UNIT-4

PUSH DOWN AUTOMATA

Course Outcome: Design of PDA and equivalence with its CFG

Topics:
1. Definition of PDA
2. Instantaneous discription of PDA
3. context free languages
4. Equivalance between PDA and CFG
5. Deterministic push down Automata
6. Regular languages and DPDA
7. Context free languages and DPDA

Defination of PDA:

We can define the PDA with seven touple representation as shown below.

FA – 5 touple ⟶ $(\Sigma, q_0, Q, F, \delta)$

PDA – 7 touple ⟶ $(\underbrace{\Sigma, q_0, Q, F, \delta}_{\text{FA}}, \underbrace{\Gamma, Z_0}_{\text{stack}})$

PDA = FA + 1 stack
(5) (2)

where

Σ = input alphabet

q_0 = initial state

Q = set of all states

F = set of all final states, $F \subseteq Q$

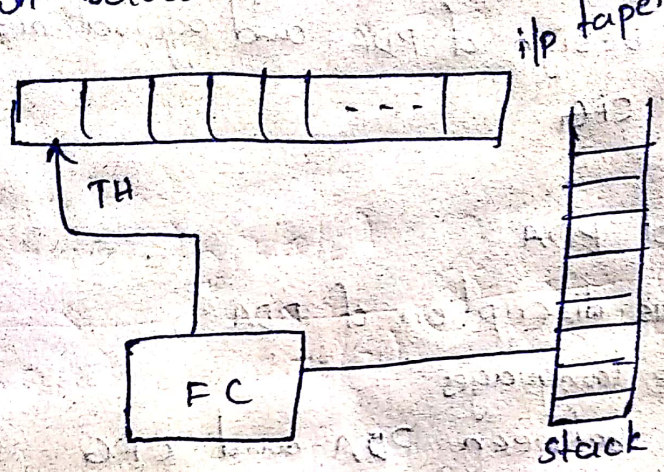
δ = Transition function, represented as

$$\delta : Q \times \{\Sigma \cup \in\} \times \Gamma \longrightarrow Q \times \Gamma^*$$

Γ = stack alphabet

Z_0 indicates the stack empty or bottom of the stack.

Push down automata is represented in diagramatical way as shown below.



Context free languages and instantaneous Description.

Context free languages are represented in two ways

1. state transition diagram
2. Instantaneous description.

* Design PDA to accept the language i) $L = \{a^n b^n / n \geq 1\}$

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

iii) $L = \{a^n b^n c^m d^m / m, n \geq 1\}$

iv) $L = \{a^{m+n} b^m c^n / n, m \geq 1\}$

v) $L = \{a^n b^m c^m d^n / m, n \geq 1\}$

vi) $L = \{a^n b^{2n} / n \geq 1\}$

vii) $L = \{w c w^R / w \in (a+b)^+\}$

Sol

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

Note:

Note:
We can perform three types of operations on stacks.

1. push operation : It is represented as

$$\delta(v_i, x, y) = (v_j, xy)$$

2. pop operation : It is represented as

$$\delta(v_i, x, y) = (v_j, \epsilon)$$

3. skip operation : That means we don't perform either push or pop. It is represented as

$$\delta(v_i, x, y) = (v_j, y)$$

Sol i. $L = \{a^n b^n / n \geq 1\}$

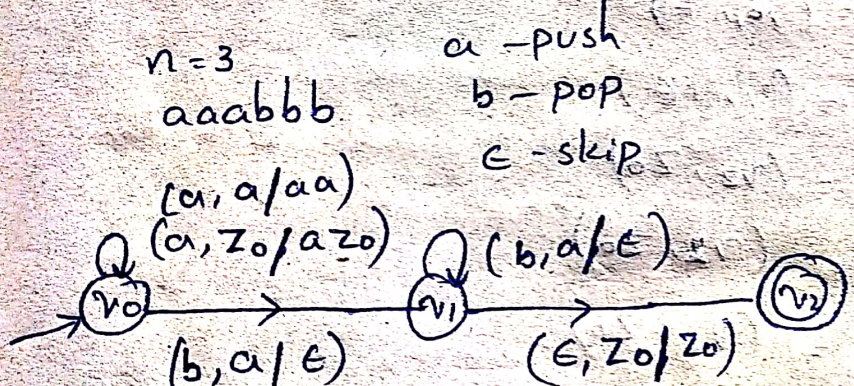
$n = 3$

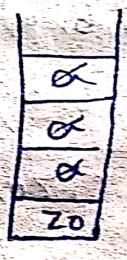
aaabbb

a - push

b - pop

ϵ - skip





For the above diagram we can represent the instantaneous description for every state on every transition as shown below.

$\delta(v_0, a, a) = (v_0, aa)$

$\delta(v_0, a, Z_0) = (v_0, aZ_0)$

$\delta(v_0, b, a) = (v_1, \epsilon)$

$\delta(v_1, b, a) = (v_1, \epsilon)$

$\delta(v_1, \epsilon, Z_0) = (v_2, Z_0)$

$\{a^m b^n c^{} / m, n \geq 1\}$

$n = 2, m = 3$

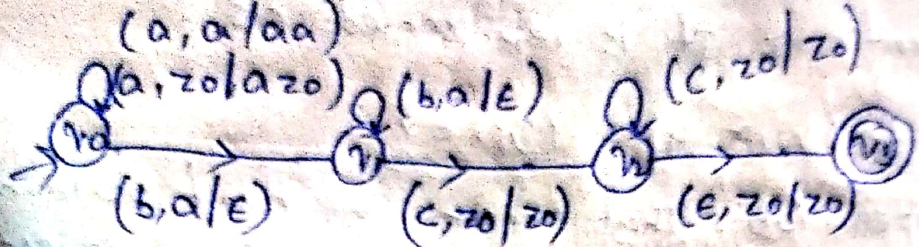
$a^2 b^2 c^3$

aabb ccc

a – push

b – pop

c – skip



Instantaneous description:

