

# COMP 330 Winter 2021

## Assignment 2 Solutions

Prakash Panangaden

### Question 1[20 points]

Give regular expressions for the following languages over  $\{a, b\}$ :

1.  $\{w \mid w \text{ contains an even number of occurrences of } a\}$
2.  $\{w \mid w \text{ contains an odd number of occurrences of } b\}$
3.  $\{w \mid w \text{ does not contain the substring } ab\}$
4.  $\{w \mid w \text{ does not contain the substring } aab\}$

### Solution

- $(b^*ab^*(b^*)^*+)^*$
- $(a^*ba^*)(ba^*ba^*)^*$
- $b^*a^*$
- $b^*a^*(bbb^*a^*)^*b^*$ . Another equivalent expression is  $b^*+b^*aa^*(bbb^*aa^*)^*b^*$ .

The last one is quite tricky. An  $a$  can be followed by two or more  $b$ 's but not by a single  $b$  if you want more  $a$ 's later.

### Question 2[20 points]

Suppose that you have a DFA  $M = (S, \Sigma, s_0, \delta, F)$ . Consider two distinct states  $s_1, s_2$  i.e.  $s_1 \neq s_2$ . Suppose further that for all  $a \in \Sigma$   $\delta(s_1, a) = \delta(s_2, a)$ . Show that for any *nonempty* word  $w$  over  $\Sigma$  we have  $\delta^*(s_1, w) = \delta^*(s_2, w)$ .

**Solution** We proceed by induction on the length of the word. The base case is when  $w$  has length 1, i.e. it is a single letter. This means we must show:

$$\forall a \in \Sigma, \delta^*(s_1, a) = \delta^*(s_2, a).$$

From the definition of  $\delta^*$  this becomes just

$$\forall a \in \Sigma, \delta(s_1, a) = \delta(s_2, a).$$

But this was given to us in the problem so the base case is immediately verified.

The inductive hypothesis is that

$$\forall w \in \Sigma^*, |w| < n \text{ implies } \delta^*(s_1, w) = \delta^*(s_2, w).$$

Now we consider a word of the form  $wa$  where  $w$  has length  $n - 1$  and  $a$  is any letter in  $\Sigma$ .

$$\delta^*(s_1, wa) = \delta(\delta^*(s_1, w), a) = \delta(\delta^*(s_2, w), a) = \delta^*(s_2, wa).$$

The first equation follows from the definition of  $\delta^*$ , the second follows from the inductive hypothesis and the last from the definition of  $\delta^*$ . This verifies the inductive case, we are done.

### Question 3[20 points]

Show that the following languages are not regular by using the pumping lemma.

1.  $\{a^n b^m a^{n+m} \mid n, m \geq 0\}$ ,
2.  $\{x \mid x \in \Sigma^* \text{ and } x = x^R\}$ , where  $x^R$  means  $x$  reversed; these strings are called *palindromes*. An example is *abba*, a non-example is *baba*.

### Solution

1. The demon picks a number  $p$ . I pick the string  $a^p b a^{(p+1)}$ . Whatever the demon tries to do he is forced to pick  $y$  to consist of a nonempty string that consists exclusively of  $as$ , because  $|xy| \leq p$  and  $|y| > 0$ , say that  $y = a^l$ . Now I choose  $i = 3$ . The new string is  $a^{(p+2l)} b a^{(p+1)}$  which is not in the language since  $2l \neq 1$ . My strategy is sure to work because I have taken into account anything that the demon might do.
2. Suppose the demon picks the number  $p$ . I choose the string  $a^p b b a^p$  which is a palindrome. Now the demon tries to divide it up into  $x, y, z$  with  $|xy| \leq p$ ,  $|y| > 0$ , in such a way that  $xy^i z$  a palindrome for all  $i \geq 0$ . It must be the case that  $x = a^k$  for some  $k \in \{0, \dots, p-1\}$ ,  $y = a^j$  for some  $j \in \{1, \dots, p\}$  and  $z = a^l b b a^p$  for some  $l \in \{0, \dots, p-1\}$ . Now I choose  $i = 2$  and we see that  $xy^2 z = a^{p+j} b b a^p$ , which is not a palindrome, since  $j > 0$ . Therefore, this language is not regular.

