COMP 330 Assignment 3

Name: Yuhao Wu
ID Number: 260711365

2018-10-6

Question 1:

Are the following statements true or false? Prove your answer in each case. We have some fixed alphabet Σ with at least two letters. In the following A and B stand for languages, i.e. subsets of Σ^*.

- If A is regular and $A \subseteq B$ then B must be regular.
- If A and AB are both regular then B must be regular.
- If $\{A_i \mid i \in \mathbb{N}\}$ is an infinite family of regular sets then $\bigcup_{i=1}^{\infty} A_i$ is regular.
- If A is not regular it cannot have a regular subset.

(a):

FALSE

We take $A = \{\varepsilon\}$, $B = \{\ a^n\ b^n | n \geq 0\ \}$

Then we have A is regular $\quad A \subseteq B \quad B$ is not regular.

(b):

FALSE:

Suppose that our alphabet Σ is $\{\ a, b\ \}$.
Then I choose $A = a^* \qquad B = \{a^i \mid i \text{ is a prime number }\}$,
then we have $B \subseteq A$. Thus, we can say that $AB = a^*$, which is regular. And A is regular as well.
While as we have seen in class, $B = \{a^i \mid i \text{ is a prime number }\}$ is not regular.
Thus, we can say that If A and AB are both regular then B can be non-regular.

(c):

FALSE:

Suppose that A_i is an infinite family of regular languages. Then this statement cannot be true.
Because every language is the union of some set of regular languages. Let L be an arbitrary language whose words are $w_1, w_2, w_3, \ldots$.
Let A_i be the set of singleton languages $\{\{w_1\}, \{w_2\}, \{w_3\}, \ldots\}$, such that $w_i \in L$. The number of elements of A_i is equal to the cardinality of L.
Each individual element of A_i is a language that contains a single string, so for each element of A_i it is a regular set containing only one word.
$L = \bigcup_{i=1}^{\infty} A_i$. As not all languages are regular, it must exist case that $\bigcup_{i=1}^{\infty} A_i$ is not to be regular.

(d):

$$A \text{ is not regular} \implies \text{A cannot have a regular subset}$$
$$\neg\left(\text{ A cannot have a regular subset }\right) \implies \neg\left(\text{ } A \text{ is not regular }\right)$$
$$\text{A have a regular subset} \implies A \text{ is regular}$$

which is the same as Question.(a):

We take $A = \{\varepsilon\}$, $B = \{\ a^n\ b^n | n \geq 0\ \}$

Then we have A is regular $\quad A \subseteq B \quad B$ is not regular.

Question 2:

Show that the following language is not regular using the pumping lemma.

$$L = \{a^n b a^{2n} | n > 0\}$$

- Demon picks a random positive integer p
- I pick the string $w = a^p\ b\ a^{2p}$. Obviously, we have $|w| \geq p$
- Demon is now need to choose x, y, z such that $xyz = w$.
  As $|x\ y| \leq p$ and $|y| > 0$, Demon is forced to pick y to consist of a nonempty string that consists of "a" only
  We say that $y = a^k \quad 0 < k \leq p$
- Now I choose $i = 2$. The new string is $a^{p+k}\ b\ a^{2p}$ which is not in the language as

$$\frac{p+k}{2p} = \frac{1}{2} + \frac{k}{2p} > \frac{1}{2}$$

Question 3:

Show that the language

$$F = \{a^i\ b^j\ c^k \mid i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$$

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.

Show "F" is not regular:
Suppose F is regular, then I will choose another regular language L. As we have seen in class, if F, L is regular, then $F \cap L$ is regular.
I choose L to be $\left(a\ b^*\ c^*\right)$. Obviously, L is regular.
As the word in L starts with exactly only one a, if it is also in F, we must have $j = k$, which means

$$F \cap L = \{a\ b^j\ c^j \mid j \geq 0\}$$

As we have seen in class, $F \cap L$ is not regular, which means F is not regular.

Show "F" satisfies the statement of the Pumping Lemma

- I choose a number $p,\ p \in \mathbb{N},\ p > 0$
- Demon chooses a random word w in F, such that $|w| \geq p$
- I choose x, y, z and $w = xyz$, such that $|xy| \leq p$ & $|y| > 0$
- Demon randomly choose number $n \in \mathbb{N}$, we need to show if $xy^n z$ is also in F

I choose $p = 2$

[1]: Consider Demon chooses word $w = b^j c^k$, which means $i = 0$ Then choose $x = \varepsilon$ and

- $y = b$ if $j \neq 0$
- $y = c$ if $j = 0$ which means w contains c only.

If $y = b$, then $z = b^{j-1}c^k$, and $xy^n z = b^n b^{j-1} c^k$, which is clearly in F for any n.

If $y = c$, then $z = c^{k-1}$, and $xy^n z = c^n c^{k-1}$, which is clearly in F for any n.