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

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, a) = (q_0, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, c, z_0) = (q_2, z_0)$

$\delta(q_2, c, z_0) = (q_2, z_0)$

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

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

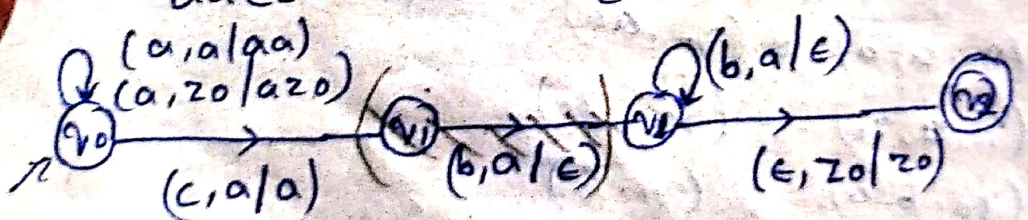
a – push

b – pop

c – skip

$n = 2$

aacbb



Instantaneous description.

$\delta(q_0, aa) = (q_0, aa)$

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, c, a) = (q_1, a)$

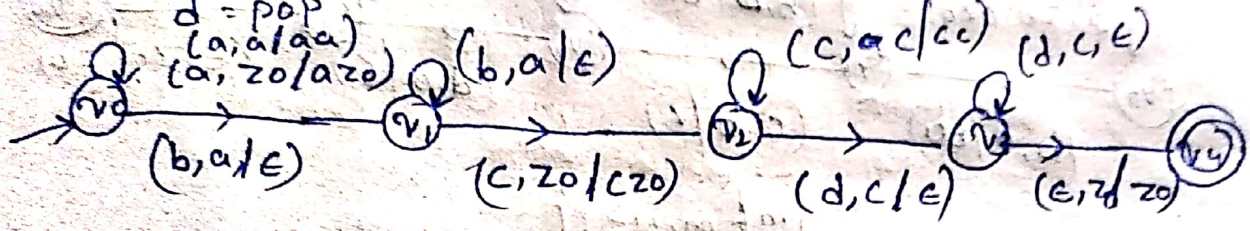
$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(v_2, b, a) = (v_2, \epsilon)$

