Q. Prove that the language $L = \{ a^n b^n \mid n \geq 1 \}$ is Non-Regular

⇒ 1) Assume L is regular.

2) F.A exists → n states.

3) $a^n b^n \rightarrow n$

2n (length)

4) $a^{n-1} \quad a \quad b^n$ or $\underbrace{aa}_{x}\underbrace{ab}_{y}\underbrace{bb}_{z}$ aaabbb

$x y^i z \Rightarrow aaababbb \notin L$

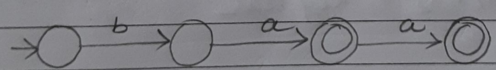
5) $a^{n-1} \cdot a^2 \cdot b^n$

$a^{n+1} \cdot b^n \notin L$

Language is non-Regular

Q. There exist a Regular language L and ~~its~~ F.A all strings in the language having length less than no. of states of its finite Automata, Then the language is known as Finite Language

Example -



GATE 2019

For $\Sigma = \{a, b\}$, let us consider the regular language

$$L = \{ x \mid x = a^{2+3k} \text{ (or) } x = b^{10+12k}, k \geq 0 \}$$

which one of the following can be a pumping length for L?
↓ No of states

a) 3

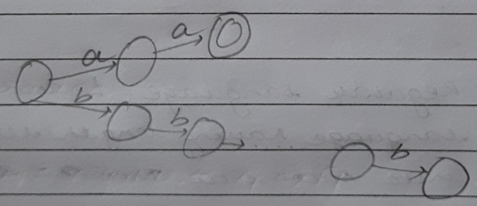
b) 5

c) 9

✓d) 24

Put $k = 0$.

$$\left\{ \begin{matrix} a^2 \dots \\ b^{10} \dots \end{matrix} \right\}$$



13 states है।
तो 3, 5 या 9 तो हो ही नहीं सकता।

CLOSURE PROPERTIES OF Various LANGUAGES

OPERATION	Regular	DCFL	CFL	CSL	REC	RE
UNION	YES	NO	YES	YES	YES	YES
INTERSECTION	YES	NO	NO	YES	YES	YES
COMPLEMENT	YES	YES	NO	YES	YES	NO
CONCATENATION	YES	NO	YES	YES	YES	YES
HOMOMORPHISM	YES	NO	YES	NO	NO	YES
SUBSTITUTION	YES	NO	YES	YES	NO	YES
Inv. HOMOMORPHISM	YES	YES	YES	YES	YES	YES
REVERSE	YES	NO	YES	YES	YES	YES
INTERSECTION with a Regular language.	YES	YES	YES	YES	YES	YES

Subset operation on Regular Languages.

Suppose $L = (a+b)^*$ (Regular)

Subset of $L = a^n b^n$ (Non-regular)

Subset operation is not closed under Regular Languages

Subset of regular languages ~~is~~ may or may not be Regular.

Q. Which of the following is true?

a) Subset of Non-Regular set is regular
b) Subset of any infinite set is regular
c) Subset of any regular set is regular
✓d) None

L-31

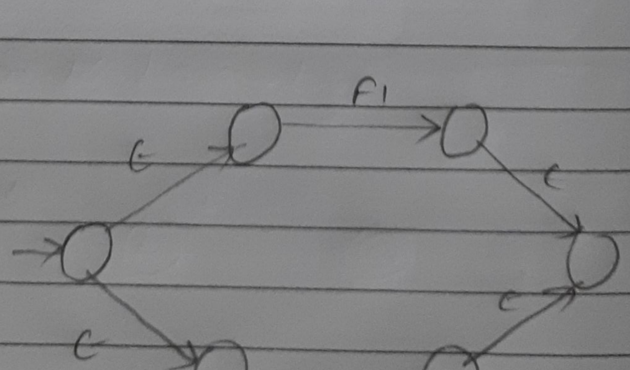
Union Operation

$L_1 = \text{regular} = r_1 \Rightarrow F1$

$L_2 = \text{regular} = r_2 \Rightarrow F_2$

$L_1 \cup L_2 = r_1 + r_2 = \text{Regular}$

Always Regular



$\cup \Rightarrow$ OR $(+)$
$\cap \Rightarrow$ AND $(\times)$

classmate
Date
Page

Concatenation Operation

$$L_1 = \{a^n \mid n \geq 1\} = \text{Regular} = a^+$$
$$L_2 = \{b^n \mid n \geq 1\} = \text{Regular} = b^+$$

$L_1 \cdot L_2 \Rightarrow$ Regular $= a^+b^+$
Closed

Q. If $L_1 = \{a^n \mid n \geq 0\}$ and $L_2 = \{b^n \mid n \geq 0\}$, Consider

I) $L_1 \cdot L_2$ is a regular language. a^*b^*
II) $L_1 \cdot L_2 = \{a^nb^n \mid n \geq 0\}$

which one of the following is CORRECT ?
✓a) Only I (b) Only II
b) Both I & II (d) Neither I nor II

