CS 4124
Solutions to Homework Assignment 1
Collin McDevitt

February 2, 2024

[25] 1. Textbook Problem 5 in B.1 on Page 520.

Prove Lemma A.4: For all alphabets Σ and all languages $L \subseteq \Sigma^\star$, the equivalence relation $\equiv_L$ is right-invariant.

The lemma and the definition of the equivalence relation are on Page 505.

Proof. Let Σ be an alphabet and $L \subseteq \Sigma^*$ and $\equiv_L$ be an equivalence relation as defined by (A.3). Let $x, y, z, w \in \Sigma^*$ and $x \equiv_L y$ then we have $xz, yz \in L$ as $zw \in \Sigma^*$ we have $xzw, yzw \in L$ which implies $xz \equiv_L yz$. □

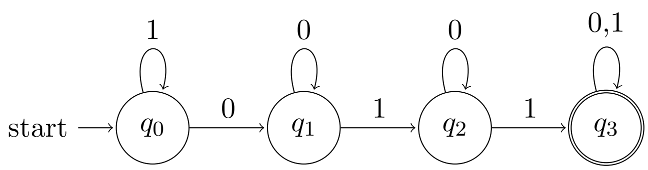
---

[25] 2. Textbook Problem 3(c) in B.2 on Page 522.

Design an FA M, with alphabet $\Sigma = \{0, 1\}$, that recognizes (c) the set of all strings that contain the string 011, in that order, but not necessarily consecutively.

Be certain you understand the set (language) of M before you start designing. Constructing some examples of strings both in $L(M)$ and not in $L(M)$ can be helpful.

You may use an algebraic specification or a transition diagram (labeled directed graph) specification to present your design for M. Be certain to explain why your design works.



This design for a **FA** works as reading through the input string we have it only enters state q_1 if it has read 0 and will loop otherwise. Then for each of the next states q_1, q_2 they will loop if they receive a 0 otherwise they will go to q_2 and q_3 respectively. q_3 is an accepting state and will loop for any input which ensures that the string only need to contain 011 not end in that.

---

[25] 3. Consider the OA M_4 in Figure 3.4 on Page 60. Give a complete and careful algebraic specification

$$M_4 \quad = \quad (Q, \Sigma, \delta, q_0, F)$$

for M_4.

$Q = \{C, A_1, B_1, A_2, B_2, ...\}$ is the set of states for M_4.

$\Sigma = \{a, b\}$ is M_4's input alphabet.

$\delta : Q \mapsto Q$ is M_4's state transition function. Where $\delta(A_k, a) = A_{k+1}$ and $\delta(A_k, b) = B_k$ for $k \geq 2$ and $\delta(A_0, b) = \delta(B_1, b) = \delta(B_1, a) = C$ and $\delta(B_k, b) = B_{k-1}$ for $k \geq 2$ and $\delta(B_K, a) = C$ for $k \geq 1$.

$q_0 = A_0$ the starting state.

$F = \{A_0, B_1\}$ is the final/accepting states. These are only reached if the input are from the set $\{a^n b^n : n \in \mathbb{N}\}$.

---

[25] 4. Textbook Problem 7(c) in B.2 on Pages 522 and 523.

Use a (fooling set)-plus-(Continuation Lemma) argument to prove that the following language is *not* Regular:

$$L_5 \quad = \quad \{a^i b^j c^k \mid i = j \text{ OR } j = k\}.$$

Your argument will be a proof by contradiction.

The Continuation Lemma is Lemma 3.2 on Page 57, while the fooling set kind of argument is first utilized on Pages 76 and 77.

Proof. Assume that L_5 is regular then there is a finite automata

$$M = \{Q, \{a, b, c\}, \delta, q_0, F\}$$

such that $L(M) = L_5$. As Q is finite and $\{a^k : k \in \mathbb{N}\}$ has infinite cardinality then the function $\delta : J \subseteq Q \times \{a^k : k \in \mathbb{N}\} \mapsto Q$ cannot be injective by the pigeonhole principle. Then for some $k, j \in \mathbb{N}$ with $k \neq j$ we have $\delta(q_0, a^k) = \delta(q_0, a^j) = q$ where $q \in Q$. Then we have by the continuation lemma $\delta(q, b^i) = \delta(q_0, a^k b^i) = \delta(q_0, a^j b^i) = q_1$ where $i \in \mathbb{N}$. Again by the pigeonhole principle we have that there exists $t, s \in \mathbb{N}$ where $t \neq s$ and $\delta(q_1, c^s) = \delta(q_1, c^t)$. Then this would imply $\delta(q_0, a^k b^i c^s) = \delta(q_0, a^j b^i c^t)$ □