$\delta(v_1, \epsilon, z_0) = (v_3, z_0)$

iii. $L = \{ a^n b^n c^m d^m \mid m, n \geq 1 \}$

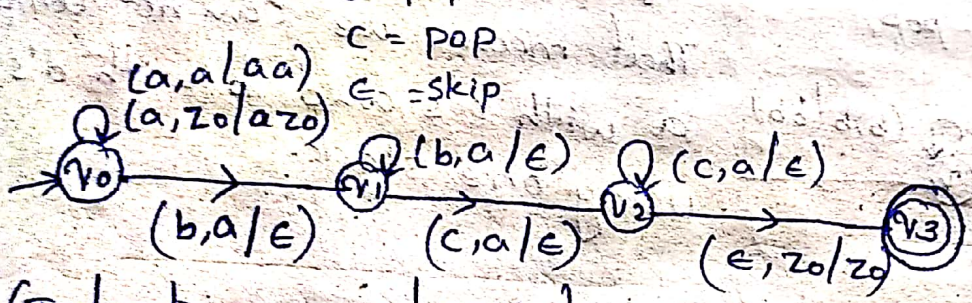
a = push
b = pop
c = push
d = pop



iv. Design PDA to accept the language $L = a^{m+n} b^m c^n \mid m, n \geq 1$

a = push
b = pop
c = pop
ϵ = skip

let $m = 2, n = 3$



(Instantaneous description

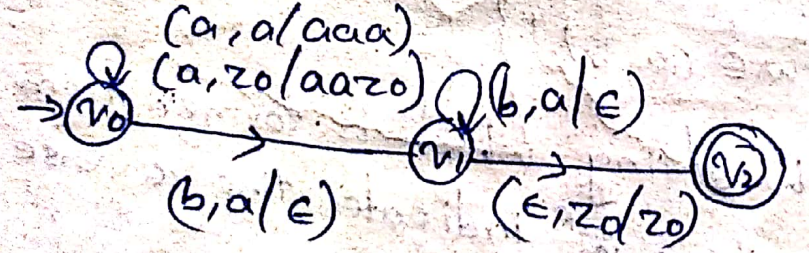
~~$\delta(v_0, a, z)$~~

vi. $L = \{ a^n b^{2n} \mid n \geq 1 \}$

let $n = 2$

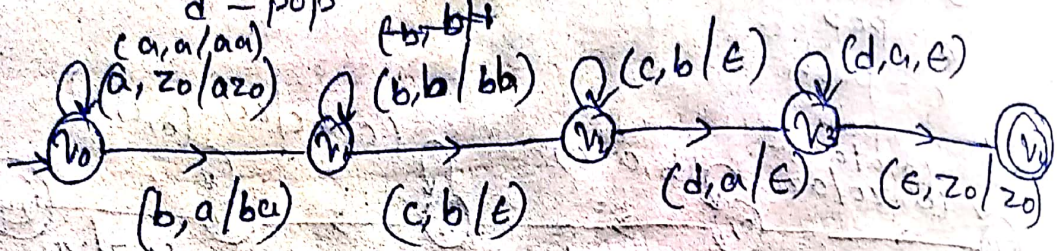
$a^2 b^4$

aabbbb

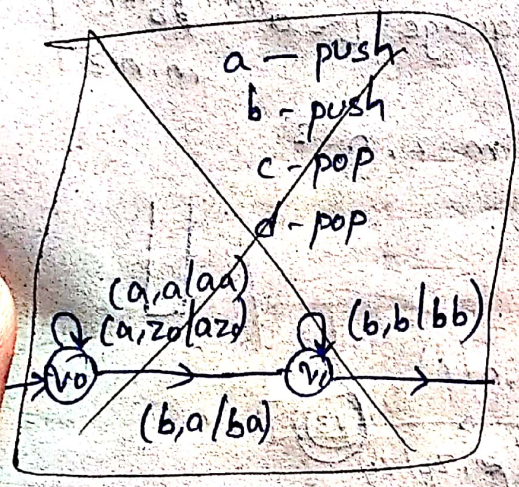


vii. $L = \{a^n b^m c^m d^n / m, n \geq 1\}$

a – push
b – push
c – pop
d – pop



viii) $L = \{a^n b^m c^n d^m / m, n \geq 1\}$



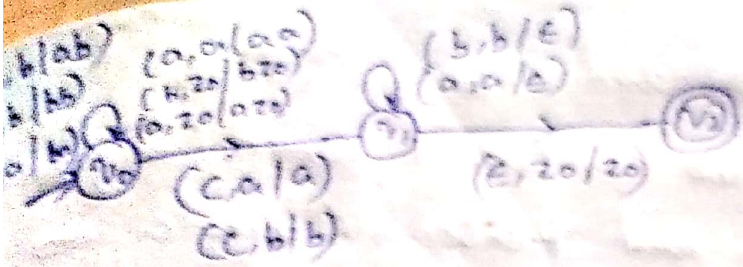
It is not possible. Because the given language is not CFL. That means we can not compare a with c's and b with d's using one stack.

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

It is not a CFL. If we want to compare two variables one stack is enough. But If we compare three variables two stack. But the PDA uses only one stack.

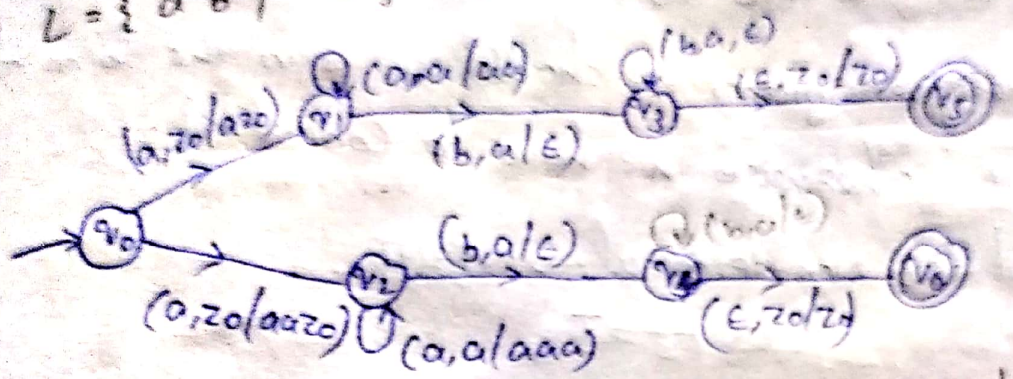
$L = \{wcw^R / w \in (a, b)^+\}$

Here w is the string which is formed from input symbols a's and b's. w^R indicates the reverse of the string and c is the constant.



Design PDA for the following language

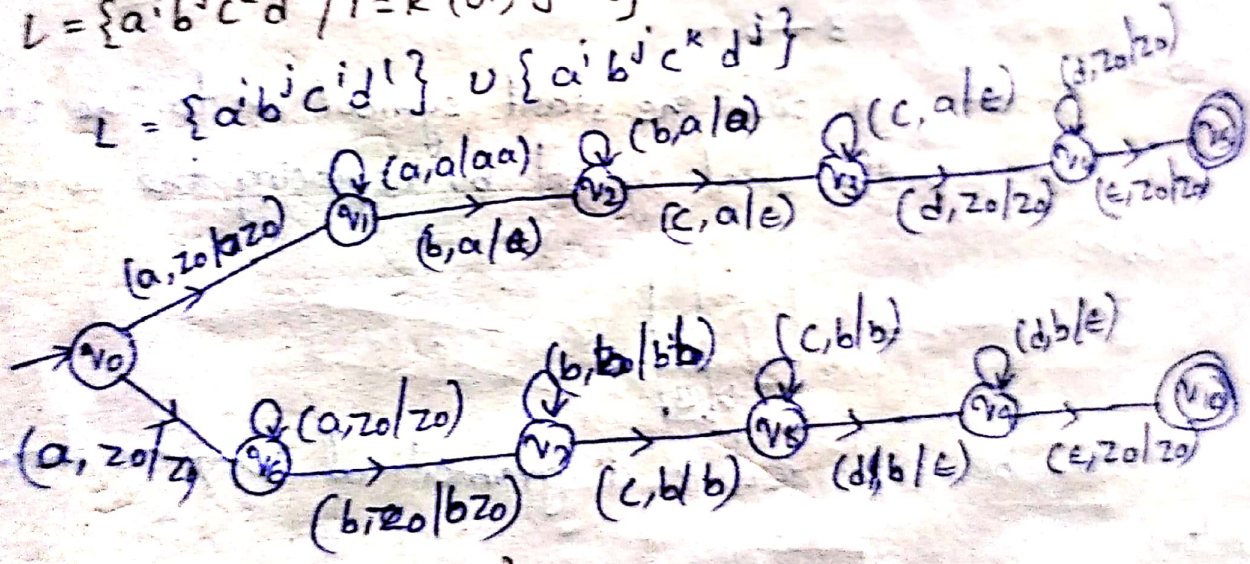
i) $L = \{a^n b^n / n \geq 1\} \cup \{a^n b^{2n} / n \geq 1\}$



