Don't take tension, one or two are enough →

Deterministic finite automata: On each input there is one & only state to which automation can transition from its current state. It consists of

$(Q, \Sigma, \delta, q_0, F) \Longrightarrow$ Notation

A finite set of states, Q
A finite set of input symbols, Σ
Transition function, δ. $\rightarrow \delta(q, a) = P$
Initial state q_0
A set of final or accepting states F

q = Present state
a = input
P = next state

Notation of DFA:

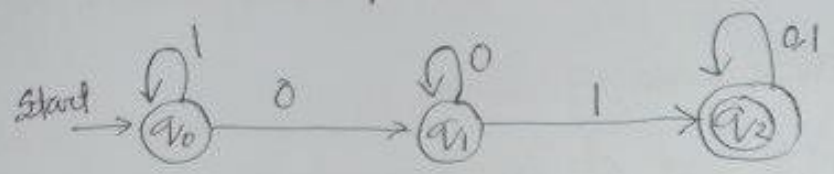
Transition Diagram
Transition Table.

Q) How a DFA processes a string?

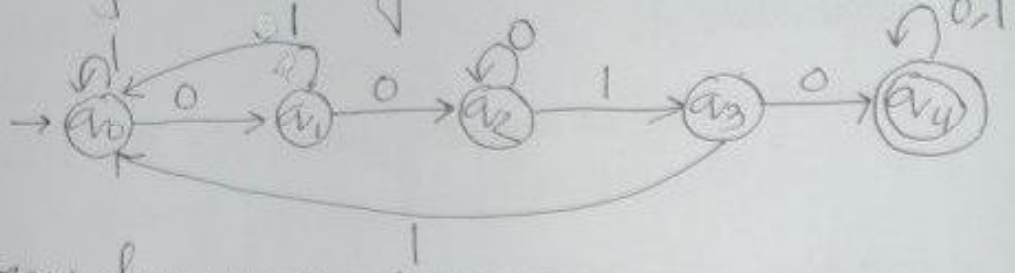
→ We start out with DFA in q_0. Transition function, $\delta(q_0, a) = q_1$ we process next symbol a_2 by evaluating $\delta(q_1, a_2)$. If the state is q_2. Then common format is $\delta(q_{i-1}, a_i) = q_i$. If

q_n is the final state. Then it is accepted. Otherwise rejected.

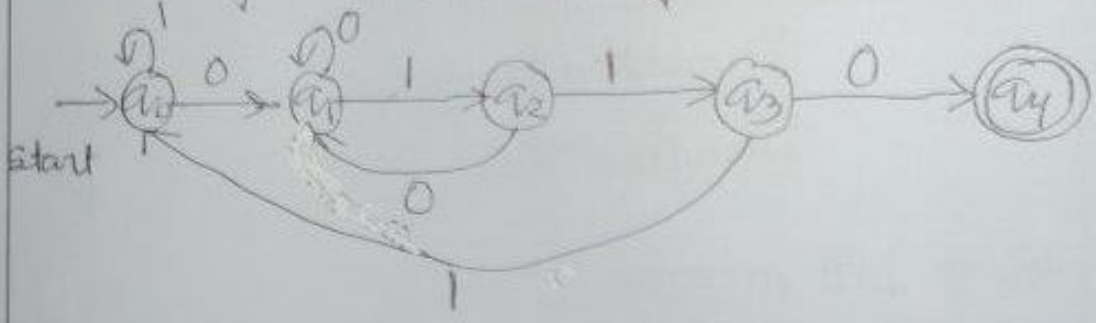
↳ Draw transition diagram for the DFA accepting string with substring 01.



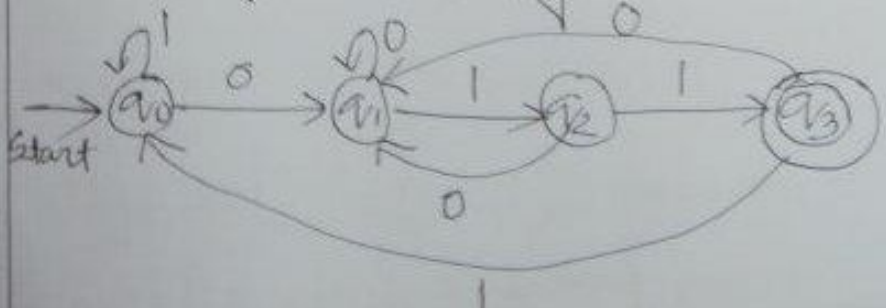
↳ Draw for substring 0010



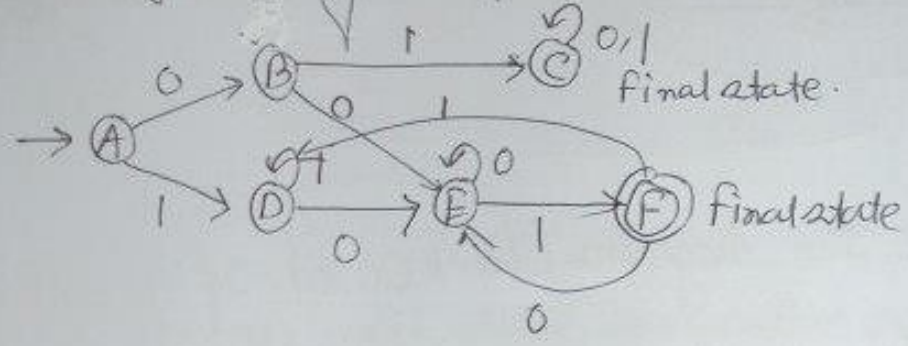
↳ Draw for 0110 substring



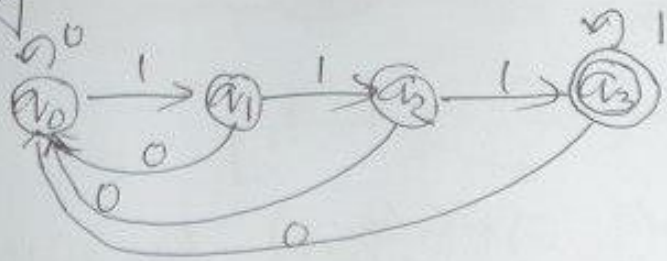
↳ Draw for substring end with 01.



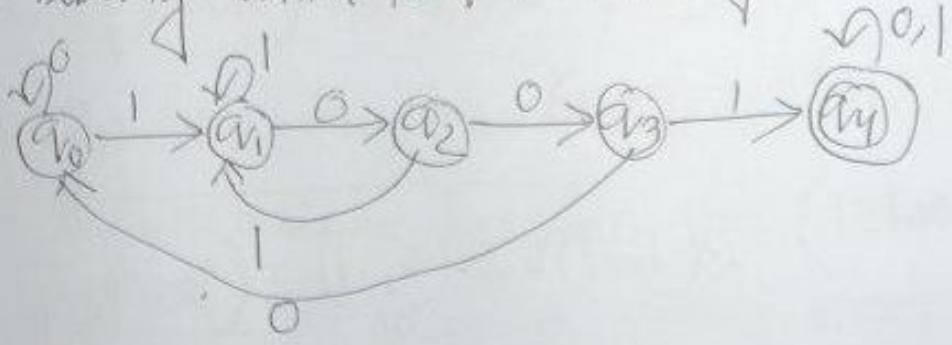
→ Set of string begin or end with 01.



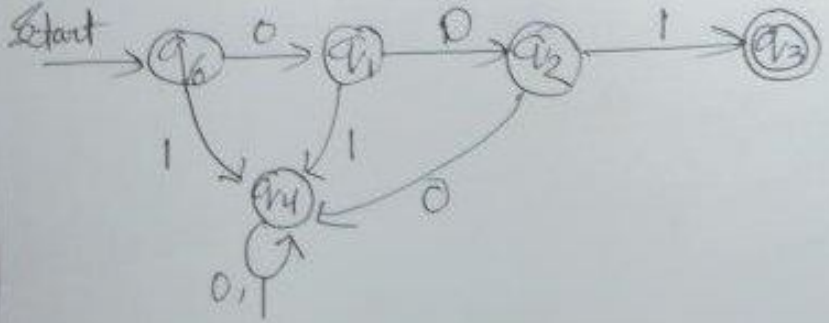
→ String end with 111



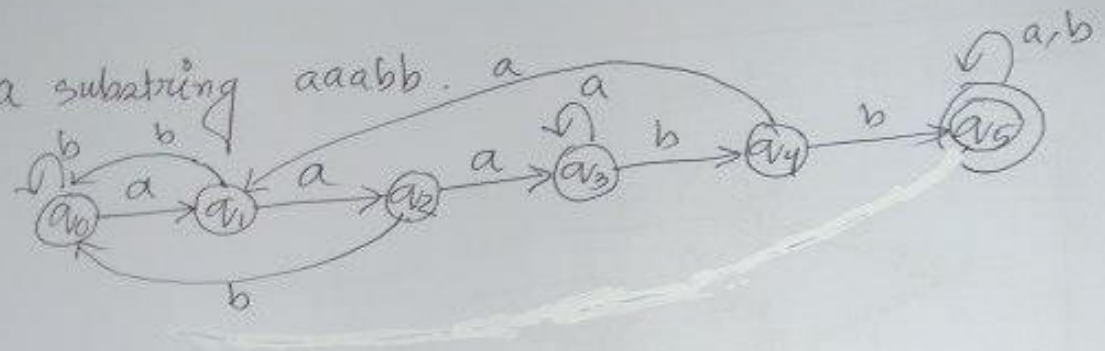
→ a string with 1001 substring



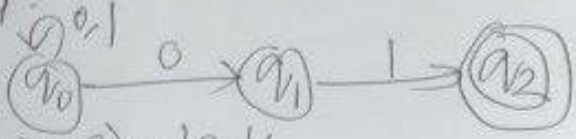
→ a string begin with 001



is a substring aaabb.



