CPSC 313 — Tutorial Exercise #6
Closure Properties of Regular Languages

About This Exercise

The following lecture concerns material found in Section 1.2 of *Introduction to the Theory of Computation* and presented in the following lectures.

- Lecture #7: Regular Operations and Closure Properties of Regular Languages

This exercise will be discussed in the tutorial on Thursday, February 3. Please try to solve the problems in this exercise ***before*** attending this tutorial, so that you can participate effectively in discussions about solutions of these problems with other students and the teaching assistants.

Problems To Be Solved

1. Recall that, during this lecture, it was proved that

   - If $L_1, L_2 \subseteq \Sigma^\star$ are regular languages then so is $L_1 \cup L_2$.
   - If $L_1, L_2 \subseteq \Sigma^\star$ are regular languages then so is $L_1 \circ L_2$.
   - If $L_1 \subseteq \Sigma^\star$ is a regular language then so is $L_1^\star$.

   Use these ***closure properties of regular languages*** to prove that the following languages are regular languages — by expressing them as unions, concatenations and stars of simpler regular languages.

   (a) The set of strings in $\Sigma_1^\star$, for $\Sigma_1 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, consisting of one $\mathtt{a}$, followed by one or more $\mathtt{b}$'s, followed by two $\mathtt{c}$'s.

   (b) The set of strings in $\Sigma_1^\star$, for Σ_1 as above, that include *at most* one $\mathtt{a}$.

2. The ***complement*** L^C of a language $L \subseteq \Sigma^\star$ is the set of strings in $\Sigma^\star$ that ***do not*** belong to L:

$$L^C = \Sigma^\star \setminus L = \{\omega \in \Sigma^\star \mid \omega \notin L\}.$$

   Prove that the set of regular languages is also closed under ***complementation,*** that is, if $L \in \Sigma^\star$ is a regular language then so is L^C.

   Hint: Consider a ***deterministic*** finite automaton whose language is L. How would you change it, to get a DFA whose language is L^C instead?

A Final Question for Further Study

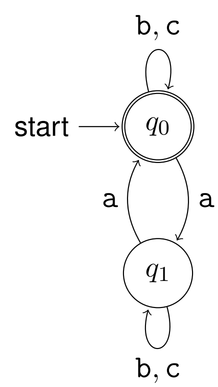
It is unlikely that there will be time for a discussion of this final question. Interested students are welcome to discuss it with the instructor during office hours.

3. Students who are reading the *Introduction to the Theory of Computation* may have noticed that a ***different*** proof is included in this reference to show that if $L_1, L_2 \subseteq \Sigma^\star$ are regular languages then so is $L_1 \cup L_2$. This makes use of a ***product construction*** that is follows.

   Suppose that $L_1 \subseteq \Sigma^\star$ is a regular language and let $M_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$ be a DFA for M. For example, if $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and

$$L_1 = \{\omega \in \Sigma^\star \mid \omega \text{ includes an even number of a's}\}$$

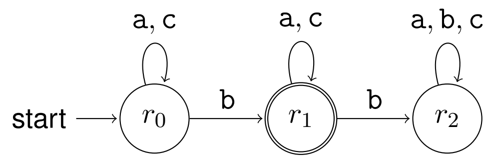
   then M_1 might be as follows.



   Suppose, as well, that $L_2 \subseteq \Sigma^\star$ is a regular language, and let $M_2 = (Q_2, \Sigma, \delta_2, r_0, F_2)$ be a DFA for L_2. For example, if

$$L_2 = \{\omega \in \Sigma^\star \mid \omega \text{ includes exactly one } \mathtt{b}\}$$

then M_2 might be as follows.



Now suppose we want to design a DFA for $L_1 \cup L_2$. The DFA we are constructing will remember the following:

Which state of M_1 has now been reached *and* which state of M_2 has now been reached, if both automata are run on the string being processed?

Thus we will construct a DFA $M = (Q, \Sigma, \delta, \widehat{q}_0, F)$ as follows.

- Q will be the set of all ***ordered pairs*** of states in M_1 and in M_2:
$$Q = Q_1 \times Q_2 = \{(q, r) \mid q \in Q_1 \text{ and } r \in Q_2\}.$$
- The start state is the ordered pair including the start states of M_1 and of M_2:
$$\widehat{q}_0 = (q_0, r_0).$$
- For each pair of states $q \in Q_1$ and $r \in Q_2$, and each symbol $\sigma \in \Sigma$, $\delta((q, r), \sigma) = (s, t)$, where $s \in Q_1$ is the state reached by following the transition out of q for σ in M_1, and $t \in Q_2$ is the state reached by following the transition out of r for σ in M_2. That is,
$$\delta((q, r), \sigma) = (\delta_1(q, \sigma), \delta_2(r, \sigma)).$$
- Since the language of M should be $L_1 \cup L_2$, a state $(q, r) \in Q_1 \times Q_2 = Q$ will be an accepting state if and only if either $q \in F_1$ or $r \in F_2$ (or both):
$$F = \{(q, r) \in Q_1 \times Q_2 \mid q \in F_1 \text{ or } r \in F_2 \text{ (or both)}\}.$$

Thus, if M_1 and M_2 are as above then M will be as shown in Figure 1 on page 4.

With a bit of work (and another proof by mathematical induction, on the length $|\omega|$ of an input string ω) one can prove that

$$\delta^\star(\widehat{q}_0, \omega) = (\delta_1^\star(q_0, \omega), \delta_2^\star(r_0, \omega))$$

for every string $\omega \in \Sigma^\star$ — so that if F is as defined above, then the language of M is $L_1 \cup L_2$ as claimed.

Something nice about this "product construction" is that it can be modified (by changing the set of accepting states) in order to prove the set of regular languages is closed under *other* operations.

With that noted, modify this construction to prove the following properties.

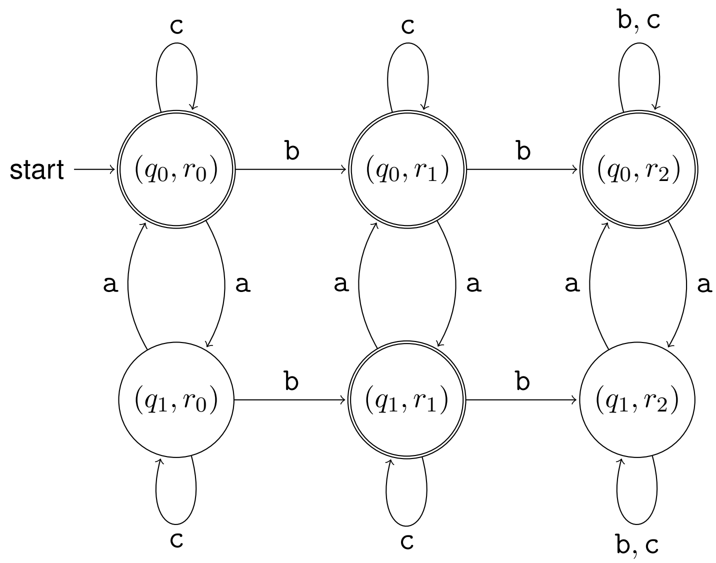


Figure 1: DFA Resulting from Product Construction

(a) ***Intersection:*** If $L_1, L_2 \subseteq \Sigma^\star$ are regular languages then so is

$$L_1 \cap L_2 = \{\omega \in \Sigma^\star \mid \omega \in L_1 \text{ and } \omega \in L_2\}.$$

(b) ***Exclusive-Or:*** If $L_1, L_2 \subseteq \Sigma^\star$ are regular languages then so is

$$L_1 \oplus L_2 = \{\omega \in \Sigma^\star \mid \omega \text{ belongs to exactly one of the languages } L_1 \text{ and } L_2\}.$$

(c) ***Set Difference:*** If $L_1, L_2 \subseteq \Sigma^\star$ are regular languages then so is

$$L_1 \setminus L_2 = \{\omega \in \Sigma^\star \mid \omega \in L_1 \text{ and } \omega \notin L_2\}.$$

Note: It might be helpful to start by proving this for the specific languages L_1 and L_2, using (and modifying) the DFA in Figure 1 as needed: It might be easier to see how to prove this for *arbitrary* regular languages L_1 and L_2 after that.