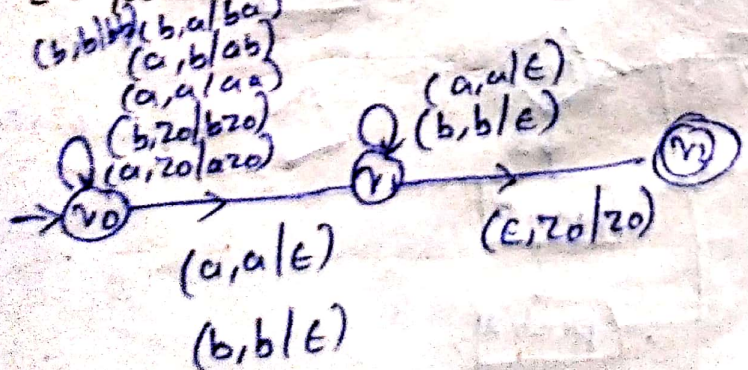
Design PDA state transition diagram for the language

ii) $L = \{a^i b^j c^k d^l / i = k \text{ (or) } j = l\}$

$L = \{a^i b^j c^i d^l\} \cup \{a^i b^j c^k d^j\}$



iii) $L = \{w w^R / w \in (a,b)^+\}$



Equivalence between PDA and CFG

It is nothing but we can convert context free grammers into pushdown Automata and Push Down Autorrata into Context free grammers.

Conversion of CFG into PDA

step-1 The given grammer should be is in the form of CNF. otherwise we have to convert it.

That is Non-terminal $\rightarrow$ NT.NT

NT $\rightarrow$ T

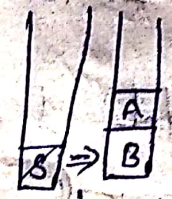
Ex: $S \rightarrow AB$

$A \rightarrow a$

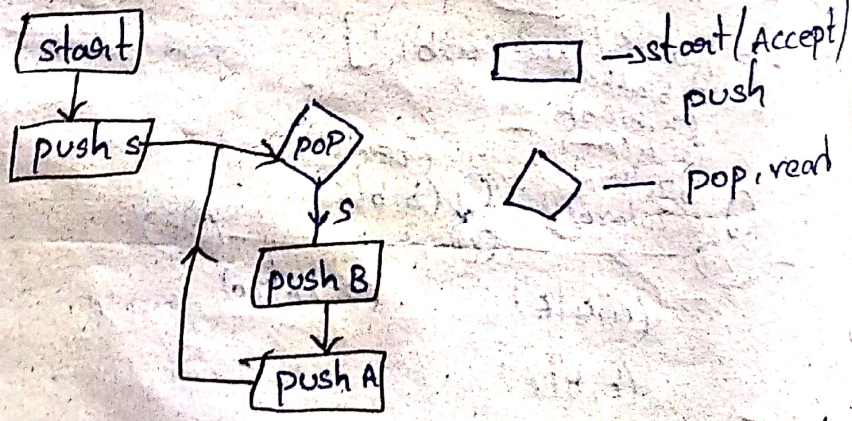
$B \rightarrow +$

step-2 Insert (or) push the start symbol on to the stack and then pop the start symbol by replacing the corresponding variables.

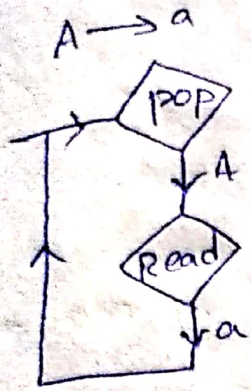
Ex: $S \rightarrow AB$



It can be explained in diagramatical way as shown below



step-3 If the production is in the form of non-terminal tends to terminal then the operation is explained as shown in the following diagram.



step-4 write the instantaneous description for given context free grammer.

Convert the following CFG into PDA

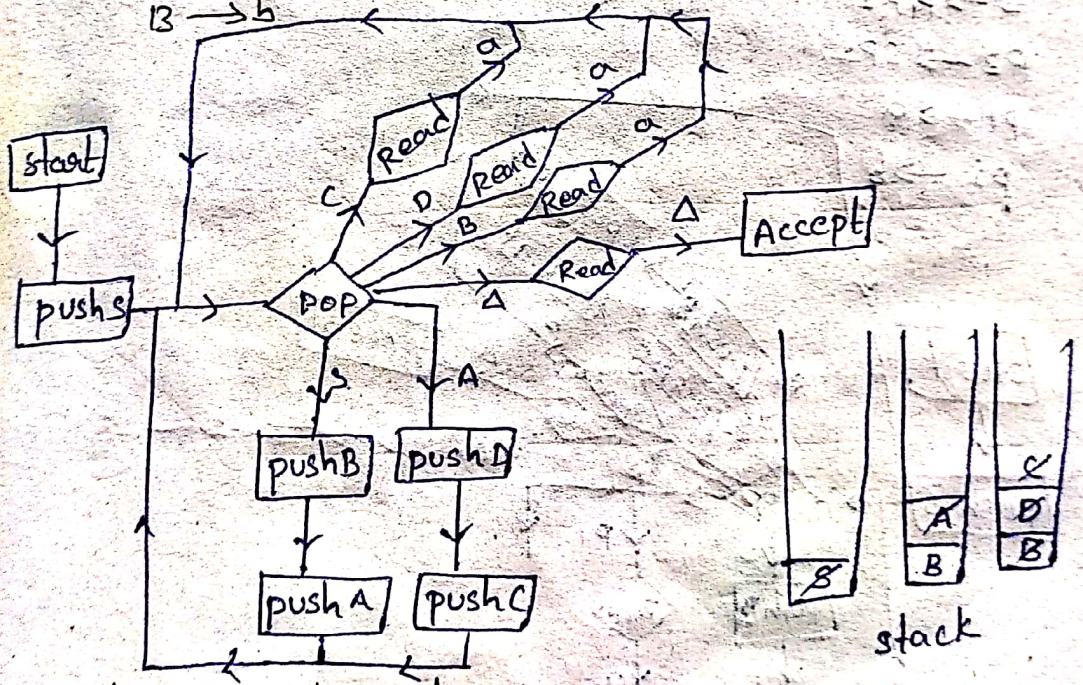
$S \rightarrow AB$

$A \rightarrow CD$

$D \rightarrow a$

$C \rightarrow a$

$B \rightarrow b$



Instantaneous description.

