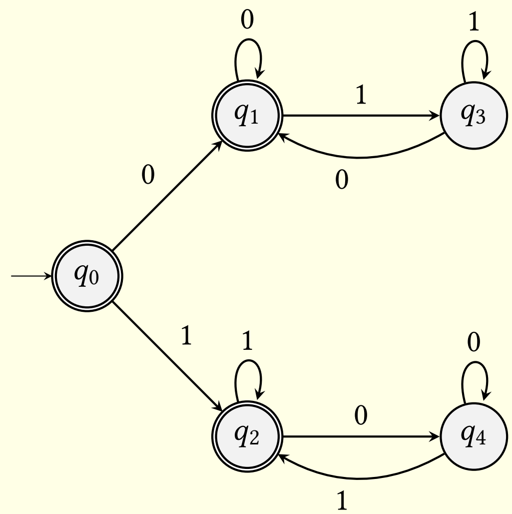
1. **[More NFA/DFA constructions]** Show that the following languages are regular. You can use NFAs, DFAs or the closure operations of regular languages.

(a) $L = \{w \mid w \text{ has equal number of occurrences of } 01 \text{ and } 10\}$.

Solution:

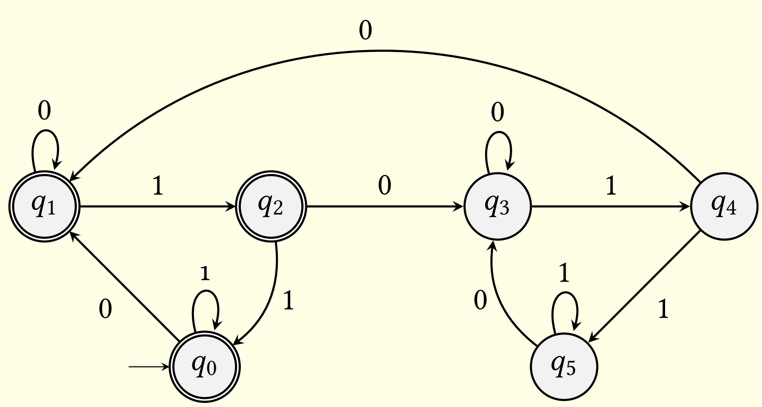


Let's divide the language L into two parts, strings beginning with 0 and strings beginning with 1. If w begins with 0, and if the DFA sees a 1, it should wait to see 0 only then the occurrences of 101 and 10 becomes equal. Therefore the DFA waits in state q3. Similarly if w begins with 1 and if the DFA sees a 0, it should wait to see a 1, to make the number of occurrences of 01 and 10 equal. Therefore the DFA waits in q4. Others strings are accepted by the DFA since there is no comparisons between number of occurrences of 01 and 10 in other strings.

(b) $L = \{w \mid w \text{ contains an even number of occurrences of } 010\}$.

The occurrences can overlap here. For instance, the string $01010 \in L$.

Solution:

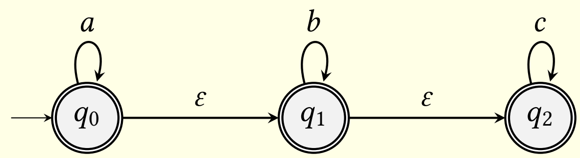


We can think of the DFA as divided into two sections - one corresponding to the states $\{q_0, q_1, q_2\}$ and the other corresponding to the states $\{q_3, q_4, q_5\}$. The former corresponds to the case when there are even number of occurrences of the substring 010, and the latter when there are an odd number of occurrences of 010.

Note that this is not a proof of the construction. Please fill in the gaps yourself or talk to me or one of the TAs if you have doubts.

(c) The alphabet $\Sigma = \{a, b, c\}$, and $L = \{w \mid w = a^i b^j c^k, i, j, k \geq 0\}$.

Solution:

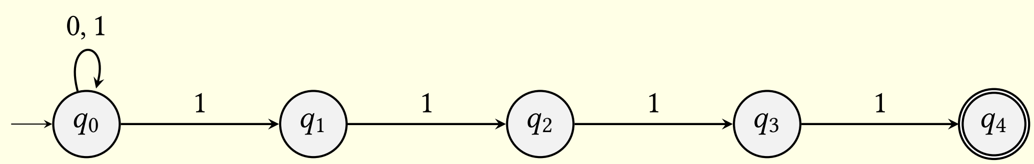


Dead states are not being indicated.