↳ Given the diagram. Check 00101 if accepted or not.



$\hat{\delta}(q_0, \epsilon) = \{q_0\}$

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

$\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = (q_0, q_1) \cup \phi = \{q_0, q_1\}$

$\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0, q_2\}$

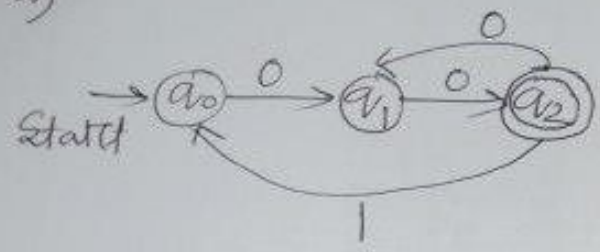
$\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$

$\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0, q_2\}$

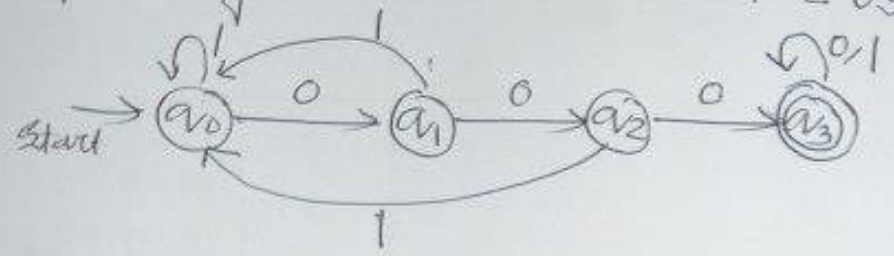
[accepted]

2.2.4 → Design DFA for set of all strings ending in 00.

a)



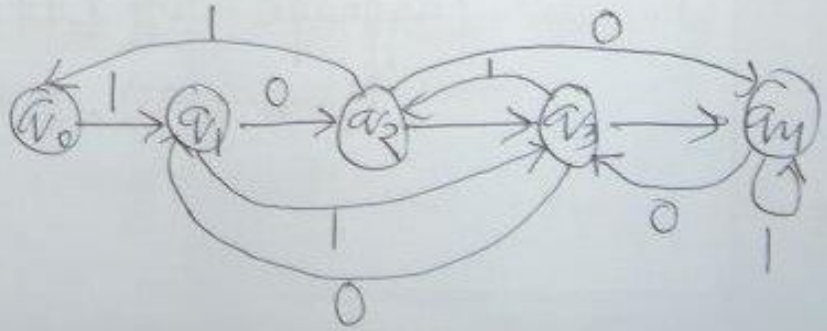
b) Set of strings with three consecutive 0's.



2.2.6 → a) The set of all strings beginning with 1 and is a multiple of 5.

→ If we have any number of form 5a+b, b is the remainder between 0 & 4. then $2(5a+b) = 10+2b$. 10a is surely divisible by 5. Now we can tabulate.

	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



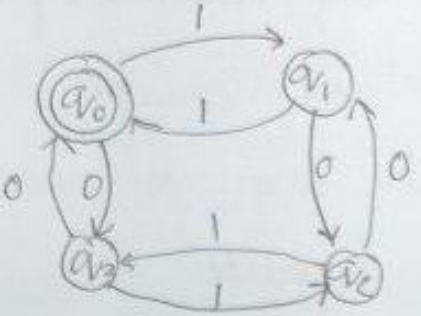
↳ Design a DFA that accept the language.

$L = \{w \mid w$ has both an even number of 0's & even number of 1's$\}$

→ Here, we count the number of 0's or 1's modulo 2. So number of 0's can be even or odd. Number of 1's can also be even or odd. So, there are four states.

$q_0 \rightarrow$ No. of 0's even, No. of 1's even
$q_1 \rightarrow$ No. of 0's even, No. of 1's odd.
$q_2 \rightarrow$ No. of 0's odd, No. of 1's even.
$q_3 \rightarrow$ No. of 0's odd, No. of 1's odd.

So, diagram will be



উপরদিকে 1
ডানেবামে 0.

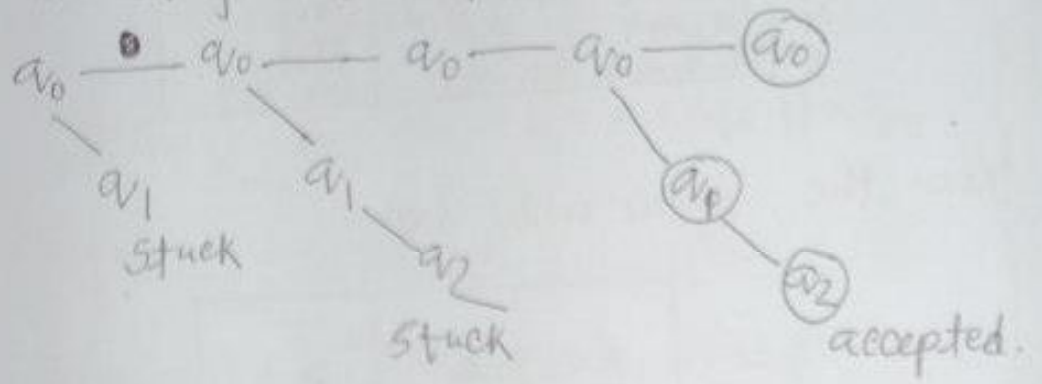
So, the language of the DFA

$= \{(q_0, q_1, q_2, q_3), \{0,1\}, \delta, q_0, \{a_0\}\}$.

Nondeterministic finite automata: On an input and state it returns a set of states (zero, one or more).

. An NFA accepting all strings end in 01.

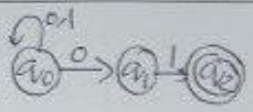
* Check for 00101



Difference between DFA & NFA

DFA	NFA
It has 2^n states.	It has n states.
It is a bit harder to construct	It is easier than a DFA construction.
Transition function is single valued	multivalued transition function
It can't use empty string	Can use empty string.

↳ Draw a DFA diagram from the ~~table~~ diagram



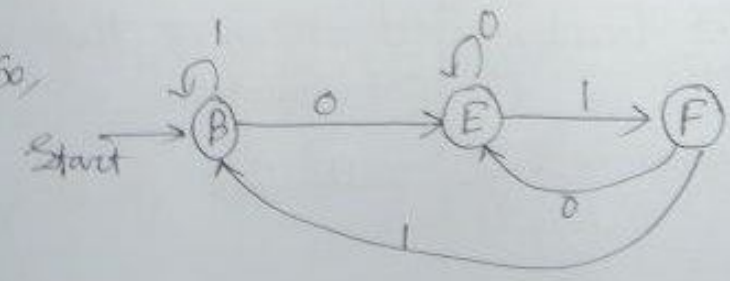
→

		0	1
A	ϕ	ϕ	ϕ
B	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
C	$\{q_1\}$	ϕ	q_2
D	$\{q_2\}$	ϕ	ϕ
E	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
F	$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
G	$\{q_1, q_2\}$	ϕ	$\{q_2\}$
H	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Now the table will be

A	A	A
B	E	B
C	A	D
D	A	A
E	E	F
F	E	B
G	A	A
H	E	F

So,



→ Extended transition function describes what happen if we start in any state and follow any sequence of inputs.

$\hat{\delta}$ → Extended transition function.

For diagram 

Check $w = 11011$.

$$\hat{\delta}(q_0, \epsilon) = q_0$$

$$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \epsilon), 1) = \delta(q_0, 1) = q_0$$

$$\hat{\delta}(q_0, 11) = \delta(\hat{\delta}(q_0, 1), 1) = \delta(q_0, 1) = q_0$$

$$\hat{\delta}(q_0, 110) = \delta(\hat{\delta}(q_0, 11), 0) = \delta(q_0, 0) = q_1$$

$$\hat{\delta}(q_0, 1101) = \delta(\hat{\delta}(q_0, 110), 1) = \delta(q_1, 1) = q_2$$

$$\hat{\delta}(q_0, 11011) = \delta(\hat{\delta}(q_0, 1101), 1) = \delta(q_2, 1) = q_2$$

Accepted

For a table,

	0	1
q_0	q_2	q_1
q_1	q_2	q_0
q_2	q_0	q_3
q_3	q_1	q_2

for the 1000.

