

**CISC 603-51- A-2021/SUMMER - THEORY OF COMPUTATION**

**Assignment - 3**

**Regular Expressions, Regular Languages**

**Context-Free Languages**

**By,**

**Satheesh Swaminathan**

**Student ID:** **266582**

**Assignment- 3 Solutions**

**3.1 #1**

L((a + bb)∗) and |w|=5

Consider (a + bb)

This language will accept either ‘a’ or ‘bb’

\* means that the symbol can oocur any number of times preceeded by the symbol \*.

Now this language will accept L((a + bb)∗) and |w|=5

String with only a’s

“aaaaa”

String with one set of ‘bb’ and length of string 5

“aaabb”

“aabba”

“abbaa”

“bbaaa”

Strings with two sets of ‘bb’

“abbbb”

“bbabb”

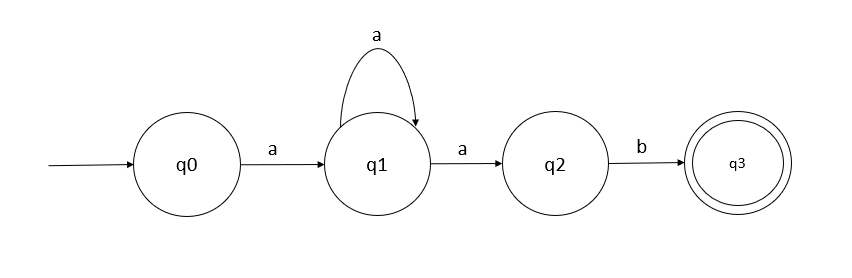
“bbbba”

**3.1 #3**

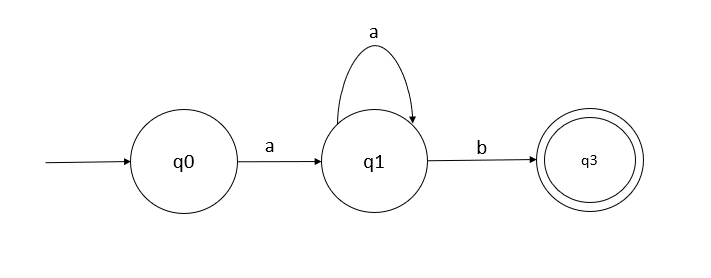
L(aa ∗ (ab + b))

(aa ∗ (ab + b)) = aa\*ab + aa\*b

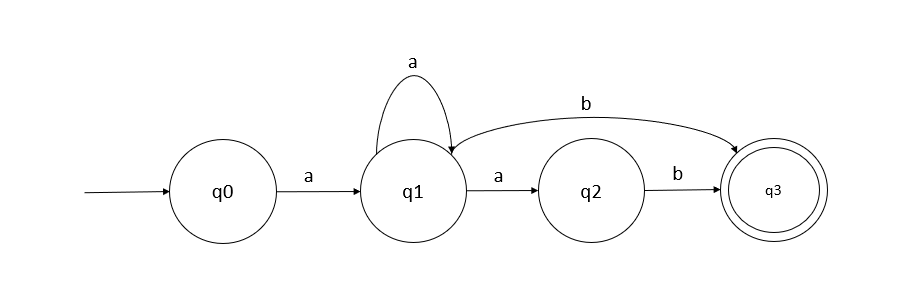
aa\*ab



aa\*b



aa\*ab + aa\*b



**3.1 #5**

((0 + 1) (0 + 1)∗)∗ 00 (0 + 1)∗

(0 + 1) (0 + 1)∗)∗

This expression accepts 0, 1, 00, 01, 11, 10

(0+1)\* accepts 0, 1, 00, 01, 11, 10

((0 + 1) (0 + 1)∗)∗ 00 (0 + 1)∗ accepts

L1 = {00,100,000,001,0100,1000,1100,0001,.....}

(0 + 1)∗ 00 (0 + 1)∗

L2 = { 00,000,100,001,0000,1000,0001,.....}

Therefore, this expression denotes the language in example 3.5

**3.1 #7**

{anbm : n ≥ 3, m is odd}

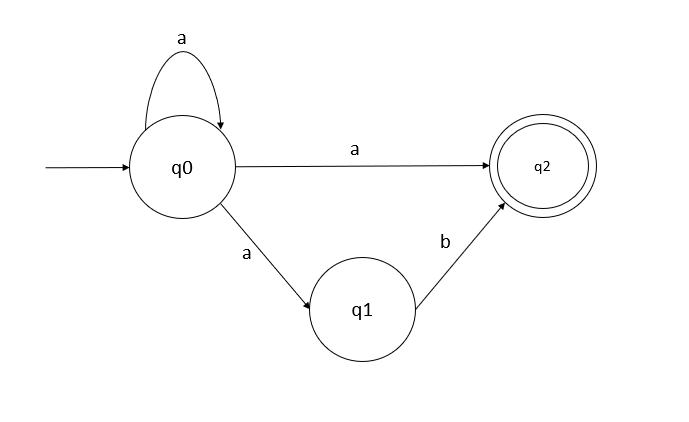
aaaa\* -> an where a >=3

(bb)\*b ensures that the number of b is odd

Therefore aaaa\*(bb)\*b is the regular expression for this set.