(d) $L = \{w \mid$ at least one of the last four positions is a $0\}$.

Solution:

To prove L is regular, let us construct an NFA for $\overline{L}$, where $\overline{L} = \{w \mid$ w ends with a $1111\}$. By closure properties, since $\overline{L}$ is regular, L is regular.



2. Prove the following two properties of regular languages.

(a) **[An easy one]** Show that every infinite regular language L contains an infinite proper subset $L' \subset L$ such that L' is also regular.

Solution: Let $w \in L$. Then $L' = L - \{w\}$ is regular using the closure properties of regular languages. Since L is infinite, L' is also infinite and is a proper subset of L.

(b) **[A slightly harder one]** Show that every infinite regular language L contains a proper subset $L' \subset L$ such that both L' and $L - L'$ are infinite and regular.

Solution: Let $M = (Q, \Sigma, \delta, q_o, F)$ be a DFA accepting L, and let $|Q| = k$. Let $w = \sigma_1\sigma_2 \ldots \sigma_n \in L$. Let $q_i = \widehat{\delta}(q_0, \sigma_1\sigma_2 \ldots \sigma_i)$ for $1 \leq i \leq n$. Assume that $n \gg k$ - such a w must exist since L is infinite. Thus, there exists i, j such that $q_i = q_j$ (pigeonhole principle). Let y be the substring of w such that $\widehat{\delta}(q_i, y) = q_j$. Thus, we can write $w = xyz$. Furthermore, for every $i \in \mathbb{N}$, $xy^iz \in L$. Let $L' = \{xy^iz \mid i \text{ is even}\}$. The language L' is regular - you will get the DFA from M (verify this). Furthermore $L - L'$ is infinite since it contains the set $\{xy^iz \mid i \text{ is odd}\}$. By the closure property of regular languages $L - L'$ is also regular.

3. **[Alternate definition for an NFA]** Let us define a ∀NFA as a 5-tuple $N = (Q, \Sigma, \Delta, q_0, F)$ where $\Delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$, $q_0 \in Q$, and $F \subseteq Q$. The extended transition function is defined

exactly as we did in class. The language accepted by N is defined as follows.

$$L(N) = \{w \mid \widehat{\Delta}(q_0, w) \subseteq F\}.$$

In other words, a string is accepted iff all the non-deterministic choices lead to a final state.

Show that the languages accepted by ∀NFAs are precisely the class of regular languages.

Solution:

($\Leftarrow$) Let L be a regular language. Then there is a DFA M such that $L(M) = L$. Since $\widehat{\delta}(q_0, w)$ is a single state, M is already a ∀NFA.

($\Rightarrow$) Let L be a language such that it is accepted by a ∀NFA $N = (Q, \Sigma, \Delta, q_0, F)$. Consider a DFA M defined as follows: $M = (\mathcal{P}(Q), \Sigma, \delta, \{q_0\}, F')$, where $\mathcal{P}(Q)$ is the power set of Q. We will define δ as follows.

$$\delta(Q', \sigma) = \widehat{\Delta}(Q', \sigma)$$

This is similar to the subset construction. We will define $F' = \mathcal{P}(F)$ as the power set of F - this is the only difference from the subset construction taught in class. Verify that $L = L(N)$ iff $L = L(M)$ (go back and look at the proof of correctness of the subset construction).

4. **[More examples of closure properties of regular languages]** These are problems that prove additional closure properties of regular languages, similar to some of the examples we saw in class.

(a) We saw in class that if L is regular, then $\sqrt{L}$ is also regular. Can you prove a similar statement for $\sqrt[3]{L}$? How will you generalize this to $\sqrt[k]{L}$ for a constant k?

(b) For a languages $L \subseteq \Sigma^*$, define $L_{\frac{1}{2}-}$ as follows.

$$L_{\frac{1}{2}-} = \{x \mid \exists\, y \text{ such that } |x| = |y| \text{ and } xy \in L\}.$$

Show that if L is regular, then $L_{\frac{1}{2}-}$ is regular.

Solution: This is quite similar to the construction for $\sqrt{L}$. In the case of $\sqrt{L}$, the string y was x itself.

Let L be accepted by the DFA $M = (Q, \Sigma, \delta, q_0, F)$. Our NFA $N = (Q', \Sigma, \Delta, S, F)$ will be as follows:

- $Q' = Q \times Q \times Q$,
- $S = \{(g, q_0, g) \mid g \in Q\}$,
- $F = \{(g, g, q) \mid q \in F\}$,
- $\Delta((g, q, q'), \sigma) = \{(g, \delta(q, \sigma), \delta(q', \sigma')) \mid \sigma' \in \Sigma\}$ (This is the only place where the construction differs from that of $\sqrt{L}$).

Please read the proof for the regularity of $\sqrt{L}$ and try to imitate that proof in this setting.

(c) For two languages $L_1, L_2 \subseteq \Sigma^*$ define the *quotient* of L_1 by L_2, denoted as L_1/L_2, as follows.

$$L_1/L_2 = \{x \mid \exists y \in L_2 \text{ such that } xy \in L_1\}.$$

Show that if L_1 is regular, and L_2 is any language (*not necessarily regular*), then L_1/L_2 is regular.

Solution: Let $M_1 = (Q, \Sigma, \delta, q_0, F)$ be the DFA accepting L_1. Define $M = (Q, \Sigma, \delta, q_0, F')$ where the states Q and transition functions of M are same as M_1. The final states F' are defined as follows: $F' = \{q \in Q \mid \exists y \in L_2 \text{ s.t } \widehat{\delta}(q, y) \in F\}$.

It should be easy to prove that $L(M) = L_1/L_2$ now.

(d) For a language $L \subseteq \Sigma^*$, define $\text{rot}(L)$, the *rotation* of L, as follows.

$$\text{rot}(L) = \{xy \mid yx \in L\}.$$

Show that if L is regular, then $\text{rot}(L)$ is regular.

Solution:

The idea is as follows: We will first guess a state g and start the simulation from that state. Now, while reading w, if there is a prefix x of w such that $\widehat{\delta}(g, x) \in F$, then we will nondeterministically move to the start state of the DFA q_0, and verify whether $\widehat{\delta}(q_0, y) = g$, where y is the remainder of the string w. The explicit decsription of the NFA that does this is given below.

Let $L(M) = L$ be regular such that $M = (Q, \Sigma, \delta, q_0, F)$.

We will construct an NFA $N = (Q', \Sigma, \Delta, q_0', F')$ such that $L(N) = \text{rot}(L)$.

Define $Q' = (Q \times Q \times \{\text{before}, \text{after}\}) \cup \{q_0'\}$, where first item of the tuple stores the guessed state, the second stores the current state in the simulation of M on input w. The value before denotes that we have not completed scanning y and after denotes whether we have finished scanning y. The transition function is defined as below.

- $\Delta(q_0', \varepsilon) = \{(g, g, \text{before}) | g \in Q\}$,
- $\Delta((g, f, \text{before}), \varepsilon) := \{(g, q_0, \text{after})\}, \forall g \in Q$ and $\forall f \in F$,
- $\Delta((g, q, \text{before}), \sigma) := \{(g, \delta(q, \sigma), \text{before})\}$,
- $\Delta((g, q, \text{after}), \sigma) := \{(g, \delta(q, \sigma), \text{after})\}$.

The final states are defined as follows: $F' = \{(g, g, \text{after}) | g \in Q\}$.

Correctness: To prove $L(N) = rot(L)$

$(\Rightarrow) L(N) \subseteq \text{rot}(L)$

Suppose that $w \in L(N)$. There exists a path say P that we follow while reading w such that we start from q_0' and we reach a final state that is of the form (g, g, after). By construction, P must start by taking an ε transition to a state of the form (g, g, before), followed by some walk inside $\{g\} \times Q \times \{\text{before}\}$. At some point, an ε transition must be taken before leaving (g, a, before) where $a \in F$. Let x be such that $(g, a, \text{before}) \in$

$\widehat{\Delta}(q_0', x)$. Following this it goes to (g, q_0, after) and at this point reading the rest of the input must lead to (g, g, after). Let y be the string such that $(g, g, \text{after}) \in \widehat{\Delta}((g, q_0, \text{after}), y)$. Thus, we know that $\hat{\delta}(q_0, y) = g$ and $\hat{\delta}(g, x) \in F$. Hence $yx \in L$. $\square$

($\Leftarrow$) $\text{rot}(L) \subseteq L(N)$

Suppose $w \in \text{rot}(L)$. Then there exists y, x such that $w = xy$ and $yx \in L$. We need to show an accepting path from q_0' to $f' \in F'$, that is a state of the form (g, g, after). Note that $\widehat{\delta}(q_0, yx) = \widehat{\delta}(\widehat{\delta}(q_0, y), x) \in F$.