Check 11010.

$\hat{\delta}(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \epsilon), 1) = \delta(q_0, 1) = q_1$

$\hat{\delta}(q_0, 11) = \delta(\hat{\delta}(q_0, 1), 1) = \delta(q_1, 1) = q_0$

$\hat{\delta}(q_0, 110) = \delta(\hat{\delta}(q_0, 11), 0) = \delta(q_0, 0) = q_2$

$\hat{\delta}(q_0, 1101) = \delta(\hat{\delta}(q_0 110), 1) = \delta(q_2, 1) = q_3$

$\hat{\delta}(q_0, 11010) = \delta(\hat{\delta}(q_0 1101), 0) = \delta(q_3, 0) = q_1$

$\hat{\delta}(q_0, 11010) = \delta(\hat{\delta}(q_0, 11010), 1) = \delta(q_1, 1) = q_0$ rejected.

Q. What is the language of DFA?

→ Language of DFA, $A = \{Q, \Sigma, \delta, q_0, F\}$ is defined by

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \text{ is in } F\}$$

it is set of strings w that take start state to one of the accepting states.

For the language

Q. Define Context Free Grammars. [7]

→ A CFG consists of
- Finite set of symbols / terminals
- Finite set of variables
- A start symbol.
- A finite set of productions.

So, four components of CFG

$G = \{V, T, P, S\}$

Given.

$E \to I$
$E \to E+E$
$E \to E*E$
$E \to (E)$
$I \to a$
$I \to b$
$I \to Ia$
$I \to Ib$
$I \to I0$
$I \to I1$

Derive $a*(a+b00)$

Leftmost derivation

$E \Rightarrow E*E$
$lm \Rightarrow I*(E+E)$
$lm \Rightarrow a*(I+E)$
$lm \Rightarrow a*(a+I)$
$lm \Rightarrow a*(a+I0)$
$lm \Rightarrow a*(a+I00)$
$lm \Rightarrow a*(a+b00)$

Rightmost derivation

$E \Rightarrow E*E$
$rm \Rightarrow E*(E)$
$rm \Rightarrow E*(E+E)$
$rm \Rightarrow E*(E+I)$
$rm \Rightarrow E*(E+I0)$
$rm \Rightarrow E*(E+I00)$
$rm \Rightarrow E*(E+b00)$
$rm \Rightarrow E*(I+b00)$
$rm \Rightarrow E*(a+b00)$
$rm \Rightarrow I*(a+b00)$
$rm \Rightarrow a*(a+b00)$

What is left most derivation & rightmost derivation?

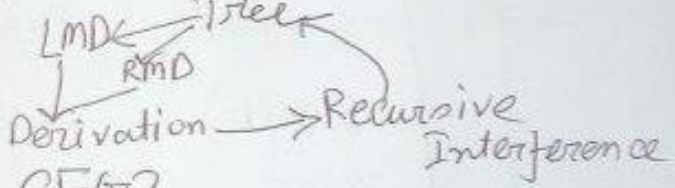
→ LMD: At each step we replace the left most variable by one of it's production bodies.

RMD: At each step we replace the rightmost variable by one of it's production bodies.

↳ What is recursive inferences & derivation?

→ In this recursive inference approach we infer the strings from body to head.

In derivation we approach from head to body.



↳ What is language of CFG?

→ If $G=(V,T,P,S)$ is a CFG, then

language $L(G)=\{w \text{ in } T \mid S \underset{G}{\Rightarrow} w\}$

↳ What is parser?

→ A function that discover structure of program

↳ What is ambigious grammar?

→ When a grammar fails to provide unique structure, it is ambigious grammar. It happens due to
— precedence of operator is neglected.
— A sequence of identical operator can group either from left or from right.

↳ Derive a CFG for prefix expression with x, y operands & +, −, * operators.

→
$E \to x$
$E \to y$
$E \to -EE$
$E \to +EE$
$E \to xEE$

✕ এটি Infix এর জন্য $E \to E+E$, $E*E$ হবে।

✕ Postfix এর জন্য EE+, EE−, EE*

Now, derive $-*+-yxyxy$ using leftmost derivation.

$E \Rightarrow -EE$
$\Rightarrow_{lm} -*EEE$
$\Rightarrow_{lm} -*+EEEE$
$\Rightarrow_{lm} -*+-EEEEE$
$\Rightarrow_{lm} -*+-yEEEE$
$\Rightarrow_{lm} -*+-yxEEE$
$\Rightarrow_{lm} -*+-yxyEE$
$\Rightarrow_{lm} -*+-yxyxE$
$\Rightarrow_{lm} -*+-yxyxy$.

What is factor, term, expression?

→ Factor is an expression that can't be broken apart by any adjacent operator.

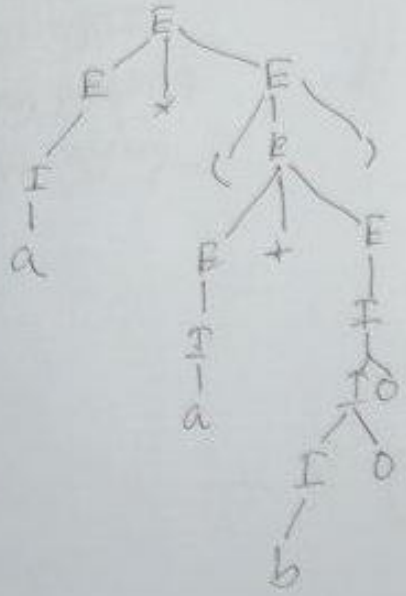
↳ Term can't be broken by + operators

↳ Expression is sum of one or more terms

What is parse tree?

- A graphical representation for a derivation is called parse tree. E.G: Parse tree for

a*(a+b00)



→ Design a CFG for $L = \{0^n 1^n, n>1\}$ set of all strings of one or more zeros followed by an equal number of 1's.

Ans → $S \to 0S1 | S | \varepsilon$

** Design CFG for language consisting of all strings over {a,b} containing an unequal number of a's, b's.

→ $S \to U | V$

$U \to TaU | TaT$

$V \to TbV | TbT$

$T \to aTbT | bTaT | \varepsilon$

→ Now, Using infix expression add following productions

$S \to z$

$S \to y$

$S \to z$

$S \to S+S$

$S \to S-S$

$S \to S*S$

$S \to S/S$

$S \to (S)$

derive $(x+y)*x-z*y/(x+z)$

$S \Rightarrow S-S$

$\Rightarrow S*S-S/S$

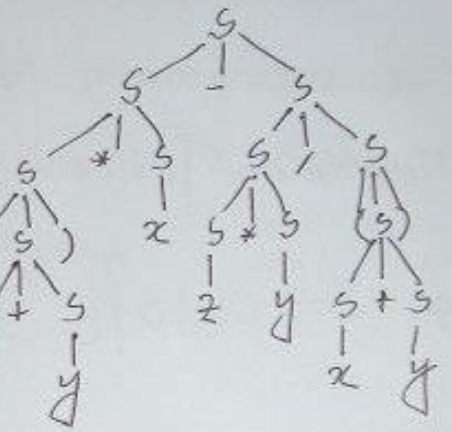
$\Rightarrow (S)*S-S/(S)$

$\Rightarrow (S+S)*S-S/(S+S)$

$\Rightarrow (S+S)*S-S*S/(S+S)$

$\Rightarrow (x+y)*x-z*y/(x+y)$

Parse tree



Given, $S \to ABIC$

$A \to aAb \mid ab$

$B \to cBd \mid cd$

$C \to aCd \mid aDd$

$D \to bDc \mid bc$

Now we have to prove it is ambigious for aabbccdd.

$S \to AB$

$S \to aAbB$

$S \to aabbcBd$

$S \to aabbccdd$

$S \to C$

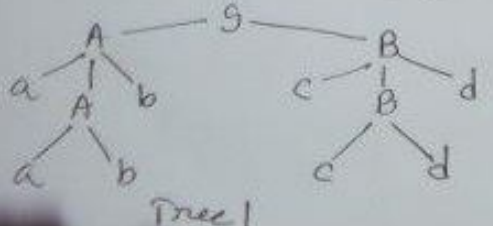
$S \to aCd$

$S \to aaDdd$

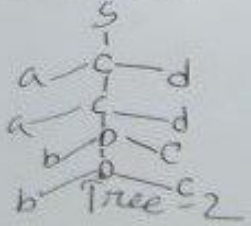
$S \to aabDcdd$

$S \to aabbccdd$

Here two parse tree are produced So it is inherently ambigious.



Tree 1



Tree 2

↳Given a=2, Now ataxa can be 8 or 6 when we precede ③ * then the output will be minimized.

↳What is inherently ambigious?

→ A context free language is said to be inherently ambigious if all its grammars are ambigious. Two or more parse tree will be created for inherently ambigious language.