Intersection Operation

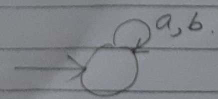
L_1 = regular $\Rightarrow F_1$
L_2 = regular $\Rightarrow F_2$

$L_1 \cap L_2 \Rightarrow$ Regular $= F_1 \times F_2$ = and construction
Always Regular

Q. How many states in minimal DFA that accepts $L_1 \cap L_2$, where $\begin{cases} L_1 = (a+b)^* a \rightarrow \text{Ending with } a \\ L_2 = (a+b)^* b \ \text{Ending with } b \end{cases}$

a) 1
b) 2
c) 3
d) 4

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

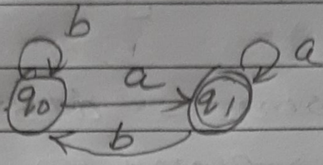


Q. How many states in minimal DFA for $L_1 \cap L_2$

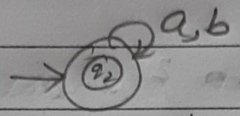
$$L_1 = (a+b)^* a$$
$$L_2 = (a+b)^*$$

a) 1
✓ b) 2
c) 3
d) 4

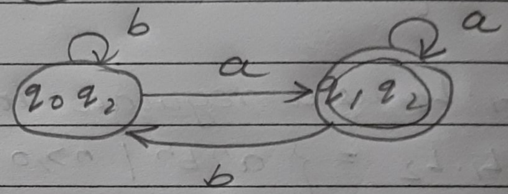
$L_1 =$



$L_2 =$



$L_1 \times L_2 =$



Superset $\cap$ subset = Subset

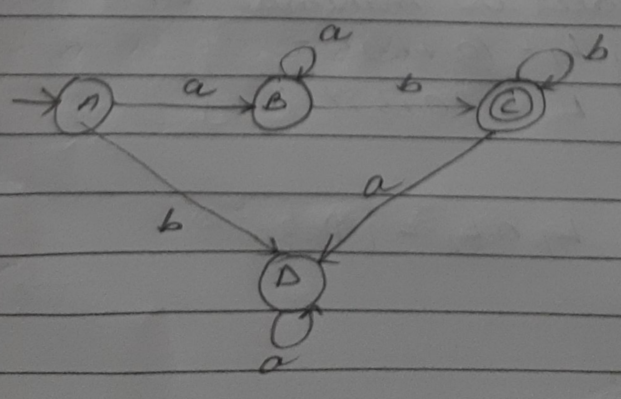
Complement operation

$\Sigma^* - L$ = DFA (interchanging final & non-final)
Regular

Reverse operation

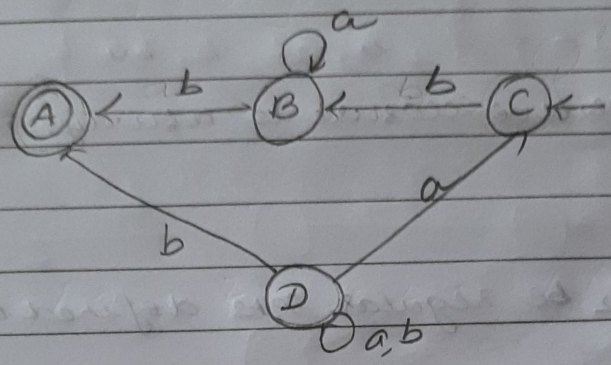
L = Regular
L^R =

L = reg $= \{ a^n b^m \mid n, m \geq 1 \}$



L^R = Regular ⇒

Interchange Final & Initial states only.



Reverse the transitions (selfloops remain as it is)

Q. Which of the following is false.

a) If L is regular $L \cdot L^R$ is also regular
b) If L is regular, $L \cap L^R$ is also regular
c) If L is regular $\{vu \mid v \in L, u \in L^R\}$ is regular
✓d) $L = \{ww^R \mid w \in L\}$ L is regular

Reversal Operation on Regular Expression

1. $(E+F)^R = E^R + F^R$
2. $(E.F)^R = F^R . E^R$
3. $(E^*)^R = (E^R)^*$

Date
Page

Kleene Closure Operation

$L \Rightarrow$ Regular Language $= r$

$L^* \Rightarrow r^*$

Regular

$L^+ =$ Positive closure is always regular

Q. Let R_1 and R_2 be regular sets defined over the alphabet then

a) $R_1 \cap R_2$ is not regular — Regular

b) $R_1 \cup R_2$ is not regular — Regular

✓ c) $\Sigma^* - R_1$ is regular (Complement)

d) R_1^* is not regular — Regular

Infinite Union Operation

$L_1, L_2, L_3 \ldots\ldots$ = Regular language

$L_1 \cup L_2 \cup L_3 \ldots\ldots$ } Non-Regular/ Regular

May or maynot be regular (Not closed)

$\{ab\} \cup \{a^2b^2\} \cup \{a^3b^3\} \cup \ldots\ldots = \{a^nb^n\}$

Note -

Regular languages are closed under finite union but may or may not ~~closed~~ regular under infinite union.