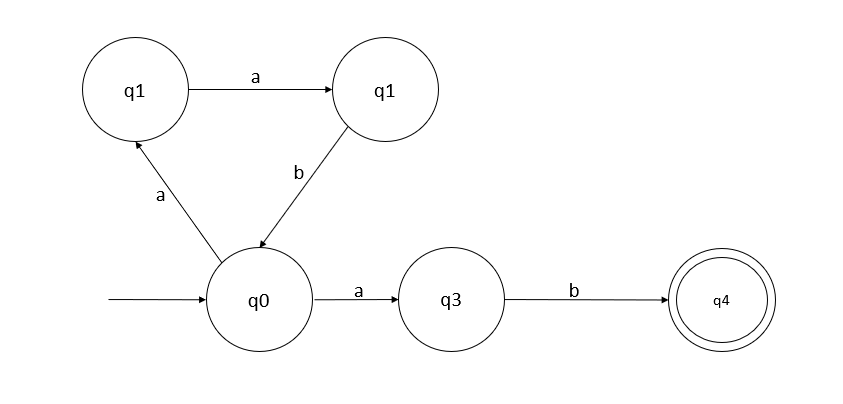
**3.2 #1**

L(a∗a + ab)



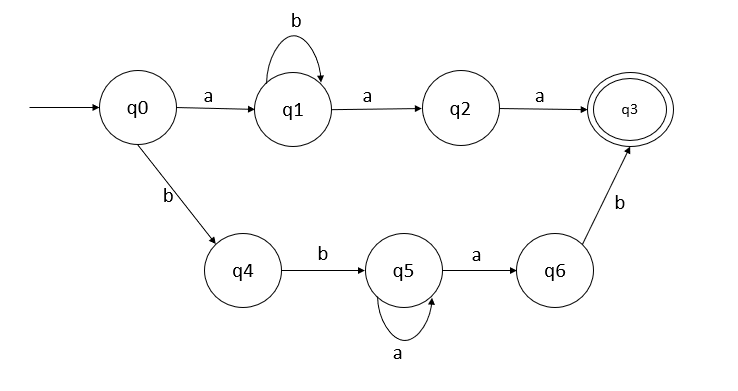
**3.2 #2**

L((aab)∗ab)



**3.2 #3**

L(ab∗aa + bba∗ab)



**5.1 #1**

(a) L = anbn, n is even.

L = { ε, aabb, aaaabbbb, …. }

V = {s}

Ε = (a,b)

P = {S🡪 aaSbb, S🡪 ε }

(b) L = anbn, n is odd.

L = {ab, aaabbb,aaaabbbbb,….}

G = <V, ε, S, P >

V = {S}

Ε = (a,b)

P = {S🡪 aaSbb, S🡪ab }

(c) L = anbn, n is a multiple of three.

This language generates equal number of a and a and they are multiple of 3’s.

If n=3

“aaabbb”

If n=6

“aaaaaabbbbbb”

G = (V, T, S, P)

V: finitie set of symbols or variables {S, A, B, C, D, E, F}

T: finitie set of terminal symbols {a, b, ε}

P: finite set of Productions

{

S 🡪 aA

A🡪aB

B🡪as/aC

C🡪bD

D🡪bE

E🡪bF/bC

F🡪bG/ ε

}

S – Start Symbol {S}

(d) L = anbn, n is not a multiple of three.