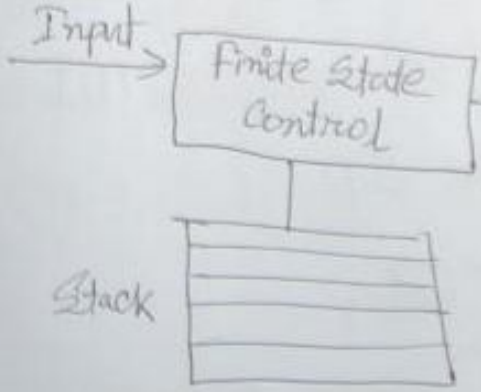
↳ If $\Sigma = \{0,1\}$

what is Σ^3?

→ $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

↳What is Pushdown Automata?

→ A Pushdown automata is $\in$NFA with addition of stack. Stack can be pushed, popped.



There are two versions of PDA. 1) Acceptance by final state Acceptance by 2. empty stack

↳ What is notation of PDA?

→ $P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$

Q: A finite set of states

Σ: A finite set of input symbols

Γ: A finite stack alphabet

δ: Transition function

$\delta(q, a, x)$

state → q, input symbol → a, stack symbol → x

q_0: Start state

Z_0: Start symbol

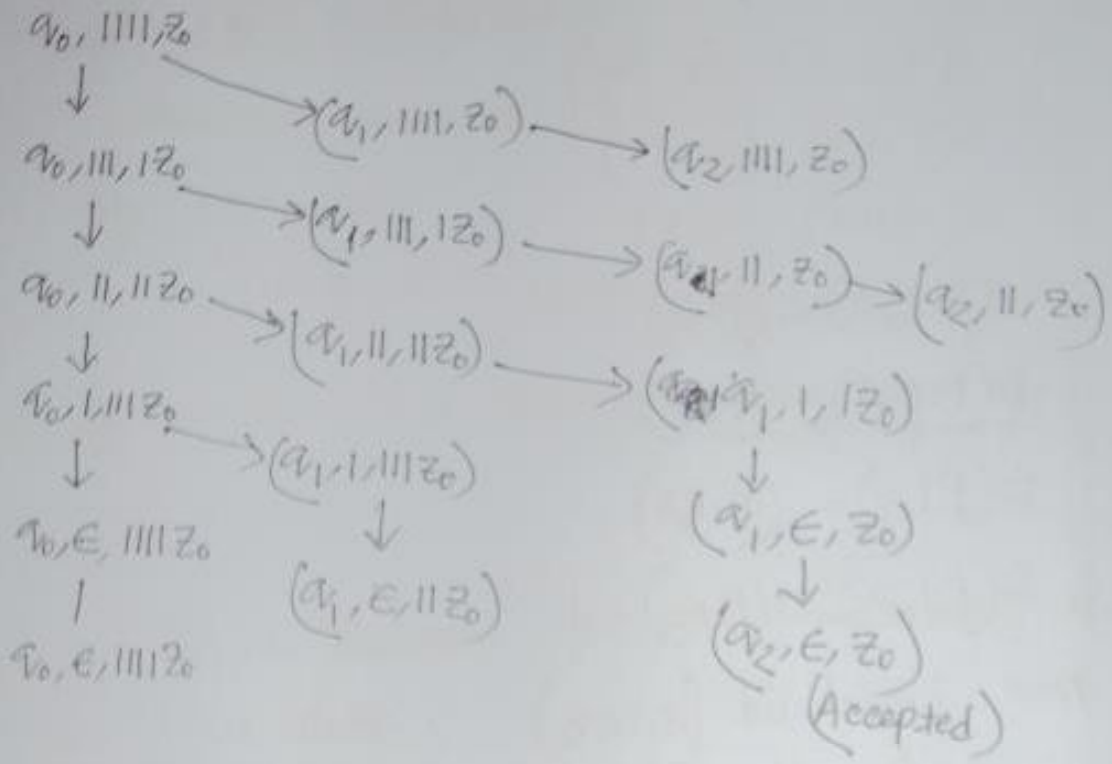
F: Set of accepting states.

↳ Given a PDA $(\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$.

Initial ID (q_0, w, Z_0) & Input is 1111.

Find all reachable ID.

Next page →



Some rules:

When state is q_0 you can't pop. You can only transfer.

$(q_0, 11, 11Z_0) = (q_0, 1, 111Z_0)$

When state is q_1 you can pop but can't transfer elements

$(q_1, 11, 11Z_0) = (q_1, 1, 1Z_0) =$ ✱

↓

Popped

and no input is remaining

When stack is empty, then it will go q_2

$(q_1, \in, Z_0) = (q_2, \in, Z_0)$

It will be accepted.

Given the transition functions

$\delta(q, 0, Z_0) = (q, XZ_0)$

$\delta(q, 0, X) = (q, XX)$

$\delta(q, 1, X) = (q, X)$

$\delta(q, \epsilon, X) = (P, \epsilon)$

$\delta(P, \epsilon, X) = (P, \epsilon)$

$\delta(P, 1, X) = (P, XX)$

$\delta(P, 1, Z_0) = (P, \epsilon)$

Now, initial id (q, w, Z_0)

Find for i) 01 ii) 0011 iii) 010 all reachable ID.

i) $(q, 01, Z_0) \vdash (q, 1, XZ_0) \vdash (q, \epsilon, XZ_0) \vdash (P, \epsilon, Z_0)$
$\vdash (P, \epsilon, \epsilon)$ (accepted)

ii) $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (q, 11, XXZ_0) \vdash$
$(q, 1, XXZ_0) \vdash (q, \epsilon, XXZ_0) \vdash (P, \epsilon XZ_0) \vdash (P, \epsilon, Z_0)$
$\vdash (P, \epsilon, \epsilon)$ (accepted)

iii) $(q, 010, Z_0) \vdash (q, 10, XZ_0) \vdash (q, 0, XZ_0) \vdash (q, \epsilon, XXZ_0)$
$\vdash (P, \epsilon, XZ_0) \vdash (P, \epsilon, Z_0) \vdash (P, \epsilon, \epsilon)$ (accepted)

What is language of PDA?

↳ Let $P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$

then $\{w \mid (q_0, w, Z_0) \vdash (q, \epsilon, \alpha)\}$ is acceptance by final state.

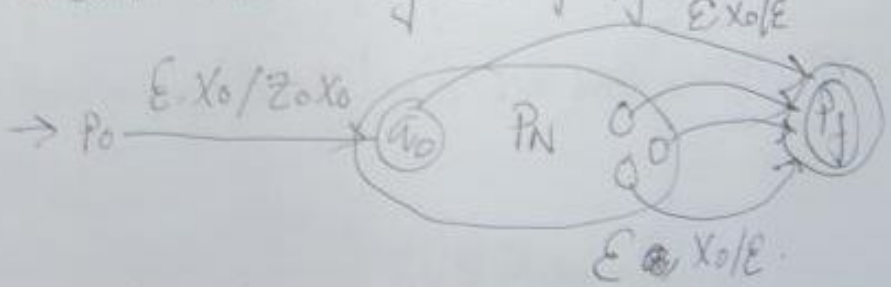
$\{w \mid (q_0, w, Z_0) \vdash (q, \epsilon, \epsilon)\}$ is acceptance by empty stack. These are language of PDA.

↳ Write a theorem for PDA from empty stack to final state.

→ If $L = N(P_N)$ for some PDA. $P_N = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ Then there is a PDA P_F such that

$$L = L(P_F).$$

Here, we use a symbol X_0 which must not be a symbol of Γ. X_0 is a marker of the stack and start symbol of P_F.

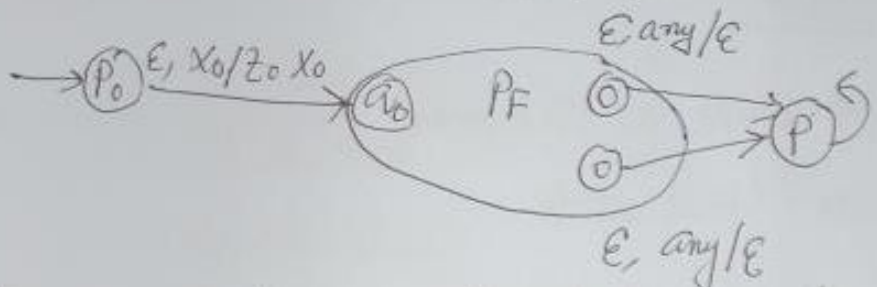


Here, a new state P_0 push Z_0 on the top of stack and enter state q_0.

$\therefore P_F = \{Q \cup \{P_0, P_F\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, P_0, X_0, \{P_F\}\}$.

↳ Write theorem of PDA from final state to empty stack.

Here Let L_e be $L(P_F)$ for some PDA $P_F = \{Q, \Sigma, \Gamma, \delta_F, q_0, z_0, F\}$. then there is a PDA P_N such that $L = N(P_N)$



we use X_0, a marker on the bottom of stack. It is P_N's start symbol. state P_0 is used to push z_0 in stack.

$\therefore P_N = \{Q \cup \{P_0, P_0\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, P_0, X_0\}$.

↳ Design a PDA using if else.

→ Here, $\delta_N(q, i, z) = (q, zz)$, pushes z when we see if

$\delta_N(q, e, z) = (q, \varepsilon)$, pops z when we see else.

Thus when (else) > (if +1) stack is emptied and input string is accepted.

eg. $\delta(q, iee, z) \vdash \delta(q, ee, zz) \vdash (q, e, z) \vdash (q, \varepsilon\varepsilon)$
[accepted]

↳ Define Turing machine.

→ There are some problems which are undecideable. An abstract computing machine can solve. It is called turing machine.

It consists of
- A finite control
- A tape divided into cell.
- Read/write head.

↳ What's the notation of turing machine?

→ $M = (S, \Sigma, T, \delta, q_0, B, F)$

S = Finite set of states

Σ = A finite set of input symbols

T = tape symbols.

δ = Transition Function

q_0 = Start state

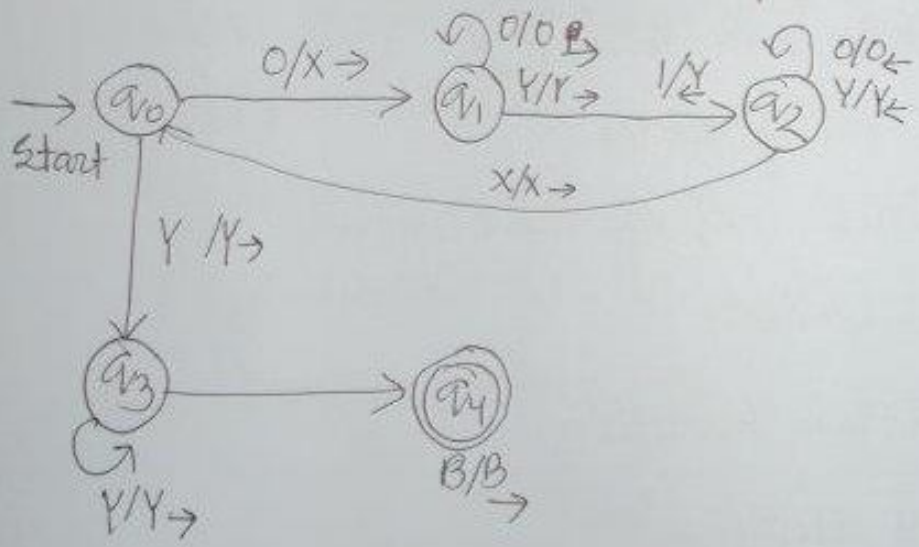
B = Blank symbol

F = Set of accepting states.

↳ Design a Turing machine for the language.

$L(m) = \{0^n 1^n \mid n \geq 1\}$

State	0	1	X	Y	B
q_0	(q_1, X, R)			(q_3, Y, R)	
q_1	$(q_1, 0, R)$	(q_2, Y, L)		(q_1, Y, R)	
q_2	$(q_2, 0, L)$		(q_0, X, R)	(q_2, Y, L)	
q_3				(q_3, Y, R)	(q_4, B, R)
q_4					



For the language we are checking

000111

$q_0 000111 \vdash Xq_1 00111 \vdash X0q_1 0111 \vdash X00q_1 111 \vdash X0q_2 011$
$\vdash Xq_2 00Y11 \vdash q_2 X00Y11 \vdash Xq_0 00Y11 \vdash XXq_1 0Y11 \vdash$
$XX0q_1 Y11 \vdash XX0Yq_1 11 \vdash XX0q_2 YY1 \vdash XXq_2 0YY1 \vdash$
$Xq_2 X0YY1 \vdash XXq_0 0YY1 \vdash XXXq_1 YY1 \vdash XXXYq_1 Y1 \vdash$
$XXXYYq_1 1 \vdash XXXYq_2 YY \vdash XXXq_2 YYY \vdash XXq_2 XYYY$
$\vdash XXXq_0 YYY \vdash XXXYq_3 YY \vdash XXXYYq_3 Y \vdash$
$XXXYYYq_3 B \vdash XXXYYYBq_4 B$

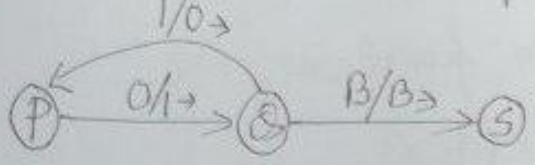
→ Consider the TM = $(\{P, S, R\}, \{0,1\}, \{0,1, B\}, \delta, P, B, \{S\})$

$\delta(P,0) = (B, 1, R)$

$\delta(B,1) = (P, 0, R)$

$\delta(B,B) = (S, B, R)$

Design transition table & diagram



(R)

(R has no connection)

	0	1	B
P	(S, 1, R)		
S		(P, 0, R)	(S, B, R)
R			
S			

↳ Convert the grammar.

$S \to 0S1 \mid A$

$A \to 1A0 \mid S \mid \varepsilon$ to a PDA that accepts

Now, the PDA will be = $\{\{q\}, \{0, 1\}, \{0, 1, A, S\}, \delta, q, S\}$

where transition function

$\delta(q, \varepsilon, S) = \{(q, 0S), (q, A)\}$

$\delta(q, \varepsilon, A) = \{(q, 1A0), (q, S), (q, \varepsilon)\}$

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

$\delta(q, 1, 1) = \{(q, \varepsilon)\}$

↳ Convert the grammar

$S \to aAA$

$A \to aS \mid bS \mid a$

to PDA that accepts same language by empty stack.

↳ What is chomsky normal form?

→ Every non empty CFL without ϵ has grammar G in which all production are in one of the following forms.

$$A \to BC$$

$$A \to a$$

where A is variable & a is terminal.

* We must eliminate useless symbols.
* We must eliminate ϵ productions. $(A \to \epsilon)$
* We must eliminate unit productions. $(A \to B)$

↳ Eliminate unit productions & derive it in chomsky form. for following example.

$$I \to a|b|Ia|Ib|I0|I1$$

$$F \to I|(E)$$

$$T \to F|T*F$$

$$E \to T|E+T$$

Now, we consider the unit pairs (E,E), (T,T), (F,F), (I,I)

(E,E) & the production E→T gives unit pair (E,T)
(E,T) & the production T→F gives unit pair (E,F)
(E,F) & the production F→I gives unit pair (E,I)
~~(E,I) & the production~~
(I,T) & the production T→F gives unit pair (I,F)
(I,F) & the production F→I gives Unit pair (I,I)
(F,F) & the production F→I gives unit pair (F,I)

Now, the resulting grammar:

Pair	Productions
(E,E)	E→ E+T
(E,T)	E→ T*F
(F,F)	E→ (E)
(E,I)	E→ a|b|Ia|Ib|I0|I1
(I,T)	T→ T*F
(I,F)	T→ (E)
(I,I)	T→ a|b|Ia+Ib|I0|I1
(F,F)	f→ (E)
(F,I)	f→ a|b|Ia|Ib|I0|I1
(I,I)	I→ a|b|Ia|Ib|I0|I1

Now the grammar constructed of the unit production elimination.

$E \to E+T \mid T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$T \to T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$F \to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Now for chomsky Normal form. Let the terminals defined by variables.

$A \to a,\ B \to b,\ Z \to 0,\ 0 \to 1,\ P \to +,\ M \to *,\ L \to (,$

$R \to)$

∴ The grammar will be following

$E \to EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$T \to TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$F \to LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$I \to a \mid b \mid IA \mid IB \mid IZ \mid IO$

$A \to a$

$B \to b$

$Z \to 0$

$0 \to 1$

$P \to +$

$M \to *$

$L \to ($

$R \to)$

But there are bodies of length 3. EPT. TMF, LER.

Let, $PT \to C_1$, $MF \to C_2$, $ER \to C_3$.

$\therefore E \to EC_1 | TC_2 | LC_3 | a | b | IA | IB | IZ | IO$

$T \to TC_2 | LC_3 | a | b | IA | IB | IZ | IO$

$F \to LC_3 | a | b | IA | IB | IZ | IO$

$A \to a$

$B \to b$

$Z \to 0$

$O \to 1$

$P \to +$

$M \to *$

$L \to ($

$R \to)$

$C_1 \to PT$

$C_2 \to MF$

$C_3 \to ER$

It is Chomsky form of the grammar

Find grammar for

$S \to AB|CA$

$A \to a$

$B \to BC|AB$

$C \to aB|b$

Here, B has no terminal, so, B is useless

So, it will be

$S \to CA$

$A \to a$

$C \to b$

↳ What are the conditions & rules to convert from grammar to PDA?

→ $PDA = (Q, \Sigma, V_n \cup T, S, q, \delta)$

where $\delta(q, \in A) = \{(q, B) | A \to B$ is a production of $G\}$

$\delta(q, aA) = \{(q, A)\}$

Now, Convert the expression to PDA

$I \to a|b|Ia|Ib|I0|I1$

$E \to I | E \times E | E + E | (E)$

a) $\delta(q, \in, I) = \{(q,a), (q,b), (q,Ia), (q,Ib), (q,I0), (q,I1)\}$

b) $\delta(q, \in, E) = \{(q,I), (q, E+E), (q, E*E), (q, (E))\}$

c) $\delta(q,a,a) = \{(q, \epsilon)\}$

$\delta(q,b,b) = \{(q, \epsilon)\}$

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

$\delta(q,1,1) = \{(q, \epsilon)\}$

$\delta(q,(,() = \{(q, \epsilon)\}$

$\delta(q,),)) = \{(q, \epsilon)\}$

$\delta(q,+,+) = \{(q, \epsilon)\}$

$\delta(q,*,*) = \{(q, \epsilon)\}$

↳ Theorem for PDA to grammar:

If $P = (S, \Sigma, T, \delta, q_0, Z_0)$ be a PDA - then there is a Context free grammar $L(G) = N(P)$

Proof: we shall construct, $G = \{V, \Sigma, R, S\}$

where variable V consists

1) Special symbol S, start symbol.

2) All symbols of the form $[pXq]$ where p & q are states in Q and X is a stack symbol in Γ.

Now productions of G are

a) for all states p, G has production

$S \rightarrow [q_0 Z_0 p]$

b) Let $\delta(q, a, X)$ contain the pair $(r, Y_1, Y_2, \ldots Y_k)$

1) where a is a symbol in Σ or $a = \epsilon$.

2) k can be any number. If $k = 0$, pair (r, ϵ).

then for list of states $r_1, r_2, \ldots, r_k$.

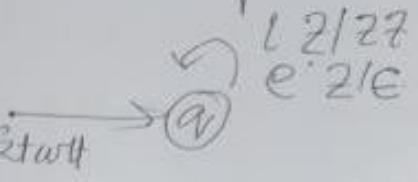
G has production

$[qX r_k] \rightarrow a[rY_1 r_1][r_1 Y_2 r_2] \ldots [r_{k-1} Y_k r_k]$

↳ Given, PDA, PN = ({q}, {i, e}, {z}, δ_N, q, z)

We have to convert grammar.

→ PN has only one state & stack symbol.



1) Production for S is $S \to [qzq]$

2) δ_N (q, i, z) contains (q, zz).

So, $[qzq] \to i\,[qzq][qzq]$

3) δ_N (q, e, z) contains (q, ε) so,

$[qzq] \to e$.

If we replace [qzq] by a simple symbol A. then,

$S \to A$

$A \to iAA$

$A \to e$.

It may be simplified $S \to iSS$

$S \to e$

∴ Grammar, G = ({S}, {i, e}, {$S \to iSS \mid eS$}).

Undecidebility code:

Our goal is to devise code for TM. with alphabet {0,1}

Now, $X_1 = 0$

$X_2 = 1$

$X_3 = B$

we shall refer L as D_1 & R as D_2

Now $\delta(q_i, x_j) = (q_k, X_l, D_m)$

Now code for string $0^i 10^j 10^k 10^l 10^m$

So code is

$C_1 11 C_2 11 C_3 11 C_4 11 \ldots 11 C_n$

e.g: Let $M = (\{q_1, q_2, q_3\}, (0,1), (0,1,B), \delta, q_1, B, \{q_2\})$

1) $\delta(q_1, 1) = (q_3, 0, R) \Rightarrow \delta(q_1, X_2) = (q_3, X_1, D_2)$

2) $\delta(q_3, 0) = (q_1, 1, R) \Rightarrow \delta(q_3, X_1) = (q_1, X_2, D_2)$

3) $\delta(q_3, 1) = (q_2, 0, R) \Rightarrow \delta(q_3, X_2) = (q_2, X_1, D_2)$

4) $\delta(q_3, B) = (q_3, 1, L) \Rightarrow \delta(q_3, X_3) = (q_3, X_2, D_1)$

Code for 1, $C_1 = 0^1 10^2 10^3 10^1 10^2 = 0100010001001$

$C_2 = 0^3 10^1 10^1 10^2 10^2 = 0001010010010$

$C_3 = 0^3 10^2 10^2 10^1 10^2 = 0001001001010$

$c_4 = 0^3 10^3 10^3 10^2 10^1 = 000100010001001 0$

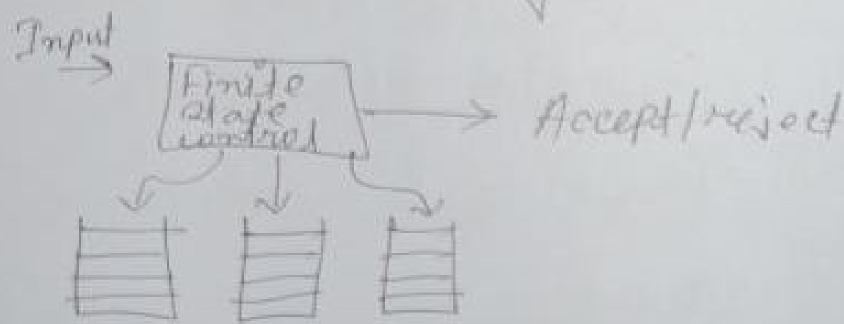
∴ Final code is

0100100010100110001010100100110001001001010011
000100010001010

↳ what is multstack machine & counter machine?

→ Multistack machine: A MS machine has a finite control, which is in one of a finite set of states. It has stack alphabet used for all stacks. Move of MS machine is based on

- The state of finite control
- The input symbol read
- The top stack symbol on each of its stacks



Counter machine: Counter machine has same structure as multistack machine but in place of stack, there is a counter.

Counters hold any nonnegative number.
But we can distinguish between zero & nonzero counters in a move. counter machine

1) Change state
2) Add or substract 1 from counters.

But it will not substract from 0 to avoid the negative result.

↳ What is subroutine?

→ A turing machine subroutine is a set of states that perform some useful procces. The call of subroutine occurs whenever there is a transition to initial state. Set of states include start & ~~final~~ another state that have no moves.

↳ Simulating TM by computer
Simulating computer by TM → (Self)

↳ What is mark up language?

→ Strings in such language are documents with certain marks (tags) in them.

Eg: HTML

↳ Define Grammar, Alphabet, String, language.

Grammar: is processing data with recursive structure.

Alphabet: A set of symbols. $\Sigma = \{0,1\}$

String: A sequence of symbol from an alphabet. 0110

Language: A set of strings from alphabet.

↳ What is power of alphabet?

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

then $\Sigma^0, \Sigma^1, \Sigma^2 \ldots \Sigma^n$ are power of alphabet.

$\Sigma^R = \{00, 01, 10, 11\}$

Given the transition functions

$\delta(q, 0, Z_0) = (q, XZ_0)$

$\delta(q, 0, X) = (q, XX)$

$\delta(q, 1, X) = (q, X)$

$\delta(q, \epsilon, X) = (P, \epsilon)$

$\delta(P, \epsilon, X) = (P, \epsilon)$

$\delta(P, 1, X) = (P, XX)$

$\delta(P, 1, Z_0) = (P, \epsilon)$

Now, initial id (q, w, Z_0)

Find for i) 01 ii) 0011 iii) 010 all reachable ID.

i) $(q, 01, Z_0) \vdash (q, 1, XZ_0) \vdash (q, \epsilon, XZ_0) \vdash (P, \epsilon, Z_0)$
$\vdash (P, \epsilon, \epsilon)$ (accepted)

ii) $(q, 0011, Z_0) \vdash (q, 011, XZ_0) \vdash (q, 11, XXZ_0) \vdash$
$(q, 1, XXZ_0) \vdash (q, \epsilon, XXZ_0) \vdash (P, \epsilon XZ_0) \vdash (P, \epsilon, Z_0)$
$\vdash (P, \epsilon, \epsilon)$ (accepted)

iii) $(q, 010, Z_0) \vdash (q, 10, XZ_0) \vdash (q, 0, XZ_0) \vdash (q, \epsilon, XXZ_0)$
$\vdash (P, \epsilon, XZ_0) \vdash (P, \epsilon, Z_0) \vdash (P, \epsilon, \epsilon)$ (accepted)

What is language of PDA?

↳ Let $P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$

then $\{w \mid (q_0, w, Z_0) \vdash (q, \epsilon, \alpha)\}$ is acceptance by final state.

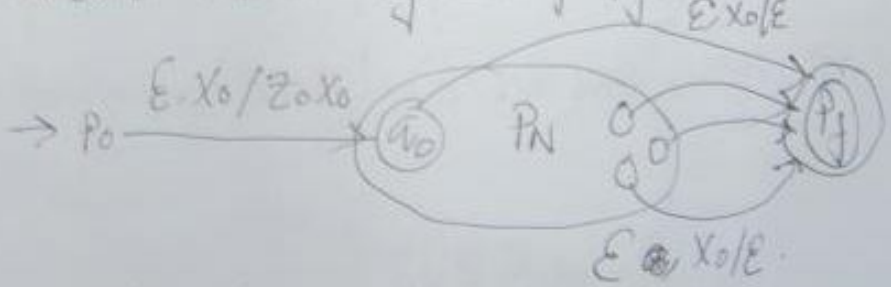
$\{w \mid (q_0, w, Z_0) \vdash (q, \epsilon, \epsilon)\}$ is acceptance by empty stack. These are language of PDA.

