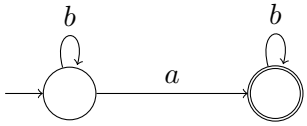
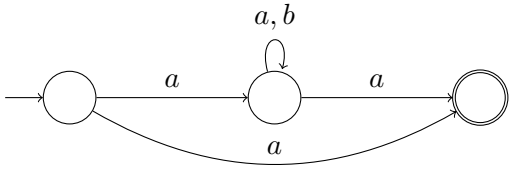
Sample solution to HW 2

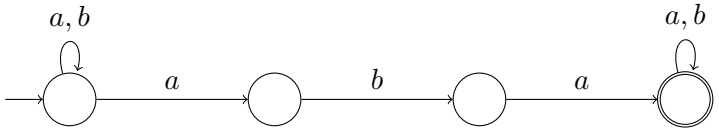
(1) (a) The language L that consists of all the words in which a appears exactly once.



(b) The language L that consists of all the words that starts with a and ends with a.

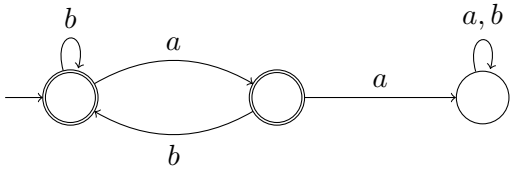


(c) The language L that consists of all the words that contains aba.



(d) The language L that consists of all the words that do *not* contain aa.

A neat way to do this is to construct first a DFA that accepts all words that contain aa, and then take its complement.

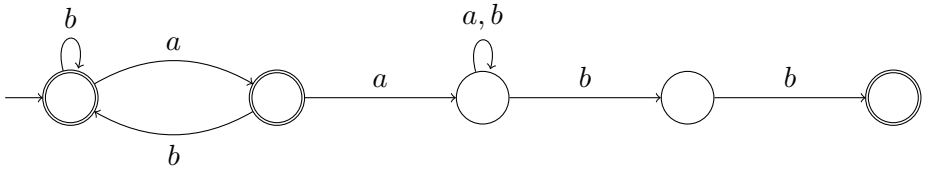


(e) The language L that consists of all the words w such that if w contains aa, then w ends with bb.

Define the following two languages.

$$\begin{aligned} L_1 &= \{w \mid w \text{ does not contain } aa\} \\ L_2 &= \{w \mid w \text{ contains } aa \text{ and ends with } bb\} \end{aligned}$$

Note that $L = L_1 \cup L_2$. This observation already gives us a hint on how to construct an NFA for L. Below is one example.



(2) (a) b^*ab^*.

(b) $\big(a \cup (a\Sigma^*a)\big)$.

(c) $\Sigma^*aba\Sigma^*$.

(d) There are two ways to construct the regex for this language. One is by constructing the regex from the DFA in question (1.d) via the procedure described in the proof of Theorem 4.1. Another way is to observe the following: w does not contain aa, if and only if one of the following holds.

- w does not contain any a, i.e., it contains only b's.
- every a *not* in the end of w is immediately followed by b.

With this observation, we can immediately construct the desired regex e_1 for L.

$$e_1 \quad = \quad b^* \ \cup \ (b^*abb^*)^*(a \cup \emptyset^*).$$

The last part $(a \cup \emptyset^*)$ is to allow the word w to end with a.

(e) Recall the two languages L_1 and L_2:

$$\begin{aligned} L_1 &= \{w \mid w \text{ does not contain } aa\} \\ L_2 &= \{w \mid w \text{ contains } aa \text{ and ends with } bb\} \end{aligned}$$

That is, $L = L_1 \cup L_2$. The regex for L_1 is e_1, as in question (2.d). The regex for L_2 is:

$$e_2 = \Sigma^*aa\Sigma^*bb.$$

So the desired regex for L is $e = e_1 \cup e_2$, which is:

$$e \quad = \quad \Big(b^* \ \cup \ (b^*abb^*)^*(a \cup \emptyset^*)\Big) \quad \cup \quad \Sigma^*aa\Sigma^*bb.$$

(3) (a) L consists of all the words in which a appears exactly 3 times.
L is regular with regex:

$$b^*ab^*ab^*ab^*$$

(b) L consists of all the words in which a appears even number of times.
L is regular with regex:

$$b^* \quad \cup \quad \big(b^*ab^*ab^*\big)^*$$

(c) L consists of all the words of even length.
L is regular with regex:

$$\big(\Sigma\Sigma\big)^*$$

(d) $L = \{a^mba^n \mid 0 \leqslant m \leqslant n\}$.
L is not regular. The proof is via pumping lemma. Suppose to the contrary that L is regular. Let $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ be its NFA.
Consider the word a^mba^n, where $m \geqslant |Q|$. By Lemma 3.7, we can partition a^m into three parts uvw such that for each $i \geqslant 0$, uv^iwba^n is accepted by $\mathcal{A}$, which is not possible. We can choose an $i \geqslant n$ which makes the number of a's on the left hand side of b bigger than n. Thus, there is no such NFA that accepts L.

