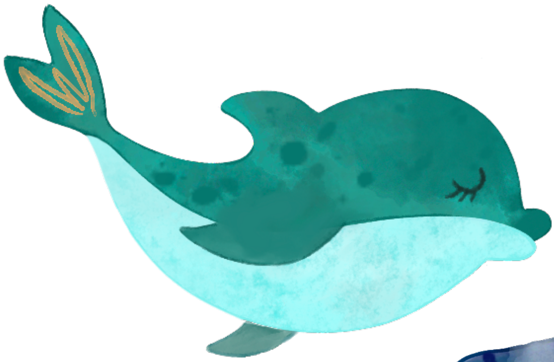
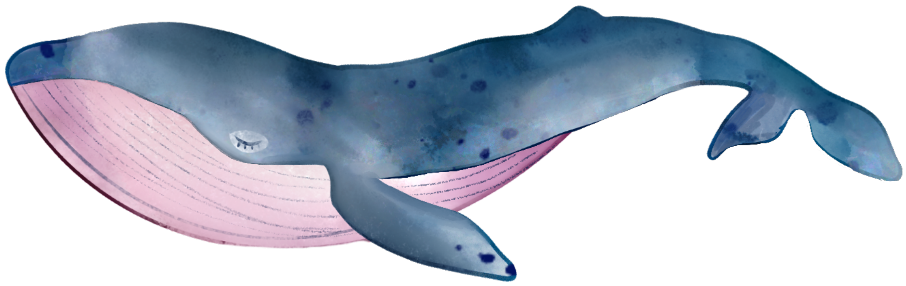
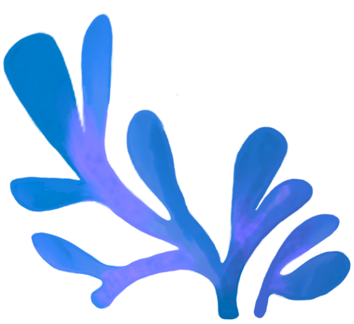
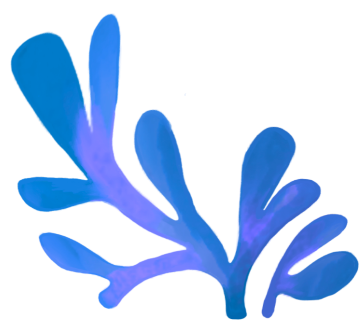




THEORY OF COMPUTATION





Subject:

Topic: **Module - 1**

Date: [redacted]

Introduction

QUESTION

Prove that for any $n \geq 1$

$$1^2+2^2+3^2+\dots+n^2 = \frac{n(n+1)(2n+1)}{6}$$

ANSWER

for $n=1$,

$$\frac{1(1+1)(2\cdot 1+1)}{6} = \frac{2\cdot 3}{6} = 1$$

Replace 'n' with 'K'

$$\Rightarrow 1^2+2^2+3^2+\dots+K^2 = \frac{K(K+1)(2K+1)}{6}$$

$n=K+1$,

$$1^2+2^2+3^2+\dots+K^2+(K+1)^2$$

$$LHS = \frac{K(K+1)(2K+1)}{6}+(K+1)^2$$

$$= \frac{6(K+1)^2+K(K+1)(2K+1)}{6}$$

$$= \frac{6(K^2+2K+1)+K(K+1)(2K+1)}{6}$$

$$= \frac{6K^2+12K+6+2K^3+2K^2+K^2+K}{6}$$

$$= \frac{2K^3+9K^2+13K+6}{6}$$

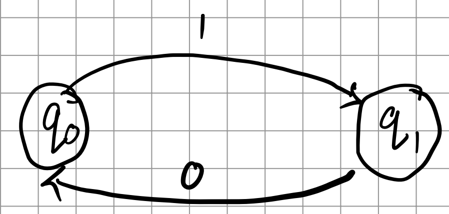
$$RHS = \frac{(K+1)(K+2)(2K+2+1)}{6}$$

$$= \frac{(K+1)(K+2)(2K+3)}{6}$$

$$= \frac{(K^2+3K+2)(2K+3)}{6}$$

$$= \frac{2K^3+9K^2+13K+6}{6}$$

$$\Rightarrow LHS = RHS$$



Transition diagram

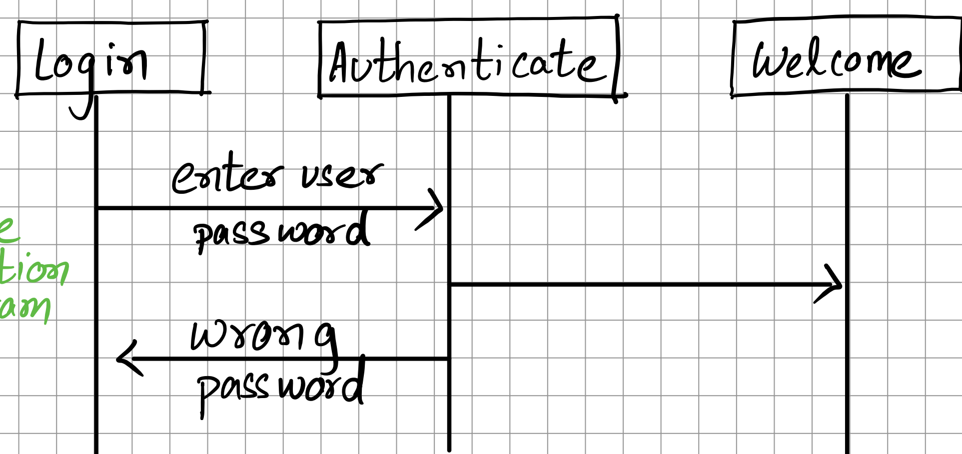
$\delta(q_0,1) \rightarrow q_1$

$\delta(q_1,0) \rightarrow q_0$

Transition table

$Q \backslash \Sigma$	0	1
q_0	ϕ	q_1
q_1	q_0	ϕ

State transition diagram



Three branches of Theory of Computation

→ Automated Theory

→ Computability Theory

→ Complexity Theory

Inductive Proof

The proof by Induction can be done in 3 steps $\left(1+2+3\dots+n = \frac{n(n+1)}{2}\right)$

1. **Basic**:- here we assume the initial value. $n=1$

   $LHS = 1$

   $RHS = \frac{1(1+1)}{2} = 2/2 = 1$

2. Inductive Hypothesis:- assign value of n in K in the equation

   Replace 'n' with K

   $$1+2+3+4\dots+K = \frac{K(K+1)}{2} \quad \text{— (2)}$$

HOMEWORK

$1+4+7+\ldots+(3n-2) = \frac{n(3n-1)}{2}$

for all values of $n \geq 1$

ANSWER

for $n=1$;

$\frac{1(3.1-1)}{2} = \frac{1(3-1)}{2} = \frac{2}{2} = 1$

for $n=k$

$1+4+7+\ldots+(3k-2) = (k+1)\left(\frac{3k+2}{2}\right)$

for $n=(k+1)$

$1+4+7+\ldots+(3k-2)+3(k+1)-2$

$\Rightarrow \frac{k(3k-1)}{2} + 3(k+1) - 2 = \frac{(k+1)(3k+2)}{2}$

$\Rightarrow \frac{3k^2-k+6k+6-4}{2} = \frac{3k^2+2k+3k+2}{2}$

$\Rightarrow \frac{3k^2+5k+2}{2} = \frac{3k^2+5k+2}{2}$

$\Rightarrow$ LHS = RHS

3. Inductive step

Substitute $n = k+1$

$LHS = 1+2+3 \ldots k+k+1$

$= \frac{k(k+1)}{2} + (k+1)$

$= \frac{k(k+1)+2(k+1)}{2}$

$RHS = \frac{n(n+1)}{2}$; $n = k+1$

$= \frac{(k+1)(k+1+1)}{2}$

$= \frac{(k+1)(k+2)}{2}$

$\Rightarrow$ LHS = RHS

HOMEWORK

$(1*2)+(2*3)+(3*4)\ldots + n(n+1) = 1/3\{n(n+1)(n+2)\}$ for all values of $n \geq 1$

Subject: Topic: Basic Terminologies

Date: Previous Topic:

QUESTION

Let x, y are strings 'abc' and '123' find xy^R

ANSWER

$x = abc$

$y = 123$

$y^R = 321$

$xy^R = abc321$

⟨1⟩ Symbols - a, 1, +, −, % (Infinite)

⟨2⟩ Alphabet (Σ) (finite)

$\Sigma = \{a, b, c \ldots z\}$

$\Sigma = \{0, 1, 2 \ldots\}$

⟨3⟩ String (W) (finite)

$\Sigma = \{0, 1\}$

$W = 01$
$W = 011$
$W = 01010011$

Length of the string ($|w|$)

$w = 011$
$|w| = 3$

Reverse of the String (w^R)

$w = 011$
$w^R = 110$

Power of an alphabet

$\Sigma = \{a, b\}$

$\Sigma^0 = \{\phi\}$

$\Sigma^1 = \{a, b\}$

$\Sigma^2 = \{a, b\}\{a, b\} = \{aa, ba, ab, bb\}$

$\Sigma^3 = \Sigma^2 . \Sigma = \{aa, ba, ab, bb\}\{a, b\}$

$= \{aaa, aba, baa, bba, abb, aab, bab, bbb\}$

no of elements $= n(\Sigma)^N$

n = number of symbols in a alphabet
N = power of alphabet

Kleene closure

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots \cup \Sigma^\infty$

$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$

Positive Closure

$\Sigma^+ = \Sigma^* - \text{'}\varepsilon\text{'}$ (excluding the empty string 'ε')

⟨4⟩ Language

a set of strings over an alphabet. i.e subset of Σ^*

$\Sigma = \{a, b\}$

$L = ab$
$L = abb$

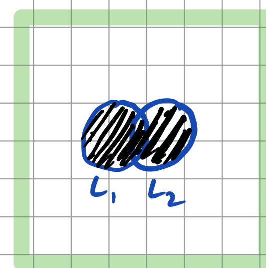
Operations on languages

⟨1⟩ Union

$L_1 = \{ab, bc, cd\}$

$L_2 = \{12, 23, 34\}$

$L_1 \cup L_2 = \{ab, bc, cd, 12, 23, 34\}$



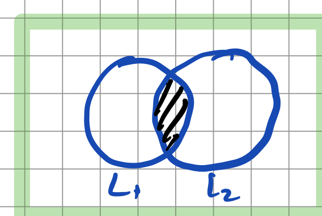
$L_1 \cup L_2 = \{x / x \text{ is in } L_1 \text{ or } L_2\}$