**Question 4** [20 points] Show that the following languages are not regular by using the pumping lemma.

1.  $\{x \in \{a, b, c\}^* \mid |x| \text{ is a square.}\}$  Here  $|x|$  means the length of  $x$ .
2.  $\{a^{2^n}b^n\}$ .

**Solution**

1. The demon picks some  $p$ . I choose the string  $a^{p^2}$  which has length  $p^2$  which is indeed a square. Now the demon has to pick  $y$  to consist of  $a$ 's exclusively because there are no other letters in the string. Let  $|y| = q > 0$  and  $q \leq p$ . Now I choose  $i = 2$ . Then we get  $a^{p^2+q}$  as the pumped string. This string has length  $p^2 + q$ . Now  $q \leq p < 2p + 1$ , hence  $p^2 + q < p^2 + 2p + 1 = (p + 1)^2$ . Thus  $p^2 + q$  cannot be a square since it is strictly less than the *next* square after  $p^2$ . Thus, I win and the language is not regular.

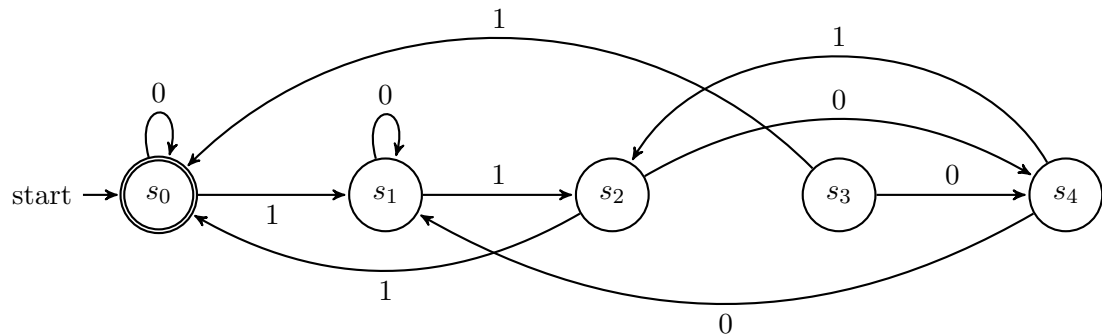
2. The demon picks some  $p$ . I pick  $a^{2p}b^p$ . The demon is forced to pick  $y$  consisting exclusively of  $a$ 's and  $|y| \leq p$ ; let  $|y| = q > 0$ . I pick  $i = 0$  so the "pumped" string is  $a^{2p-q}b^p$  and clearly this string is not in the language.

**Question 5** [20 points] We are using the alphabet  $\{0, 1\}$ . We have a DFA with 5 states,  $S = \{s_0, s_1, s_2, s_3, s_4\}$ . The start state is  $s_0$  and the only accepting state is also  $s_0$ . The transitions are given by the formula

$$\delta(s_i, a) = s_j \text{ where } j = i^2 + a \pmod{5}.$$

Draw the table showing which pairs of states are inequivalent and then construct the minimal automaton. Remember to remove useless states right from the start, before you draw the table. I am happy with a drawing of the automaton.

**Solution** Here is the original automaton:



It is clear from this picture that  $s_3$  is an unreachable state so we remove it right away.

Here is the table:

	$s_4$	$s_2$	$s_1$	$s_0$
$s_4$	/	x	x	x
$s_2$		/	x	x
$s_1$			/	x
$s_0$				/

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

After running the algorithm the pairs that cannot be equivalent are marked with an X. The diagonal is marked with a / and of course we do not need to do anything above the diagonal. There is only one pair left unmarked. According to the algorithm, these states are equivalent. This means that  $s_1$  and  $s_4$  can be lumped together to give a 3-state machine. The minimized machine is shown below:

