

CS/ECE 374 A (Spring 2022)

Homework 2 Solutions

Problem 2.1: For each of the following languages, give a regular expression that describes that language, and briefly argue why your expression is correct. Below, $\#_0(x)$ denotes the number of 0's in x , and $\#_1(x)$ denotes the number of 1's in x .

- (a) All strings in $\{0, 1\}^*$ of length at least 5 such that the symbols at positions divisible by 5 are all 0's. For example: the string 1001**0**1111**00**101**0**11 is in the language because the bold symbols are all 0's.
- (b) All strings $x \in \{0, 1\}^*$ such that x does not begin with 101 and $\#_0(x)$ is divisible by 3.
- (c) All strings in $\{0, 1\}^*$ that do not contain 0111 as a substring.
- (d) All strings in $\{0, 1, 2\}^*$ that have at least four 0's and exactly two 1's.

Solution:

- (a) The regular expression $((0 + 1)^4 0)^+$ corresponds to all strings with positive length divisible by 5 such that every 5th symbol is a 0. To get strings of length not divisible by 5, we append $(\varepsilon + 0 + 1)^4$ (all strings length at most 4) at the end. Answer:

$$((0 + 1)^4 0)^+ \cdot (\varepsilon + 0 + 1)^4.$$

[Note: there are many alternative ways to rewrite this regular expression, e.g., $((0 + 1)^4 0)^+ \cdot (\varepsilon + (0 + 1) + (0 + 1)^2 + (0 + 1)^3 + (0 + 1)^4)$.]

- (b) We divide into the following cases, which cover all possibilities for strings not beginning with 101 (assuming that the string x has length at least 3): (i) x begins with 0, (ii) x begins with 11, (iii) x begins with 100. In case (i), we want the number of 0's after the initial 0 to be congruent to 2 mod 3. In case (ii), we want the number of 0's after the initial 11 to be congruent to 0 mod 3. In case (iii), we want the number of 0's after the initial 100 to be congruent to 1 mod 3. for strings of length less than 3, In addition, the following extra strings of length less than 3 need to be included: ε , 1. Answer:

$$\begin{aligned} & 0 \cdot (1^* 0 1^* 0 1^* 0)^* 1^* 0 1^* 0 1^* \\ & + 11 \cdot (1^* 0 1^* 0 1^* 0)^* 1^* \\ & + 100 \cdot (1^* 0 1^* 0 1^* 0)^* 1^* 0 1^* \\ & + \varepsilon + 1. \end{aligned}$$

[Note: there are many alternative ways to rewrite this regular expression, e.g., $(0 1^* 0 1^* 0 + 11 + 100 1^* 0) 1^* (0 1^* 0 1^* 0)^* + \varepsilon + 1$.]

- (c) Break the input string into alternating blocks of 0's and blocks of 1's. The criterion of not containing 0111 is equivalent to say that all blocks of 1's have length at most 2, except for the first block (the string may begin with a block of 1's of arbitrary length).

Answer:

$$1^*(0^+(1+11))^*0^*.$$

[Note: there are many equivalent answers, e.g., $1^*(0+01+011)^*$.]

- (d) Let x be the number of 0's before the first 1. Let y be the number of 0's between the two 1's. Let z be the number of 0's after the second 1. Since we want $x+y+z \geq 4$, we divide into the following cases:

- $x = 0, y = 0, z \geq 4$
- $x = 0, y = 1, z \geq 3$
- $x = 0, y = 2, z \geq 2$
- $x = 0, y = 3, z \geq 1$
- $x = 0, y \geq 4, z \geq 0$
- $x = 1, y = 0, z \geq 3$
- $x = 1, y = 1, z \geq 2$
- $x = 1, y = 2, z \geq 1$
- $x = 1, y \geq 3, z \geq 0$
- $x = 2, y = 0, z \geq 2$
- $x = 2, y = 1, z \geq 1$
- $x = 2, y \geq 2, z \geq 0$
- $x = 3, y = 0, z \geq 1$
- $x = 3, y \geq 1, z \geq 0$
- $x \geq 4, y \geq 0, z \geq 0$

Answer:

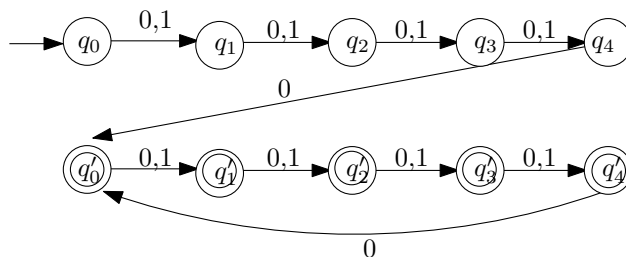
	2^*	1	2^*	1	$2^*02^*02^*0(0+2)^*$
+	2^*	1	2^*02^*	1	$2^*02^*02^*0(0+2)^*$
+	2^*	1	$2^*02^*02^*$	1	$2^*02^*0(0+2)^*$
+	2^*	1	$2^*02^*02^*02^*$	1	$2^*0(0+2)^*$
+	2^*	1	$2^*02^*02^*02^*0(0+2)^*$	1	$(0+2)^*$
+	2^*02^*	1	2^*	1	$2^*02^*02^*0(0+2)^*$
+	2^*02^*	1	2^*02^*	1	$2^*02^*0(0+2)^*$
+	2^*02^*	1	$2^*02^*02^*$	1	$2^*0(0+2)^*$
+	2^*02^*	1	$2^*02^*02^*0(0+2)^*$	1	$(0+2)^*$
+	$2^*02^*02^*$	1	2^*	1	$2^*02^*0(0+2)^*$
+	$2^*02^*02^*$	1	2^*02^*	1	$2^*0(0+2)^*$
+	$2^*02^*02^*$	1	$2^*02^*0(0+2)^*$	1	$(0+2)^*$
+	$2^*02^*02^*02^*$	1	2^*	1	$2^*0(0+2)^*$
+	$2^*02^*02^*02^*$	1	$2^*0(0+2)^*$	1	$(0+2)^*$
+	$2^*02^*02^*02^*0(0+2)^*$	1	$(0+2)^*$	1	$(0+2)^*$

