Miguel Garza

Homework 3

1a.Consider the language L = {w | w begins with a 1 and ends with a 0}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = 1 \sum *0$$

$$= 1(0+1)*0$$

The regular expression accepts string such as 100000, 10100, 1010.

Thus, the regular expression is 1(0+1)*0 .

1b. Consider the language L = {w | w contains at least three 1s}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = \sum *1 \sum *1 \sum *1 \sum *$$

$$= (0+1)*1(0+1)*1(0+1)*1(0+1)*$$

The regular expression accepts string such as 111, 101001, 10101, 0000001111,1111.

Therefore, the regular expression is (0+1)*1(0+1)*1(0+1)*1(0+1)* .

2c. Consider the language L = {w | w contains the substring 0101 (i.e, w = x0101y for some x and y)}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = \sum *1010 \sum *$$

$$= (0+1)*1010(0+1)*$$

The regular expression accepts string such as 0101, 001011, 00001010111, 01010101010101.

Therefore, the regular expression is (0+1)*1010(0+1)* .

2d. Consider the language L = {w | w has length at least 3 and its third symbol is a 0}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = \sum \sum 0 \sum *$$

$$= (0+1)(0+1)0(0+1)*$$

The regular expression accepts string such as 000, 0000, 110, 110000

Therefore, the regular expression is (0+1)(0+1)0(0+1)*.

3e. Consider the language L = {w | w starts with 0 and has odd length, or starts with 1 and has even length}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = 0(\sum \sum) * +1 \sum (\sum \sum)*$$

$$= 0((0+1)(0+1)) * +1(0+1)((0+1)(0+1))*$$

The regular expression accepts string such as 0, 011, 010, 00101, 10, 11.
Therefore, the regular expression is 0((0+1)(0+1))*+1(0+1)((0+1)(0+1))*
.

3f. Consider the language L = {w | w doesn't contain the substring 110}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = 0 * (10^{+}) * 1*$$

The regular expression accepts string such as 010, 011, 0101.
Therefore, the regular expression is 0*(10^{+})*1* .

4g. Consider the language L = {w | w The length of w is at most 0}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = \varepsilon + \sum + \sum\sum + \sum\sum\sum + \sum\sum\sum\sum + \sum\sum\sum\sum\sum$$

+

$$= \varepsilon+(0+1)+(0+1)(0+1)+(0+1)(0+1)(0+1)+(0+1)(0+1)(0+1)(0+1)+(0+1)(0+1)(0+1)(0+1)(0+1)$$

The regular expression accepts string such as ε,0, 00, 000, 0000, 0000,101, 1010 the empty string.
Therefore, the regular expression is ε+(0+1)+(0+1)(0+1)+(0+1)(0+1)(0+1)+(0+1)(0+1)(0+1)(0+1)+(0+
.

4h. Consider the language L = {w | w is any string except 11 and 111}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = \varepsilon + \sum +0\sum +10 + 0\sum\sum +10\sum +110 + (\sum)^{3}(\sum)^{+}$$

$$= \varepsilon+(0+1)+0(0+1)+10+0(0+1)(0+1)+10(0+1)+110+((0+1))^{3}((0+1))^{+}$$

The regular expression accepts string such as ε,101, 110, 1010, ...
Therefore, the regular expression is $\varepsilon+(0+1)+0(0+1)+10+0(0+1)(0+1)+10(0+1)+110+((0+1))^{3}((0+1))^{+}$
.

5i. Consider the language L = {w | every odd position of w is a 1}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = (1\sum) * (\varepsilon + 1)$$

$$= (1(0+1)) * (\varepsilon + 1)$$

The regular expression accepts string such as ε,101, 111, 1010, ...
Therefore, the regular expression is (1(0+1))*(ε+1) .

5j. Consider the language L = {w | w contains at least two 0s and at most one 1}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = 00*00*(\varepsilon+1) + 00*(\varepsilon+1)00*+(\varepsilon+1)00*00*$$

The regular expression accepts string such as 001, 010, 100, ... Note that for 00*00*(ε+1), there are two mandatory zeros and at most one 1. The one can be placed at the start, middle or at the end.

Therefore, the regular expression is 00*00*(ε+1)+00*(ε+1)00*+(ε+1)00*00* .

6k. Consider the language L = {ε,0}. In all parts, the alphabet is {0,1} . Let R be the regular expression that generates the language l. The regular expression is as follows:

$$R = 0 + \varepsilon$$

Therefore, the regular expression is 0+ε.

7b. Consider the language, A_2= { www | w ϵ {a,b}* }.

Assume $A_2$is a regular language.

Let P be the pumping length given by the pumping lemma.

Consider a string S = $a^P ba^P ba^P b$ ϵA_2

By pumping lemma, this string can be divided into three pieces xyz such that |xy| $\leq$p, |y| > 0 and xy'z ϵ $A_2 \forall$ i $\geq$0

So S = $a^p ba^p ba^p b$= xyz

Let aabaabaab be the string that belongs to A_2.The pumping length of the string is 2. To satisfy the conditions of the pumping lemma, x = a, y = a, z = baabaab

$$S = aabaabaab$$

$$= \underline{a}_{x}\,\underline{aa}_{y}\,\underline{baabaab}_{z}$$

Pump the middle part such that xy^iz (i$\geq$ 0). ....For i = 2, the y becomes aa. The string after pumping is aaabaabaab.

$$S = (a)(a)^i baabaab$$

$$= \underline{a}_{x}\,\underline{aa}_{y}\,\underline{baabaab}_{z}$$

when [i = 2]

The string aaabaabaab $\notin$ A_2. It is a contradiction. So, the pumping lemma is violated.

Therefore, $A_2$is not a regular language.

8. We need to prove that B_n= {a^k| k is a multiple of n } is regular for each n $\geq$1.

Suppose k=ni, where i is any positive integer. First let's consider when i = 1.

When i = 1 and n = 1:

$$B_1 = \{a^k\}$$

$$= \{a^{ni}\}$$

$$= \{a^{|x|}\}$$

$$= \{a\}$$

Then, the string comes out to be {a}
Next, we consider i = 1 and n = 2:

$$B_2 = \{a^k\}$$

$$= \{a^{ni}\}$$

$$= \{a^{2x1}\}$$

$$= \{aa\}$$

Then, the string comes out to be {aa}.
Now we consider when when i = 1 and n = 3

$$B_3 = \{a^k\}$$

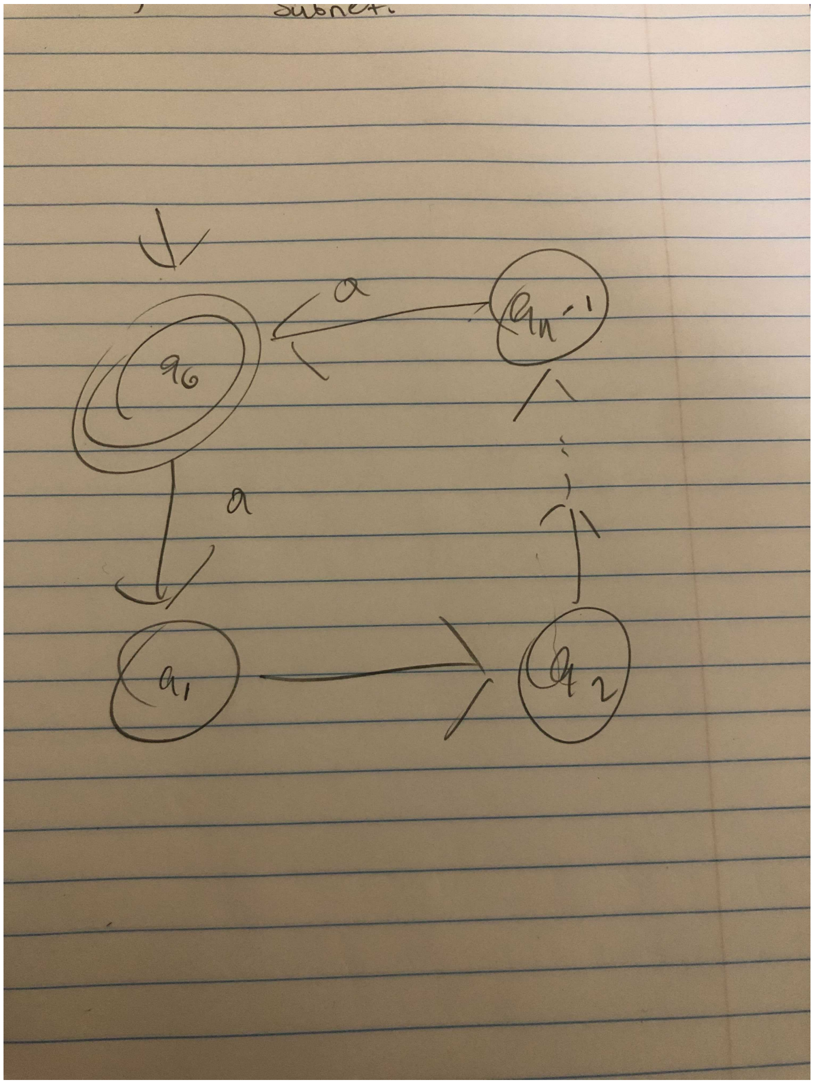
$$= \{a^{ni}\}$$

$$= \{a^{3x1}\}$$

$$= \{aaa\}$$

Then the string comes out to be {aaa} and so forth for increased values of n.

Looking at the finite automaton,



$q_0$is the initial and final state and $q_1, q_2, q_3, q_{n-1}$are the subsequent states.

By the closure property of the regular expression, it is clearly seen that the specific expression is a regular expression when the value of n is greater than and equal to 1.

Closure property includes various operations such as union, intersection, set complement, set reversal, set difference and many more. Assume that $B_n$is regular.

- Union of $B_1$and B_2results in the third string and it is also a regular expression.

- Likewise, if user applies any property of closure, then the result is the regular expression.

Hence, it is proved that the above expression is the regular expression.

9.

Pumping lemma iteration:

For a regular language A with the pumping length P, if S is any string of A with minimum length of P, then the string S can be divided into 3 pieces x, y, and z represented as S = xyz should satisfy the following conditions:

for each |y| $>$ 0, and |xy| $\leq$P.

a. Consider the language L = $\{0^n 1^m 0^n$ | m, n ≥ 0).

Assume that L is regular language and a string S $=$ $0^P 10^P$. Divide the string into three pieces x, y and z. So, S $=$ $0^P 10^P$= xyz where P is the pumping length.

Assume that x=0^{P-K}, y $=$ $0^K$and z $=$ 10^P(where K $>$ 0)

Now, $xy^0z = 0^{P-K}(0^K)^0 10^P$

$= 0^{P-K} 10^P \notin$L

The string xy^0z does not belong to L because P-K $<$ P.

So, the assumption that L is regular is a contradiction. Thus, by using pumping lemma, it is proved that L is not regular.

10.

Without DFA:

If A is a finite language, then it contains a finite number of strings a0,a1,$\cdots$,an. The language $\{a_i\}$ consisting of a single literal string ai is regular. The union of a finite number of regular languages is also regular. Therefore, A=$\{a_0\}\bigcup\{a_1\}\bigcup\cdots\bigcup\{a_n\}$ is regular.

With DFA:

Let P be the set of all prefixes of words in A. Since A is finite, so is P. We construct a DFA whose states are $\{q_p$: p$\in$P$\}\bigcup\{$q´$\}$. The initial state is qϵ. A state qp is final if and only if p$\in$A. When at state qp and reading σ, if p$\sigma\in$P then we move to qpσ, otherwise we move to q´. When at state q´, we always stay in q´.

What I described above is the minimal DFA. You can get a somewhat simpler DFA by taking an arbitrary superset of P, such as the set of all words of length at most n, where n is the length of the longest word in A.