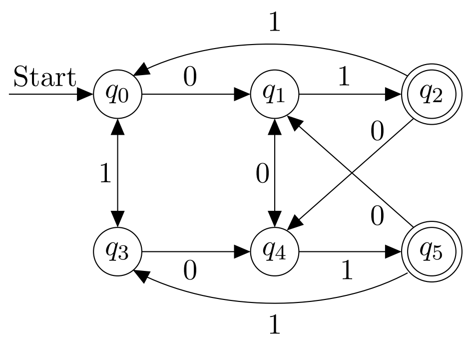
Master Degree in Computer Engineering

Final Exam for Automata, Languages and Computation

January 25th, 2021

1. **[6 points]** Consider the DFA A whose transition function is graphically represented as follows (arcs with double direction represent two arcs in opposite directions)



   (a) Provide the definition of equivalent pair of states for a DFA.

   (b) Apply to A the tabular algorithm for detecting pairs of equivalent states, reporting **all the intermediate steps**.

   (c) Specify the minimal DFA equivalent to A.

Solution

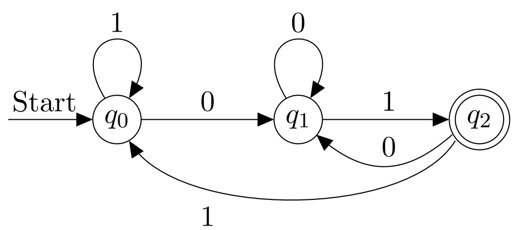
(a) The required definition can be found in the textbook.

(b) The textbook describes an inductive algorithm for detecting distinguishable state pairs. On input A, the algorithm constructs the table reported below.

q_1	Y				
q_2	X	X			
q_3		Y	X		
q_4	Y		X	Y	
q_5	X	X		X	X
	q_0	q_1	q_2	q_3	q_4

We have marked with X the entries in the table corresponding to distinguishable state pairs that are detected in the base case of the algorithm, that is, state pairs that can be distinguished by the string ε. We have then marked with Y distinguishable state pairs detected at the next iteration by some string of length one. At the successive iterations, strings of length larger than one do not provide any new distinguishable state pairs.

(c) From the above table we get the following state equivalences: $q_0 \equiv q_3$, $q_1 \equiv q_4$ and $q_2 \equiv q_5$. The minimal DFA equivalent to A is then



2. **[8 points]** For a symbol X and a string x, we write $\#_X(x)$ to denote the number of occurrences of X in x. Consider the following languages, defined over the alphabet $\Sigma = \{a, b, c\}$

$$\begin{aligned} L_1 &= \{w \mid w \in \Sigma^*,\ \#_a(w) < \#_b(w),\ \#_a(w) < \#_c(w)\} \ ; \\ L_2 &= \{w \mid w \in \Sigma^*,\ \#_a(w) < \#_b(w) \text{ or } \#_a(w) < \#_c(w)\} \ . \end{aligned}$$

State whether these languages are context-free, and provide a mathematical proof of your answer.

Solution

(a) L_1 is not a context-free language. To prove this statement, we start by intersecting L_1 with the language generated by the regular expression $\mathbf{a}^*\mathbf{b}^*\mathbf{c}^*$. It is easy to see that the resulting language is

$$L_1' = \{a^n b^p c^q \mid n, p, q \geq 0,\ n < p,\ n < q\} \ .$$

If L_1 is a context-free language, then L_1' should also be a context-free language, because of the closure of context-free languages under the intersection with regular languages. We now prove that L_1' is not a context-free language, establishing therefore a contradiction. We use the pumping lemma for context-free languages.

Let N be the pumping lemma constant. We choose the string $z = a^N b^{N+1} c^{N+1} \in L_1'$ and consider all possible factorizations $z = uvwxy$ satisfying the conditions $|v| + |x| \geq 1$ and $|vwx| \leq N$. Because of the latter condition, we have that vx can contain occurrences of at most two symbols from Σ, and these two symbols can be either a and b or else b and c, but not a and c. We separately discuss all possible cases in what follows.

- If v contains two symbol X and Y from Σ, it is easy to see that any string uv^kwx^ky with $k \geq 2$ will not belong to L_1', because of alternating occurrences of X and Y. A similar argument holds if x contains two symbol from Σ.

- If vx contains at most one symbol X from Σ, we distinguish two scenarios. If $X = a$, the string uv^kwx^ky with $k \geq 2$ will not belong to L'_1, because the number of occurrences of b or c will not be larger than the number of occurrences of a. If $X = b$ or $X = c$, the string uv^kwx^ky with $k = 0$ will not belong to L'_1, because the number of occurrences of b or c will not be larger than the number of occurrences of a.
- If v contains only X and y contains only Y, $X \neq Y$, then there must be a symbol $Z \in \Sigma$ such that Z does not occur in v and in x. We have already excluded the case $Z = b$, because $|vwx| \leq N$. If $Z = a$, the string uv^kwx^ky with $k = 0$ will not belong to L'_1, because the number of occurrences of b or c will not be larger than the number of occurrences of a. If $Z = c$, the string uv^kwx^ky with $k \geq 2$ will not belong to L'_1, because the number of occurrences of a will be larger than the number of occurrences of c.

