COMP 330 Fall 2021
Assignment 2 Solutions

Prakash Panangaden

Question 1[20 points]
Give regular expressions for the following languages over $\{a, b\}$:

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

Solution

- $(b^*ab^*ab^*)^*+b^*$ There are either no a's at all or there are some a's which must come in pairs. There can be any number (including possibly 0) of b's before the first a, between the two a's and after the second a. This pattern can be repeated.
- $(a^*ba^*)(ba^*ba^*)^*$ Similar reasoning to the previous case except that we have to have at least one b.
- b^*a^* Once you see an a you can never go back to seeing b's.
- $b^*a^*(bbb^*a^*)^*b^*$. Another equivalent expression is $b^*+b^*aa^*(bbb^*aa^*)^*b^*$. An a can be followed by two or more b's but not by a single b if you want more a's later.

The last one is quite tricky.

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 Σ 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 δ^* 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 Σ.

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

The first equation follows from the definition of δ^*, the second follows from the inductive hypothesis and the last from the definition of δ^*. 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. $\{a^n b^m a^{n+m} | n, m \geq 0\}$,
2. $\{x | x = x^R, x \in \Sigma^*\}$, 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^pbba^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^iz a palindrome for all $i \geq 0$. It must be the case that $x = a^k$ for some $k \in \{0, ..., p-1\}$, $y = a^j$ for some $j \in \{1, ..., p\}$ and $z = a^lbba^p$ for some $l \in \{0, ..., p-1\}$. Now I choose $i = 2$ and we see that $xy^2z = a^{p+j}bba^p$, 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. $\{x \in \{a, b, c\}^* | |x| \text{ is a square.}\}$ Here $|x|$ means the length of x.
2. $\{a^{2n}b^n\}$.

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] Show that the language

$$F = \{a^ib^jc^k | i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$$

is not regular. Show, however, that it satisfies the statement of the pumping lemma as I proved it in class, i.e. there is a p such that all three conditions for the pumping lemma are met. Explain why this does not contradict the pumping lemma.

Solution: First, I will show that F is not regular. If F were regular, than the intersection of F and any regular language would also be regular. We know that $L = \{ab^ic^j | i, j \geq 0\}$ is a regular language corresponding to the regular expression ab^*c^*, and clearly $L \cap F = \{ab^nc^n\}$, since any word in L

must begin with exactly one a, and so if it is also in F then $i = j$. But we know that $\{ab^n c^n\}$ is not regular by an easy variation of the example done in class, so F must not be regular.

Next I will show that F satisfies the pumping lemma. Let $p = 2$. Then the demon gives you a word w from F, of the form $a^i b^j c^k$. First, consider the case where $i = 0$, so $w = b^x c^y$. Then choose $x = \epsilon$ and $y = b$ if the first letter is a b, or $y = c$ otherwise. Let z be the rest of the word. If $y = b$, then $z = b^{x-1}c^y$, and pumping results in a string of the form $b^n b^{x-1} c^y$, which is clearly in F for any n. If $y = c$, then $z = c^{y-1}$. So pumping results in a string of the form $c^n c^{y-1}$ which is also clearly in F.

Next consider the case where $i = 1$, so $w = ab^j c^j$. Again let $x = \epsilon$, $y = a$, and $z = b^j c^j$. Then pumping results in a string of the form $a^n b^j c^j$, which is clearly in F.

Next consider the case where $i = 2$, so $w = aab^j c^k$. Now we can't choose $x = \epsilon$ and $y = a$, because if we do, then $xy^0 z = ab^j c^k$ is not necessarily in our language. So choose $x = aa$. Then pumping cannot result in a string of the form $ab^j c^k$, because we either pump down and have no a's, or we pump up and have more than one a. So pumping results in strings which are still in our language.

Finally, consider the case where $i > 2$. Again choose $x = \epsilon$ and $y = a$. If we pump down, we remove only one a, so our string still has at least two a's, and j and k don't have to match. If we pump up, the string is clearly still in our language. This proves that the language satisfies the pumping lemma.

This does not contradict the pumping lemma because the pumping lemma claims that all regular languages are "pumpable," which is not the same as claiming that all languages that are "pumpable" must be regular.