(e) L consists of all the words in which the number of occurrences of a is a prime number.
L is not regular. The proof is again via pumping lemma. Suppose to the contrary that L is regular. Let $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$ be its NFA.
Consider the word a^m, where m is a prime number bigger than $|Q|$. By Lemma 3.6, we can partition a^m into three parts uvw such that for each $i \geqslant 0$, uv^iw is accepted by $\mathcal{A}$.

Now, uvw consists of only a's. So, the number of a's in uv^iw is precisely its length, which is:

$$|uv^iw| \quad = \quad |u| + i|v| + |w|$$

If we consider $i = |u| + |w|$, we have:

$$\begin{aligned} |uv^iw| &= |u| + (|u| + |w|)|v| + |w| \\ &= (|u| + |w|)(|v| + 1) \end{aligned}$$

Thus, $|uv^iw|$ cannot be a prime number. However, it is supposed to be accepted by $\mathcal{A}$. Therefore, there cannot be such NFA $\mathcal{A}$ that accepts L.

(4) For a language $L \subseteq \Sigma^*$ (not necessarily regular), we define the equivalence relation $\sim_L$ on Σ^* as follows. $u \sim_L v$, if the following holds: For every $w \in \Sigma^*$, $uw \in L$ if and only if $vw \in L$.

(a) That $\sim_L$ is an equivalence relation is quite straightforward.

(b) Prove that if $u \sim_L v$, then either both $u, v \in L$ or both $u, v \notin L$.

Proof: Suppose $u \sim_L v$. By definition, for every $w \in \Sigma^*$, $uw \in L$ if and only if $vw \in L$. In particular, $u\epsilon \in L$ if and only if $v\epsilon \in L$. Since $u\epsilon = u$ and $v\epsilon = v$, either both $u, v \in L$ or both $u, v \notin L$.

(c) Suppose L is regular, and $\mathcal{A}$ is its DFA, i.e., $L(\mathcal{A}) = L$. For a word w, we denote by $\mathcal{A}(w)$ the state of $\mathcal{A}$ after reading w. Or, more formally, if $w = a_1 \cdots a_n$ and $q_0a_1q_1 \cdots a_nq_n$ is the run of $\mathcal{A}$ on w, then $\mathcal{A}(w) = q_n$.

Prove that if u and v are words such that $\mathcal{A}(u) = \mathcal{A}(v)$, then $u \sim_L v$.

Proof: Let u and v be such that $\mathcal{A}(u) = \mathcal{A}(v)$. Let $u = a_1 \cdots a_n$ and $v = b_1 \cdots b_m$.

We have to show that for every $w \in \Sigma^*$, $uw \in L$ if and only if $vw \in L$. Let $w = c_1 \cdots c_k$. Consider the run of $\mathcal{A}$ on uw:

$$p_0 \; a_1 \; p_1 \; \cdots\cdots \; a_np_n \; c_1r_1 \; \cdots\cdots c_kr_k$$

Likewise, consider the run of $\mathcal{A}$ on vw:

$$s_0 \; b_1 \; s_1 \; \cdots\cdots \; b_ms_m \; c_1t_1 \; \cdots\cdots c_kt_k$$

Here both p_0, s_0 is the initial state of $\mathcal{A}$. Since $\mathcal{A}(u) = \mathcal{A}(v)$, we have $p_n = s_m$. Furthermore, $\mathcal{A}$ is deterministic. Thus, $r_1 = t_1, \ldots, r_k = t_k$, and therefore,

$$\mathcal{A}(uw) \quad = \quad \mathcal{A}(vw)$$

So, we have proved that for every word $w \in \Sigma^*$, $uw \in L$ if and only if $vw \in L$.

(d) Following (c), prove that if L is regular with $\mathcal{A}$ being its DFA, then $\sim_L$ has finitely many equivalence classes and $\#(\sim_L) \leqslant |Q|$, where Q is the set of states of $\mathcal{A}$.