We thus conclude that L'_1 is not a context-free language.

(b) L_2 is a context-free language. To prove this statement, let us define

$$\begin{aligned} L'_2 &= \{w \mid w \in \Sigma^*,\ \#_a(w) < \#_b(w)\}\ , \\ L''_2 &= \{w \mid w \in \Sigma^*,\ \#_a(w) < \#_c(w)\}\ , \end{aligned}$$

and observe that $L_2 = L'_2 \cup L''_2$. It is easy to show that both L'_2 and L''_2 are context-free languages, by providing push-down automata that recognize these languages. Since the class of context-free languages is closed under union, we have that L_2 is a context-free language.

3. **[5 points]** Assess whether the following statements are true or false, providing a mathematical proof for all of your answers.

(a) If $L_1 \cup L_2$ is a context-free language then also L_1 and L_2 are context-free languages.

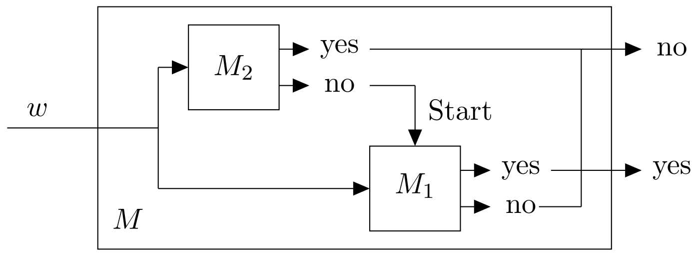
(b) Given two languages L_1 and L_2 in REC, the language $L_1 \smallsetminus L_2$ is always in REC.

(c) Given two languages L_1 and L_2 in RE, the language $L_1 \smallsetminus L_2$ is always in RE.

Solution

(a) False. We know that if L_1 and L_2 are context-free languages, then $L_1 \cup L_2$ is a context-free language as well. However, the inverse implication which is proposed in the exercise does not hold in general. To see this, we can consider a counterexample. Let $\Sigma = \{a, b, c\}$ and let $L_1 = \{a^nb^nc^n \mid n \geq 0\}$ and $L_2 = \{a^ib^jc^k \mid i, j, k \geq 0,\ i, j, k \text{ are not equal}\}$. It is not difficult to see that $L_1 \cup L_2 = \{a^ib^jc^k \mid i, j, k \geq 0\}$, which is a regular language and therefore a context-free language. However, both L_1 and L_2 are not context-free languages, as can easily be shown using the pumping lemma for context-free languages.

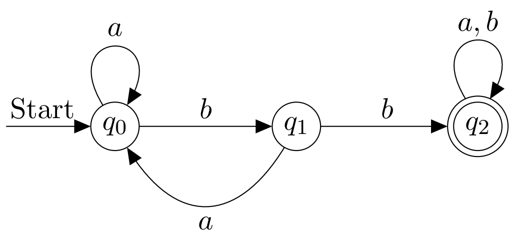
(b) True. Since L_1 and L_2 are in REC, there exists TMs M_1 and M_2 that always halt and such that $L(M_1) = L_1$ and $L(M_2) = L_2$. We construct a TM M that always halts and such that $L(M) = L_1 \smallsetminus L_2$. The main idea underlying the definition of M is exemplified by the following block digram, using the technique of lazy evaluation already exploited in several exercises in the course lectures



On input w, M simulates M_2 on w. If M_2 accepts, then M rejects, according to the definition of set difference. If M_2 rejects, then M starts the simulation of M_1 on w. If M_1 accepts, then M also accepts; otherwise M rejects, according to the definition of set difference. It is easy to show that M always halts.

(c) False. Consider the following counterexample. Let $L_1 = \{0,1\}^*$, which is a regular language and therefore also a RE language. Let also L_2 be some language in RE$\smallsetminus$REC. It is not difficult to see that $L_1 \smallsetminus L_2 = \overline{L_2}$, where $\overline{L_2}$ is the complement of L_2. We know from a theorem in the textbook that the complement of a language in RE$\smallsetminus$REC is not an RE language. We therefore conclude that $\overline{L_2}$ is not in RE.

4. **[6 points]** Consider the DFA A whose transition function δ is graphically represented as



(a) Describe in words the language L recognized by A.