[2]: Consider Demon chooses word $w = ab^j c^j$, which means $i = 1$ Then choose $x = \varepsilon, y = a, z = b^j c^j$

We will get the string in the form of

$$x\ y^n\ z = a^n\ b^j\ c^j$$

As the number of b's is the same as the number of a's, regardless of whether pumping up or pumping down, $xy^n z$ will always be in F

[3]: Consider Demon chooses word $w = a^2 b^j c^k$, which means $i = 2$ Then choose $x = \varepsilon, y = a^2, z = b^j c^j$

We will get the string in the form of

$$x\ y^n\ z = a^{2n}\ b^j\ c^k$$

Thus, regardless of pumping up or pumping down, we will never have the case with only one a, so we don't need to worry about whether j equals k or not.
Thus, xy^nz will always be in F.

[4]: Consider Demon chooses word $w = a^i b^j c^k$, which means $i \geq 3$ Then choose $x = \varepsilon, y = a, z = a^{i-1} b^j c^j$

We will get the string in the form of

$$x\ y^n\ z = a^n\ a^{i-1}\ b^j\ c^k$$

Thus, when pumping down, we will get rid of only one a, so the number of a's will never be 1. So we don't need to worry about whether j equals k or not.
Thus, xy^nz will always be in F.

Show why this does not contradict the Pumping Lemma:
Pumping Lemma says:

$$L \text{ is regular} \implies L \text{ can be pumped}$$

which is the same as (contrapositive):

$$L \text{ can't be pumped} \implies L \text{ is not regular}$$

Pumping Lemma **doesn't** claims that:

if L can be pumped, then it is regular

so this does not contradict the pumping lemma.

Question 4:

Let D be the language of words w such that w has an even number of a's and an odd number of b's and does not contain the substring ab.

- Give a DFA with only five states, including any dead states, that recognizes D
- Give a regular expression for this language.

Question (a):

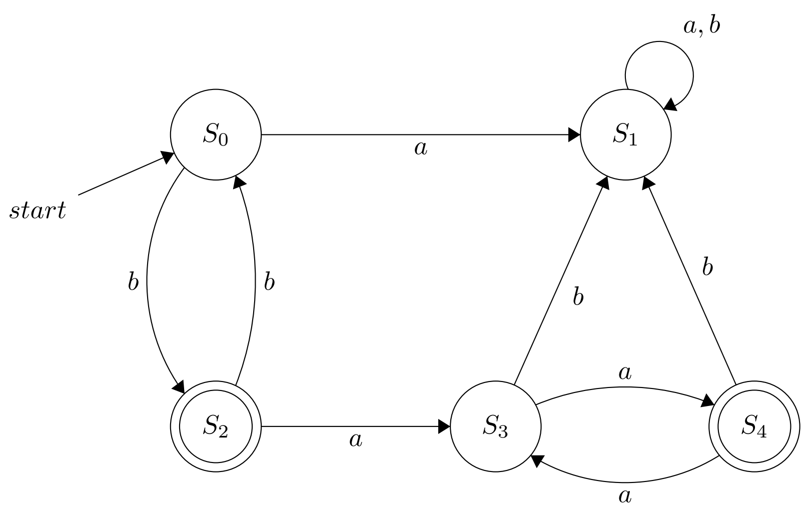
- If we see an "a", then we can never have a "b" after that.
  Thus, if our first letter is an "a", then it has to go to a reject state(which is S_1 in the picture), as we can never have odd numbers of "b"
  Once you go to S_1, which means either we can't have odd numbers of b's or the word contains ab, thus regardless what we read after we will keep in S_1

- According to above, the word has to start with b, then after reading "b" it will go to S_2, which means it contains add number of b's and no "a" and no "ab"

- From S_2, if I read in "a", I will go to another state S_3, which means w has an odd number of a's and an odd number of b's and does not contain the substring ab.
  Thus, if S_3 reads in "a", then w has an even number of a's and an odd number of b's and does not contain the substring ab, which is our accept state S_4

- S_4 reads in a, which means w has an odd number of a's and an odd number of b's and does not contain the substring ab, which go back to state S_3

- As both S_3 and S_4 end in "a", if we read in "b", then the word contains "ab". After that, regardless of what you read, it won't be in the language. So, it goes to S_1



Question (b):

$$b\left(bb\right)^*\left(aa\right)^*$$

Question 5:

Consider the language $L = \{a^n\ b^m | n \neq m\}$; as we have seen this is not regular. Recall the definition of the equivalence $\equiv_L$ which we used in the proof of the Myhill-Nerode theorem. Since this language is not regular $\equiv_L$ cannot have finitely many equivalence classes. Exhibit explicitly, infinitely many distinct equivalence classes of $\equiv_L$.

Given any language $L \subseteq \Sigma^*$, not necessarily regular, we define an equivalence relation R_L on Σ^* as follows

$$x\ R_L\ y \text{ if and only if } \forall z, xz \in L \Leftrightarrow yz \in L$$

.

We just need to construct infinite many equivalence relation R_L:
To begin with, I choose $x = a^{k_1} \quad y = a^{k_2} \quad k_1 \neq k_2 \quad k_1, k_2 \in \mathbb{N}$
Then obviously, we have infinite many pairs of such x, y.
Then we choose $z = b^{k_1}$, thus, we have $xz \notin L \quad yz \in L$, which means x, y not in equivalence relationship.
As there are infinite pairs of such x, y, we can say: infinitely many distinct equivalence classes of R_L.