Problem 2.2: Describe a DFA that accepts each of the following languages. Describe briefly what each state in your DFA *means*. For (a)–(c), you should draw the complete DFA. For (d), do not attempt to draw your DFA, since the number of states could be huge; instead, give a mathematically precise description of the states Q , the start state s , the accepting states A , and the transition function δ .

- (a) (20 pts) All strings in $\{0,1\}^*$ of length at least 5 such that the symbols at positions divisible by 5 are all 0's.
- (b) (20 pts) All strings $x \in \{0,1\}^*$ such that x does not begin with 101 and $\#_0(x)$ is divisible by 3.
- (c) (20 pts) All strings in $\{0,1\}^*$ that do not contain 011001 as a substring.
- (d) (40 pts) All strings $x \in \{0,1\}^*$ such that x contains 0^9 as a substring, or $|x|$ is divisible by 8, or $\#_0(x) - \#_1(x)$ is divisible by 7.

Solution:

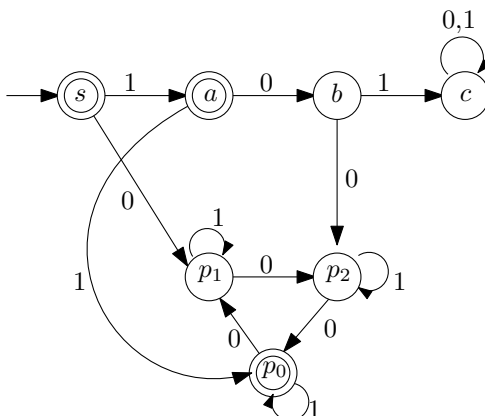
(a)



Meaning of the states:

- q_i ($i \in \{0, 1, 2, 3, 4\}$): the string read so far has length i .
- q'_i ($i \in \{0, 1, 2, 3, 4\}$): the string read so far has length at least 5 and congruent to $i \bmod 5$, and symbols at positions divisible by 5 are all 0's.

(b)



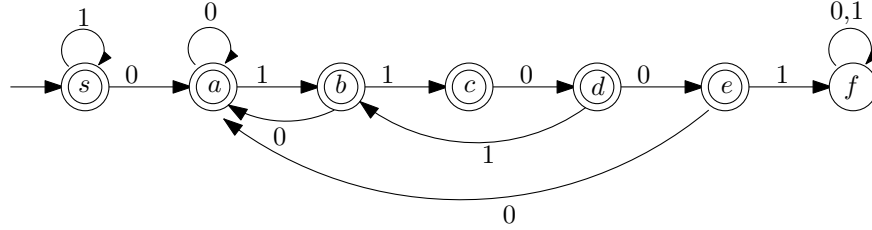
Meaning of the states:

- s : start state.

- a : read 1.
- b : read 10.
- c : string begins with 101.
- p_i ($i \in \{0, 1, 2\}$): string begins with 0 or 11 or 100, and the number of 0's seen so far is congruent to $i \pmod 3$.

[Note: an alternative solution is to use a product construction, though the number of states would be a little bigger.]

(c)



Meaning of the states:

- f : 011001 has already appeared as a substring.
- e : the last 5 symbols read is 01100 but none of the above is true.
- d : the last 4 symbols read is 0110 but none of the above is true.
- c : the last 3 symbols read is 011 but none of the above is true.
- b : the last 2 symbols read is 01 but none of the above is true.
- a : the last symbol read is 0 but none of the above is true.
- s : none of the above.

(d) Define DFA $M = (Q, \Sigma, s, \delta, A)$ where $\Sigma = \{0, 1\}$, and

$$\begin{aligned}
 Q &= \{(i, j, k) : 0 \leq i \leq 8, 0 \leq j < 8, 0 \leq k < 7\} \cup \{z\} \\
 s &= (0, 0, 0) \\
 A &= \{(i, j, k) \in Q : j = 0 \text{ or } k = 0\} \cup \{z\}.
 \end{aligned}$$

(The number of states is $9 \cdot 8 \cdot 7 + 1 = 505$.)

The transition function $\delta : Q \times \Sigma \rightarrow Q$ is defined as follows: for all $(i, j, k) \in Q$,

$$\begin{aligned}
 \delta((i, j, k), 0) &= \begin{cases} (i + 1, (j + 1) \bmod 8, (k + 1) \bmod 7) & \text{if } i < 8 \\ z & \text{if } i = 8 \end{cases} \\
 \delta((i, j, k), 1) &= (0, (j + 1) \bmod 8, (k - 1) \bmod 7) \\
 \delta(z, 0) &= z \\
 \delta(z, 1) &= z.
 \end{aligned}$$

Explanation: The state z means that we have encountered 0^9 as a substring. The state $(i, j, k) \in Q$ means that we haven't encountered 0^9 as a substring yet, and the number of consecutive 0's most recently seen is i , and the total number of symbols read so far is congruent to $j \pmod 8$, and the number of 0's minus the number of 1's read so far is congruent to $k \pmod 7$.