⟨2⟩ Intersection

$L_1 = \{ab, bc, cd\}$

$L_2 = \{cd, ef, gh, ij\}$

$L_1 \cap L_2 = \{cd\}$



$L_1 \cap L_2 = \{x / x \text{ in } L_2 \text{ and } L_1\}$

⟨3⟩ Concatenation

$L_1 = \{ab\}$

$L_2 = \{1, 2\}$

$L_1 \cdot L_2 = \{ab12\}$

Subject: Topic: Module - 2

Date: [redacted] Topic: finite State Automata

QUESTION

Design a DFA which accepts only 101 over the input set $\Sigma = \{0, 1\}$

ANSWER

Define DFA

Let 'M' is a DFA where

Q = finite Series $\{q_1, q_2, q_3\}$

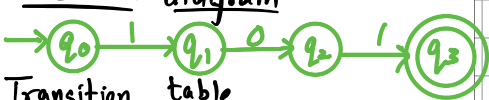
Σ = Input Symbols - $\{0,1\}$

$\delta = Q \times \Sigma$

S = Starting state - $\{q_0\}$

F = Final State - $\{q_3\}$

Transition diagram



Transition table

Q \ Σ	0	1
q_0	ϕ	q_1
q_1	q_2	ϕ
q_2	ϕ	q_3
q_3	ϕ	ϕ

$W = 101$

$\delta(q_0, 101) = \delta(\delta(q_0, 1), 01)$
$= \delta(q_1, 01)$
$= \delta(\delta(q_1, 0), 1)$
$= \delta(q_2, 1)$
$= q_3$

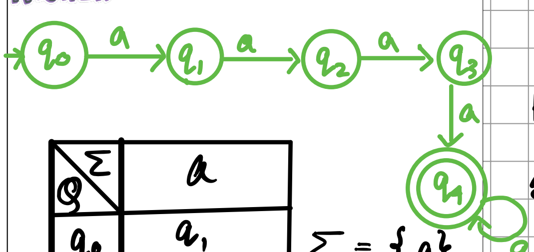
Hence, the answer is accepted

QUESTION

Design a DFA for

$L(G) = \{a^n / n \geq 4\}$

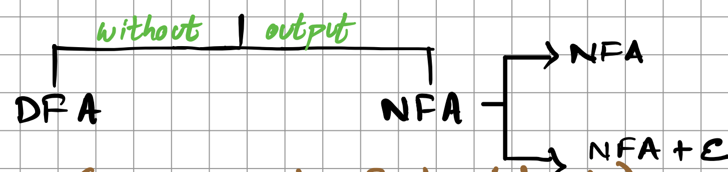
ANSWER



Q \ Σ	a
q_0	q_1
q_1	q_2
q_2	q_3
q_3	q_4
q_4	q_4

$\Sigma = \{a\}$

finite State automata



DFA (Deterministic finite Automata)

$'M' = \{Q, \Sigma, \delta, S, f\}$

Q - finite state $\{a_0, a_1, a_3\}$

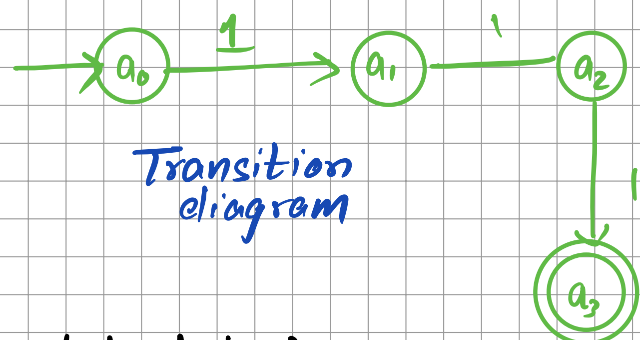
Σ - input Symbols $\{1\}$

δ - Transition $Q \times \Sigma$

S - Starting state - $\{a_0\}$

F - Final State - $\{a_3\}$

five Tuples



Transition diagram

Q \ Σ	1
a_0	a_1
a_1	a_2
a_2	a_3
a_3	ϕ

Correct Input '111'

$\delta(a_0, 111) = \delta(\delta(a_0, 1), 11)$
$= \delta(a_1, 11)$
$= \delta(a_2, 1)$
$= a_3$, final state

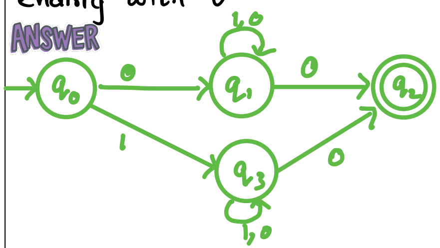
Hence, Input is accepted.

$\delta(a_0, 11) = \delta(\delta(a_0, 1), 11)$
$= \delta(a_1, 1)$
$= a_2$ - not accepted, as it doesn't reach final state

QUESTION

Design a DFA of all strings ending with '0'

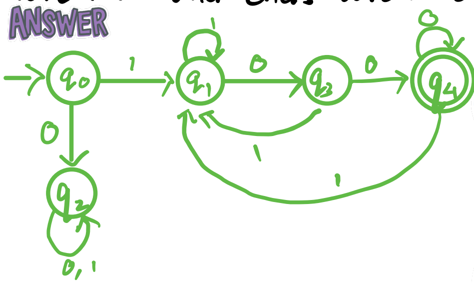
ANSWER



QUESTION

Design a DFA that starts with '1' and ends with '00'

ANSWER

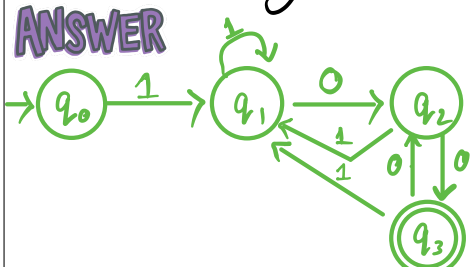


Q \ Σ	0	1
q0	q2	q1
q1	q3	q1
q2	q2	q2
q3	q4	q1
q4	q4	q1

QUESTION

Design a DFA starting with 1 and ending with even no. of zeroes

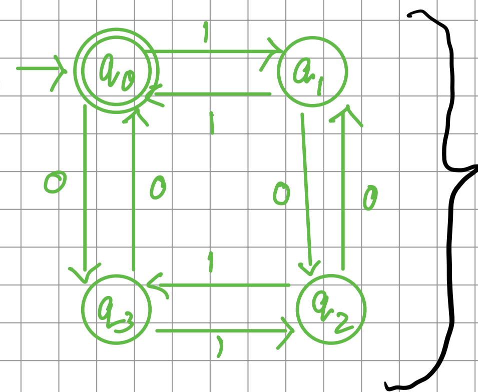
ANSWER



QUESTION

accepts even '1's and '0's

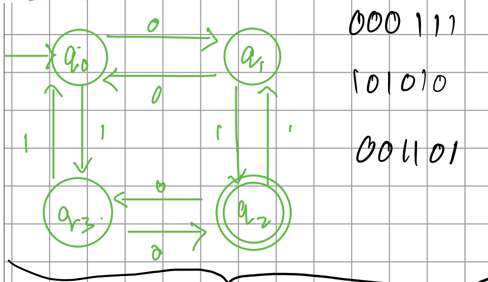
ANSWER



QUESTION

accepts odd '1's and '0's

ANSWER



000111

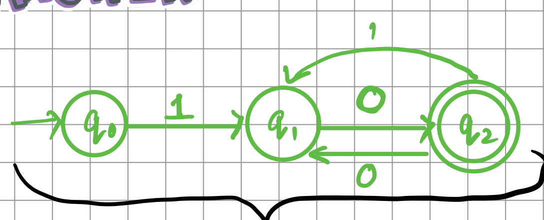
101010

001101

QUESTION

Starting with one and ending with odd number of zeroes.

ANSWER



QUESTION

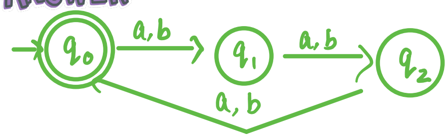
Design a DFA for the following language.

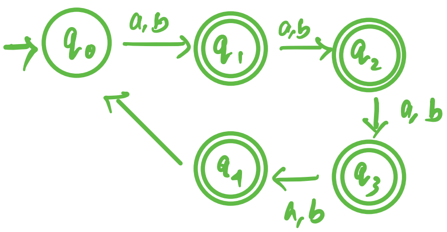
$L = \{w : |w| \bmod 3 = 0;\ w \in \{a,b\}^*\}$

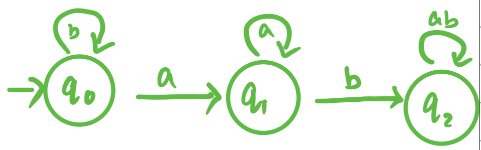
$L = \{w : |w| \bmod 5 \neq 0;\ w \in \{a,b\}^*\}$

$L = \{w_1\, ab\, w_2;\ w_1, w_2 \in \{a,b\}^*\}$

ANSWER



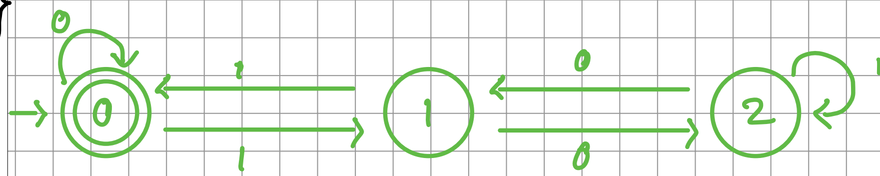




QUESTION

DFA that accepts binary strings divisible by 3

ANSWER



	0	1
* $q(0)$	$q(0)$	$q(1)$
$q(1)$	$q(2)$	$q(0)$
$q(2)$	$q(1)$	$q(2)$

$1011010 \longrightarrow q0$

$\delta(\delta(q_0, 1), 011010)$

$\delta(\delta(q(1), 0), 11010)$

$\delta(\delta(q(2), 1), 1010)$

$\delta(\delta(q(2), 1), 010)$

$\delta(\delta(q_2, 0), 10)$

$\delta(\delta(q_1, 1), 0)$

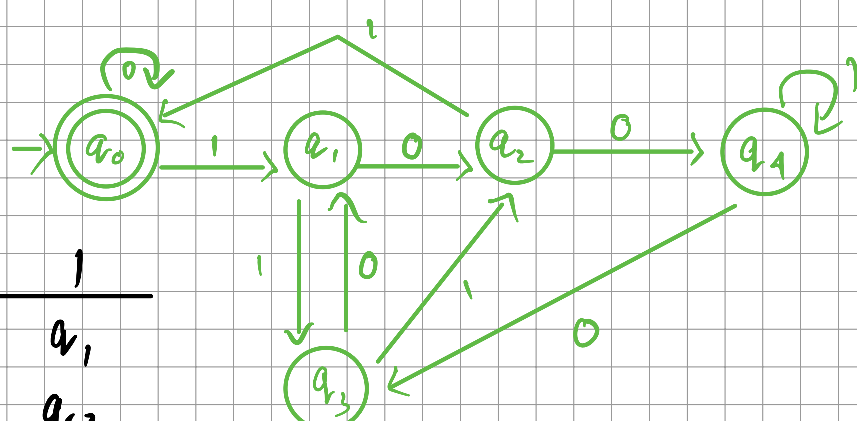
$\delta(\delta(q(0), 0) = \underline{q(0)}$

hence, 1011010 is accepted.