Infinite Intersection

$L_1, L_2, L_3 \ldots \}$ Regular languages

$L_1 \cap L_2 \cap L_3 \cap \ldots \}$ Non-Regular / Regular.

$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

Hence, $L_1 \cap L_2 \cap L_3 \cap \ldots = \overline{L_1} \cup \overline{L_2} \cup \overline{L_3} \cup \ldots$

Hence, Not closed

Note:-

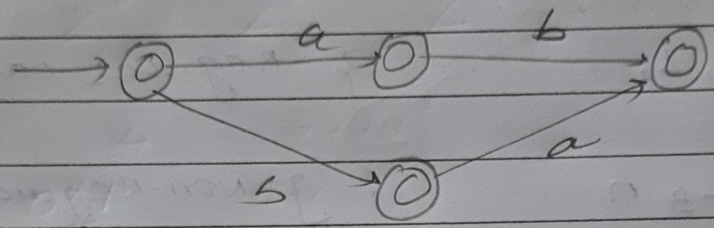
Not closed under Regular languages

1) Subset Operation
2) Infinite union
3) Infinite Intersection
4) Infinite Concatenation

Prefix operation (or) Init operation

$L = \{ab, ba\}$

Prefix$(L) = \{\epsilon, a, ab, b, ba\}$



Prefix Automata is constructed by making all States as final (excluding Dead state).

[Closed]

Suffix operation

Suffix(L) = Reverse(Prefix(Reverse(L)))

Example -

L = "TOC"   Suffix of $(L) = \epsilon$, C, OC, TOC

Reverse(L) = COT

Prefix of (Reverse of (L)) = ϵ, C, CO, COT

Reverse(Prefix(Reverse(L))) = ϵ, C, OC, TOC

[Closed]

08-07-2021

Q. If L_1 is regular & $L_2 \subseteq L_1$, then which of the following has to be regular ?

a) L_2 (b) $L_1 \cap L_2$ (c) L_2^n ✓(d) L_1^n

L_2 is a subset of L_1

may or may not be regular

Quotient Operation

$$L_1/L_2 = \{ x \mid xy \in L_1, y \in L_2 \} \qquad \frac{L_1}{L_2} = \frac{xy}{y} = x$$

E.g $\frac{010}{10} = 0$, $\frac{101}{11} = \phi$, $\frac{01010}{10} = 010$

$\frac{0101}{0101} = \epsilon$, $\frac{010}{\epsilon} = 010$, $\frac{\epsilon}{010} = \phi$

$$\frac{0^*}{0} = \{ \epsilon, 0, 00, 000, \dots \}$$

$\{ \epsilon, 0, 00, 000 \}$

$= 0^*$

$$\frac{\Sigma^*}{\Sigma^*} = \Sigma^* \Rightarrow \{ \epsilon, a, b, aa, ab, bb, ba \}$$

$\{ \epsilon, a, b \} = \Sigma^*$

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

$$\Rightarrow \Sigma^*$$

$$\Sigma^* = \frac{\Sigma^*}{\Sigma^*} = \frac{\Sigma^+}{\Sigma^*} = \frac{\Sigma^*}{\Sigma^+} = \frac{\Sigma^+}{\Sigma^*}$$

Q. Let $L_1 = 0^*1$
$L_2 = 1^*0$

(a) 0^*11^*0 (b) 0^*
(c) 0^*1 ✓(d) ϕ

Q. Let $L_1 = a^*ba^*$ $L_2 = b^*a$ then $L_1/L_2 =$

✓(a) $a^*ba^* + a^*$ (b) a^*
(c) a^*ba^* (d) $a^*b + a^+$

$$\frac{a^*ba^*}{b^*a}$$

$$\frac{aba}{a} \Rightarrow ab$$

$$\frac{a^*ba}{ba} = a^*$$

assume a^* से ϵ आया

$$\frac{ba^*}{b^*a} \Rightarrow \frac{ba}{ba} = \epsilon$$ तो आयेगा ही आयेगा।

Q. $L_1 = a^*b$, $L_2 = b^*$

then $L_1 / L_2 = ?$

a) $a^* b^+$

√b) $a^*(b+\epsilon)$

c) $a^* b$

d) b^*

$$\frac{a^*b}{b^* \to \epsilon} \Rightarrow a^*b$$

$$\frac{a^*b}{b} \Rightarrow a^*$$

$$a^*b + a^*$$

$$a^*(b+\epsilon)$$

Quotient Operation is closed for regular languages.

