CSE 105: Homework #1

Due on Jan 19, 2023 at 23:59pm

Professor Minnes

Ray Tsai

A16848188

Problem 1

With $\Sigma_1 = \{0, 1\}$ and $\Sigma_2 = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$ and $\Gamma = \{0, 1, x, y, z\}$

(a) Give an example of the shortest string over Σ_1 that is meaningful to you in some way, and explain why it's meaningful to you.

Solution. The empty string $\epsilon_1 \in \Sigma_1^*$ is meaningful to me because it's the shortest string. □

(b) List all examples of strings of length 1 over Σ_2 and explain why your list is exhaustive.

Solution. Since a string of length 1 is simply a string of a single symbol, each symbol and only the symbol in Σ_2 are strings of length 1, namely

$$a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z.$$

□

(c) Calculate the number of distinct strings of length 3 over Γ and explain your calculation.

Solution. Since there are 5 choices for each character there are 3 characters in the string, there are $5^3 = 125$ distinct strings of length 3 over Γ. □

(d) With the ordering $x < y < z < 0 < 1$, list the first ten strings over Γ in string order.

Solution. $\epsilon, x, y, z, 0, 1, xx, xy, xz, x0$ □

(e) Give an example of a finite set that is a language over Σ_1 and over Σ_2 and over Γ, or explain why there is no such set.

Solution. Since $\Sigma_1^* \cup \Sigma_2^* \cup \Gamma^* = \{\epsilon\}$, $\{\epsilon\}$ is a language over all three alphabets. □

(f) Give an example of an infinite set that is a language over Σ_1 and over Γ, or explain why there is no such set.

Solution. $(\Sigma_1 \cap \Gamma)^*$ is a language over Σ_1 and over Γ. Since $\Sigma_1 \cap \Gamma$ is nonempty, $(\Sigma_1 \cap \Gamma)^*$ is an infinite set. □

Problem 2

(a) Give three regular expressions that all describe the set of all strings over $\{a, b\}$ that have even length. Ungraded bonus challenge: Make the expressions as different as possible!

Solution. $((a \cup b)(a \cup b))^*, ((aa) \cup (ab) \cup (ba) \cup (bb))^*, (aa)^* \cup (ab)^* \cup (ba)^* \cup (bb)^*$. □

(b) A friend tells you that each regular expression that has a Kleene star (*) describes an infinite language. Are they right? Either help them justify their claim or give a counterexample to disprove it and then fix the formula.

Proof. Yes they are correct. Suppose for the sake of contradiction that Σ^* is finite, for a nonempty set Σ. Let σ be concatenation of all strings in Σ^*. Consider σa, for some $a \in \Sigma$. σa is a string over Σ, so $\sigma a \in \Sigma^*$. However, σa does not equal to any string in Σ^*, contradiction. Therefore, Σ^* must be infinite. □

Problem 3

For languages L_1, L_2 over the alphabet $\Sigma_1 = \{0, 1\}$, we have the associated sets of strings

$$SUBSTRING(L_1) = \{w \in \Sigma_1^* \mid \text{there exist } a, b \in \Sigma_1^* \text{ such that } awb \in L_1\}$$

and

$$L_1 \circ L_2 = \{w \in \Sigma_1^* \mid w = uv \text{ for some strings } u \in L_1 \text{ and } v \in L_2\}$$

(a) Specify an example language A over Σ_1 such that $A \neq \emptyset$ and yet $\textsc{Substring}(A) = \emptyset$, or explain why there is no such example. A complete solution will include either (1) a precise and clear description of your example language A and a precise and clear description of the result of computing $\textsc{Substring}(A)$ using relevant definitions to justify this description and to justify the set equality with $\emptyset$, or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

Solution. There are no such examples. Since $A \neq \emptyset$, let $w \in A$. There exists $a = \epsilon$, $b = \epsilon$ in Σ_1^* such that $awb = w \in A$, so $w \in \textsc{Substring}(A)$. Thus, $\textsc{Substring}(A)$ must be nonempty. □

(b) Specify example languages B, C over Σ_1 such that $B \neq \Sigma_1^*$ and $C \neq \Sigma_1^*$ and yet $B \circ C = \Sigma_1^*$, or explain why there are no such examples. A complete solution will include either (1) a precise and clear description of your example languages B, C and a precise and clear description of the result of computing $B \circ C$ using relevant definitions to justify this description and to justify the set equality with Σ_1^*, or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

Solution. Consider $B = \{w \in \Sigma_1^* \mid |w| \text{ is even}\}$ and $C = \{w \in \Sigma_1^* \mid |w| \text{ is odd}\} \cup \{\epsilon\}$. Let $b \in B$ and $c \in C$. When $c \neq \epsilon$, we know bc is of odd length. Otherwise, $bc = b$ is of even length. Hence, we get

$$B \circ C = \{w \in \Sigma_1^* \mid |w| \text{ is even}\} \cup \{w \in \Sigma_1^* \mid |w| \text{ is odd}\} = \Sigma_1^*.$$

