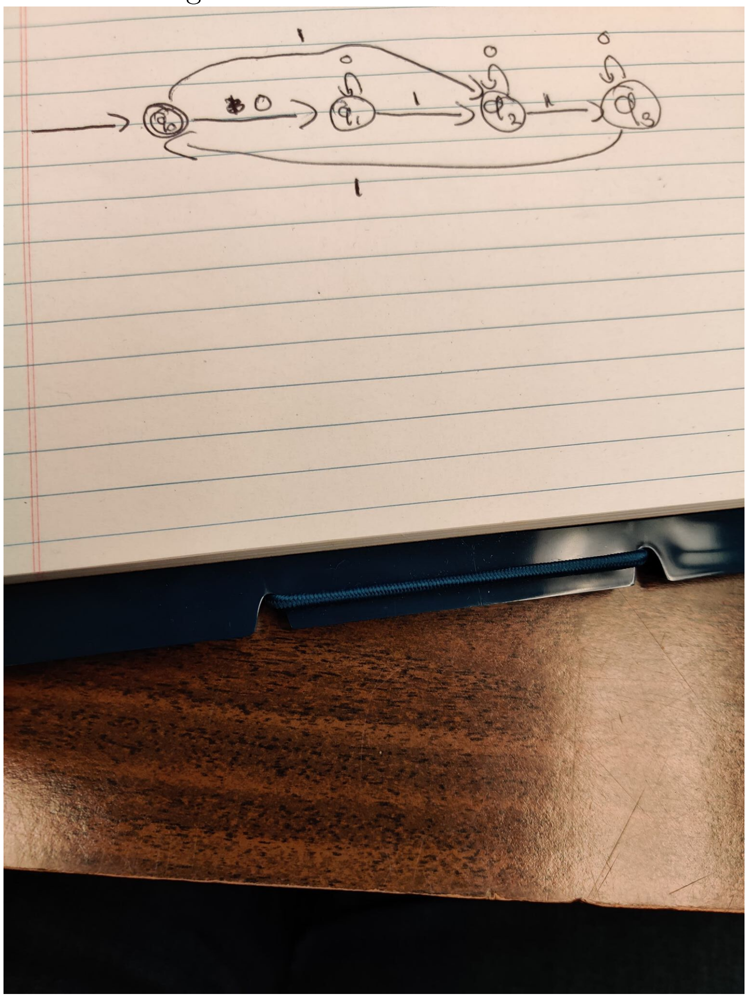
Problem 1.7c: Supply a state diagram for an NFA with six states that recognizes the language of words containing either an even number of 0s or at least two 1s.

The state diagram is as follows:



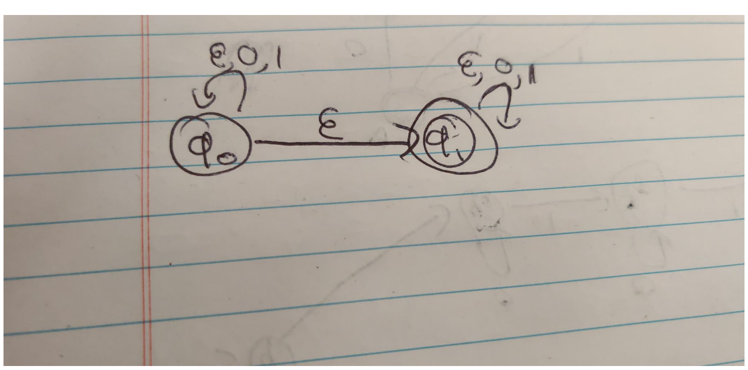
Problem 1.10a: Supply a state diagram for an NFA that recognizes $\{w* \mid w$ contains at least three 1s$\}$.

The state diagram is as follows:



Problem 1.14b: Show that swapping the accept and non-accept states of an NFA that recognizes a language L does not necessarily form an NFA that recognizes $L^{\complement}$. Is the class of languages recognized by NFAs closed under complement?