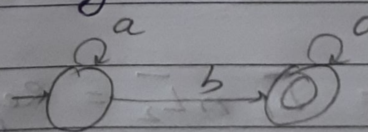
Quotient Finite Automata

1) L_1/L_2 Finite Automata is L_1 Automata only but there maybe a change in final state.

2) To know first state is final state or not ?
Starting from first state consider the total automata if it accepts any string of L_2 then first state is final otherwise non-final

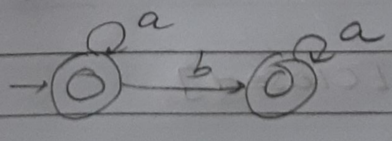
3) Repeat this process for every state

$L_1 = a^*ba^*$ ⇒



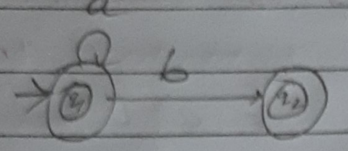
$L_2 = b^*a$

ba
accepted



L_1/L_2

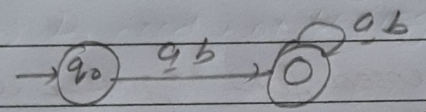
$L_1 = a^* b$ $L_2 = b^*$



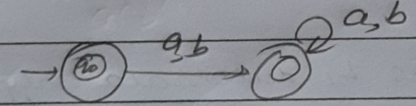
b accepted
make q_1 final

ϵ accepted at q_2
make q_2 final

* $\Sigma^* = \dfrac{\Sigma^+}{\Sigma^+} = \dfrac{(a+b)^+}{(a+b)^+}$



accepted L_2 a, b



$= \Sigma^*$

Difference operation.

L_1 = Regular
L_2 = Regular

$L_1 - L_2 = L_1 \cap \overline{L_2}$ = always Regular

[Closed]

classmate
Date
Page

Substitution operation

Symbol replaced by language (set of strings)

Suppose ⇒ $0^* 1^*$

↓ ↓

Replaced by

$(ab)^* (ba)^*$

[closed]

Homomorphism Operation.

~~Symbols~~ ~~String~~ will be replaced by one string

$0 \ 1 \Rightarrow ab \ ba$ [closed]

Inverse Homomorphism operation

Strings will be replaced by Symbols

$ab \ ba \Rightarrow 0 \ 1$

[closed]

Q. $L_1 \cap L_2$ is regular and L_1 is Regular. what can be said about L_2 = ?

a) L_2 should be Regular

b) L_2 need not be Regular

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

Regular

$\{ab\}$ Regular

$\{a^n b^n\}$ Non Regular

Q. L_1 is regular and $L_1 \cup L_2$ is regular then L_2 is ___?

a) Should be regular

✓b) Need not be regular

$$L_1 \cup L_2 = (a+b)^*$$

$L_1 \to (a+b)^*$, $L_2 \to (a^n b^n)$, $(a+b)^* \leftarrow$ Regular

Q. L_1 is regular, $L_1 . L_2$ is regular then L_2 is

a) L_2 should be regular

✓b) L_2 need not be regular

$$L_1 . L_2 = \phi$$

$L_1 \to \phi$ Regular, $L_2 \to a^n b^n$, ϕ Regular

GRAMMAR

TYPES OF GRAMMAR

Type 3 G (or) Regular Grammar	TYPE 2 Grammar (or) Context Free Grammar	TYPE 1 Grammar (or) Context Sensitive Grammar	Type 0 G (or) Unrestricted Grammar
Right linear Grammar $A \to xB/x$	$A \to \alpha$	$\alpha \to \beta$	$\alpha \to \beta$
Left Linear Grammar $A \to Bx/x$	$\alpha \in (V+T)^*$	$\lvert\alpha\rvert \leq \lvert\beta\rvert$	$\alpha \to (V+T)^+$
		$\alpha, \beta \in (V+T)^+$	$\beta \in (V+T)^*$

Regular Grammar

Regular Grammar = Regular Expression = Finite Automata = Regular Language.

Regular Grammar $\qquad x \in \Sigma^*$

Left Linear Grammar	Right Linear Grammar
$A \to Bx/x$	$A \to xB/x$

Linear Grammar-

In any grammar LHS has exactly one Non-terminal and RHS atmost one Non terminal exists is known as linear grammar

A Regular Grammar can be either Left Linear or Right Linear ; but not both, i.e, mixing of both is not allowed.