$\delta(q, w, S) = (q, AB)$

$\delta(q, w, A) = (q, CD)$

$\delta(q, a, C) = (q, \in)$

$\delta(q, a, D) = (q, \in)$

$\delta(q, b, B) = (q, \in)$

2. Convert CFG into PDA

$S \rightarrow aSb$

$S \rightarrow ab$

let us consider

$P1 \rightarrow a$

$P2 \rightarrow b$

$\Rightarrow S \rightarrow P1SP2$

$S \rightarrow P1P2$

$P1 \rightarrow a$

$P2 \rightarrow b$

let us consider $SP2 \rightarrow P3 \rightarrow SP2$

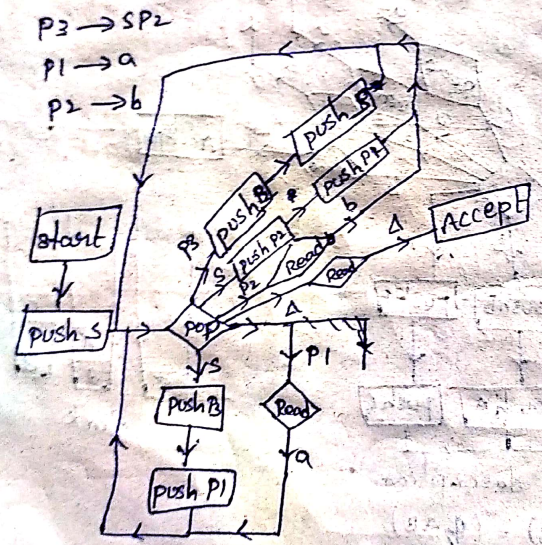
$\Rightarrow S \rightarrow P1P3$

$S \rightarrow P1P2$

$P3 \rightarrow SP2$

$P1 \rightarrow a$

$P2 \rightarrow b$



instantaneous description

$\delta(q, w, S) = (q, P1P3)$

$\delta(q, w, S) = (q, P1P2)$

$\delta(q, w, P3) = (q, SP2)$

$\delta(q, a, P1) = (q, \epsilon)$

$\delta(q, b, P2) = (q, \epsilon)$

Deterministic PDA and non-deterministic PDA

context free languages are classified into two catagories

1. Deterministic context free languages
2. Non-deterministic context free languages.

DCFL is recognised by deterministic push down Automata (DPDA)

*NCFL is recognised by NPDA

* By default push down Automata is the NPDA
* The relationship between DPDA and NPDA is DPDA ⊆ NPDA

* Every DPDA is NPDA but not vice versa
* Every language L is a regular language then it is accepted by finite automata.
* The language which is regular then it is accepted by DPDA as well as NPDA
* The relationship between FA, DPDA, NPDA is represented as

FA ⊆ DPDA ⊆ NPDA

Identify the following languages which are RL, DCFL, CFL
(FA) (DPDA) (NPDA)

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

In this example one comparision is required (a ↔ b)
But in regular language there is no need of stack. But
In this example one stack is required

RL — ✗

DCFL — ✓

CFL — ✓

ii) $L=\{a^n b^n c^m / m,n \geq 1\}$

RL ×
DCFL ✓
CFL ✓

iii) $L=\{a^{m+n} b^m c^n / m,n \geq 1\}$

RL ×
DCFL ✓
CFL ✓

iv) $L=\{a^n b^{m+n} c^m / m,n \geq 1\}$

RL ×
DCFL ✓
CFL ✓

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

RL ×
DCFL ×
CFL ×

vi) $L=\{a^n b^n / n \geq 1\} \cup \{a^n b^{2n} / n \geq 1\}$

RL ×
DCFL ×
CFL ✓

The language is not DCFL because the string (or) language has two paths to get the final state.

vii) $L=\{wcw^R / w \in (a,b)^+\}$

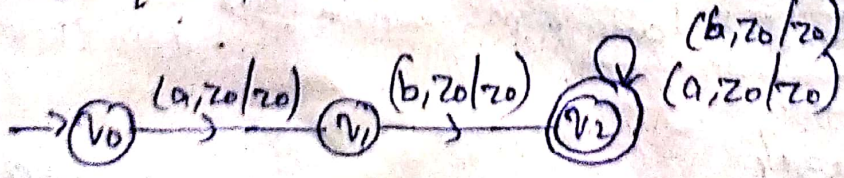
RL ×
DCFL ✓
CFL ✓

viii) $L=\{ww^R / w \in (a,b)^+\}$

RL ×
DCFL ×
CFL ✓

Design the DPDA diagram for regular language, the language which starts with ab over the alphabet $\Sigma=\{a,b\}$

$L=\{aba, abb, abaa, \ldots\}$



UNIT-5

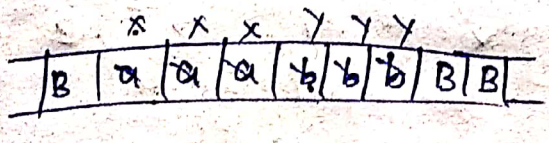
Turing Machine

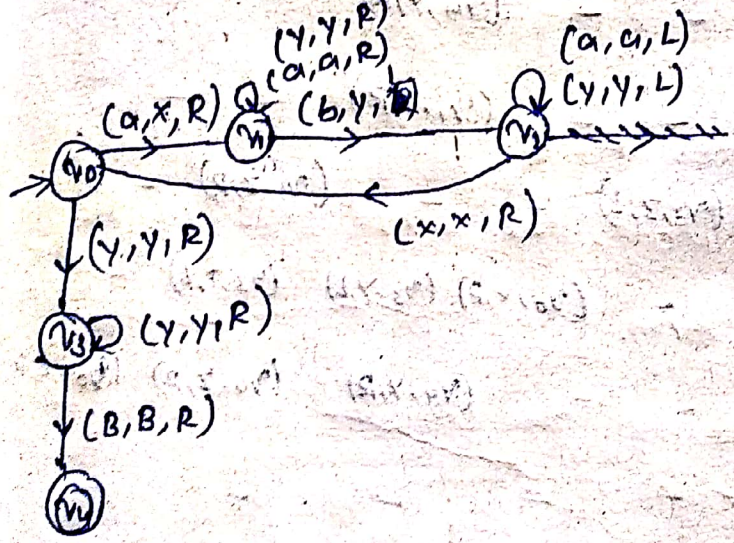
Topics

* Definition of TM
* Model
* Representation of Turing Machines
  1. Instantaneous description
  2. Transition tables
  3. Transition diagram
* language acceptance of a TM
* Design of TM
* Types of TM
*