QUESTION

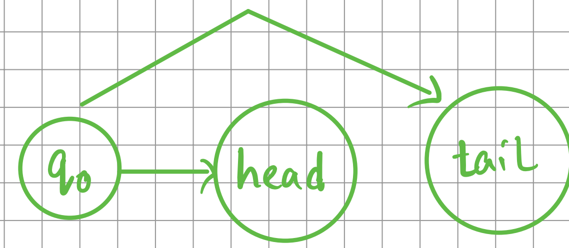
DFA that accepts binary strings divisible by 5.

ANSWER



	0	1
* q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

Non-deterministic finite Automata (NFA)



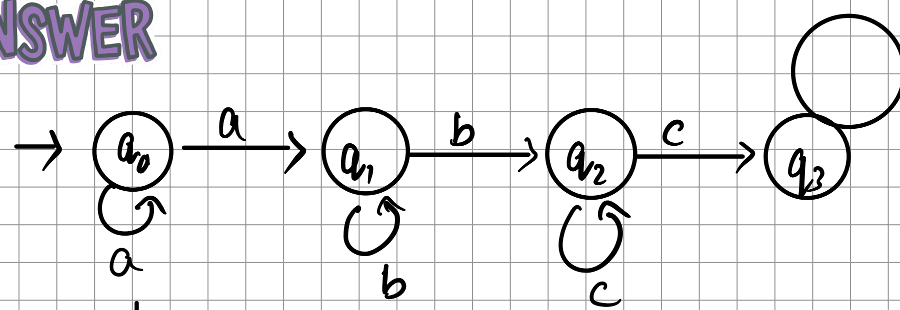
input – transit to multiple States (1 or more) [NFA]

input - transit to one state [DFA]

QUESTION

NFA for the language, $L=\{a^n b^m c^p / m,n,p \geq 1\}$

ANSWER



	a	b	c
q_0	$\{q_0, q_1\}$	ϕ	ϕ
q_1	ϕ	$\{q_1, q_2\}$	
q_2	ϕ	ϕ	$\{q_2, q_3\}$
q_3	ϕ	ϕ	ϕ

$$\delta(q_0, abbcc) = \delta(\delta(q_0, a), bbcc)$$

$$= \delta(\{q_0, q_1\}, bbcc) = \delta(\delta(q_0, b) \cup \delta(q_1, b), bcc)$$

$$= \delta(\phi \cup \{q_1, q_2\}, bcc) = \delta(\delta(q_1, b) \cup \delta(q_2, b), cc)$$

$$= \delta(\{q_1, q_2\} \cup \phi, cc) = \delta(\delta(q_1, c) \cup \delta(q_2, c), c)$$

$$= \delta(\phi \cup \{q_2, q_3\}, c) = \delta(\delta(q_2, c) \cup \delta(q_3, c))$$

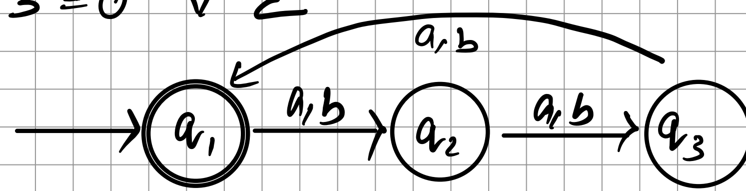
$$= \delta(\{q_2, q_3\} \cup \phi) = \{q_2, \underline{\underline{q_3}}\}$$

QUESTION

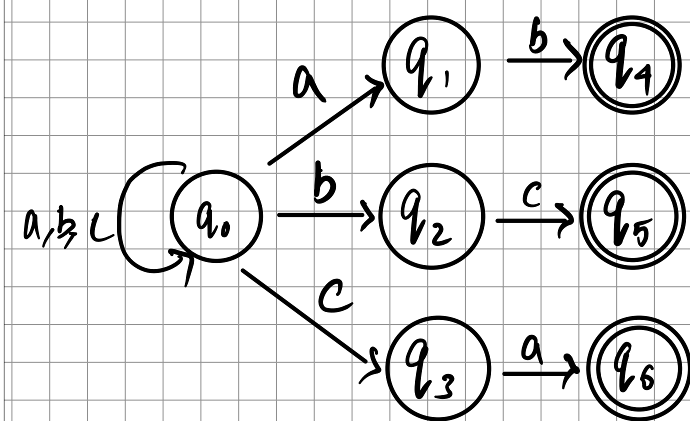
NFA for $\Sigma = \{a, b\}^*$, where

ANSWER

$|w| \bmod 3 = 0 \ \forall \Sigma$



NFA for $\Sigma = \{a, b, c\}$ ends with ab, bc, ca



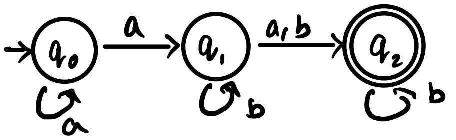
$\delta(q_0, abc)$

$= \delta(\delta(q_0, a), bc) = \delta(a$

$=$

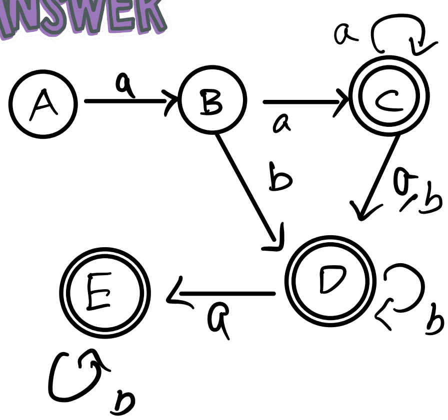
QUESTION

Convert the given NFA to DFA.

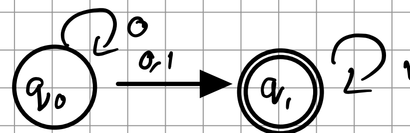


ANSWER

aa bab



Conversion of NFA to DFA



Step 1: Define NFA 'M'

$M = \{Q, \Sigma, \delta, S, F\}$

Q = Finite States $- \{q_0, q_1\}$

Σ = Input Symbol $- \{0, 1\}$

$\delta = Q \times \Sigma = 2^Q$

$S = \{q_0\}$, $F = \{q_1\}$

Step 2: Let M' be the DFA with δ' the transition of DFA.

$\delta'(q_0, 0) - \delta(q_0, 0) - \{q_0, q_1\}$ — (1)

$\delta'(q_0, 1) - \delta(q_0, 1) - \{q_1\}$ — (2)

Lets find the state for (1) with $\Sigma = \{0, 1\}$

$\delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\}$

Lets find states for (2) with $\Sigma = \{0, 1\}$

$\delta'(q_1, 0) = \{\phi\} = \phi$

$\delta'(q_1, 1) = \{q_1\}$

Lets rename;

Step 3: transition table

	0	1
q_0	$\{q_0, q_1\}$	q_1
q_0, q_1	$\{q_0, q_1\}$	q_1
q_1	ϕ	q_1

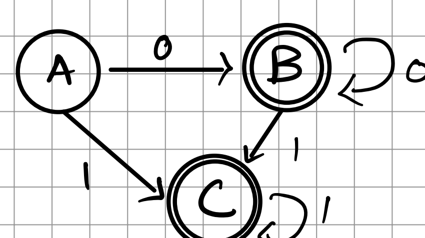
$q_0 \rightarrow A$

$q_0, q_1 \rightarrow B$

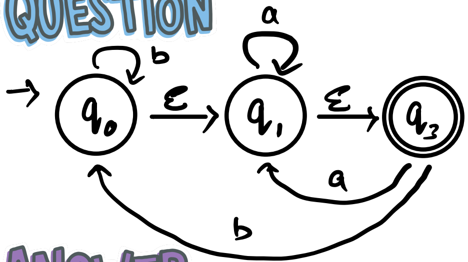
$q_1 \rightarrow C$

	0	1
A	B	C
* B	B	C
* C	ϕ	C

Step 4:- transition diagram



QUESTION



ANSWER

Step 1 :- Define NFA-ε-machine 'M'

$M = \{Q, \Sigma, \delta, S, F\}$

$Q = \{q_0, q_1, q_2, q_3\}$

$\delta = \{\varepsilon, a, b\}$

$S = \{q_0\}$

$F = \{q_3\}$

Step 2 :- Find $\hat{\delta}$ (DFA) for machine 'M'

$\varepsilon\ closure(q_0) = \{q_0, q_1\}$ — (A)

Apply Σ in (A)

$\hat{\delta}(A,a) = \varepsilon closure(\delta(A,a))$

$\varepsilon\ closure(\delta(q_0,q_1), a)$

$\varepsilon\ closure(\delta(q_0,a) \cup \delta(q_1,a))$

$\varepsilon\ closure(\phi \cup q_1)$

$\varepsilon\ closure(q_1 q_2)$ — B

$\hat{\delta}(A,b) = \varepsilon closure(\delta(A,b))$

$\varepsilon closure(\delta(q_0,q_1,q_2), b)$

$\varepsilon closure(\delta(q_0,b) \cup \delta(q_1,b) \cup \delta(q_2,b))$

$\varepsilon closure(q_0 \cup \phi \cup q_0)$

$\varepsilon\ closure(q_0) = \{q_0,q_1,q_2\}$ (A)

Find $\hat{\delta}$ for B with Σ

$\hat{\delta}(B,a)\ \varepsilon\ closure(\delta(q_1,a) \cup \delta(q_2,a))$

$= \varepsilon\ closure(q_1 \cup q)$

$= \varepsilon\ closure(q_1) = \{q_1,q_2\}$ — B

$\hat{\delta}(B,b) = \varepsilon\ closure(\delta(q_1,b) \cup \delta(q_2,b))$

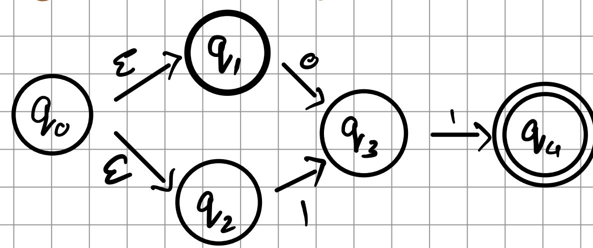
$= \varepsilon closure(\phi \cup q_0)$

$\varepsilon\ closure(q_0) = \{q_0,q_1,q_2\}$ - A

Step 3 :- transition table

Q \ Σ	a	b
* A	B	A
* B	B	A

Convert NFA-ε to DFA



Step 1 :- Define NFA-ε-machine 'M'

$M = \{Q, \Sigma, \delta, S, F\}$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\delta = \{\varepsilon, 0, 1\}$

$S = \{q_0\}$

$F = \{q_4\}$

Step 2 :- Find $\hat{\delta}$ (DFA) for machine 'M'