(b) For each state q of A, provide a definition for properties $\mathcal{P}_q$ in such a way that, for any string $x \in \{a,b\}^*$, we have

$$\mathcal{P}_q(x) \Leftrightarrow \hat{\delta}(q_0, x) = q \ .$$

(c) Using mutual induction, prove the relation $\mathcal{P}_{q_2}(x) \Rightarrow \hat{\delta}(q_0, x) = q_2$.

Solution

(a) DFA A accepts the language L defined as the set of all strings over $\{a,b\}$ that contain bb as a substring.

(b) We define the required properties as follows

- For every $x \in \{a,b\}^*$, $\mathcal{P}_{q_0}(x)$ holds if and only if $x \notin L$ and x does not end with a b.
- For every $x \in \{a,b\}^*$, $\mathcal{P}_{q_1}(x)$ holds if and only if $x \notin L$ and x ends with a b.
- For every $x \in \{a,b\}^*$, $\mathcal{P}_{q_2}(x)$ holds if and only if $x \in L$.

(c) Proof of $\mathcal{P}_{q_2}(x) \Rightarrow \hat{\delta}(q_0, x) = q_2$. The proof is by induction on the length of x.

Base. We have $x = \varepsilon$. Since $\mathcal{P}_{q_2}(x)$ is false, the implication is true.

Induction. Let $|x| = n > 0$. If $\mathcal{P}_{q_2}(x)$ is true then $x \in L$. Let us write $x = yY$ with $Y \in \{a, b\}$ and $y \in \{a, b\}^*$, $|y| = n - 1$. We distinguish two cases.

- If $y \in L$, then $\mathcal{P}_{q_2}(y)$ holds true. We then apply the inductive hypotheses ($|y| = n - 1$) and conclude that $\hat{\delta}(q_0, y) = q_2$. We can write $\hat{\delta}(q_0, x) = \delta(\hat{\delta}(q_0, y), Y) = \hat{\delta}(q_2, Y) = q_2$ for any $Y \in \{a, b\}$.
- If $y \notin L$, then the only possible scenario is that $Y = b$ and y ends with a b. By definition we have $\mathcal{P}_{q_1}(y)$ and, using mutual induction ($|y| = n - 1$), we have $\hat{\delta}(q_0, y) = q_1$. We can write $\hat{\delta}(q_0, x) = \delta(\hat{\delta}(q_0, y), b) = \hat{\delta}(q_1, b) = q_2$.

5. **[8 points]** Let $L_1 = \{0^n 1^n \mid n \geq 1\}$. Define the following property of the RE languages

$$\mathcal{P} \;=\; \{L \mid L \in \text{RE},\ L \cap L_1 = \emptyset\}\ ,$$

and let $\overline{\mathcal{P}}$ be the complement of $\mathcal{P}$. Assess whether the languages $L_{\mathcal{P}}$ and $L_{\overline{\mathcal{P}}}$ belong to the classes REC, RE$\smallsetminus$REC, or are outside of RE, and provide a proof of your answers.

Solution The property $\mathcal{P}$ is not trivial. There are several ways to prove this. For instance, the language $L = \emptyset$ is a regular language and therefore also a RE language. We have $L \in \mathcal{P}$, since $L \cap L_1 = \emptyset$. Furthermore, the language $L' = \{0, 1\}^*$ is a regular language and therefore also a RE language. We have $L' \notin \mathcal{P}$, since $L' \cap L_1 = L_1 \neq \emptyset$. By applying Rice's theorem, we can conclude that $L_{\mathcal{P}}$ is not in REC.

We have $\overline{\mathcal{P}} = \{L \mid L \in \text{RE},\ L \cap L_1 \neq \emptyset\}$. Thus, a TM M accepts a language in $\overline{\mathcal{P}}$ if and only if $\mathsf{enc}(M) \notin L_{\mathcal{P}}$. We then cnclude that $L_{\overline{\mathcal{P}}} = \overline{L_{\mathcal{P}}}$.

We know from a theorem in the textbook that the complement of a language not in REC cannot be in REC. Since we have already assessed that $L_{\mathcal{P}}$ is not in REC, we must conclude that $L_{\overline{\mathcal{P}}}$ is not in REC as well.

We have that $L_{\overline{\mathcal{P}}}$ is a RE language. To see this, consider a NTM $N_{\overline{\mathcal{P}}}$ that, on input $\mathsf{enc}(M)$, guesses a string $w \in L_1$ and simulates M on w, accepting if and only if M accepts. It is easy to see that $L(N_{\overline{\mathcal{P}}}) = L_{\overline{\mathcal{P}}}$. Since $N_{\overline{\mathcal{P}}}$ can be transformed into a (deterministic) TM, we have thus shown that $L_{\overline{\mathcal{P}}}$ is a RE language.

Since $L_{\overline{\mathcal{P}}}$ is in RE$\smallsetminus$REC, we can apply a known theorem from the textbook and conclude that $L_{\mathcal{P}}$ is not an RE language. This concludes the exercise.