Both not allowed

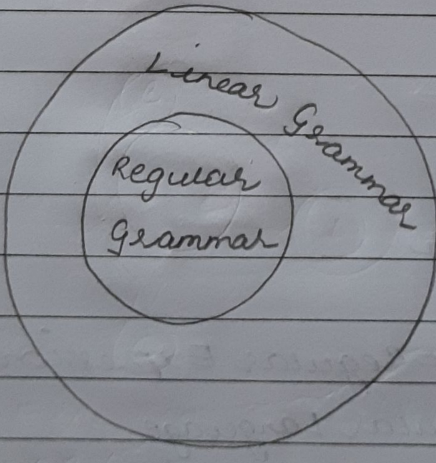
Example ⇒ $S \rightarrow aS / Sb / a$

Linear but not Regular

mixing of LL & RL

⇒ $S \rightarrow aSb / ab$

Linear but not Regular



Every regular grammar is linear but every linear grammar need not be regular

NOTE :-

1) For every regular language regular grammar can be constructed.

2) For every regular language left linear grammar can be constructed.

3) For every regular language right linear grammar can be constructed.

Left Linear = Right Linear

Can be constructed

4) For every left linear grammar right linear grammar can be constructed.

5) For every right linear grammar left linear grammar can be constructed.

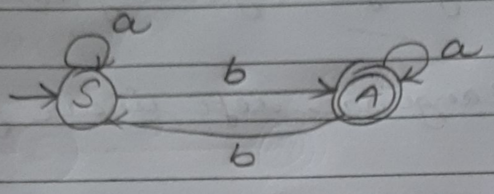
6) For every left linear grammar regular expression can be constructed.

7) For every right linear grammar regular expression can be constructed.

8) For every F.A right linear grammar / left linear grammar can be constructed.

Q. Construct regular grammar for the following finite automata.

(i)



Right linear Grammar

$G = (N, T, P, S)$

Non-Terminal = {States}

N = States

$T = \Sigma$

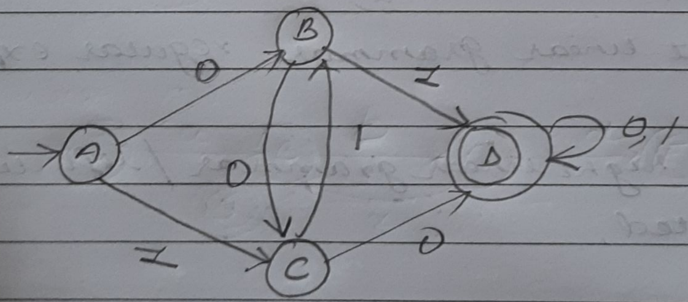
S = {Initial State}

P = transitions/productions

$S \to aS \,/\, bA$

$A \to aA \,/\, bS \,/\, \epsilon$

(ii)



$A \to 0B \,/\, 1C$

9 Productions

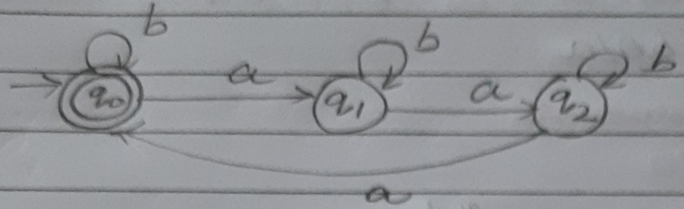
$B \to 1D \,/\, 0C$

$C \to 0D \,/\, 1B$

$D \to 0D \,/\, 1D \,/\, \epsilon$

classmate
Date
Page

(iii)



$q_0 \to bq_0 / aq_1 / \epsilon$

$q_1 \to aq_2 / bq_1$

$q_2 \to aq_0 / bq_2$

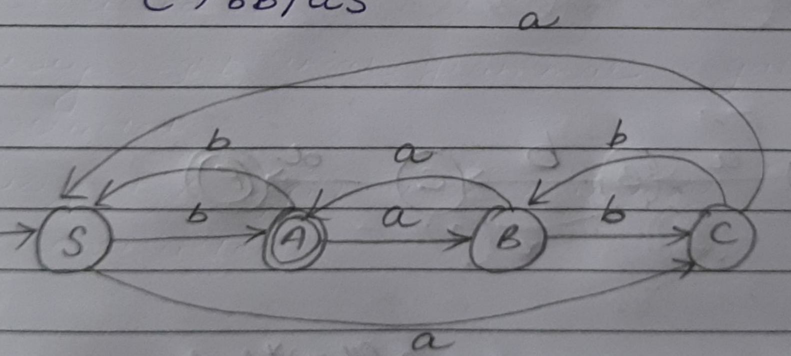
Q. Construct Finite Automata for the following regular grammar