Consider the following NFA:

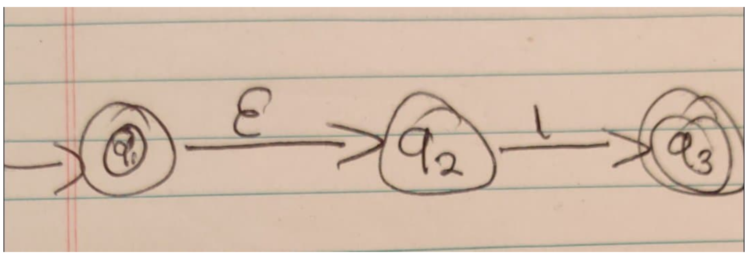


The string ε is recognized in both the given NFA and the NFA obtained with switching the accept and non-accept states of the given, as required. However, the class of languages recognized by NFAs is closed under complement, as demonstrated.

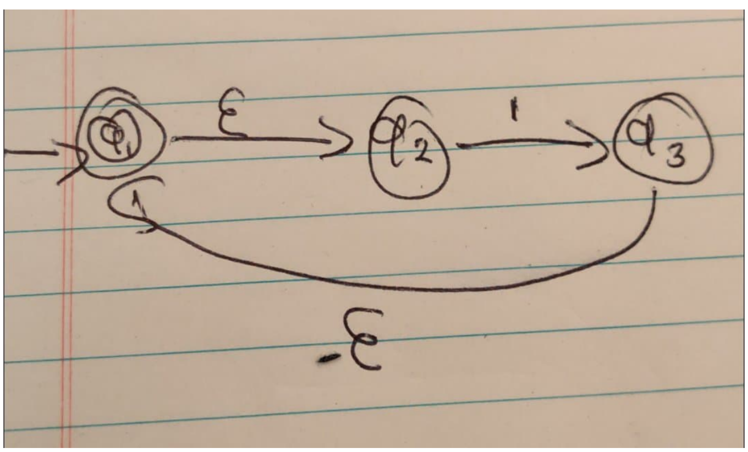
Proof. Let M be an NFA. We know that an equivalent DFA M' exists, i.e. $L(M) = L(M')$. Trivially, there must exist a DFA N that recognizes the complement of $L(M')$. N is an NFA that recognizes the complement of $L(M') = L(M)$, as required. $\square$

Problem 1.15: Show that the given construction does not prove that the set of languages recognized by an NFA is closed under the Kleene star.

Consider the following NFA:



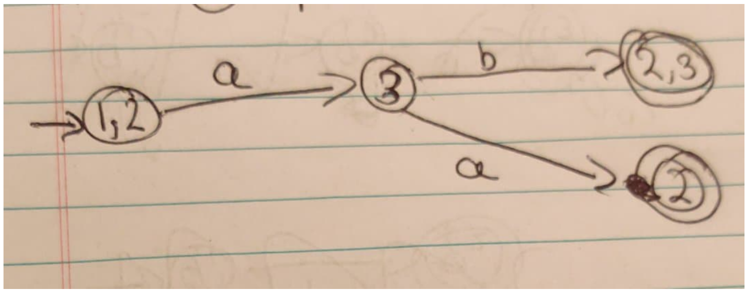
The NFA constructed with the given rules , is as follows:



Clearly, it fails at ε, and thus does not recognize the Kleene-closure of the language of the initial NFA.

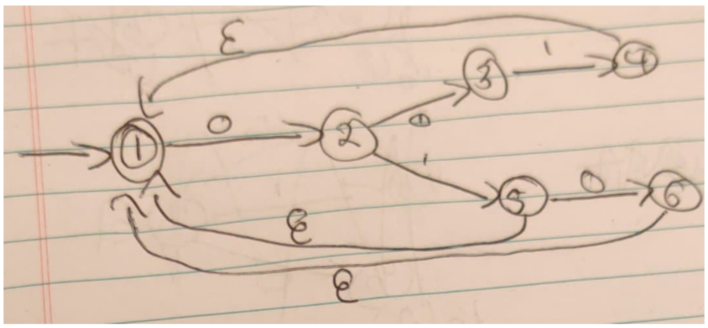
Problem 1.16b: Supply a DFA equivalent to the given NFA.

This is given below:

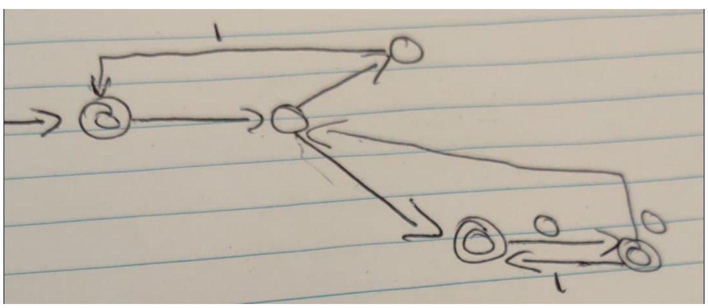


Problem 1.17: Complete the following.

(a) **Supply an NFA for the given language.**



(b) **Convert the NFA from (a) to a DFA.**



Problem 1.18: For parts (i) and (l), supply regular expressions detecting the given language.

For Part (i), the regular expression $(10 \cup 11)^*$ describes the language. For Part (l), the regular expression $((ab^*ab^*)^* \cup 01^*01^*)$ describes the given language.

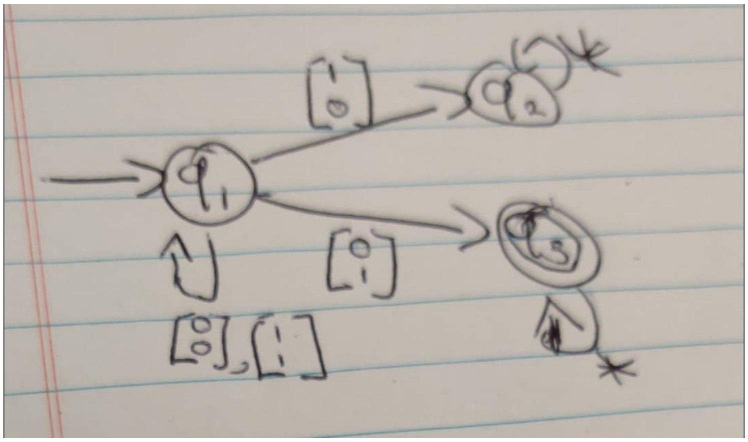
Problem 1.31, Claim: If A is regular, then so is A^R.

Proof. Let A be a regular language, and suppose the DFA M has $L(M) = A$. Suppose w.l.o.g. that $M = (Q, \Sigma, \delta, q_0, \{f\})$, i.e. M only has one accepting state. Then, the DFA $M' = (Q, \Sigma, \delta^R, \{f\}, \{q_0\})$ recognizes A^R. Here, δ^R is the transition relation defined by $\delta(q, a) \mapsto q$, i.e. the *reversed* transitions of δ.

M' as defined recognizes A^R, as required. $\square$

Problem 1.34: Show that the given language is regular.

The given language is recognized by the following DFA:



Problem 1.51, Claim: $\equiv_L$ is an equivalence relation.

Proof. We show that $\equiv_L$ is reflexive, symmetric and transitive.

Let x be a string and $w \in L$ be arbitrary for some language L. In general, either $xw \in L$ or $xw \notin L$. So, $x \equiv_L x$ for all strings x. Therefore, $\equiv_L$ is reflexive.

Let x, y be strings. Suppose that $x \equiv_L y$; then, for all $w \in L$, either $xw, yw \in L$ or $xw, yw \notin L$. Equivalently, $yw, xw \in L$ or $yw, xw \notin L$. Thus, $y \equiv_L x$, so $\equiv_L$ is symmetric.

Let x, y, z be strings, and suppose $x \equiv_L y$ and $y \equiv_L z$. Then, for all $w \in L$, $xw \in L$ iff $yw \in L$, and $yw \in L$ iff $zw \in L$. Thus, $xw \in L$ iff $zw \in L$ for all w. Thus, $\equiv_L$ is transitive, as required. $\square$