1. $L = \{a^n b^n \mid n \geq 1\}$

n = 3

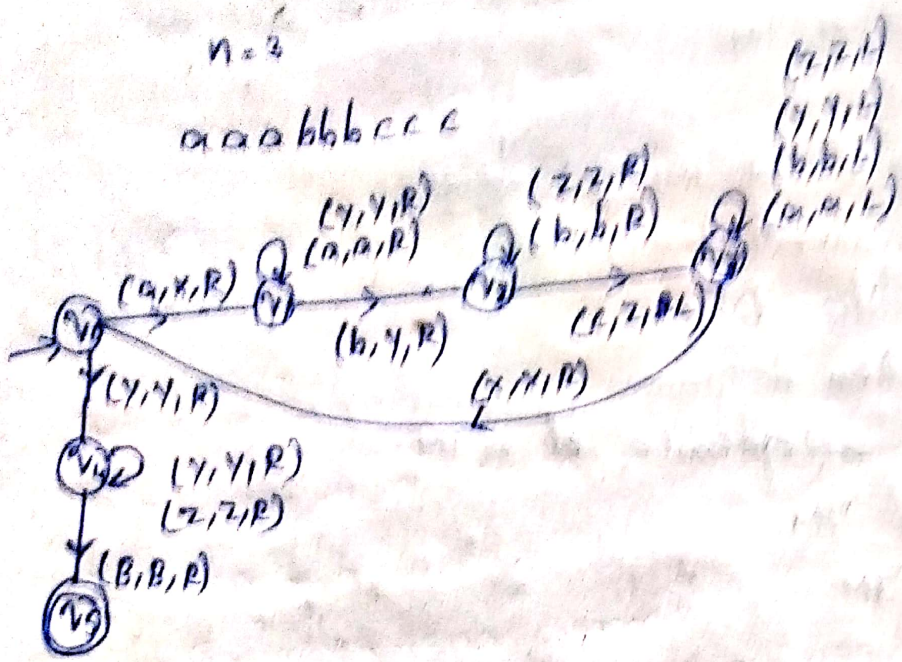




2. $L = \{a^n b^n c^n / n \geq 1\}$

$n = 3$

aaabbbccc



Transition Table.

	a	b	c	X	Y	Z	B
q_0	(q_1, X, R)	-	-	-	(q_{14}, Y, R)	-	-
q_1	(q_1, a, R)	(q_2, Y, R)	-	-	(q_1, Y, R)	-	-
q_2	-	(q_2, b, R)	(q_3, Z, L)	-	-	(q_2, Z, R)	-
q_3	(q_2, a, L)	(q_3, b, L)	-	(q_0, X, R)	(q_3, Y, L)	(q_3, Z, L)	-
q_4	-	-	-	-	(q_4, Y, R)	(q_4, Z, R)	(q_5, B, R)
q_5	-	-	-	-	-	-	-

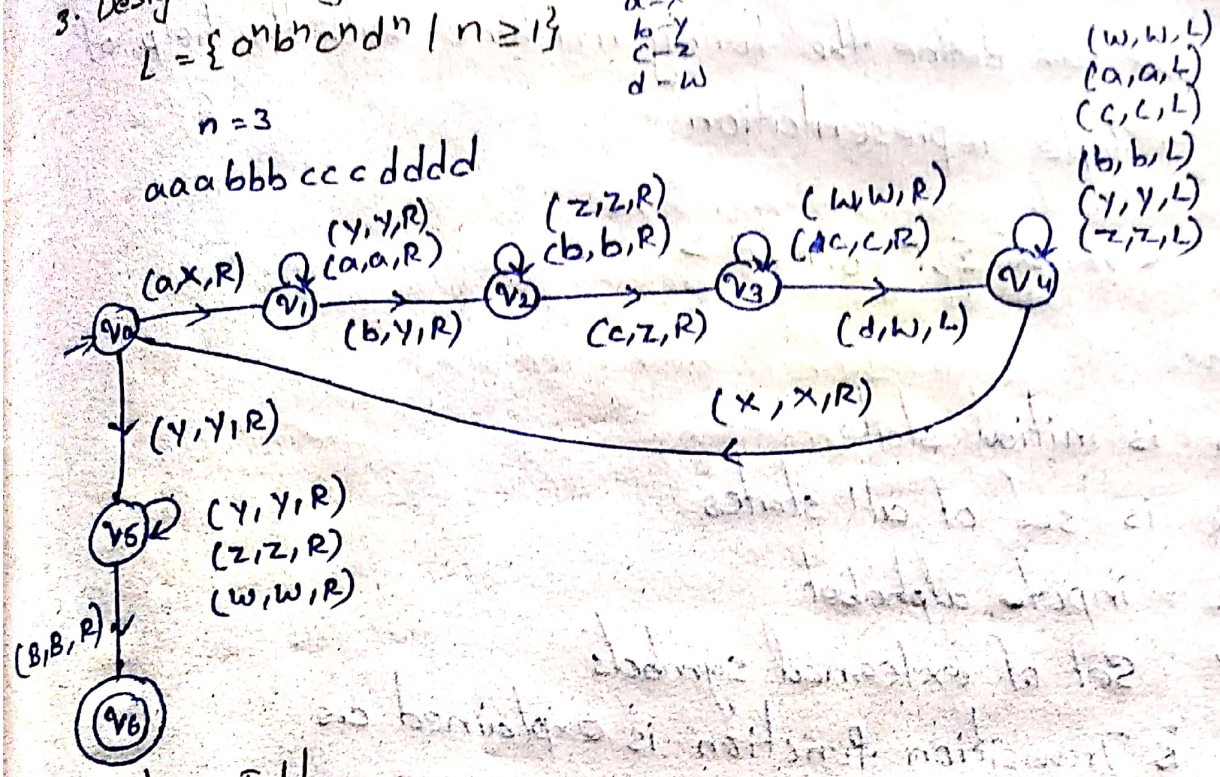
3. Design turing machine to accept the language