$\varepsilon\ closure(q_0) = \{q_0, q_1, q_2\}$ — (A)

Apply Σ in (A)

$\hat{\delta}(A,0) = \varepsilon\ Closure(\delta(A,0))$
$\varepsilon\ Closure(\delta(q_0,q_1,q_2), 0)$
$\varepsilon\ Closure(\delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_2,0))$
$\varepsilon\ Closure(\phi \cup q_3 \cup \phi)$
$\varepsilon\ Closure(q_3) = \{q_3\}$ —(B)

$\hat{\delta}(A,1) = \varepsilon\ Closure(\delta(q_0,q_1,q_2), 1)$
$\varepsilon\ Closure(\delta(q_0,1) \cup \delta(q_1,1) \cup (q_2,1))$
$\varepsilon\ Closure(\phi \cup \phi \cup q_3)$
$\varepsilon\ Closure(q_3) = \{q_3\}$ —(B)

Find $\hat{\delta}$ for B with $\Sigma = \{0,1\}$

$\hat{\delta}(B,0) = \varepsilon\ Closure(\delta(q_3,0))$
$= \varepsilon\ closure(\phi) = \phi$

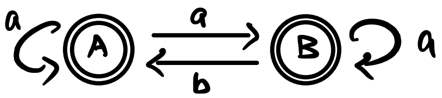
$\hat{\delta}(B,1) = \varepsilon\ closure(\delta(q_3,1)) = \{q_4\}$
$= \varepsilon\ closure(q_4)$

Find $\hat{\delta}$ for C with $\Sigma = \{0,1\}$

$\hat{\delta}(C,0) = \varepsilon\ Closure(\delta(q_4,0))$
$= \varepsilon\ closure(\phi) = \phi$

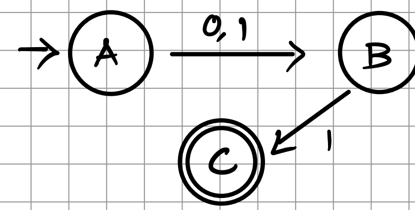
$\hat{\delta}(C,1) = \varepsilon\ closure(\delta(q_4,1))$
$= \varepsilon\ closure(\phi) = \phi$

Step 4 :- transition diagram

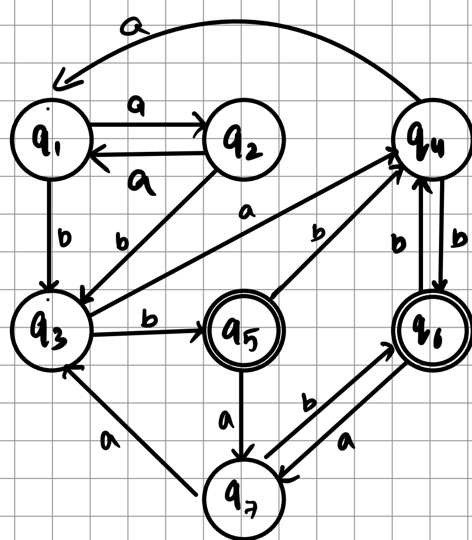


Obtained DFA :-

Q \ Σ	0	1
A	B	B
B	ϕ	C
C	ϕ	ϕ



Minimization of DFA



Step 1 : Define the DFA

$M' = \{Q, \Sigma, \delta, S, F\}$

Q - finite state $\{q_0, q_1, q_3\}$

Σ - input Symbols $\{1\}$

δ - Transition $Q \times \Sigma$

S - Starting state - $\{q_0\}$

F - Final State - $\{q_3\}$

Step 2 : transition table

	a	b
$\rightarrow q_1$	q_2	q_3
q_2	q_1	q_3
q_3	q_4	q_5
q_4	q_1	q_6
q_5	q_7	q_4
q_6	q_7	q_4
q_7	q_3	q_6

Step 3 : find equivalence

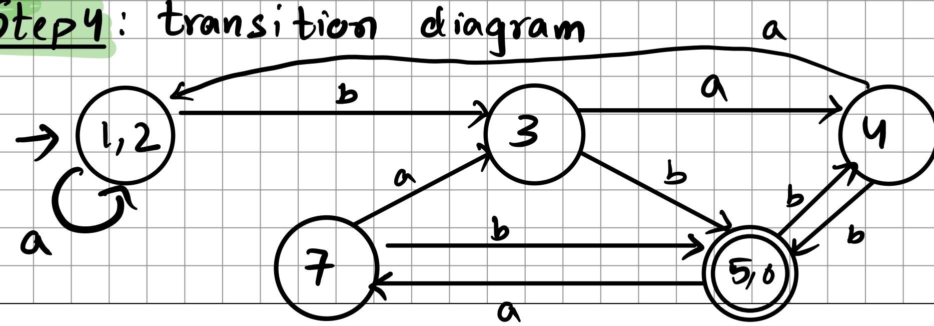
$0\,eq \rightarrow \{q_1, q_2, q_3, q_4, q_7\}\ \{q_5, q_6\}$

$1\,eq \rightarrow \{1, 2, 5, 6\}\ \{3, 4, 7\}$

$2\,eq \rightarrow \{1, 2\}\ \{5, 6\}\ \{3, 7\}\ \{4\}$

$3\,eq \rightarrow \{1, 2\}\ \{5, 6\}\ \{3\}\ \{4\}\ \{7\}$

Step 4 : transition diagram

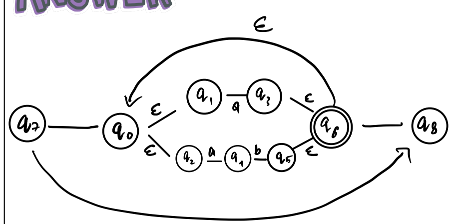


Subject:
Topic: Module-3
Date:
Previous Topic:

QUESTION

$(ab + a)^*$

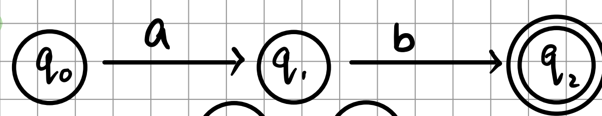
ANSWER



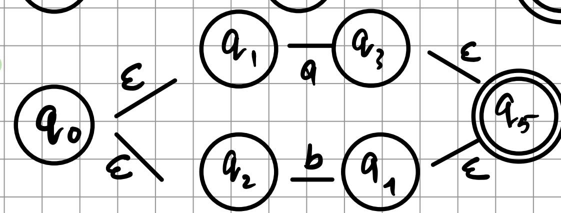
Regular Expressions

Language accepted by finiti automata Can be easily described by simple expressions Called regular expression.