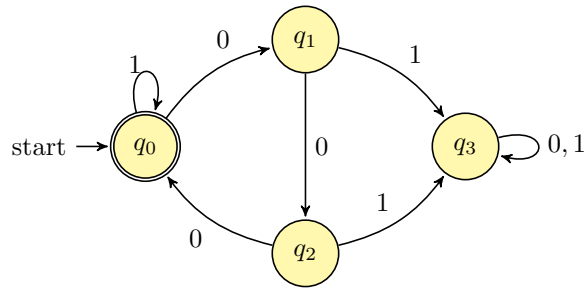
□

(c) Specify example **finite** languages L_1, L_2 over Σ_1 such that $L_1 \circ L_2 \neq L_1$ but $|L_1 \circ L_2| = |L_1|$, or explain why there are no such examples. A complete solution will include either (1) a precise and clear description of your example languages L_1, L_2 and a precise and clear description of the result of computing $L_1 \circ L_2$ using relevant definitions to justify this description and to justify the cardinality claims and set (in)equality claims, or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

Solution. Consider $L_1 = \{\epsilon\}$ and $L_2 = \{1\}$. $L_1 \circ L_2 = \{1\} \neq L_1$, but $\mid L_1 \circ L_2 \mid = \mid \{1\} \mid = \mid L_1 \mid$. □

Problem 4

Consider the finite automaton $(Q, \Sigma, \delta, q_0, F)$ whose state diagram is depicted below



where $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, and $F = \{q_0\}$, and $\delta : Q \times \Sigma \to Q$ is specified by the look-up table

	0	1
q_0	q_1	q_0
q_1	q_2	q_3
q_2	q_0	q_3
q_3	q_3	q_3

(a) A friend tries to summarize the transition function with the formula

$$\delta(q_i, x) = \begin{cases} q_0 & \text{when } i = 0 \text{ and } x = 1 \\ q_3 & \text{when } 0 < i \leq 3 \text{ and } x = 1 \\ q_j & \text{when } j = (i+1) \mod 3 \text{ and } x = 0 \end{cases}$$

Are they right? Either help them justify their claim or give a counterexample to disprove it.

Solution. The output $\delta(q_3, 0) = q_1$ is incorrect. It should be q_3 instead. □

(b) Give a regular expression R so that $L(R)$ is the language recognized by this finite automaton. Justify your answer by referring to the definition of the semantics of regular expressions and computations of finite automata. Include an explanation for why each string in $L(R)$ is accepted by the finite automaton *and* for why each string not in $L(R)$ is rejected by the finite automaton.

Solution. Consider $R = (1 \cup 000)^*$. Suppose $w \in L(R)$. We may assume $w \neq \epsilon$, as the start state is the end state so ϵ would be accepted. Then, w consists of sequences of 1 and 000. Denote $\delta(\delta(\delta(q_0, 0), 0), 0)$ as $\delta(q_0, 000)$. At the start state, if the automaton reads 000, we get $\delta(q_0, 000) = q_0$, so we end up at the start state. Otherwise, the automaton reads 1 and $\delta(q_0, 1) = q_0$. Thus, w is guaranteed to end up at the start state, which is also the end state, so any string $w \in L(R)$ would be accepted.

Now suppose that $w \notin L(R)$. Since the prefix of the form 1 or 000 takes no effect, we may removed them from w. If $w = 0$ or $w = 00$, then $\delta(q_0, 0) = q_1$ and $\delta(\delta(q_0, 0), 0) = q_2$, which would get rejected because it doesn't end in the accept state. Thus, we may assume w starts with 01 or 001. Then, $\delta(\delta(q_0, 0), 1) = \delta(\delta(\delta(q_0, 0), 0), 1) = q_3$. However, note that $\delta(q_3, x) = q_3$ for all $x \in \Sigma$, so the automaton would get stuck in q_3 once it is reached, which implies that w would never end up in the accept state. Thus, w would get rejected by the automaton. □

Problem 4 continued on next page. . .

(c) Keeping the same set of states $Q = \{q_0, q_1, q_2, q_3\}$, alphabet $\Sigma = \{0, 1\}$, same start state q_0, and same transition function δ, choose a new set of accepting states F_{new} so that the new finite automaton that results accepts at least one string that the original one rejected **and** rejects at least one string that the original one accepted, or explain why there is no such choice of F_{new}. A complete solution will include either (1) a precise and clear description of your choice of F_{new} and a precise and clear the two example strings using relevant definitions to justify them, or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

Solution. Consider $F_{new} = \{q_3\}$. Since 01 ends in q_3, it is accepted by the new automaton but not by the original one. Since all strings that the original automaton accepts end in q_0, they would all be rejected by the new automaton. $\square$