↳ Write a theorem for PDA from empty stack to final state.

→ If $L = N(P_N)$ for some PDA. $P_N = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ Then there is a PDA P_F such that

$$L = L(P_F).$$

Here, we use a symbol X_0 which must not be a symbol of Γ. X_0 is a marker of the stack and start symbol of P_F.

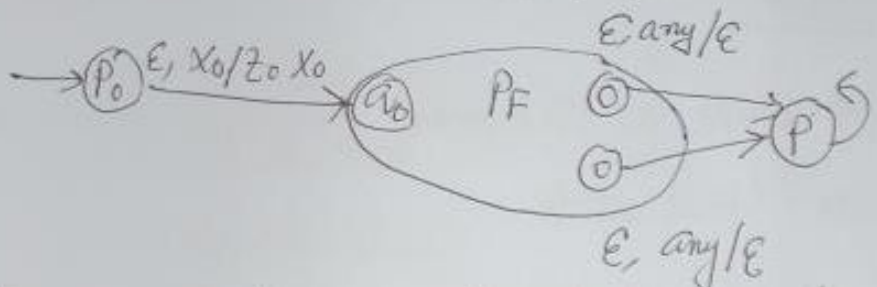


Here, a new state P_0 push Z_0 on the top of stack and enter state q_0.

$\therefore P_F = \{Q \cup \{P_0, P_F\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, P_0, X_0, \{P_F\}\}$.

↳ Write theorem of PDA from final state to empty stack.

Here Let L_e be $L(P_F)$ for some PDA $P_F = \{Q, \Sigma, \Gamma, \delta_F, q_0, z_0, F\}$. then there is a PDA P_N such that $L = N(P_N)$



we use X_0, a marker on the bottom of stack. It is P_N's start symbol. state P_0 is used to push z_0 in stack.

$\therefore P_N = \{Q \cup \{P_0, P_0\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, P_0, X_0\}$.

↳ Design a PDA using if else.

→ Here, $\delta_N(q, i, z) = (q, zz)$, pushes z when we see if

$\delta_N(q, e, z) = (q, \varepsilon)$, pops z when we see else.

Thus when (else) > (if +1) stack is emptied and input string is accepted.

eg. $\delta(q, iee, z) \vdash \delta(q, ee, zz) \vdash (q, e, z) \vdash (q, \varepsilon\varepsilon)$
[accepted]

↳ Define Turing machine.

→ There are some problems which are undecideable. An abstract computing machine can solve. It is called turing machine.

It consists of
- A finite control
- A tape divided into cell.
- Read/write head.

↳ What's the notation of turing machine?

→ $M = (S, \Sigma, T, \delta, q_0, B, F)$

S = Finite set of states

Σ = A finite set of input symbols

T = tape symbols.

δ = Transition Function

q_0 = Start state

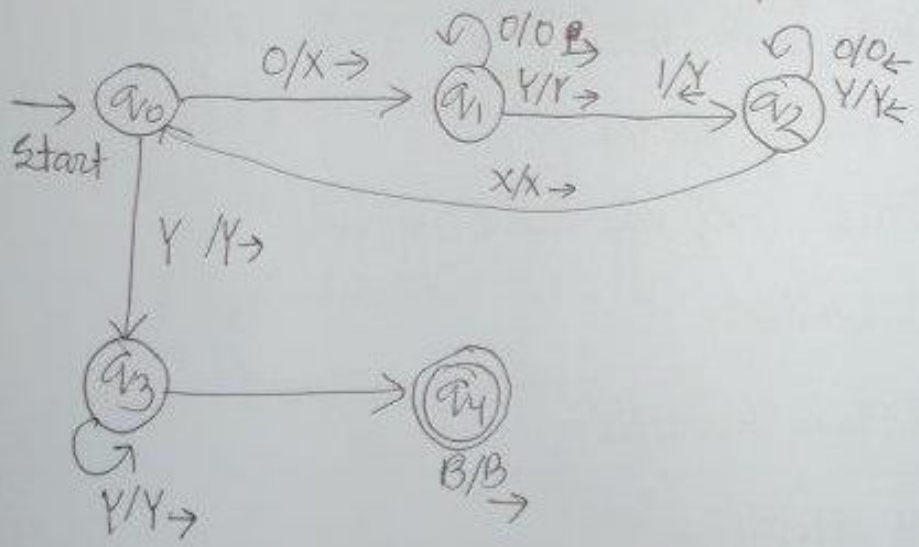
B = Blank symbol

F = Set of accepting states.

↳ Design a Turing machine for the language.

$L(m) = \{0^n 1^n \mid n \geq 1\}$

State	0	1	X	Y	B
q_0	(q_1, X, R)			(q_3, Y, R)	
q_1	$(q_1, 0, R)$	(q_2, Y, L)		(q_1, Y, R)	
q_2	$(q_2, 0, L)$		(q_0, X, R)	(q_2, Y, L)	
q_3				(q_3, Y, R)	(q_4, B, R)
q_4					



For the language we are checking

000111

$q_0 000111 \vdash Xq_1 00111 \vdash X0q_1 0111 \vdash X00q_1 111 \vdash X0q_2 011$
$\vdash Xq_2 00Y11 \vdash q_2 X00Y11 \vdash Xq_0 00Y11 \vdash XXq_1 0Y11 \vdash$
$XX0q_1 Y11 \vdash XX0Yq_1 11 \vdash XX0q_2 YY1 \vdash XXq_2 0YY1 \vdash$
$Xq_2 X0YY1 \vdash XXq_0 0YY1 \vdash XXXq_1 YY1 \vdash XXXYq_1 Y1 \vdash$
$XXXYYq_1 1 \vdash XXXYq_2 YY \vdash XXXq_2 YYY \vdash XXq_2 XYYY$
$\vdash XXXq_0 YYY \vdash XXXYq_3 YY \vdash XXXYYq_3 Y \vdash$
$XXXYYYq_3 B \vdash XXXYYYBq_4 B$

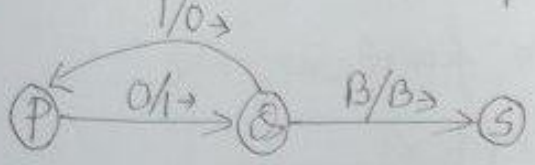
→ Consider the TM = $(\{P, S, R\}, \{0,1\}, \{0,1, B\}, \delta, P, B, \{S\})$

$\delta(P,0) = (B, 1, R)$

$\delta(B,1) = (P, 0, R)$

$\delta(B,B) = (S, B, R)$

Design transition table & diagram



(R)

(R has no connection)

	0	1	B
P	(S, 1, R)		
S		(P, 0, R)	(S, B, R)
R			
S			

↳ Convert the grammar.

$S \to 0S1 \mid A$

$A \to 1A0 \mid S \mid \varepsilon$ to a PDA that accepts

Now, the PDA will be = $\{\{q\}, \{0, 1\}, \{0, 1, A, S\}, \delta, q, S\}$

where transition function

$\delta(q, \varepsilon, S) = \{(q, 0S), (q, A)\}$

$\delta(q, \varepsilon, A) = \{(q, 1A0), (q, S), (q, \varepsilon)\}$

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

$\delta(q, 1, 1) = \{(q, \varepsilon)\}$

↳ Convert the grammar

$S \to aAA$

$A \to aS \mid bS \mid a$

to PDA that accepts same language by empty stack.

↳ What is chomsky normal form?

→ Every non empty CFL without ϵ has grammar G in which all production are in one of the following forms.

$$A \to BC$$

$$A \to a$$

where A is variable & a is terminal.

* We must eliminate useless symbols.
* We must eliminate ϵ productions. $(A \to \epsilon)$
* We must eliminate unit productions. $(A \to B)$

↳ Eliminate unit productions & derive it in chomsky form. for following example.

$$I \to a|b|Ia|Ib|I0|I1$$

$$F \to I|(E)$$

$$T \to F|T*F$$

$$E \to T|E+T$$

Now, we consider the unit pairs (E,E), (T,T), (F,F), (I,I)

(E,E) & the production E→T gives unit pair (E,T)
(E,T) & the production T→F gives unit pair (E,F)
(E,F) & the production F→I gives unit pair (E,I)
~~(E,I) & the production~~
(I,T) & the production T→F gives unit pair (I,F)
(I,F) & the production F→I gives Unit pair (I,I)
(F,F) & the production F→I gives unit pair (F,I)

Now, the resulting grammar:

Pair	Productions
(E,E)	E→ E+T
(E,T)	E→ T*F
(F,F)	E→ (E)
(E,I)	E→ a|b|Ia|Ib|I0|I1
(I,T)	T→ T*F
(I,F)	T→ (E)
(I,I)	T→ a|b|Ia+Ib|I0|I1
(F,F)	f→ (E)
(F,I)	f→ a|b|Ia|Ib|I0|I1
(I,I)	I→ a|b|Ia|Ib|I0|I1

Now the grammar constructed of the unit production elimination.

