CS320: Problems and Solutions for Day 5, Winter 2023

Problem 1 Let L be the language over alphabet $\{a, b, c\}$ consisting of all strings that do not contain ab as a substring.

(a) Construct a finite automaton M that accepts L. If such an automaton M does not exist, explain why.

Answer: See figure Figure 1.

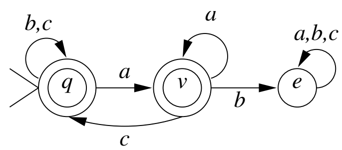

Figure 1:

(b) If you constructed an automaton M in your answer to part (a), is M deterministic? Justify briefly your answer.

Answer: Yes. In its transition graph, exactly one arc labeled by each alphabet symbol is incident out of every node, while there are no other arcs. This corresponds to a total state-transition function that maps state-symbol pairs to states.

Problem 2 Let L be the language defined by the regular expression

$$\boldsymbol{a^* (ab \cup ba \cup e) b^*}$$

(a) Construct a finite automaton M that accepts L.

Answer: See Figure 2.

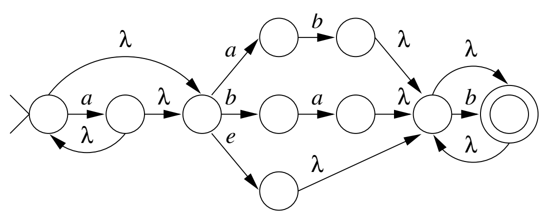

Figure 2:

(b) Is the finite automaton M that you constructed in your answer to part (a) deterministic? Justify briefly your answer.

Answer: No. For example, it does not have a b-transition out of the initial state; it has λ-transitions.

Problem 3 Let M be the finite automaton represented by the state diagram on Figure 3, and let L be the language accepted by M.

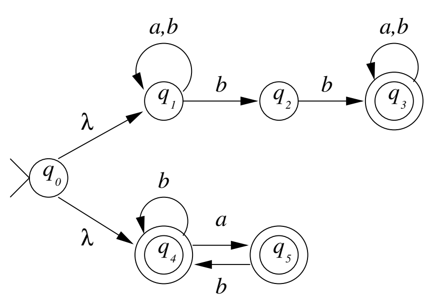

Figure 3:

(a) Write 10 distinct strings that belong to L.

(b) Write 10 distinct strings over alphabet $\{a, b\}$ that do not belong to L.

Answer: Note that L is the set of strings over $\{a, b\}$ that do not contain aa as a substring or contain bb as a substring. $\bar{L}$ is, therefore, the set of strings over $\{a, b\}$ that do not contain bb as substring, but do contain aa as a substring.

$\in L$	$\notin L$
a	aa
b	aab
$bbaa$	$aaab$
ba	$aabababab$
bb	$baab$
$bbaaa$	$babaa$
$babab$	$baabab$
$abbb$	$abaa$
$bbab$	$aabaa$
$aaaaaaaabb$	$abaaab$

Problem 4 Let M be the finite automaton represented by the state diagram on Figure 4, and let L be the language accepted by M.

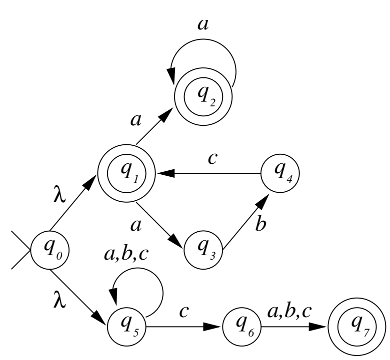

Figure 4:

(a) Write 10 distinct strings that belong to L.

(b) Write 10 distinct strings over alphabet $\{a, b, c\}$ that do not belong to L.