Proof: From (c), we know that if $\mathcal{A}(u) = \mathcal{A}(v)$, then $u \sim_L v$. Thus, $\#(\sim_L) \leqslant |Q|$.

(5) **(bonus point)** Let L be a language over Σ, where $\sim_L$ has finitely many equivalence classes $C_1, \ldots, C_m$. Using the notation in Lecture 1, we can represent each C_i as $[w]$, for every $w \in C_i$.

In the following, we assume that $L \neq \emptyset$.

(a) Prove that there is $i_1, \ldots, i_k \subseteq \{1, \ldots, m\}$ such that $L = C_{i_1} \cup \cdots \cup C_{i_k}$. (You can use (4.b) here.)

Proof: What this statement means is that L is a union of some of the equivalence classes of $\sim_L$.

We will show that for every equivalence class C_i, either all words in C_i belong to L, or none of the words in C_i belong to L. More formally, we will show that if $C_i \cap L \neq \emptyset$, then $C_i \subseteq L$.

Suppose $C_i \cap L \neq \emptyset$, and let $w \in C_i \cap L$. By definition, $w \sim_L v$, for every word $v \in C_i$. By (4.b), if $w \sim_L v$, then either both of them are in L, or both of them are not in L. Since $w \in L$, it means v is also in L. Thus, we have shown that $v \in L$, for every $v \in C_i$.

From here, it follows that L is simply a union of some of the equivalence classes of $\sim_L$. That is, there is $i_1, \ldots, i_k \subseteq \{1, \ldots, m\}$ such that $L = C_{i_1} \cup \cdots \cup C_{i_k}$.

(b) Consider the following DFA $\mathcal{A} = \langle \Sigma, Q, q_0, F, \delta \rangle$.

- $Q = \{p_1, \ldots, p_m\}$, i.e., the number of states is precisely the number of equivalence classes in $\sim_L$.
- q_0 is p_j, where j is such that $\epsilon \in C_j$.
- $F = \{p_{i_1}, \ldots, p_{i_k}\}$, where $i_1, \ldots, i_k$ are the indices in (5.a).
- $\delta : Q \times \Sigma \to Q$ is defined as follows. For every $p_i \in Q$, for every $a \in \Sigma$, we pick an arbitrary $w \in C_i$, and define $\delta(p_i, a) = p_j$, where $[wa] = C_j$.

Prove that δ is a well-defined function, i.e., for every $w_1, w_2 \in C_i$, $[w_1 a] = [w_2 a]$. In other words, the end result p_j remains the same for whichever w we pick, as long as w is from C_i.

Proof: Let $w_1, w_2 \in C_i$, i.e., w_1, w_2 belong to the same equivalence class, which means that $w_1 \sim_L w_2$. We will show that $w_1 a \sim_L w_2 a$, for every $a \in \Sigma$.

Suppose to the contrary that there is a such that $w_1 a \not\sim_L w_2 a$. This means there is v such that either:

- $w_1 av \in L$, but $w_2 av \notin L$, or
- $w_1 av \notin L$, but $w_2 av \in L$.

But this contradicts the fact that $w_1 \sim_L w_2$. This means that $w_1 a \sim_L w_2 a$.

(c) Let $\mathcal{A}$ be as in (b). Prove that $L(\mathcal{A}) = L$.

Proof: Recall the definition of $\mathcal{A}(w)$. That is, $\mathcal{A}(w)$ is the state of $\mathcal{A}$ after reading w starting from the initial state.

From the construction of $\mathcal{A}$, for every word $w \in \Sigma^*$, if $[w] = C_j$, then $\mathcal{A}(w) = p_j$. Now, by (5.a),

$$w \in L \quad \text{if and only if} \quad w \in C_{i_1} \cup \cdots \cup C_{i_k}$$

Now,

$$w \in C_{i_1} \cup \cdots \cup C_{i_k} \quad \text{if and only if} \quad A(w) \text{ is one of } p_{i_1}, \ldots, p_{i_k}.$$

Thus, $w \in L$ if and only if $w \in L(\mathcal{A})$, and hence, $L = L(\mathcal{A})$.

(d) **[Myhill-Nerode theorem]** Prove that a language L is regular if and only if $\sim_L$ has finitely many equivalence classes.

Proof: The "only if" direction comes from (4.d), while the "if" direction comes from (5.c).