(i) $S \to bA / aC$ Also identify the language

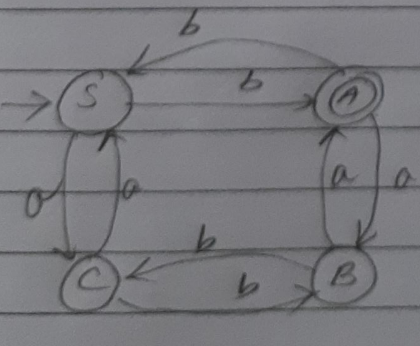
$A \to bS / aB / \epsilon$

$B \to aA / bC$

$C \to bB / aS$



can also be constructed as

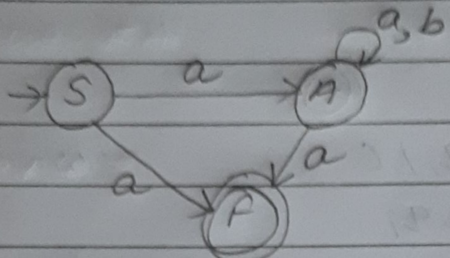


{ odd b's and even a's }

classmate
Date
Page

(ii) $S \rightarrow aA/a$

$A \rightarrow aA/bA/a$



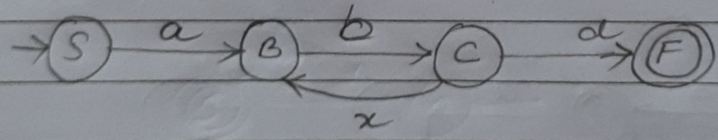
Starting & ending with 'a'.

$a(a+b)^* a + a$ Regular Expression.

(iii) $S \rightarrow aB$

$B \rightarrow bC$

$C \rightarrow xB/d$



Regular Expression = $a(bx)^* bd$

Q. Which production should be ~~able~~ added to given grammar so that resultant grammar generates all strings ending with 0

$S \rightarrow 0A$

$A \rightarrow 0A \,/\, 1B \,/\, \epsilon$

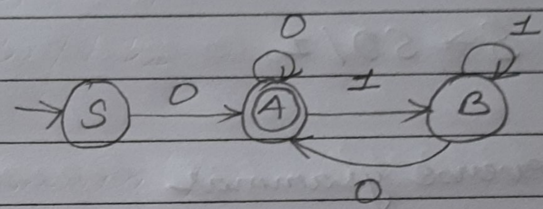
$B \rightarrow 1B$ 0110

a) $B \rightarrow 1S$

✓ b) $B \rightarrow 0A$

c) $S \rightarrow 1C$

d) $B \rightarrow 1B$



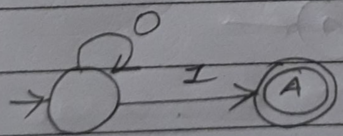
Regular Expression $= 0(0+1)^*0 + 0$

or

$0(0+11^*0)^*$

Q. Construct FA for the following left linear grammar.

$S \rightarrow S0/1 \quad = 10^*$



$= 0^*1$ ×

wrong.

classmate
Date
Page

Left Linear Grammar Construction

1) Reverse grammar
2) Construct FA
3) Reverse FA

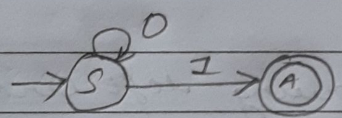
Example →

$S \rightarrow S0 / 1$

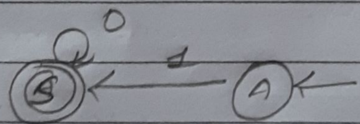
1) Reverse grammar

$S \rightarrow 0S / 1$

2) Construct FA



3) Reverse FA



Q. $S \to aB / bC / bA$

$A \to bB / aC$

$C \to aB / bC / a$

$B \to aS / bA / b$

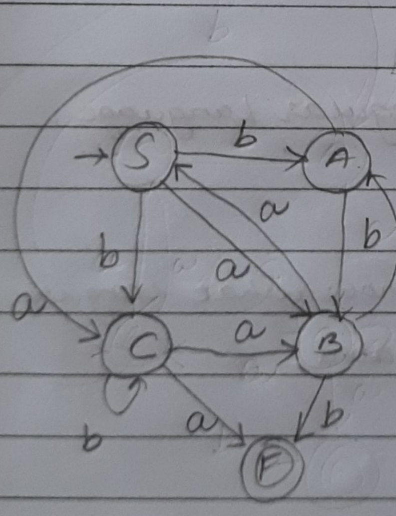
Language of this grammar is

a) $L = \{ x \mid x \in \{a, b\}^*$ Comparison $n_a(x) = 3 n_b(x) \}$

b) $L = \{ x \mid x \in \{a, b\}^*$ Comparison $3 n_a(x) = n_b(x) \}$

c) $L = \{ x \mid x \in \{a, b\}^*$ $n_a(x) = 3 + n_b(x)$ Comparison

d) None



Not required to construct क्योंकि Comparison exist करता है।

L-34

09-07-21

Regularity Problem

1) Regular grammar

$S \to aS/bS/\epsilon$

$(a+b)^*$

2) $S \to aSb/ab$ Context Free Grammar

$\{a^n b^n | n \geq 1\}$ (Non-Regular)

3) $S \to AB$ Context free grammar

$A \to a$

$B \to b$

(Regular)

CFG sometimes generates Regular Language.

\# $S \to AB/BC$

$A \to BAC/BC/a$ Don't know what language is generated.

$B \to CC/b$

$C \to AB/b$

Regularity problem - means checking whether language generated by given CFG is regular or not?

* Regularity problem of CFG is undecidable problem.

* There is no algorithm for Regularity problem.

Q. $G1 \Rightarrow$ $S \to Sab / AB$

$A \to aA / \epsilon$ a^*

$B \to bB / \epsilon$ b^*

Regular $a^*b^*(ab)^*$

$G2 \Rightarrow$ $S \to aA$

$A \to aAB / \epsilon$

$B \to c$ $a\,a^n c^n$

Non-regular

$G3 =$ $S \to aAb / \epsilon$

$A \to c$ $\{acb, \epsilon\}$

Regular

Which of the above grammar generates a regular language?

a) G1 and G2 only
b) G2 only
c) G2 & G3 only
d) G1 and G3 only ✓

Simplification of CFG

1) Eliminate useless variable (Non-terminals)
2) Eliminate unit production
3) Eliminate null Production

* Elimate useless variables -

a) Eliminate non-terminals which are not deriving any string

b) Eliminate non-terminals which are not required for derivation.

Q. Eliminate useless variables for the following grammar?

(i) $S \rightarrow AB/a$ $\quad A\phi = \phi$

$A \rightarrow a$

$\times$ $B \rightarrow bB \rightarrow \phi$

$C \rightarrow a$

$S \rightarrow a$

$\left.\begin{array}{l} A \rightarrow a \\ C \rightarrow a \end{array}\right\} \times$

$S \rightarrow a$

(ii) $S \rightarrow A\overset{\phi}{B}/a$

$A \rightarrow B\overset{\phi}{C}/b$

$\left.\begin{array}{l} B \rightarrow bC/Ba \\ C \rightarrow aB/bC \end{array}\right\} \times$ no string generation $= \phi$

$D \rightarrow b$

$S \to a$ $\quad\quad\quad\quad$ $S \to a$

$A \to b$

$D \to b$

(iii) $S \to AB/\cancel{AC}$ ϕ

$A \to aAb/bAa/a$

$B \to bbA/aaB/AB$

$C \to abCa/aDb$ } useless variables $= \phi$

$D \to bD/aC$

$E \to a$ × Not required

$S \to AB$

$A \to aAb/bAa/a$

$B \to bbA/aaB/AB$

(iv) $S \to A/B$

$H \to 1/\epsilon$

$D \to \epsilon$

$A \to 0E\cancel{A}/OFC$ ϕ ϕ

$B \to 0E\cancel{B}/OFD$ ϕ

$E \to 0E\cancel{E}/OFG$ } × ϕ ϕ

$F \to 0\cancel{E}F/OFH/1$ ϕ

~~$S \to A/B$~~

~~$H \to 1/\epsilon$~~

~~$D \to \epsilon$~~

~~$A \to OFC$~~

~~$B \to OFD$~~

~~$F \to OFH/1$~~

$S \to B$ $\quad\quad$ 7 Productions

$B \to OFD$

$F \to OFH/1$

$D \to \epsilon$

$H \to 1/\epsilon$

* Elimination of Unit Production
($A \to B$)

$S \to A$ $\qquad$ $S \to a$

$A \to B$ $\qquad$ $A \to a$

$B \to C$ $\qquad$ $B \to a$

$C \to D$ $\qquad$ $C \to a$

$D \to a$ $\qquad$ $D \to a$

Now, $S \to a$

Q. Eliminate unit & useless productions.

$S \to Aa / B$

$B \to A / bb$

$A \to a / bc / B$

Eliminate unit

$S \to Aa / a / bc / bb$

$B \to a / bc / bb$ } X

$A \to a / bc / bb$

$S \to Aa / a / bc / bb$

$A \to a / bc / bb$

classmate
Date
Page

* Elimination of Null productions ($A \to \epsilon$)

If ϵ exist, unit productions are not visible & hence useless productions are not visible.

Example -

$S \to aS_1b$
$S_1 \to aS_1b / \epsilon$

$S \to aS_1b / ab$
$S_1 \to aS_1b / ab$

Example -

$S \to AaB$	$S \to AaB / Aa / aB / a$
$A \to aA / \epsilon$	$A \to aA / a$
$B \to bB / \epsilon$	$B \to bB / b$

Order of Elimination

Null $\to$ Unit $\to$ Useless
(1) (2) (3)

Normal Form of CFG

⇒ Normal form - Applying condition on RHS of CFG is known as normal form.

Normal forms.

Chomsky Normal Form	Greibach Normal Form
$A \to BC$	$A \to aV^*$
$A \to a$	

$a \in$ Terminal

Before applying Normalisation, grammar must be simplified.

Q Convert following CFG into CNF grammar.
$\downarrow$ Chomsky Normal Form.

