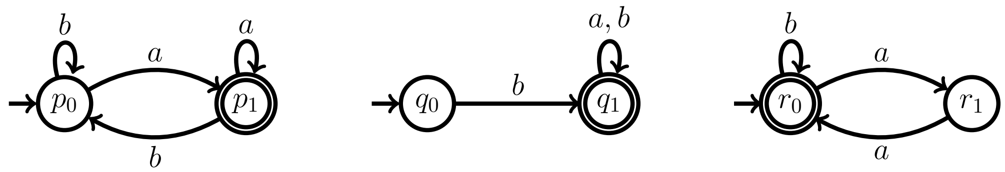
Automata and Formal Languages — Homework 4

Due 14.11.2017

Exercise 4.1
Consider the following DFAs A, B and C:



Use pairings to decide *algorithmically* whether $L(A) \cap L(B) \subseteq L(C)$.

Exercise 4.2
Let $L \subseteq \Sigma^*$ be a language accepted by an NFA A. For every $u, v \in \Sigma^*$, we say that $u \preceq v$ if and only if u can be obtained by deleting zero, one or multiple letters of v. For example, $abc \preceq abca$, $abc \preceq acbac$, $abc \preceq abc$, $\varepsilon \preceq abc$ and $aab \not\preceq acbac$. Give an NFA-ε for each of the following languages:

(a) $\downarrow L = \{w \in \Sigma^* : w \preceq w' \text{ for some } w' \in L\}$,

(b) $\uparrow L = \{w \in \Sigma^* : w' \preceq w \text{ for some } w' \in L\}$,

(c) $\sqrt{L} = \{w \in \Sigma^* : ww \in L\}$,

(d) $\mathrm{Cyc}(L) = \{vu \in \Sigma^* : uv \in L\}$.

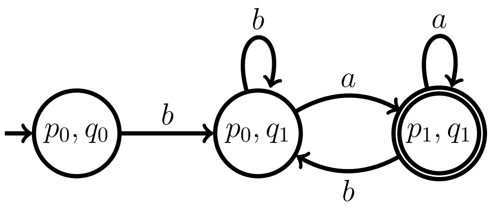
Exercise 4.3
Let Σ_1 and Σ_2 be alphabets. A *morphism* is a function $h : \Sigma_1^* \to \Sigma_2^*$ such that $h(\varepsilon) = \varepsilon$ and $h(uv) = h(u) \cdot h(v)$ for every $u, v \in \Sigma_1^*$. In particular, $h(a_1 a_2 \cdots a_n) = h(a_1)h(a_2)\cdots h(a_n)$ for every $a_1, a_2, \ldots, a_n \in \Sigma$. Hence, a morphism h is entirely determined by its image over letters.

(a) Let $L \subseteq \Sigma_1^*$ be accepted by some NFA A_1. Give an NFA-ε B_2 that accepts $h(L) = \{h(w) : w \in L\}$.

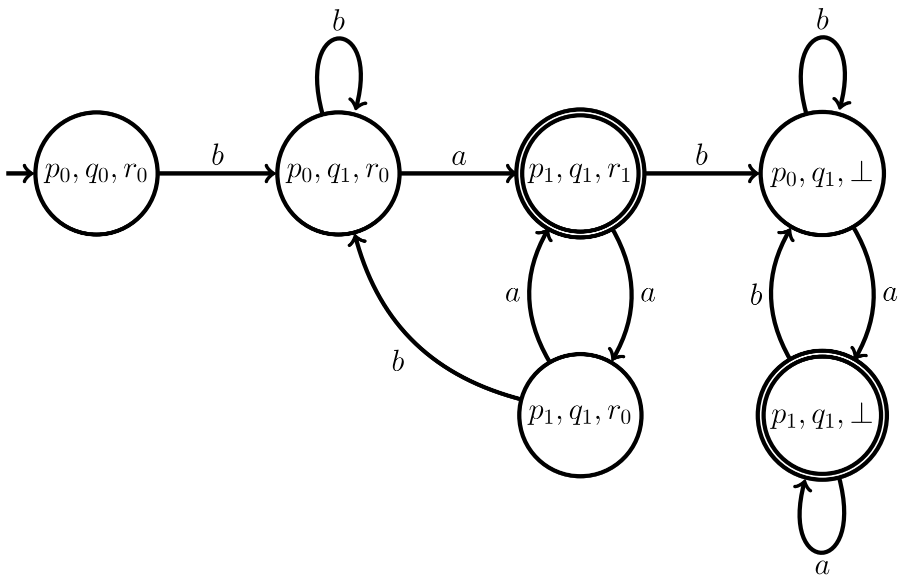
(b) Let $L \subseteq \Sigma_2^*$ be accepted by some NFA A_2. Give an NFA B_1 that accepts $h^{-1}(L) = \{w \in \Sigma_1^* : h(w) \in L\}$.

(c) Show that $L = \{(aab)^n e^m (cad)^n ef(fe)^n : m, n \in \mathbb{N}\}$ is not regular by using the fact that $\{a^n b^n : n \in \mathbb{N}\}$ is not regular.