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

a - X
b - Y
c - Z
d - W

n = 3

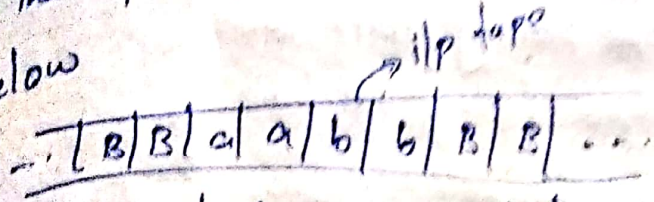
aaabbbcccddd



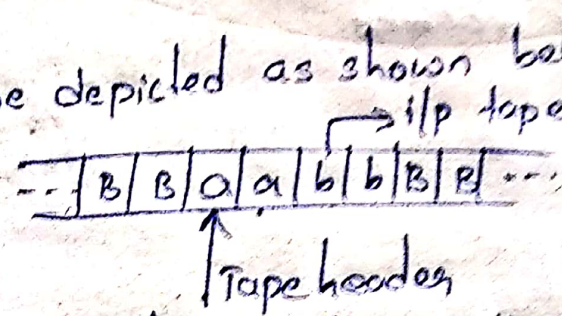
Transition Table.

	a	b	c	d	X	Y	Z	W	B
q_0	(q_1, X, R)	–	–	–	–	(q_5, Y, R)	–	–	–
q_1	(q_1, a, R)	(q_2, Y, R)	–	–	–	(q_1, Y, R)	–	–	–
q_2	–	(q_2, b, R)	(q_3, Z, R)	–	–	–	(q_2, Z, R)	–	–
q_3	–	–	(q_3, C, R)	(q_4, W, L)	–	–	–	(q_3, W, R)	–
q_4	(q_4, a, L)	(q_4, b, L)	(q_4, C, L)	–	(q_0, X, R)	(q_4, Y, L)	(q_4, Z, L)	(q_4, W, L)	–
q_5	–	–	–	–	–	(q_5, Y, R)	(q_5, Z, R)	(q_5, W, R)	(q_6, B, R)
q_6	–	–	–	–	–	–	–	–	–

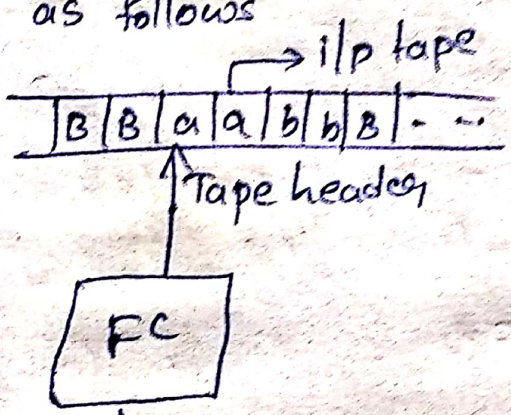
* The tape representation with input string as shown below



* The input tape is pointing by the tape header
* From the tape we can read the data or onto the tape we can write the data with the help of tape header
* It can be depicted as shown below



* The turing machine is having the finite control.
* The Block diagram (or) model of a turing machine is represented as follows



* In turing machine we can move from left to right (or) right to left ~~with~~ in perspective of tape. But incase of Finite Automata and push down automata the tape header is moving from left to right only.
* In case of FA and PDA It we can perform read operation only on to the stack. But in case of turing machine it can perform read operation as well as right operation on to the stack.

Types of turing machine:

1. One way infinite tape turing machine
2. Two way infinite tape turing machine
3. Multi tape turing machine
4. Multi head turing machine
5. multi dimensional turing machine
6. offline turing machine

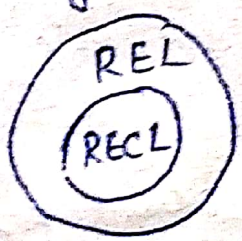
Unit - VI

Syllabus:

1. Recursive languages
2. Recursive Enumerable Language
3. closure properties of Recursive languages
4. closure properties of Recursive Enumerable languages
5. Universal Turing machine
6. Decidable and un-decidable Languages
7. Halting problem of Turing Machines
8. post correspondence problem
9. Modified post's correspondence problem.