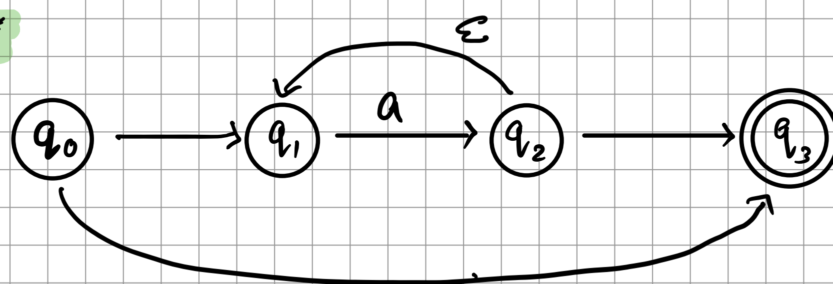
a.b



a+b



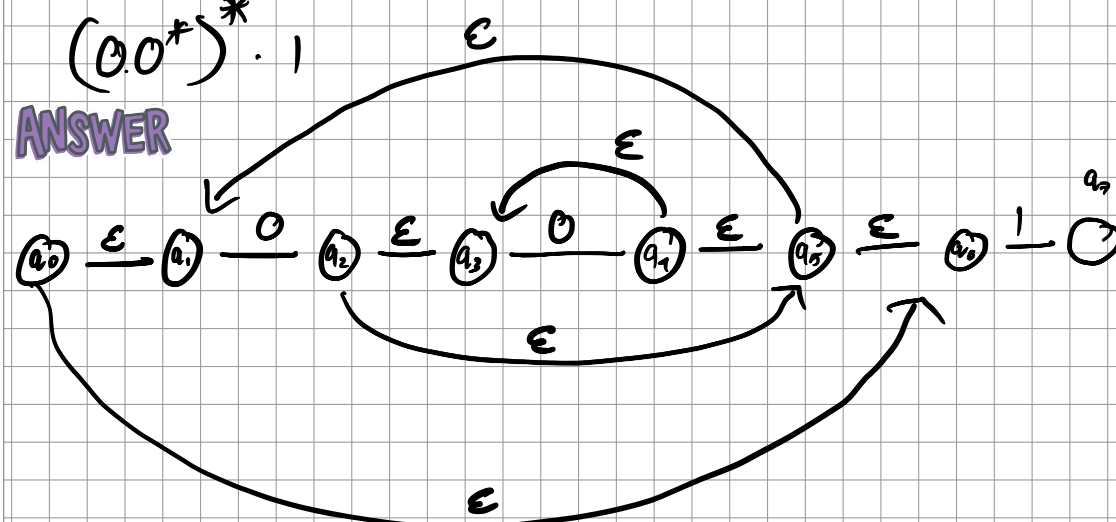
a^*



QUESTION

$(0.0^*)^* . 1$

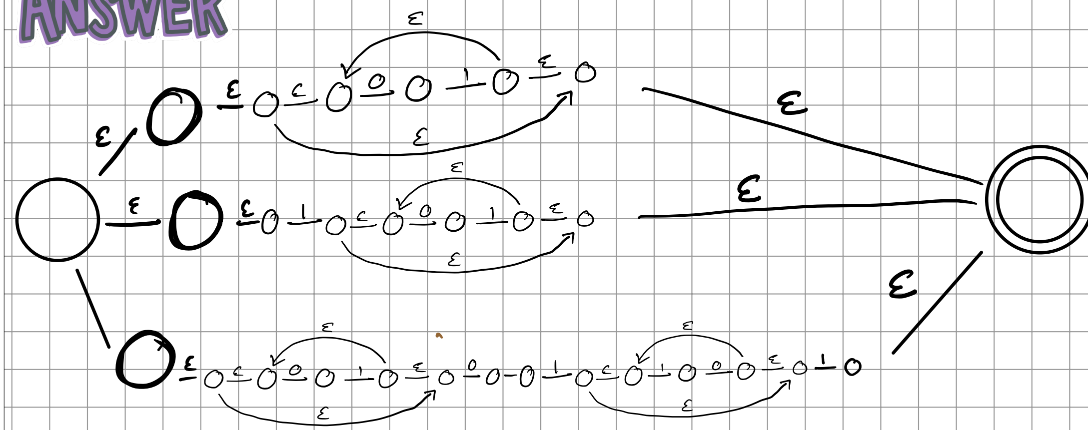
ANSWER

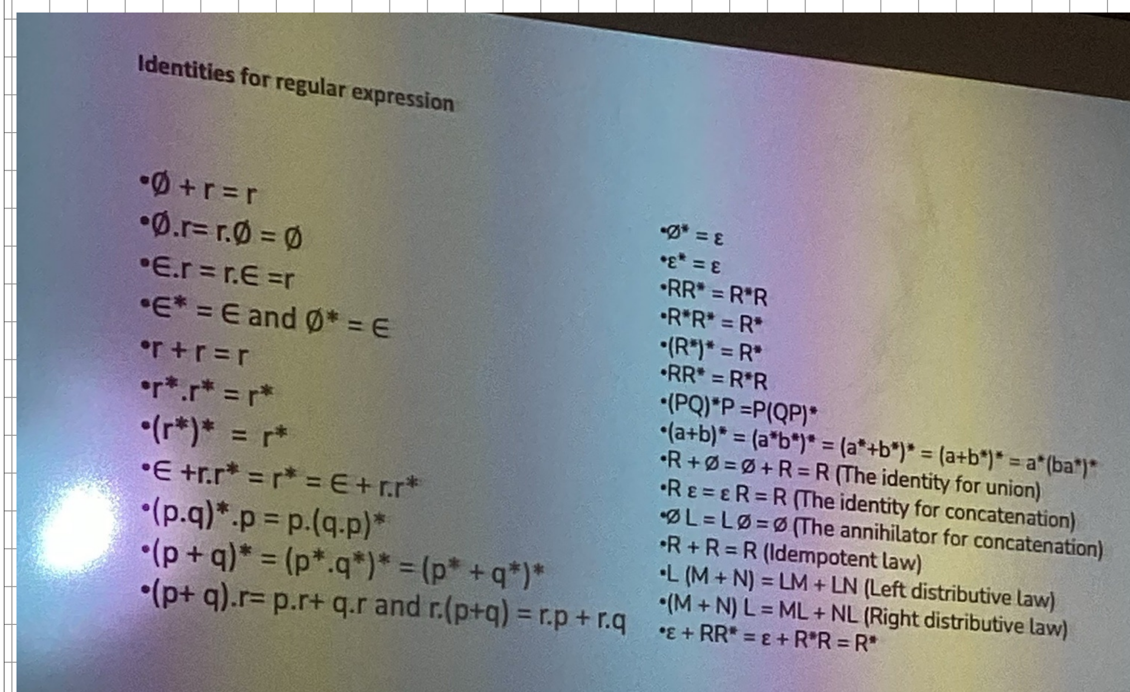


QUESTION

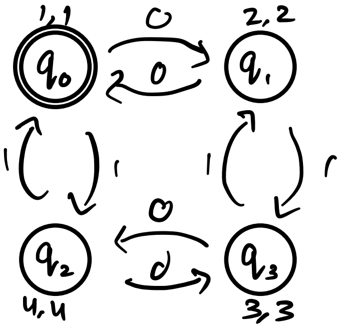
$(0.1)^* + 1(01)^* + (01)^*0.(10)^*.1$

ANSWER





QUESTION



ANSWER

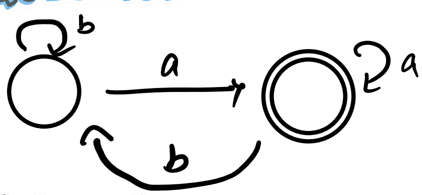
$R_{11} = \varepsilon$ $R_{21} = 0$ $R_{31} = \phi$ $R_{41} = 1$

$R_{12} = 0$ $R_{22} = \varepsilon$ $R_{32} = 1$ $R_{42} = \phi$

$R_{13} = \phi$ $R_{23} = 1$ $R_{33} = \varepsilon$ $R_{43} = 0$

$R_{14} = 1$ $R_{24} = \phi$ $R_{34} = 0$ $R_{44} = \varepsilon$

QUESTION



ANSWER

	k=0	k=1
R^0_{11}	b^*	b^*
R_{12}	a	$a b^*$
R_{21}	b	b^*
R_{22}	a^*	$a^* + ab^*$

when, $k=1$; $i=1$; $j=1$

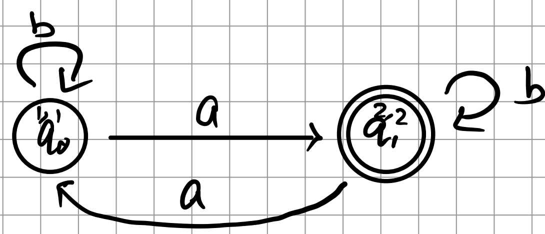
$$R^0_{11} = R^0_{11} + (R^0_{11})(R^0_{11})^* \cdot (R^0_{11})$$

$$= b^* + (b^*)(b^*)^* \cdot (b^*)$$

$$= b^* + b^* b^* b^*$$

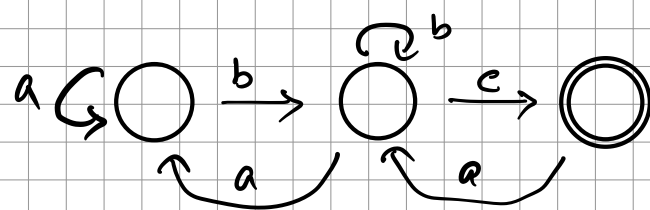
$$= b^* + b^+ = b^*$$

Converting finite Automata to regular expressions



$R_{11} = b^*$ $R_{2,2} = b^*$

$R_{12} = a$ $R_{2,1} = a$



$_1R_1 = a^*$ $_2R_1 = a$ $_3R_1 = \phi$

$_1R_2 = b$ $_2R_2 = b^*$ $_3R_2 = a$

$_1R_3 = \phi$ $_2R_3 = c$ $_3R_3 = \varepsilon$

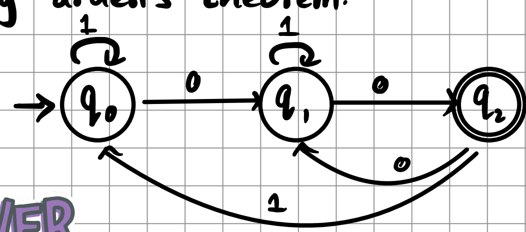
Arden's theorem

If, $R = Q + RP$

then, $R = QP^*$

QUESTION

Construct a regular expression for the following using arden's theorem.



ANSWER

Step 1: form all incoming equations

$$q_0 = \varepsilon + q_0 1 + q_2 1 \quad \text{——— ①}$$
$$q_1 = q_0 0 + q_1 1 + q_2 0 \quad \text{——— ②}$$
$$q_2 = q_1 0 \quad \text{——— ③}$$

Step 2: apply arden's theorem

$$q_1 = q_0 0 + q_1 1 + q_1 00 \qquad [\text{— ③}]$$

$$\underbrace{q_1}_{R} = \underbrace{q_0 0}_{Q} + \underbrace{q_1}_{R}\underbrace{(1+00)}_{P}$$

$$\Rightarrow q_1 = q_0 0(1+00)^* \quad \text{——— ④}$$

$$\Rightarrow q_0 = \varepsilon + q_0 1 + q_1 01 \qquad [\text{— ③}]$$
$$\Rightarrow q_0 = \varepsilon + q_0 1 + q_0 0(1+00)^* \qquad [\text{— ④}]$$

$$\Rightarrow \underbrace{q_0}_{R} = \underbrace{\varepsilon}_{Q} + \underbrace{q_0}_{R}\underbrace{(1+0(1+00)^* 01)}_{P}$$

$$\Rightarrow q_0 = \varepsilon(1+0(1+00)^* 01)^*$$

$$\Rightarrow q_0 = (1+0(1+00)^* 01)^* \qquad [\varepsilon \cdot R^* = R^*]$$

Pumping lemma for regular languages

It is used to prove that a language is not regular and cannot be used vice-versa

If L is a Regular language, then L has a pumping length 'P' such that any string 'S' where $|S| \geqslant P$ may be divided into 3 parts $S = xyz$ such that

(i) $xy^iz \in A$ for every $i \geqslant 0$
(ii) $|y| > 0$
(iii) $|xy| \leqslant P$

QUESTION

Prove that the given language is not regular

(i) $L = \{ a^n b^n \mid n \geqslant 1 \}$

(ii) $L = \{ a^p \mid p \text{ is prime} \}$

ANSWER

(i) Let L be a language and $w \in L$
$w = xyz$
$|w| \geqslant n$
$|xy| \leqslant n$

$w = xyz = a^n b^n$
$xy = a^n$; $z = b^n$; $y = a^m$

$xy^n z \in L \ \forall \ n \geqslant 1$

$xy^n z = xyy^{n-1} z = a^n (a^m)^{n-1} b^n = a^n . a^{mn-m} . b^n$

for $x=1 \rightarrow a^{n+m(1-1)} . b^n = a^n b^n$ (regular)
for $x=2 \rightarrow a^{n+m(2-1)} . b^n = a^{n+m} . b^n$ (irregular)
for $x=3 \rightarrow a^{n+m(2)} . b^n = a^{n+2m} . b^n$ (irregular)

(ii) let L be regular

let $w = a^P$; then, $|w| = P > n$

$w = xyz$; $|xy| \leqslant n$ and $|y| \geqslant 1$

$y = a^m$, $1 \leqslant m \leqslant n$; let $i = p+1 \Rightarrow p = i-1$

now, $|xy^iz| = |xyz| + |y^{i-1}| = |xyz| + |y|(i-1)$

$= p + m(i-1) = p + mp = (m+1)p$, is not prim

$\therefore xy^iz \in L$

L is not regular

QUESTION

Design a DFA for a library shelf where 1st book should be maths, 2nd book can be either physics or chemistry and third book should be computer.

ANSWER

Step 1 :- Define the DFA

$M = \{\Sigma, \delta, f, S\}$

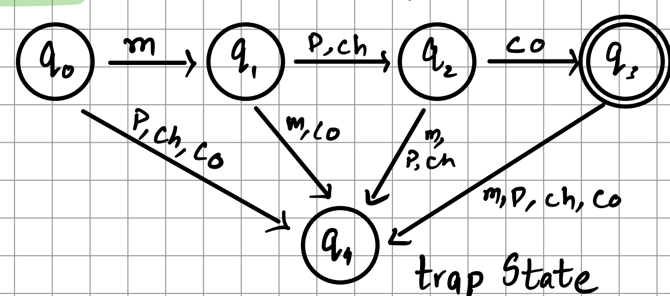
$\Sigma = \{ m, p, ch, co \}$

$\delta = Q \times \Sigma$

$S = \{q_0\}$

$f = \{q_1\}$

Step 2 :- transition diagram



Step 3 :- transition table

	m	p	ch	co
q_0	q_1	q_4	q_4	q_4
q_1	q_4	q_2	q_2	q_4
q_2	q_4	q_4	q_4	q_3
q_3	q_4	q_4	q_4	q_4

Subject: Topic: **Module -4**
Date: Previous Topic:

Context free Grammar

Let G be a grammar

$G \rightarrow \{N, T, S, P\}$

$N \rightarrow$ Non-terminal
$T \rightarrow$ Terminal
$S \rightarrow$ Starting state
$P \rightarrow$ production

QUESTION

Using the given grammar generate string.