Solution 4.1
We first build the pairing accepting $L(A) \cap L(B)$. Note that it is not necessary to explore the implicit trap states of A and B as they cannot lead to final states in the pairing. We obtain:



Now, we build the pairing accepting $(L(A) \cap L(B)) \setminus L(C)$ from the above automaton and C. Note that we must explore the implicit trap state of C as it may be part of final states in the pairing. We obtain:



Since the above automaton contains final states, its language is non empty and hence $L(A) \cap L(B) \not\subseteq L(C)$. Note that we can reach this conclusion as soon as we construct state (p_1, q_1, r_1).

Solution 4.2
Let $A = (Q, \Sigma, \delta, Q_0, F)$ be an NFA that accepts L.

(a) We add a ε-transition "parallel" to every transition of A. This simulates the deletion of letters from words of L. More formally, let $B = (Q, \Sigma, \delta', Q_0, F)$ be such that, for every $q \in Q$ and $a \in \Sigma \cup \{\varepsilon\}$,

$$\delta'(q, a) = \begin{cases} \delta(q, a) & \text{if } a \in \Sigma, \\ \{q \in Q : q \in \delta(q, b) \text{ for some } b \in \Sigma\} & \text{if } a = \varepsilon. \end{cases}$$

(b) For every state of Q, we add self-loops for each letter of Σ. This corresponds to the insertion of letters in words of L. More formally, let $B = (Q, \Sigma, \delta', Q_0, F)$ be such that $\delta'(q, a) = \delta(q, a) \cup \{q\}$ for every $q \in Q$ and $a \in \Sigma$.

(c) Intuitively, we construct an automaton B that guesses an intermediate state p and then reads w simultaneously from an initial state q_0 and from p. The automaton accepts if it simultaneously reaches p and and an accepting state q_F. More formally, let $B = (Q', \Sigma, \delta', Q'_0, F')$ be such that

$$\begin{aligned} Q' &= Q \times Q \times Q, \\ Q'_0 &= \{(p, q, p) : p \in Q, q \in Q_0\}, \\ F' &= \{(p, p, q) : p \in Q, q \in F\}, \end{aligned}$$

and, for every $p, q, r \in Q$ and $a \in \Sigma$,

$$\delta'((p,q,r),a) = \{(p,q',r') : q' \in \delta(q,a), r' \in \delta(r,a)\}.$$

(d) Intuitively, we construct an automaton B that guesses a state p and reads a prefix v of the input word until it reaches a final state. Then, B moves non deterministically to an initial state from which it reads the remainder u of the input word, and it accepts if it reaches p. More formally, let $B = (Q', \Sigma, \delta', Q'_0, F')$ be such that

$$\begin{aligned} Q' &= Q \times \{0,1\} \times Q, \\ Q'_0 &= \{(p,0,q) : p \in Q, q \in Q_0\}, \\ F' &= \{(p,1,p) : p \in Q\}, \end{aligned}$$

and, for every $p, q \in Q$ and $a \in \Sigma \cup \{\varepsilon\}$,

$$\delta'((p,b,q),a) = \begin{cases} \{(p,b,q') : q' \in \delta(q,a)\} & \text{if } a \in \Sigma, \\ \{(p,1,q') : q' \in Q_0\} & \text{if } a = \varepsilon, b = 0 \text{ and } q \in F, \\ \emptyset & \text{otherwise.} \end{cases}$$

Solution 4.3

(a) Since h is determined by its image over letters, we replace each transition (p,a,q) of A by a sequence of transitions from p to q labeled by $h(a)$. Some ε-transitions may be introduced if $h(a) = \varepsilon$ for some $a \in \Sigma$.

(b) Let $A = (Q, \Sigma_2, \delta, Q_0, F)$. We keep the states of A unchanged, but we remove its transitions. For each $p, q \in Q$ and $a \in \Sigma_1$, we add a transition (p,a,q) to B for every state q that can be reached from state p by reading $h(a)$ in A. More formally, let $B = (Q, \Sigma_1, \delta', Q_0, F)$ be such that

$$\delta'(p,a) = \{q \in Q : p \xrightarrow{h(a)}_A q\}.$$

(c) For the sake of contradiction, suppose L is regular. There exists an NFA A that accepts L. Let g be the morphism such that $g(a) = a$, $g(b) = b$, $g(c) = c$, $g(d) = d$ and $g(e) = g(f) = \varepsilon$. We have

$$g(L) = \{(aab)^n (cad)^n : n \in \mathbb{N}\}.$$

By (a), language $g(L)$ is regular. Let h be the morphism such that $h(a) = aab$, $h(b) = cad$, $h(c) = c$, $h(d) = d$, $h(e) = e$ and $h(f) = f$. We have

$$h^{-1}(g(L)) = \{a^n b^n : n \in \mathbb{N}\}.$$

By (b), language $h^{-1}(g(L))$ is regular, which is a contradiction. □

★ As discussed in class, there is a simpler solution. Suppose L is regular and let h be the morphism such that $h(b) = a$, $h(c) = b$ and $h(a) = h(d) = h(e) = h(f) = \varepsilon$. We have

$$h(L) = \{a^n b^n : n \in \mathbb{N}\}.$$

By (a), language $h(L)$ is regular, which is a contradiction. □