Recursive language:

1. A recursive language (subset of REL) can be decided by Turing machine which means it will enter into final state for the string of language and rejecting state for the strings which are not part of the language.
2. Example, $L = \{a^n b^n c^n / n \geq 1\}$ is recursive because we can construct a turing machine which will move to final state if the string is of the form $a^n b^n c^n$ else move to non-final state.
3. So the TM will always halt in this case.
4. REC languages are also called as Turing decidable languages.
5. The relationship between RE and REC languages can be shown in figure

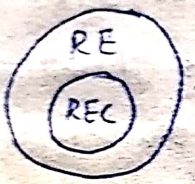


RECL — Recursive language

REL — Recursive Enumerable language

Recursive Enumerable Language:

1. Recursive Enumerable languages or type-0 languages are generated by type-0 grammers
2. A RE language can be accepted or recognized by Turing Machine which means it will enter into final state for the strings of language and may or may not enter into reject state for the strings which are not part of the language.
3. It means TM can loop forever for the strings which are not a part of the language.
4. Recursively Enumerable languages are also called as Turing undecidable / Turing recognizable languages.
5. The relationship b/w REC and RE is



closure properties of Recursive language:

* Recursive languages not closed under Homomorphism and substitution
* Union: If L_1 and L_2 are two recursive languages, their union $L_1 \cup L_2$ will also be recursive because if TM halts for L_1 and halts for L_2, it will also halt for $L_1 \cup L_2$
* Concatenation: If L_1 and L_2 are two recursive languages, their concatenation $L_1 \cdot L_2$ will also be recursive. For example

$L_1 = \{a^n b^n c^n / n >= 0\}$

$L_2 = \{d^m e^m f^m / m >= 0\}$

$L_3 = L_1 \cdot L_2$

$= \{a^n b^n c^n d^m e^m f^m / m \geq 0 \text{ and } n >= 0\}$ is also recursive.

* Kleen closure:
If L1 is recursive, its kleene closure L1* will also be recursive. For example:

$L1 = \{a^n b^n c^n | n >= 0\}$

$L1^* = \{a^n b^n c^n | n >= 0\}^*$ is also recursive.

* Intersection and complement:
If L1 and L2 are two recursive languages, their intersection $L1 \cap L2$ will also be recursive. For Example:

$L1 = \{a^n b^n c^n d^n | n >= 0$ and $m >= 0\}$

$L2 = \{a^n b^n c^n d^n | n >= 0$ and $m >= 0\}$

$L3 = L1 \cap L2$

$= \{a^n b^n c^n d^n | n >= 0\}$ will be recursive.

Recursive Enume

closure properties of Recursive Enumerable languages

* Recursive Enumerable Languages are not closed under Complementation and set difference.
* Union: If L1 and L2 are two recursive enumerable languages, their union $L1 \cup L2$ will also be recursive enumerable because if TM halts for L1 and halts for L2

* intersection
* Kleen closure
* concatination

} same as recursive languages

Post's correspondence problem:

Establishment of post's corresponding problem in 1946's.

Two equal side series of non-null string over the same characters set "Σ"

we illustrate with an example of two series of strings

x - series
y - series
} Σ = {0,1}

i	xi	yi
1	10	101
2	01	100
3	0	10
4	100	0
5	1	010

From this table Consider the substring

x-series :- $x_1 x_5 x_2 x_3 x_4 x_4 x_3 x_4$

y-series :- $y_1 y_5 y_2 y_3 y_4 y_4 y_3 y_4$

Construction of x and y is :-

x → 10101010001000100

y → 10101010001000100

Here both are same, in this instance we say instance post correspondence problem has a solution in the form of -15234434

2.

i	xi	yi
1	0	000
2	01000	01
3	01	1

x-series → $x_2 x_3$
y-series → $y_2 y_3$
x-series → $x_2 x_1 x_1 x_3$
y-series → $y_2 y_1 y_1 y_3$

Here 2 1 3 is not the solution
One of the solution for this problem is 2113

Modify post Correspondence problem:

In case of post correspondence problem the series values of both x and y should be the same.

For eg; if 2113 is the solution then x-series and y-series is represented as

x-series = x_2 x_1 x_1 x_3
y-series = y_2 y_1 y_1 y_3

But in case of modified post correspondance, problem the series correspondance, problem starting position values should be same.

for eg: we considea the x-seqies and y-seqies as

x-series - $x_2 x_1 x_5 x_7$
y-series - $y_2 y_5 y_1 y_3$

Decidable and undecidable language:

* The language which is solvable is called as decidable language.
* The language which is unsolvable is called o undecidable language.
* Based on getting the solution we can also classify the problem into two categories
  - i) Decidable problems
  - ii) Undecidable problems

In case of decidable problems, the solution is defined
In case of undecidable problems, sometimes it may be possible to define the solution or sometimes it may not be possible to find the solution.

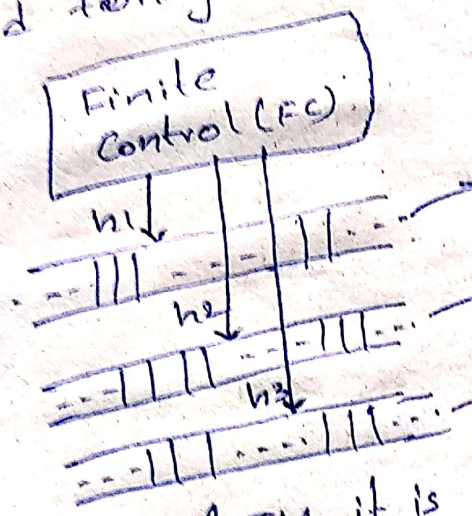
The following are the examples of decidable languages.

→ FA accept the ~~recursive language~~ regular language
→ If L1 is regular language, L2 is regular language then L1∪L2 is also regular language and L1∩L2 is also regular language
→ If L1 and L2 are two context free languages then L1∪L2 is also context free language.

The following are equations of undecidable languages
→ The CFG 'G' the language over G is L(G) is ambiguous
→ If L1 is CFL and L2 is CFL then L1∪L2 is CFL

Universal Turing Machine

* The turing machine which can perform many kinds of turing machine operations is called as universal turing machine
* The language which is accepted by universal TM is called universal language.
* The notation to represent the universal language is L_U
* We can also represent the universal language L_U as $L_U = L(U)$
* Let us consider a universal turing machine which is multi head turing machine as shown below



* In the above universal TM, it is having 3 tapes and they are pointing by the corresponding header
* The i/p tape is having the input string
* The encoded tape is having the encoded data of given input string
* The states tape gives the complete information regarding all states of the universal TM