Let $g = \widehat{\delta}(q_0, y)$. Before reading w we take an ε transition to reach (g, g, before). After reading x we must be in $(g, \widehat{\delta}(g, x), \text{before})$. We know that $\widehat{\delta}(g, x) \in F$, and hence by the construction, there exists an ε transition from $(g, \widehat{\delta}(g, x), \text{before})$ to (g, q_0, after). After this transition we read the rest of the input (*i.e.*, y) and reach $(g, \widehat{\delta}(q_0, y), \text{after}) = (g, g, \text{after})$ which is an accepting state of N. Thus, $\text{rot}(L) \subseteq L(N)$. $\square$

Here's an alternative way to express the same idea mathematically.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA accepting L. For each state g, we will define an NFA $N_g = (Q_g, \Sigma, \Delta_g, s, F_g)$ as follows:

- $Q_g = Q \cup Q'$, where $Q' = \{q' | q \in Q\}$,
- $s = g$,
- $F_g = \{g'\}$
- The transition function is defined as follows.
  - $\Delta_g(q, \sigma) = \delta(q, \sigma)$,
  - $\Delta_g(q', \sigma) = q_1'$ where $q_1 = \delta(q, \sigma)$
  - $\Delta_g(f, \varepsilon) = q_0'$, $\forall f \in F$.

Think of this as two copies of M, and ε-transitions from the final states of one copy to the start state of the other copy. All the other transitions within the copies remain the same. The start state is the state g in the first copy, and the final state is the corresponding state g' in the second copy.

The final NFA is the union of all the NFAs N_g for every $g \in Q$.

Try proving the correctness of the construction - it will be similar to the earlier proof.

5. **[Proving non-regularity of languages]** You can either use the pumping lemma for proving non-regularity, or you can prove it without explicitly using the lemma and directly giving a proof by contradiction.

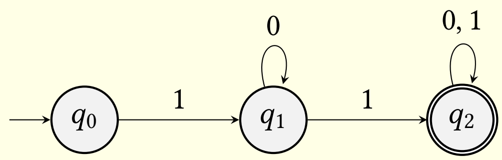
   (a) Consider the following two languages.

$$L_1 = \{1^k x \mid x \in \{0,1\}^* \text{ and } \#1(x) \geq k, k \geq 1\},$$
$$L_2 = \{1^k x \mid x \in \{0,1\}^* \text{ and } \#1(x) \leq k, k \geq 1\}.$$

   One of the languages is regular, and the other is not. Which one is regular and which is non-regular? Justify your answer.

Solution:

Language L_1 is regular. Notice that L_1 is identical to the language $L' = \{w \mid w = 1w' \text{ where } \#1(w') \geq 1\}$. Clearly, $L' \subseteq L$. For the other direction, consider any $1^k x \in L_1$. We can write this string as $1^{k'} y$ where $k' = 1$ so $\#1(y) = k - 1 + \#1(x)$. Now, $k - 1 + \#1(x)$ is at least $k' = 1$. The DFA for L_1 will be as follows.



We will show that L_2 is not regular using the pumping lemma.

Assume that the prover chooses some $k > 0$. The spoiler will choose $x = 1^{k+1}0$, $y = 1^{k+1}$, and $z = \varepsilon$. The string $xyz = 1^{k+1}01^{k+1} \in L_2$. Now, suppose that the prover chooses $u = 1^l$, $v = 1^m$, $w = 1^n$ for any l, m, n such that $l + m + n = k + 1$ and $m \neq 0$, the spoiler can choose $i = 2$ and the string $xuv^2wz = 1^{k+1}01^{k+1+m} \notin L_2$. Write down clearly why the original $xyz \in L$, and why $xuv^2wz \notin L$.

(b) Consider the following two languages.

$$L_1 = \{w \mid \exists\, x, y \in \{0,1\}^* - \varepsilon \text{ such that } w = xyx\},$$
$$L_2 = \{w \mid \exists\, x, y \in \{0,1\}^* - \varepsilon \text{ such that } w = xyx^R\}.$$

One of the languages is regular, and the other is not. Which one is regular and which is non-regular? Justify your answer.

Solution: Language L_1 is not regular. We show this using the pumping lemma as follows -

- Prover chooses $k > 0$.
- Spoiler chooses $x = 0^{k+1}1^{k+2}$, $y = 0^{k+1}$, $z = 1^{k+1}$ where $xyz = 0^{k+1}1^{k+2}0^{k+1}1^{k+1} = x'y'x'$ where $x' = 0^{k+1}1^{k+1}$ and $y' = 1$. Hence, $xyz \in L$ and $|y| = k + 1 > k$.
- Prover chooses $u = 0^l$, $v = 0^m \neq \epsilon$, $w = 0^n$ where $l + m + n = k + 1$ and $m \geq 1$ so that $y = uvw$.
- Spoiler chooses $i = 0$ so that $xuv^iwz = 0^{k+1}1^{k+2}0^{l+n}1^{k+1} = 0^{k+1}1^{k+2}0^{k+1-m}1^{k+1} \notin L$.

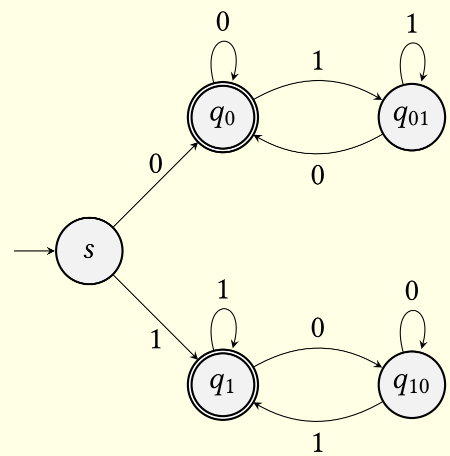
Language L_2 is regular. Observe the following characterization of L_2.

Claim 1. *$w \in L_2 \iff w$ starts and ends with the same symbol and $|w| \geq 3$*

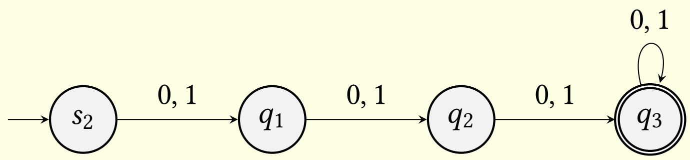
Hence, we define languages $L_3 = \{w \in \{0,1\}^* \mid w \text{ starts and ends with the same character}\}$ and $L_4 = \{w \in \{0,1\}^* \mid |w| \geq 3\}$.

Clearly $L_2 = L_3 \cap L_4$. Hence, using closure properties it suffices to show that L_3 and L_4 are regular.

Below is a DFA accepting L_3



Below is a DFA accepting L_4



(c) Show that $L = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i = 1 \Rightarrow j = k\}$ is not regular.

Solution: This is a simple extension of the proof that $\{0^n 1^n \mid n \geq 0\}$ is not regular. Using the pumping lemma,

For a choice of $k > 0$, choose $x = a, y = b^{k+1}, z = c^{k+1}$. The string $xyz = ab^{k+1}c^{k+1}$ and hence $xyz \in L$.

Now, $\forall u = b^l, v = b^m, w = b^n$ such that $l + m + n = k + 1, m \neq 0$, choose $i = 0$. Then, $xuv^iwz = xuwz = ab^{l+n}c^{k+1}$. Since $l + n < k + 1, xuv^0wz \notin L$.

(d) Show that $L = \{w \mid w \neq w^R\}$ is not regular.

Solution: To show that language L is not regular, we can equivalently show $\overline{L}$ is not regular as regular languages are closed under complementation. Note that $\overline{L} = \{w \mid w = w^R\}$.

We show that $\overline{L}$ is not regular by using the Pumping Lemma and showing a winning strategy exists for the *Spoiler*.

Prover : Chooses any $k > 0$.

Spoiler : Chooses $x = 1^k 0$, $y = 1^k$, $z = \epsilon$. Clearly, $xyz = 1^k 0 1^k \in \overline{L}$.

Prover : Chooses $u = 1^l$, $v = 1^m$, $w = 1^n$, s.t $l + m + n = k$, $m \neq 0$. From the choice of x, y, and z, this is the only possibility for the choices for u, v, and w. The values l, m, and n could be arbitrary.

Spoiler : Chooses $i = 2$: The new string becomes $1^k 0 1^{k+m}$.

Since $m \neq 0$, the new string $\notin \overline{L}$ and hence $\overline{L}$ is not regular $\implies$ L is not regular.