00101, 1001, 00011

$S \rightarrow A1B$
$A \rightarrow 0A / \varepsilon$
$B \rightarrow 0B / 1B / \varepsilon$

Step (1) Define Grammar

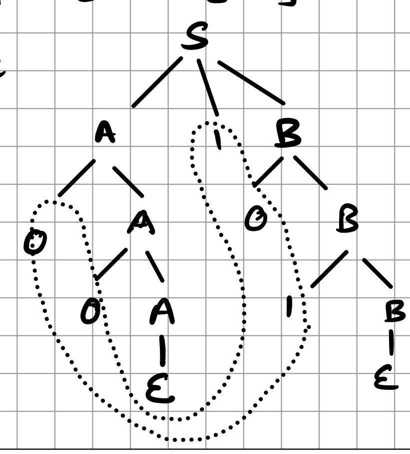
$G \rightarrow \{N, T, S, P\}$

N - Non Terminal - $\{S, A, B\}$
T - Terminal - $\{0, 1, \varepsilon\}$
S - Starting state - $\{S\}$

00101 (Left most deviation)

$S \rightarrow A1B$
$S \rightarrow 0A1B$ [$\because R_2$]
$S \rightarrow 00A1B$ [$\because R_2$]
$S \rightarrow 00\varepsilon 1B$ [$\because R_2$]
$S \rightarrow 001B$ [$\because R * \varepsilon = R$]
$S \rightarrow 0010B$ [R_3]
$S \rightarrow 00101B$ [R_3]
$S \rightarrow 00101\varepsilon$ [R_3]
$S \rightarrow 00101$ [$\because R * \varepsilon = R$]

Parse tree

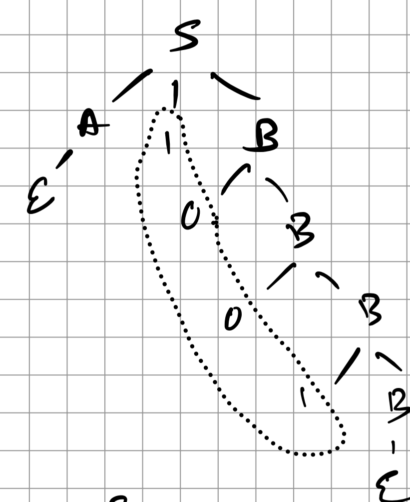


00101 (Right most derivation)

S → A1B
S → A10B [∵ R_3]
S → A101B [∵ R_3]
S → A101ε [∵ R_3]
S → 0A101 [∵ R_2]
S → 00A101 [∵ R_2]
S → 00ε101 [∵ R_2]
S → 00101

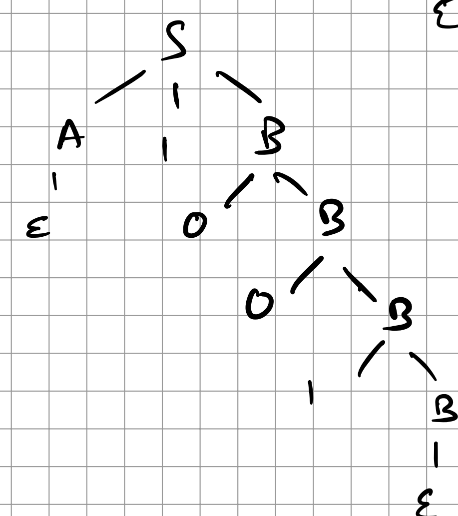
1001 (LMD)

S → A1B R_2
S → ε1B R_2
S → 1B R*ε-R
S → 10B R_3
S → 100B R_3
S → 1001B R_3
S → 1001ε R_3
S → 1001



1001 (RMD)

S → A1B
S → A10B
S → A100B
S → A1001B
S → A1001ε
S → A1001
S → ε1001
S → 1001



000 11

$S \xrightarrow{LMD}$ A1B
$S \xrightarrow{LMD}$ 0A1B
$S \xrightarrow{LMD}$ 00A1B
$S \xrightarrow{LMD}$ 000A1B
$S \xrightarrow{LMD}$ 000ε1B
$S \xrightarrow{LMD}$ 0001B
$S \xrightarrow{LMD}$ 00011B
$S \xrightarrow{LMD}$ 00011ε
$S \xrightarrow{LMD}$ 00011

$S \xrightarrow{RMD}$ A1B
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$
$S \xrightarrow{RMD}$

QUESTION

Using given grammar generate String

$E \rightarrow I \,/\, E+E \,/\, E*E \,/\, (E)'$

$I \rightarrow a \,/\, b \,/\, Ia \,/\, Ib \,/\, I0 \,/\, I1$

$(a+b)$

Step (1) Define Grammar

$G \rightarrow \{N, T, S, P\}$

N - Non Terminal -
T - Terminal -
S - Starting state - {S}

$(a+b)$

$E \xrightarrow{LMD} (E)$ R_1
$E \xrightarrow{LMD} (E+E)$ R_1
$E \xrightarrow{LMD} (I+E)$ I+I
$E \xrightarrow{LMD} (a+E)$ R_2
$E \xrightarrow{LMD} (a+I)$ I+I
$E \xrightarrow{LMD} (a+b)$ R_2

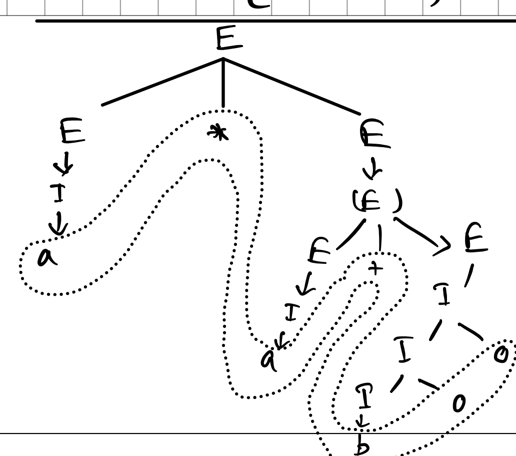
HOMEWORK

② $(a10 * b+a)$

$E \xrightarrow{LMD} (E)$
$E \xrightarrow{LMD} (E * E)$
$E \xrightarrow{LMD} (E * E+E)$
$E \xrightarrow{LMD} (E * Ia+Ib)$
$E \xrightarrow{LMD} (E * a+b)$
$E \xrightarrow{LMD} (I1 * a+b)$
$E \xrightarrow{LMD} (I1I0 * a+b)$
$E \xrightarrow{LMD} (a10 * a+b)$

① $a * (a+b00)$

$E \xrightarrow{LMD} E * E$ [R1]
$E \xrightarrow{LMD} I * E$ [R1]
$E \xrightarrow{LMD} a * E$ [R2]
$E \xrightarrow{LMD} a * (E)$ [R1]
$E \xrightarrow{LMD} a * (E+E)$ [R1]
$E \xrightarrow{LMD} a * (I+E)$ [R1]
$E \xrightarrow{LMD} a * (a+E)$ [R2]
$E \xrightarrow{LMD} a * (a+I)$ [R2]
$E \xrightarrow{LMD} a * (a+I0)$ [R2]
$E \xrightarrow{LMD} a * (a+I00)$ [R2]
$E \xrightarrow{LMD} a * (a+b00)$



Ambiguous Grammar

A context free grammar is said to be ambiguous if there exists two or more LMD or RMD or parse tree for same $w \in L(G)$

QUESTION

Find if the given grammar is ambiguous

$$x \rightarrow x+x \,/\, x*x \,/\, x \,/\, a$$

$a+a*a$

$x \xrightarrow{LMD} x+x$
$x \xrightarrow{LMD} 1+x$
$x \xrightarrow{LMD} a+x$
$x \xrightarrow{LMD} a+x*x$
$x \xrightarrow{LMD} a+1*1$
$x \xrightarrow{LMD} a+a*a$

$\therefore$ Since there is only one LMD, it is not ambiguous

Converting language to grammar

$L = a^n b^n \;\forall\; n \geq 1 \longrightarrow$

$g \rightarrow asb \,/\, \varepsilon$

$L \; w\,w^R \Rightarrow w\{a,b\}$

$w = aab$
$w^R = baa$

$g \rightarrow \{asa \,/\, bsb \,/\, \varepsilon\}$

$L = a^{2n} b^n \;\forall\; n \geq 1$

$x \longrightarrow \{aasb\}$

$L = \{a^n b^{n-3} \;\forall\; n \geq 3\}$

$g \rightarrow \{asb \,/\, aaa\}$

$L = \{a^n b^m \;\forall\; n \geq 1\}$

$g \rightarrow \{as \,/\, sb\}$

$L = \{a^n b^n c^m d^m \;\forall\; n,m \geq 1\}$

$g \rightarrow \{asb \,/\, csd\}$

Simplification of grammar

1. remove useless stuff
2. remove ε
3. remove unit production

① remove useless from given grammar

$T \rightarrow aaB \,/\, abA \,/\, aaT$

$A \rightarrow aA$
$B \rightarrow ab \,/\, b$
$C \rightarrow ad$

② $P - \{\; T \rightarrow aaB \,/\, aaT$
$\qquad B \rightarrow ab \,/\, b \;\}$

$S - \{T\}$

$S \rightarrow AB \,/\, a$
$A \rightarrow b$
$B \rightarrow D$

$S = aaB$
$aa\,ab$

S

③ $S \rightarrow aAa \,/\, bBb$
$A \rightarrow C \,/\, a$
$B \rightarrow C \,/\, b$
$\cancel{C \rightarrow CD}$
$\cancel{D \rightarrow A \,/\, B \,/\, ab}$

$S \rightarrow aAa \qquad S \rightarrow bBb$
$S \rightarrow aaa \qquad S \rightarrow bbb$

④ $S \rightarrow a \,/\, bXY$

$A \rightarrow Bda \,/\, bSX \,/\, a$

$\cancel{B \rightarrow aSB \,/\, bBX}$

$X \rightarrow \cancel{SBD} \,/\, \cancel{aBx} \,/\, ad$

$Y \rightarrow \cancel{SBa \,/\, XYb}$

$S \rightarrow bXY$

S

⑤ $S \rightarrow aZ \,/\, SY \,/\, XA$

$X \rightarrow bSZa$
$\cancel{Y \rightarrow aSY \,/\, bYZ}$
$Z \rightarrow \cancel{aYZ} \,/\, ad$
$A \rightarrow ab \,/\, \cancel{aA}$

$S \rightarrow XA$
$S \rightarrow bSZaA$
$S \rightarrow baZZaA$
$S \rightarrow baadZaA$
$S \rightarrow baadadaA$
$S \rightarrow baadadaab$