Answer: Note that L is represented by a regular expression:

$$(\boldsymbol{abc})^*\boldsymbol{a}^* \cup ((\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c})^*\boldsymbol{c}(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c}))$$

$\in L$	$\notin L$
a	b
aa	bb
abc	c
$abca$	$aabc$
$abcabc$	bbc
ca	$abcbb$
cb	cba
$abcabcaa$	bc
$abcabcbbbcc$	ac
$bbbbca$	abb

Problem 5 Let L be the language defined by the regular expression:

$$(\boldsymbol{a} \cup \boldsymbol{bc})^*(\boldsymbol{dd} \cup \boldsymbol{g})^* \cup \boldsymbol{ab}^*$$

Construct a state-transition graph of a finite automaton M that accepts L. If such automaton does not exist, prove it.

Answer: See Figure 5.

Problem 6 Let L be the language defined by the regular expression:

$$(\boldsymbol{a} \cup \boldsymbol{c})^* \, (\boldsymbol{b} \cup \boldsymbol{c})^* \cup \boldsymbol{bbb}$$

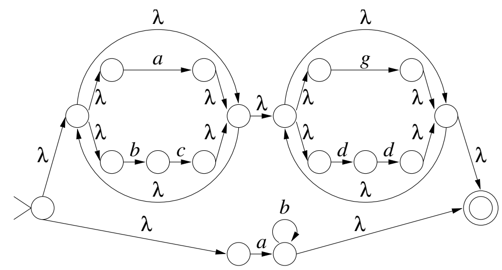

Figure 5:

(a) Construct a state-transition graph of a finite automaton that accepts L. If such automaton does not exist, prove it.

Answer: See Figure 6.

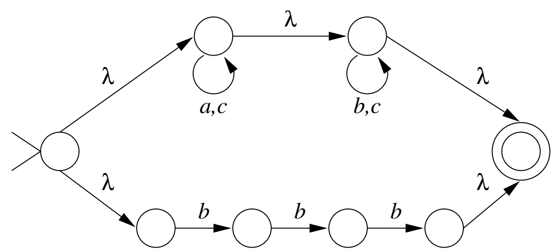

Figure 6:

(b) Does there exist an algorithm to convert an arbitrary regular expression into an equivalent finite automaton? Explain your answer briefly.

Answer: Yes. The construction is performed by a recursive decomposition of the regular expression into two expressions connected by the outer-most regular operator. If such two expressions, R_1 and R_2, are converted into two (canonical) finite automata, M_1 and M_2, with initial states q_1, q_2 respectively, and final states f_1, f_2 respectively, then their composition is converted to an automaton M, whose new initial and final states are q, f respectively, so that the composition is effected by new transitions, which are:

1. for union $R_1 \cup R_2$:

$$[q, \lambda, q_1]\,; \quad [q, \lambda, q_2]\,;$$
$$[f_1, \lambda, f]\,; \quad [f_2, \lambda, f]\,;$$

2. for concatenation $R_1 \cdot R_2$:

$$[q, \lambda, q_1]\,;$$
$$[f_1, \lambda, q_2]\,;$$
$$[f_2, \lambda, f]\,;$$

3. for Kleene star R_1^*:

$$[q, \lambda, q_1]\,;$$
$$[q_1, \lambda, f_1]\,;$$
$$[f_1, \lambda, q_1]\,;$$
$$[f_1, \lambda, f]\,;$$

In the base case, $\emptyset$ is accepted by an automaton with the empty transition set, empty string is accepted by an automaton whose only transition is of the form:

$$[q, \lambda, f]$$

and an individual letter is accepted by an automaton whose only transition is of the form:

$$[q, a, f]$$

Problem 7 Let L be the set of strings over alphabet $\{a, b, c\}$ that do not contain the substring aa.

(a) Draw a state-transition graph of a finite automaton that accepts L. If such automaton does not exist, prove it.

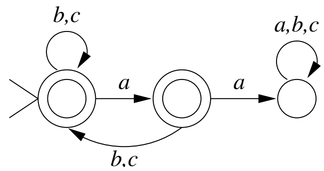

Figure 7:

Answer: See Figure 7.

(b) Is the complement $\overline{L}$ of the language L countable? Explain your answer briefly.