$E \to E+T \mid T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$T \to T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$F \to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Now for chomsky Normal form. Let the terminals defined by variables.

$A \to a,\ B \to b,\ Z \to 0,\ 0 \to 1,\ P \to +,\ M \to *,\ L \to (,$

$R \to)$

∴ The grammar will be following

$E \to EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$T \to TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$F \to LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$I \to a \mid b \mid IA \mid IB \mid IZ \mid IO$

$A \to a$

$B \to b$

$Z \to 0$

$0 \to 1$

$P \to +$

$M \to *$

$L \to ($

$R \to)$

But there are bodies of length 3. EPT. TMF, LER.

Let, $PT \to C_1$, $MF \to C_2$, $ER \to C_3$.

$\therefore E \to EC_1 | TC_2 | LC_3 | a | b | IA | IB | IZ | IO$

$T \to TC_2 | LC_3 | a | b | IA | IB | IZ | IO$

$F \to LC_3 | a | b | IA | IB | IZ | IO$

$A \to a$

$B \to b$

$Z \to 0$

$O \to 1$

$P \to +$

$M \to *$

$L \to ($

$R \to)$

$C_1 \to PT$

$C_2 \to MF$

$C_3 \to ER$

It is Chomsky form of the grammar

Find grammar for

$S \to AB|CA$

$A \to a$

$B \to BC|AB$

$C \to aB|b$

Here, B has no terminal, so, B is useless

So, it will be

$S \to CA$

$A \to a$

$C \to b$

↳ What are the conditions & rules to convert from grammar to PDA?

→ $PDA = (Q, \Sigma, V_n \cup T, S, q, \delta)$

where $\delta(q, \in A) = \{(q, B) | A \to B$ is a production of $G\}$

$\delta(q, aA) = \{(q, A)\}$

Now, Convert the expression to PDA

$I \to a|b|Ia|Ib|I0|I1$

$E \to I | E \times E | E + E | (E)$

a) $\delta(q, \in, I) = \{(q,a), (q,b), (q,Ia), (q,Ib), (q,I0), (q,I1)\}$

b) $\delta(q, \in, E) = \{(q,I), (q, E+E), (q, E*E), (q, (E))\}$

c) $\delta(q,a,a) = \{(q, \epsilon)\}$

$\delta(q,b,b) = \{(q, \epsilon)\}$

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

$\delta(q,1,1) = \{(q, \epsilon)\}$

$\delta(q,(,() = \{(q, \epsilon)\}$

$\delta(q,),)) = \{(q, \epsilon)\}$

$\delta(q,+,+) = \{(q, \epsilon)\}$

$\delta(q,*,*) = \{(q, \epsilon)\}$

↳ Theorem for PDA to grammar:

If $P = (S, \Sigma, T, \delta, q_0, Z_0)$ be a PDA - then there is a Context free grammar $L(G) = N(P)$

Proof: we shall construct, $G = \{V, \Sigma, R, S\}$

where variable V consists

1) Special symbol S, start symbol.

2) All symbols of the form $[pXq]$ where p & q are states in Q and X is a stack symbol in Γ.

Now productions of G are

a) for all states p, G has production

$S \rightarrow [q_0 Z_0 p]$

b) Let $\delta(q, a, X)$ contain the pair $(r, Y_1, Y_2, \ldots Y_k)$

1) where a is a symbol in Σ or $a = \epsilon$.

2) k can be any number. If $k = 0$, pair (r, ϵ).

then for list of states $r_1, r_2, \ldots, r_k$.

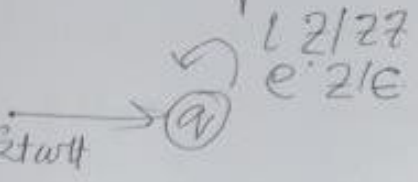
G has production

$[qX r_k] \rightarrow a[rY_1 r_1][r_1 Y_2 r_2] \ldots [r_{k-1} Y_k r_k]$

↳ Given, PDA, PN = ({q}, {i, e}, {z}, δ_N, q, z)

We have to convert grammar.

→ PN has only one state & stack symbol.



1) Production for S is $S \to [qzq]$

2) δ_N (q, i, z) contains (q, zz).

So, $[qzq] \to i\,[qzq][qzq]$

3) δ_N (q, e, z) contains (q, ε) so,

$[qzq] \to e$.

If we replace [qzq] by a simple symbol A. then,

$S \to A$

$A \to iAA$

$A \to e$.

It may be simplified $S \to iSS$

$S \to e$

∴ Grammar, G = ({S}, {i, e}, {$S \to iSS \mid eS$}).

Undecidebility code:

Our goal is to devise code for TM. with alphabet {0,1}

Now, $X_1 = 0$

$X_2 = 1$

$X_3 = B$

we shall refer L as D_1 & R as D_2

Now $\delta(q_i, x_j) = (q_k, X_l, D_m)$

Now code for string $0^i 10^j 10^k 10^l 10^m$

So code is

$C_1 11 C_2 11 C_3 11 C_4 11 \ldots 11 C_n$

e.g: Let $M = (\{q_1, q_2, q_3\}, (0,1), (0,1,B), \delta, q_1, B, \{q_2\})$

1) $\delta(q_1, 1) = (q_3, 0, R) \Rightarrow \delta(q_1, X_2) = (q_3, X_1, D_2)$

2) $\delta(q_3, 0) = (q_1, 1, R) \Rightarrow \delta(q_3, X_1) = (q_1, X_2, D_2)$

3) $\delta(q_3, 1) = (q_2, 0, R) \Rightarrow \delta(q_3, X_2) = (q_2, X_1, D_2)$

4) $\delta(q_3, B) = (q_3, 1, L) \Rightarrow \delta(q_3, X_3) = (q_3, X_2, D_1)$

Code for 1, $C_1 = 0^1 10^2 10^3 10^1 10^2 = 0100010001001$

$C_2 = 0^3 10^1 10^1 10^2 10^2 = 0001010010010$

$C_3 = 0^3 10^2 10^2 10^1 10^2 = 0001001001010$

$C_4 = 0^3 1 0^3 1 0^3 1 0^2 1 0^1 = 0001000100010010$

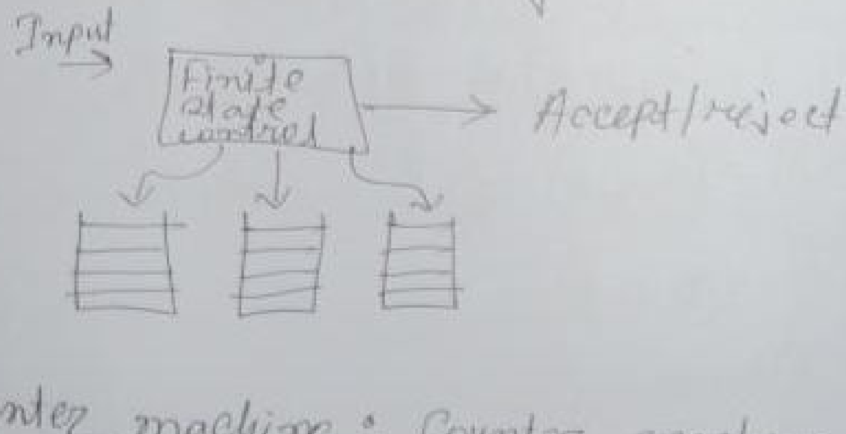
∴ Final code is

0100100010100110001010100100110001001001010011
000100010001010

↳ What is multistack machine & counter machine?

→ <u>Multistack machine</u>: A MS machine has a finite control, which is in one of a finite set of states. It has stack alphabet used for all stacks. Move of MS machine is based on

- The state of finite control
- The input symbol read
- The top stack symbol on each of its stacks



<u>Counter machine</u>: Counter machine has same structure as multistack machine but in place of stack, there is a counter.

Counters hold any nonnegative number.
But we can distinguish between zero & nonzero counters in a move. counter machine

1) Change state
2) Add or substract 1 from counters.

But it will not substract from 0 to avoid the negative result.

↳ What is subroutine?

→ A turing machine subroutine is a set of states that perform some useful procces. The call of subroutine occurs whenever there is a transition to initial state. Set of states include start & ~~final~~ another state that have no moves.

↳ Simulating TM by computer
Simulating computer by TM → (Self)

↳ What is mark up language?

→ Strings in such language are documents with certain marks (tags) in them.

Eg: HTML

↳ Define Grammar, Alphabet, String, language.

Grammar: is processing data with recursive structure.

Alphabet: A set of symbols. $\Sigma = \{0,1\}$

String: A sequence of symbol from an alphabet. 0110

Language: A set of strings from alphabet.

↳ What is power of alphabet?

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

then $\Sigma^0, \Sigma^1, \Sigma^2 \ldots \Sigma^n$ are power of alphabet.

$\Sigma^R = \{00, 01, 10, 11\}$