
Problem 1.67. Let the rotational closure of language A be $RC(A) = \{yx \mid xy \in A\}$.

Part a. Show that for any language A, we have $RC(A) = RC(RC(A))$.

Proof. To show that for any language A, $RC(A) = RC(RC(A))$, we show that all possible rotations of a string and all possible rotations of a rotation generate the same set of strings.

Let A be any language, and w be any member of A. Let n be the length of w, then $RC(A)$ contains all of the n possible rotations of the string w, which are:

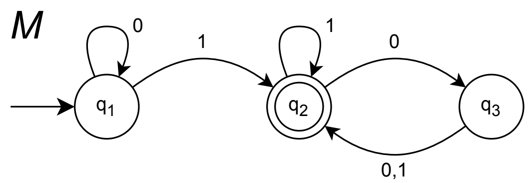
$$\begin{aligned}
&w_1 w_2 w_3 \cdots w_{n-1} w_n, \\
&w_2 w_3 \cdots w_{n-1} w_n w_1, \\
&w_3 \cdots w_{n-1} w_n w_1 w_2, \\
&\vdots \\
&w_n w_1 w_2 w_3 \cdots w_{n-1}
\end{aligned}$$

Every w_i in above strings is the i^{th} symbol of w. All of these strings share the same set of rotations. $\square$

Part b. Show that the class of regular languages is closed under rotational closure.

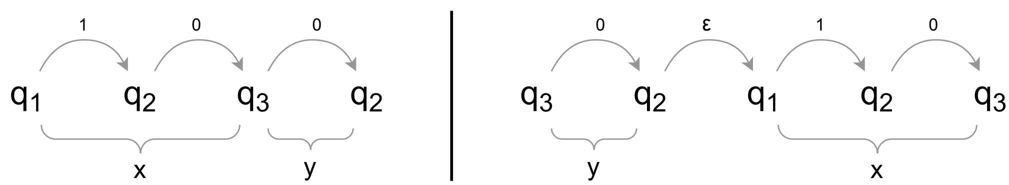
Proof Idea. Let $\Sigma = \{0, 1\}$, and define

$$A = \{w \mid w \textit{ contains at least one } 1 \textit{ and an even number of } 0s \textit{ follow the last } 1\}.$$



State diagram of DFA M that recognizes A.

Take for example the string 100, which is a member of A, and M accepts 100 by transitioning through the sequence of states q_1, q_2, q_3, q_2. The string $010 \in RC(A)$, because it is a rotation of 100 by letting $y = 0$, and $x = 10$.



Comparison of state transitions between the DFA M for A on input string 100 (left), and how a finite automaton, say M' that recognizes $RC(A)$ would compute on 010, which is a rotation of 100 (right).

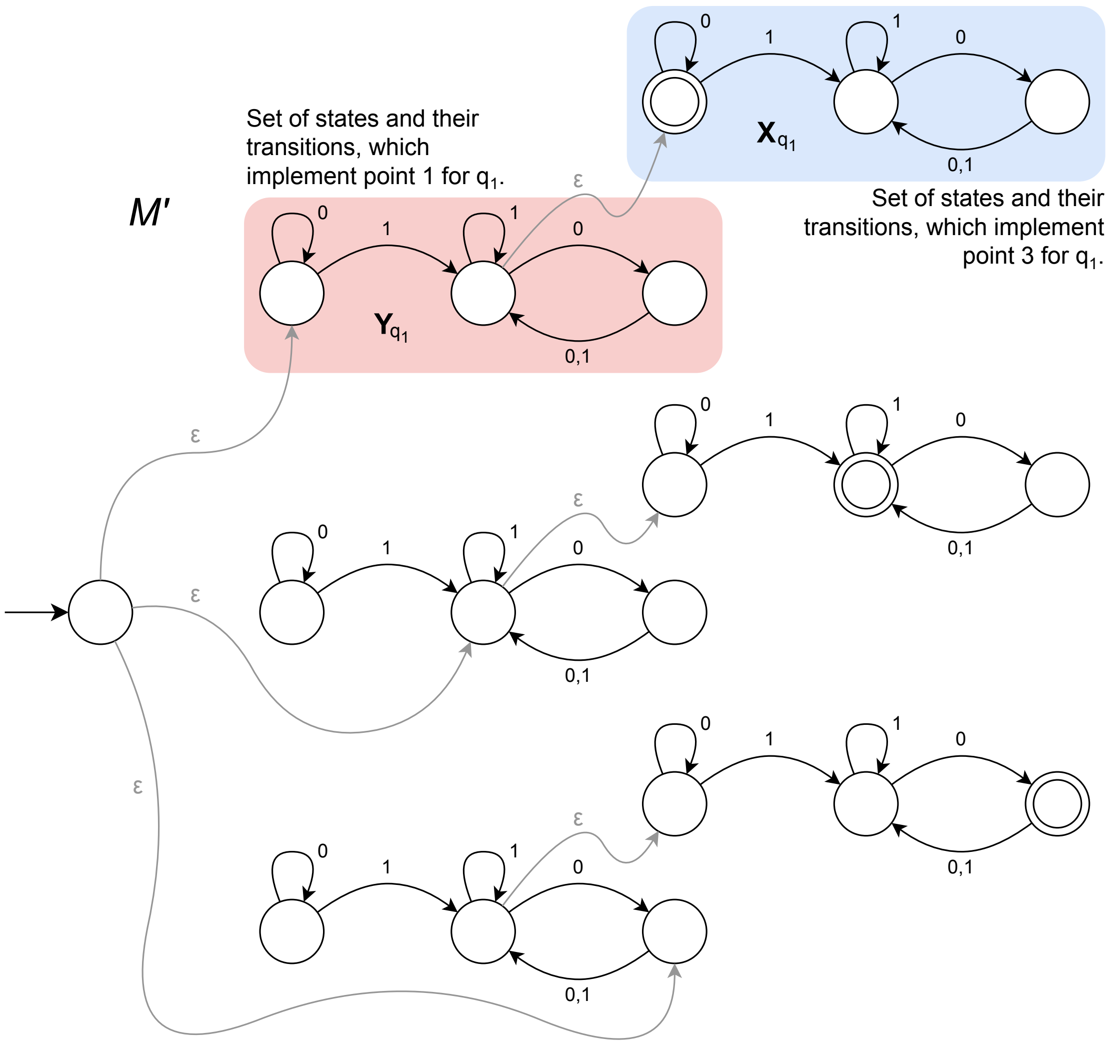
Now, let's simulate how a finite automaton M' that recognizes $RC(A)$ would compute in above case:

1. M' starts at state q_3, and keeps on computing on the input like M until an accept state is reached.

2. When an accept state is reached, which in this case is q_2, it keeps on computing like M, but also takes an ϵ-transition to the start state q_1.

3. After taking an ϵ-transition to the start state q_1, the automaton keeps on computing as M, and accepts if it finishes at state q_3.

In general, a finite automaton M' can be constructed, which recognizes $RC(A)$ by implementing above 3 points for every state in M.



State diagram of DFA M' that recognizes $RC(A)$. M' uses two copies of M, which are modified to implement points 1 and 3 for each state in M, and the ϵ-transitions implement point 2.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA that recognizes A. Construct the NFA $M' = (Q', \Sigma, \delta', q_0', F')$ to recognize $RC(A)$.

1. $Q' = \{q_s\} \cup Q_x \cup Q_y$, where

$$Q_x = \bigcup_{i=1}^{|Q|} X_{q_i} \ , \ Q_y = \bigcup_{i=1}^{|Q|} Y_{q_i}$$

   For each $q_i \in Q$

$$X_{q_i} = \bigcup_{j=1}^{|Q|} (q_i, x, q_j) \ , \ Y_{q_i} = \bigcup_{j=1}^{|Q|} (q_i, y, q_j), \ \textit{and each } q_i, q_j \in Q$$

   The set X_{q_i} is the set of states that implement point 1 for state $q_i \in Q$, and Y_{q_i} is the set of states that implement point 3 for state $q_i \in Q$. The state q_s is the new start state.

2. $q_0' = q_s$

3. $F' = \{(q_1, x, q_1), (q_2, x, q_2), \cdots, (q_n, x, q_n)\}$, where each $q_i \in Q$.

4. Define δ'(q, a) so that for any $q \in Q'$ and any $a \in \Sigma_\epsilon$:

$$\delta'(q, \ a) \ = \begin{cases} \{(q_1, y, q_1), (q_2, y, q_2), \cdots, (q_n, y, q_n)\} & q = q_s \ \textit{and} \ a \ = \ \epsilon \\ \{(q_i, x, \delta(q_j, a))\} & q \in Q_x \\ \{(q_i, y, \delta(q_j, a))\} & q \in Q_y \ \textit{and} \ q_j \notin F \\ \{(q_i, y, \delta(q_j, a))\} & q \in Q_y, \ q_j \in F \ \textit{and} \ a \neq \epsilon \\ \{(q_i, x, q_0)\} & q \in Q_y, \ q_j \in F \ \textit{and} \ a = \epsilon \end{cases}$$

$\square$