Simplification of CFG

Elimination of 'ε'

$S \rightarrow xyz$
$x \rightarrow 0x / \varepsilon$
$y \rightarrow 1y / \varepsilon$

Sub $x \rightarrow \varepsilon$

$S \rightarrow xyz / yz / xy / y$
$x \rightarrow 0x / 0$
$y \rightarrow 1y / \varepsilon$

Sub $y \rightarrow \varepsilon$

$S \rightarrow xyz / yz / xy / y / xz / z / x$
$x \rightarrow 0x / 0$
$y \rightarrow 1y / 1$

finally,

$S \rightarrow xyz / yz / xy / y / xz / x$

$x \rightarrow 0x / 0$

$y \rightarrow 1y / 1$

Chomsky's Normal Form (CNF)

$N \rightarrow NN$
$N \rightarrow T$

$N \rightarrow$ Non-Terminal
$T \rightarrow$ Terminal

QUESTION

Convert the given CFG to CNF

$X \rightarrow 0Y1 / 1Y0 / Z$
$Y \rightarrow 0Y / 1Y / \varepsilon$
$Z \rightarrow 010$
$S \rightarrow ZS / SY$

ANSWER

Step 1: Define the grammar

Let, $G = \{N, T, P, S\}$

$N \rightarrow \{S, X, Y, Z\}$
$T \rightarrow \{0, 1\}$
$S \rightarrow \{S\}$
$P \rightarrow \{\text{Productions}\}$

Step 2: remove useless parts

$X \rightarrow 0Y1 / 1Y0 / Z$
$Y \rightarrow 0Y / 1Y / \varepsilon$
$Z \rightarrow 010$
~~$S \rightarrow ZS / SY$~~ (non-terminal)

Step 3: remove ε

$X \rightarrow 0Y1 / 1Y0 / Z / 01 / 10$
$Y \rightarrow 0Y / 1Y / 0 / 1$
$Z \rightarrow 010$

Step 4: remove unit production

$X \rightarrow 0Y1 / 1Y0 / 010 / 01 / 10$
$Y \rightarrow 0Y / 1Y / 0 / 1$

Step 5: CNF Conversion

$C_1 = 0$	$D_1 = C_1 Y$	$E_1 = C_1 C_2$
$C_2 = 1$	$D_2 = C_2 Y$	

$X \rightarrow D_1 C_2 / D_2 C_1 / E_1 C_1 / C_1 C_2 / C_2 C_1$

$Y \rightarrow C_1 Y / C_2 Y / C_1 / C_2$

Greibach Normal Form (GNF)

$\boxed{A \rightarrow aX}$

$a \rightarrow$ Terminal
$X^* \rightarrow$ Non-terminal

QUESTION

Convert the following CFG to GNF

$S \rightarrow aSb$
$S \rightarrow ab$

ANSWER

Step 1 :- define grammar

Let, $G = \{N, T, P, S\}$

$N \rightarrow \{S\}$
$T \rightarrow \{a, b\}$
$S \rightarrow \{S\} \rightarrow$ Starting State
$P \rightarrow \{$ Productions $\}$

Step 2: make a new non-terminal for every terminal symbol

$S \rightarrow ASB$ $\quad A \rightarrow a$
$S \rightarrow AB$ $\quad B \rightarrow b$

Step 3 : rename the non-terminals

$S \rightarrow A_1$
$A \rightarrow A_2$
$B \rightarrow A_3$

$\rightarrow$

$A_1 \rightarrow A_2 A_1 A_3$
$A_1 \rightarrow A_2 A_3$

Step 4: use, $A_i \rightarrow A_j Y$; $j > i$

$A_1 \rightarrow a A_1 A_3$
$A_1 \rightarrow a A_3$

Closure properties of CFL

1) A CFL is closed under

(a) Union $\rightarrow$ If L_1 and L_2 are CFL, $L_1 \cup L_2$ is CFL.

(b) Concatenation $\rightarrow$ If L_1 and L_2 are CFL, $L_1 L_2$ is CFL.

(c) Kleene closure $\rightarrow$ If L_1 is CFL, L_1^* is CFL.

(d) reversal $\rightarrow$ If L is CFL, L^R is CFL

2) A CFL is not closed under

(a) Intersection

(b) Complementation

3) If L is CFL & R is regular then,

$L \cap R$ is CFL

Subject: Topic: Module 5

Date: Previous Topic:

Push down Automata (PDA)

QUESTION

Construct a PDA for $a^n b^n \mid n \geq 0$

ANSWER

Step 1 :- Define the PDA

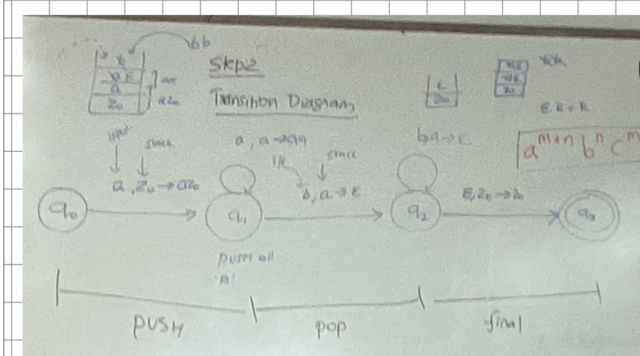
let 'M' be a PDA where 7 tuples

$M = \{Q, \Sigma, \Gamma, S, Z_0, \delta, F\}$

Q = Finite states
Σ = input symbol of FA
Γ = input symbol of stack
S = Starting state
Z_0 = Starting state of stack
δ = Transition
F = final State

{1 mark}

Step 2 :-



(drawing) → {5 marks}

a^n | b^n

push pop

Step 3 :- Transition function

$\delta(q_0, a, z_0) \vdash (q_1, aZ_a)$

$\delta(q_1, a, a) \vdash (q_1, aa)$

$\delta(q_1, b, a) \vdash (q_2, \epsilon)$

$\delta(q_2, b, a) \vdash (q_2, \epsilon_1)$

$\delta(q_2, \epsilon, Z_0) \vdash (q_3)$

{3 marks}

Step 4: Validate an input aaabbb

$\delta(q_0, aaabbb, z_0) \vdash (q_1, aabbb, az_0)$

$\delta(q_1, aabbb, az_0) \vdash (q_1, abbb, aaz_0)$

$\delta(q_1, abbb, aaz_0) \vdash (q_1, bbb, aaaz_0)$

$\delta(q_1, bbb, aaaz_0) \vdash (q_2, \epsilon bb, \epsilon aaz_0)$ {1 mark}

$\delta(q_2, bb, aaz_0) \vdash (q_2, \epsilon b, \epsilon az_0)$

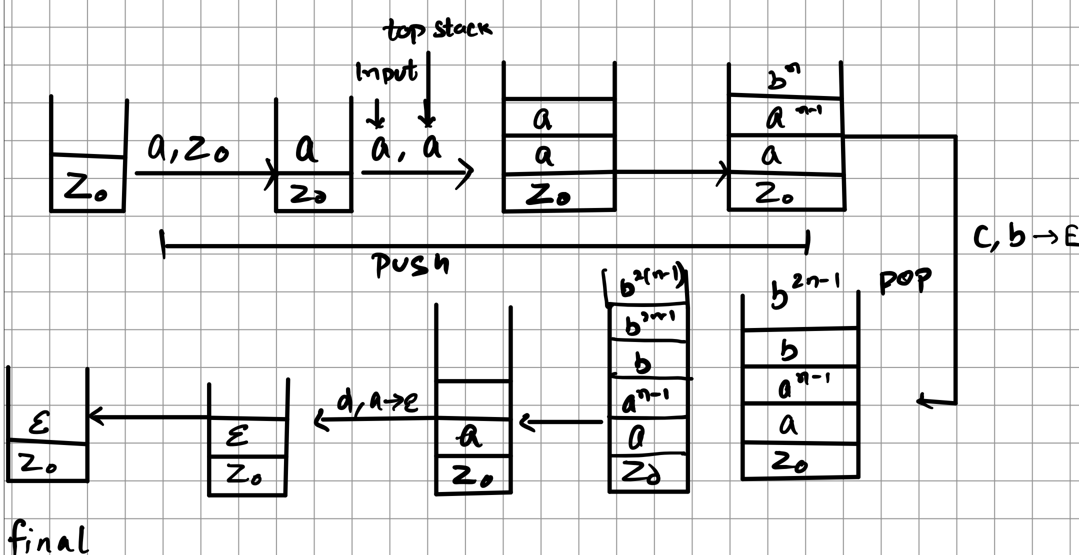
$\delta(q_2, b, az_0) \vdash (q_2, \epsilon, \epsilon z_0)$

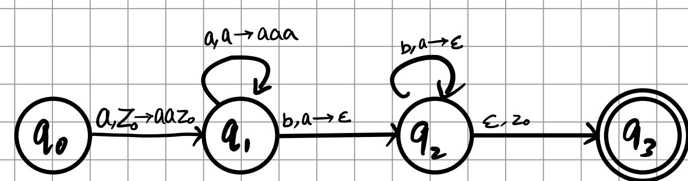
$\delta(q_2, \epsilon, z_0) \vdash (q_3, z_0)$