Answer: Yes—every language is countable, as a subset of the set Σ^*, which is countable because it is the set of all finite-length sequences over a countable set.

Problem 8 Let L be the set of strings over alphabet $\{a, b, c\}$ that do not start with c and do not end with a.

(a) Draw a state-transition graph of a finite automaton that accepts $\overline{L}$ (the complement of L). If such automaton does not exist, prove it.

Answer: See Figure 8.

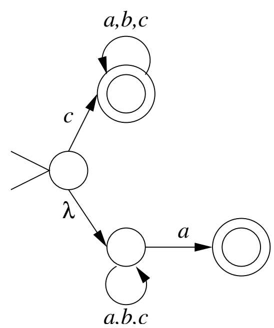

Figure 8:

(b) Does there exist an algorithm that solves the following problem:

INPUT: An arbitrary regular expression e.

OUTPUT: A finite automaton M_1 that accepts the complement of the language defined by e.

Explain your answer briefly.

Answer: Yes—such algorithm starts out by converting the regular expression into an equivalent finite automaton; next, this automaton is converted into an equivalent deterministic finite automaton; finally, the deterministic finite automaton is converted into one that accepts the complement.

Problem 9 Let L be the set of all strings over alphabet $\{a, b\}$ in which all a's come before all b's, and the number of a's is odd but the number of b's is even.

(a) Draw a state-transition graph of a finite automaton that accepts L. If such automaton does not exist, prove it.

Answer: See Figure 9.

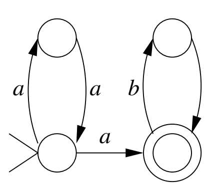

Figure 9:

(b) Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b\}$,
$V = \{S, A, B\}$, and the production set P is:

$$\begin{aligned} &S \to AB \\ &A \to aaA \mid a \\ &B \to bbB \mid \lambda \end{aligned}$$

Problem 10 Let L be the set of strings over alphabet $\{a, b, c\}$ whose total number of a's and b's is at least 2.

(a) Write a regular expression that defines L. If such a regular expression does not exist, explain why.

Answer:

$$(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c})^*(\boldsymbol{a} \cup \boldsymbol{b})(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c})^*(\boldsymbol{a} \cup \boldsymbol{b})(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c})^*$$

(b) Construct a state transition graph of a finite automaton that accepts L. If such an automaton does not exist, explain why.

Answer: See Figure 10.

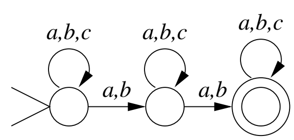

Figure 10:

(c) Write a complete formal definition of a context-free grammar G that generates L. If such a grammar does not exist, explain why.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$,
$V = \{S, B, A\}$, and the production set P is:

$$\begin{aligned} &S \to BABAB \\ &B \to BB \mid \lambda \mid a \mid b \mid c \\ &A \to a \mid b \end{aligned}$$

Problem 11 Let L be the set of all nonempty strings over alphabet $\{a, b, d\}$ whose first symbol is equal to the third symbol.

(a) Write a regular expression that defines L. If such a regular expression does not exist, prove it.

Answer:

$$\begin{gathered} \boldsymbol{a}(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{d})\boldsymbol{a}\,(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{d})^* \\ \bigcup \\ \boldsymbol{b}(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{d})\boldsymbol{b}\,(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{d})^* \\ \bigcup \\ \boldsymbol{d}(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{d})\boldsymbol{d}\,(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{d})^* \end{gathered}$$

(b) Construct a state-transition graph of a finite automaton that accepts L. If such an automaton does not exist, prove it.

Answer: See Figure 11.

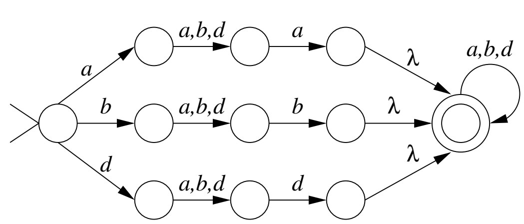

Figure 11: