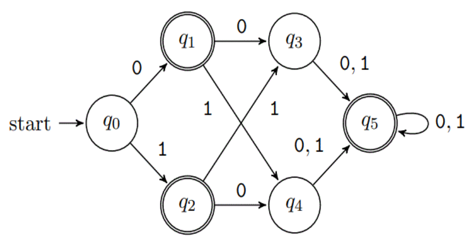
Name: Gurik Manshahia   Assignment 2
Student ID: V00863509

1. **(a) (10 Marks) Using the state partitioning algorithm presented in class, find the minimal automaton equivalent to the following:**



State table for the DFA:

	0	1
q_0	q_1	q_1
q_1	q_3	q_4
q_2	q_4	q_3
q_3	q_5	q_5
q_4	q_5	q_5
q_5	q_5	q_5

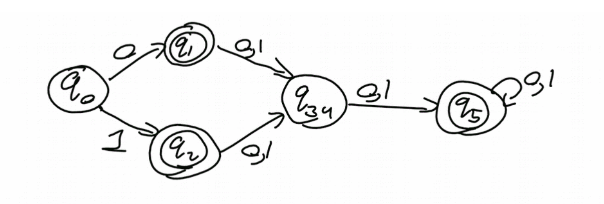
Applying state partitioning:

State 0: [q0,q3,q4][q1][q2][q5]

State 1: [q0][q3,q4][q1][q2][q5]

State 2:[q0][q3,q4][q1][q2][q5]

Minimized equivilant:



(b) (5 Marks) What is the language recognized by this automaton (Σ = {0, 1})?

L = {0,1,001,011,000,111,000,…}

L = {w | w ∈(0+1)*|w|>0,|w| !=2}

2. Prove the each of the following languages are not regular. You may use the pumping lemma, or closure properties of the regular langauges.

(a) (5 Marks) {0^n 1^m 0^n | m, n ≥ 0}

L = {0^n 1^m 0^n | m, n ≥ 0}

Assume that L is a regular language and a string S = 0^p10^p. Divide string into three stections x,y,z. which results in S = 0^p10^p = xyz. P is the pumping length.

Assume x = 0^(p-k), y = 0^k and z = 10^p where k>0

Therefore xy^0z = 0^p-k(0^k)^010^p = 0^(p-k)10^p !∈ L therefor y^0 = ∈

Therefore xy^0z is not part of L since P-K < P

By using the pumping lemma it is proved that L is not regular.

(b) (5 Marks) {0^m 1^n | m != n}

L = {0^m 1^n | m != n}

Assume L is a regular language.

(c) (5 Marks) {wuw | w, u ∈ {0, 1} +}

(HINT: One way to do this is to use closure under intersection to get a simpler pumping lemma proof.)

3. (5 Marks) Is the language {wuw | w, u ∈ {0, 1} ∗} regular? If it is, give a regular expression for the language, otherwise use the pumping lemma or closure properties of regular languages to prove that it is not.

4. Give CFGs for the following languages over σ = {0, 1}

(a) (5 Marks) {w | w = w R}

CFG is: S → 0S0|1S1|0|1|∈

(b) (5 Marks) {w | w contains the same number of 0’s and 1’s}

CFG is: S -> SASBS|SBSAS|∈

A → 0

B → 1

(c) (5 Marks) {w | w = 0n1 n , n ≥ 0}

CFG is: S → 0S1

S →∈

5. (20 Marks) Give a CFG that generates the language

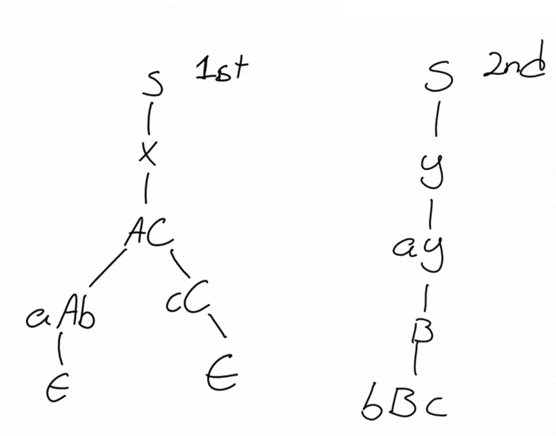
A = {a^i b^j c^k | i = j or j = k where i, j, k ≥ 0}

Is your grammar ambiguous? Why or why not?

Grammar:

S -> X | Y
X -> AC
A -> aAb | ∈
Y -> aY | B
B -> bBc | ∈
C -> cC | ∈

The grammar is ambiguous as it is able to be derived in 2 serperate ways.



6. (10 Marks) Convert the following grammar into a grammar in Chomsky normal form:

A → BAB | B | ϵ

B → 00 | ϵ

(You are eligible for partial marks only if you show each step of the normalization procedure.)

Can be written as

As the start symbol is on the right creating 1 more Start symbol S

S-> A

A->BAB | B | ε

B -> 00 | ε

remove the null productions

S->A

A-> BAB | B | 0000| 00

B -> 00 | ε

Now removing unit production S->A

S-> BAB | B | 0000| 00

B -> 00 | ε

Check if more than 2 variable are on RHS

S-> BX | B | 0000| 00

B -> 00 | ε

X->AY

Y->B

7. (15 Marks) Using the CNF version (given below) of the grammar

$E \rightarrow E * E \mid E + E \mid (E) \mid id \mid num$

show the result of running the CYK algorithm on the string w = (id + num) ∗ num. Just show the entries of the resulting table.

Note: Use the following CNF grammar:

$E \rightarrow EA \mid EB \mid LD \mid id \mid num$

$A \rightarrow ME$

$B \rightarrow P\ E$

$D \rightarrow ER$

$M \rightarrow *$

$P \rightarrow +$

$L \rightarrow ($

$R \rightarrow)$

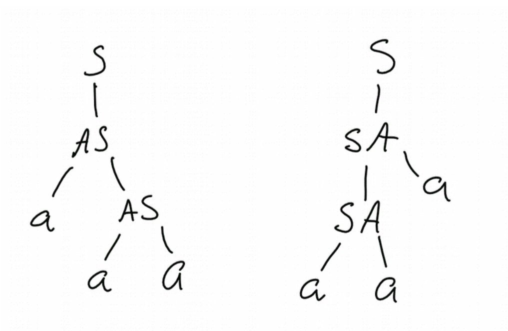
7	E						
6	-	-					
5	E	-	-				
4	-	D	-	-			
3	-	E	-	-	-		
2	-	-	B	D	-	A	
j=1	L	E	P	E	R	M	E
	1	2	3	4	5	6	7

(id + num) * num

8. (5 Marks) Is every grammar in CNF unambiguous? If your answer is ”yes”, provide a proof. If your answer is ”no”, provide a counterexample.

S → AS|SA|a

A → a



every grammar in CNG is not unambiguous above I have provided a counter example