QUESTION

$a^n b^{2n}$

ANSWER





QUESTION

$a^n b^m \mid c^n d^n$

ANSWER

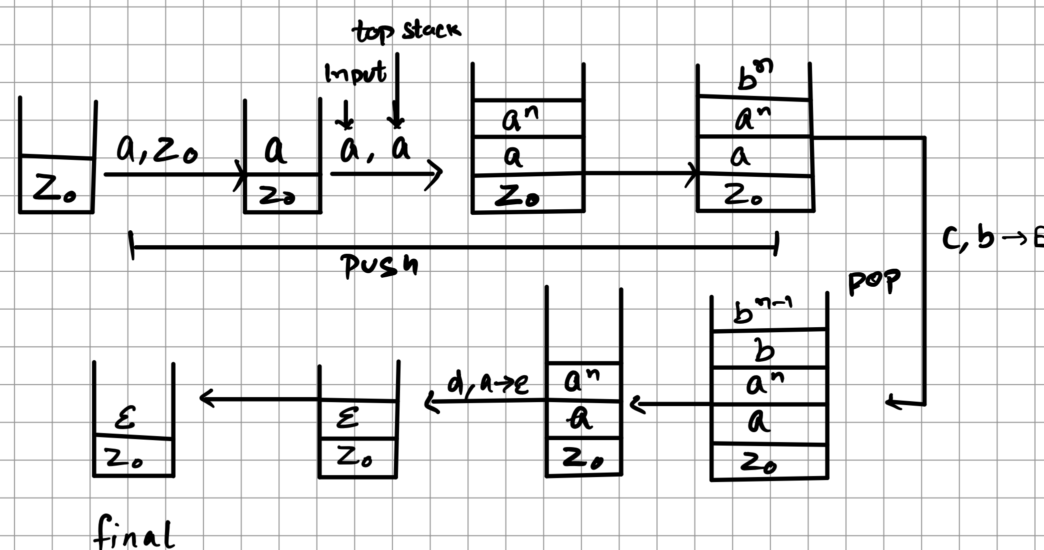
Step 1 :- define PDA ①

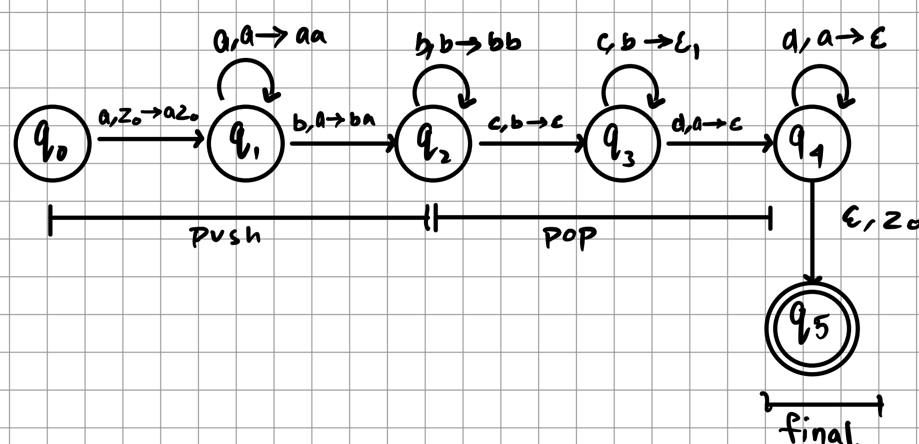
Step 2 :- transition diagram ⑤

Step 3 :- transition function ③

Step 4 :- Validate with a string ②

Step 5 :- Conclude





QUESTION

$L = wcw^R \quad \forall\, w = \{a, b\}^*$

$w = aaab$
$w^R = baaa$

ANSWER

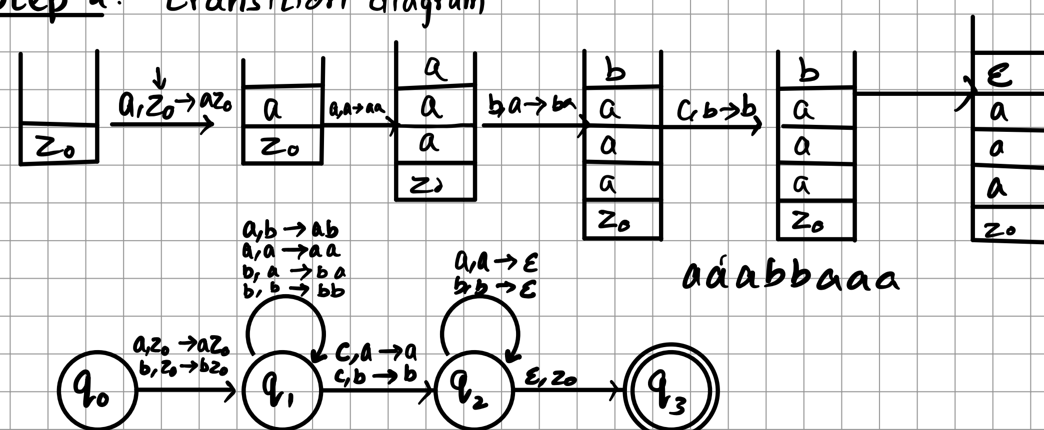
Step 1:- Define PDA

let 'M' be a PDA where 7 tuples

$M = \{Q, \Sigma, \Gamma, S, Z_0, \delta, F\}$

Q = Finite states
Σ = input symbol of FA
Γ = input symbol of stack
S = Starting state
Z_0 = Starting state of stack
δ = Transition
F = final State

Step 2:- transition diagram

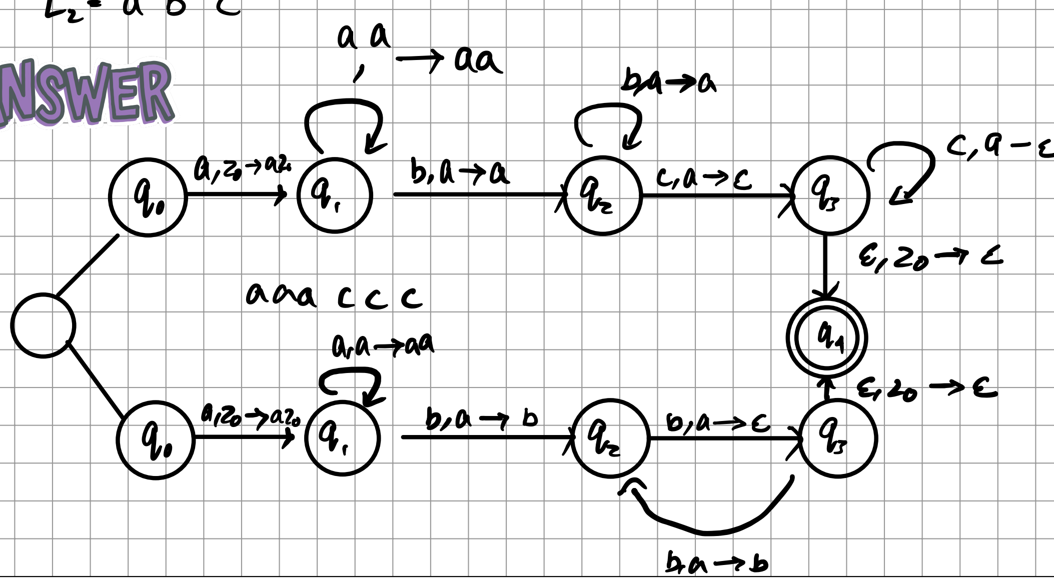


aáabbaaa

QUESTION

$L_1 = a^{2n} b^n$
$L_2 = a^n b^m c^n$

ANSWER



aaa ccc

Subject: Topic: Module - 6

Date: [redacted] Topic: Turing Machine

QUESTION

Construct a turing machine which accepts

$L = \{ a^n b^n c^n / n \geq 1 \}$

ANSWER

Step 1 : define the turing machine

$M = \{ Q, \Sigma, \Gamma, \delta, q_0, B, F \}$

$Q \rightarrow \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma \rightarrow \{a, b, c\}$

$\Gamma \rightarrow \{X, Y, Z, a, b, c, \sqcup\}$

$\delta \rightarrow$

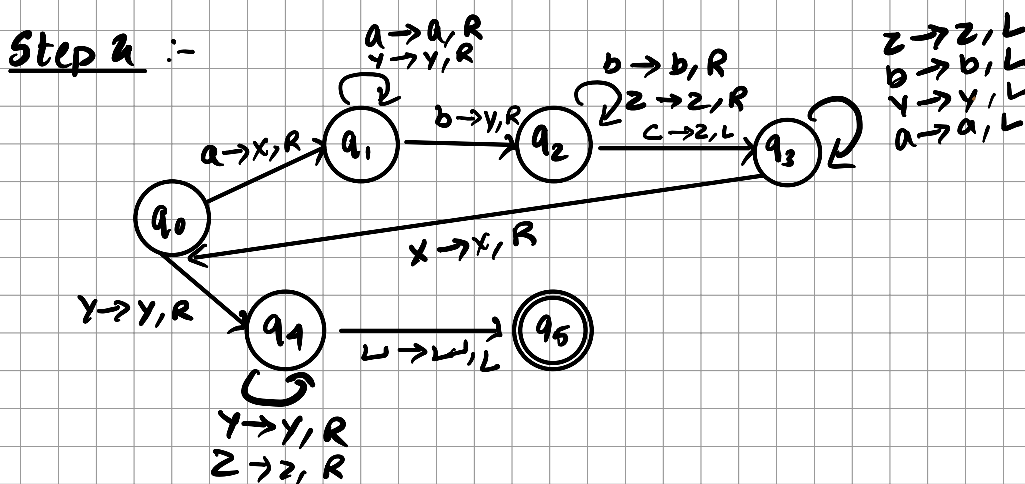
$q_0 \rightarrow \{q_0\}$

$B \rightarrow \sqcup$

$F \rightarrow \{q_5\}$

xaa ybbzcc

Step 2 :-



Step 3 :- transition state

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, Y) = (q_1, Y, R)$

$\delta(q_1, b) = (q_2, Y, R)$

$\delta(q_2, b) = (q_2, b, R)$

$\delta(q_2, Z) = (q_2, Z, R)$

$\delta(q_2, C) = (q_3, Z, L)$

$\delta(q_3, Z) = (q, Z, L)$

$\delta(q_3, b) = (q, b, L)$

$\delta(q_3, Y) = (q, Y, L)$

$\delta(q_3, a) = (q, a, L)$

$\delta(q_3, X) = (q, X, R)$

$\delta(q_0, Y) = (q, Y, R)$

$\delta(q_4, Y) = (q, Y, R)$

$\delta(q_4, Z) = (q, Z, R)$

$\delta(q_4, \sqcup) = (q, \sqcup, L)$

Step 4 : Validation 'aabbcc'

$q_0 aabbcc \vdash Xq_1abbcc \vdash Xaq_1bbcc$

$XaYq_2bcc \vdash XaYbq_2cc \vdash XaYbq_3Zc$

$\vdash XaYq_3bZc \vdash Xaq_3YbZc$

$\vdash Xq_3aYbZc \vdash Xq_0aYbZc \vdash XXq_1YbZc$

$\vdash XXYq_1bZc \vdash XXYYq_2Zc \vdash$

$\vdash XXYYZq_2c \vdash XXYYZZq_3 \vdash$

Subject:

Topic: Module - 7

Date:

Previous Topic:

Recursive → decidable

⇒ The language accepted by Turing machine for which it always halt.

Recursively Enumerable → partially decidable

⇒ The language accepted by Turing machine, may or maynot halt

Undecidable

⇒ No turing machine for this.

Properties of recursive and recursively enumerable

1) Union of two recursive is recursive
2) Union of two recursively enumerable is recursively enumerable
3) Complement of recursive is recursive
4) If L and L' are recursively enumerable, L is recursive

Post Correspondence problems

QUESTION

	List A	list B
1)	ba	ab
2)	bba	abb
3)	bba	ab
4)	aaa	aa
5)	ba	aba

4 2 1 5

4) aaabbababa
aaabbababa

aaa / bba | ba | / ba
aa / abb | ab | / aba

bba
ab

aaabba
aaab

QUESTION

	List A	List B
1	10	101
2	011	11
3	101	011

ANSWER

1) 10
101

3) 10101
1010111

QUESTION

	list A	list B
1	B	CA
2	A	AB
3	CA	A
4	ABC	C

ANSWER

2) A
AB

1) AB
ABCA

3) ABCA
ABCAA

2) ABCA A
ABCA A AB

4) ABCAAABC
ABCAAABC