S 🡪 aAb|ε

A 🡪 aBb| ε

B 🡪 aaSbb| ε

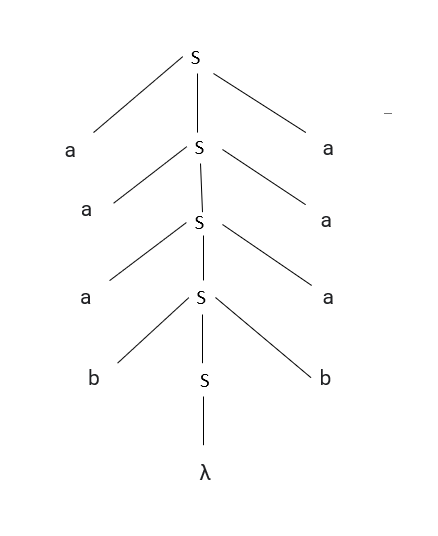
**5.1 #2**

w = aaabbaaa

S 🡪 aSa,

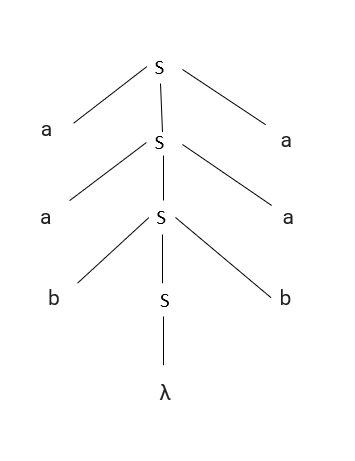
S 🡪 bSb,

S 🡪 λ



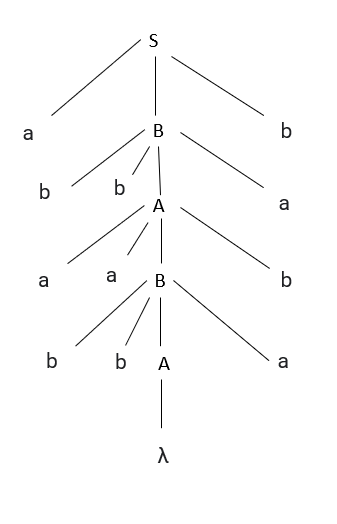
**5.1 #3**

**S = aSa --> aaSaa --> aabSaa 🡪 aabbaa**



**5.1 #4**

w = abbbaabbaba



**5.1 #5**

L(G) = {ab (bbaa)n bba (ba)n : n ≥ 0}

S 🡪 abB

A 🡪 aaBb,

B 🡪 bbAa

A🡪 λ

S 🡪 abB 🡪 abbAa 🡪 abbbaaBba 🡪 abbbaabbAaba 🡪 abbbaabbaba

**5.2 #1**

An example of regular grammar G with N = {S, A}, Σ = {a, b, c}

S → aS

S → bA

A → ε

A → cA

S → aS|bA

S → aS|bs|cA

where A → cA

A → aA | bS

so that

S → aS | bS | cA

A → aA | bS

is an s-grammar

**5.2 #2**

There is (at least) one way to prove unambiguity of a grammar G=(N,T,δ,S) for language L. It consists of two steps:

Prove L⊆L(G).

Prove [zn]SG(z)=|Ln|.

Consider the simple grammar GG with rules

S→aSa ∣ bSb ∣ ε

S → aSa ∣ bSb ∣ ε

It is clear that L(G)={wwR∣w∈{a,b}∗}

There are 2n/2 palindromes of length n if n is even, 0 otherwise.

Setting up the equation system yields

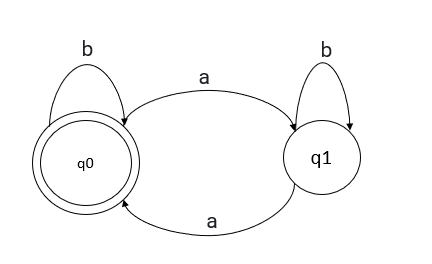
S(z)=2z2S(z)+1S(z)

whose solution is

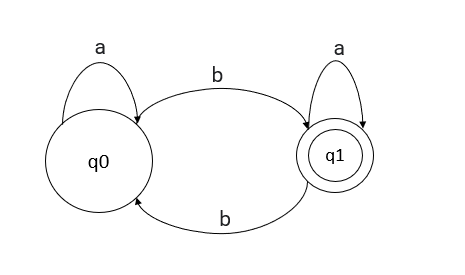


The coefficients of SG coincide with the numbers of palindromes, so G is unambiguous.

**5.2 #3**



Even number of a’s na(w) mod 2 = 0



Odd number of b’s

L(aaa\*b + ab\*)

S 🡪 aaAb | b

A 🡪 aA| ε

**5.2 #4**

L = {anbn : n ≥ 2}

For n= 2

aabb

For n= 3

aaabbb

S-grammar for this language is S🡪 aSb | aabb

**5.2 #5**

L ={anb2n : n ≥ 2}

for n =2

aabbbb

for n=3

aaabbbbbb

S-grammar for this language is S🡪 aSbb | aabbbb