$S \to aSb/ab$

1) Convert total RHS into only non-terminals by assuming terminals as new Non terminals.

2

$S \to ASB/AB$ ⇒ $S \to AX/AB$
$X \to SB$

$A \to a$ $\quad$ $A \to a$

$B \to b$ $\quad$ $B \to b$

2) $A \to \underbrace{B}_{\text{as it is}}\underbrace{CDE}_{X}$ $\quad$ $A \to BX$, $X \to C\underbrace{DE}_{Y}$ ⇒ $A \to BX$, $X \to CY$, $Y \to DE$

Q. Convert the following grammar into CNF grammar.

(i) $S \to bA / aB$

$A \to bAA / aS / a$

$B \to aBB / bS / b$

$S \to B'A / A'B$

$A \to B'AA / A'S / a$

$B \to A'BB / B'S / b$ $\Rightarrow$

$A' \to a$

$B' \to b$

$S \to B'A / A'B$

$A \to B'X / A'S / a$

$X \to AA$

$B \to A'Y / B'S / b$

$Y \to BB$

$A' \to a$

$B' \to b$

(ii) $S \to aAB / Bb$

$A \to a$

$B \to b$

$S \to AAB / BB$ $\Rightarrow$ $S \to AX / BB$

$A \to a$

$B \to b$

$X \to AB$

$A \to a$

$B \to b$

(iii) $S \rightarrow aSa \,/\, bSb \,/\, c$

$S \rightarrow ASA \,/\, BSB \,/\, c$

$A \rightarrow a$

$B \rightarrow b$

$\Rightarrow$

$S \rightarrow AX \,/\, BY \,/\, c$

$X \rightarrow SA$

$Y \rightarrow SB$

$A \rightarrow a$

$B \rightarrow b$

NOTE :

To generate n length string from CNF grammar total no. of productions required is $[2n-1]$.

(No. of steps in derivation).

10-07-21

Decision Properties of CFG

The following problems of CFG are decidable.

1. Membership problem
2. Finiteness problem
3. Emptyness problem

I Membership problem - (Parsing)

Checking String x is generated from given grammar (or) not?

Algorithm used = CYK algorithm

$O(n^3)$

Bottom up parsing algorithm

C → Cocke
Y → Younger } Dynamic programming
K → Kasami

CYK Algorithm

Q. verify the String baaba is a member of the following grammar or not?

S → AB/BC
A → BA/a
B → CC/b
C → AB/a

Substring of a String –

Consecutive Sequence of Symbols over the given String is known as substring.

Example →

String = "TOC"	"GATE"	"DELHI"
ϵ	ϵ	ϵ
T, O, C	G, A, T, E	D, E, L, H, I
TO, OC	GA, AT, T, E	DE, EL, LH, HI
TOC	GAT, ATE	DEL, ELH, LHI
	GATE	DELH, ELHI
		DELHI

$\therefore$ $\boxed{\text{Total number of substrings} = \frac{n(n+1)}{2} + 1}$

Subsequences of a String –

For a string of length n
Total no. of subsequences = 2^n

Q. How many substrings are members of the given grammar?

$S \rightarrow AB / BC$
$A \rightarrow BA / a$
$B \rightarrow CC / b$
$C \rightarrow AB / a$

String = "baaba"
Total substrings = 16
$\frac{5 \times 6}{2} + 1 = 16$

CYK Algorithm

b	a	a	b	a
B	A, C	A, C	B	A, C
A, S	B	S, C	A, S	
ϕ	B	B		
ϕ	S, C, A			
S, C, A				

* For the second row, Multiply B (column 1) with A, C (column 2) and check the RHS of production rules.

BA, BC → S, A and S

AC, AA, CC, CA → CC gives B

AB → S, A

CB

BA → A

BC → S

* For the third row
Top row First column (B) product with First diagonal (B) {Row 2 column 2} and (R-2, +C-1) × Diagonal 2.

B×B and (A, S) × (A, C)

{R-2, C-1} — second diagonal: (A, C)

BB, AA, AC, SA, SC = ϕ

Similarly, (A, C) × (S, C) and B × B

AS AC CS CC BB

CC → B

Similarly, (A, C) × (A, S) and (S, C) × (A, C)

AA, AS, CA, CS and SA, SC, CA, CC

CC → B

n^{th} row $\Rightarrow$ $n-1$ Products

* For the Fourth Row

Row 1 - column-1 X First diagonal of the required cell

and

Row2 - column-1 X Second diagonal

and

Row3 - column-1 X Third diagonal

$\therefore$

(B×B) and (A,S) X (S,C) and ϕ X (B)

BB, AS, AC, SS, SC

$= \phi$.

(A,C) XB and B X (A,S) and B X (A,C)

AB, CB, BA, BS, BA, BC

↓ ↓

S,C, A

* For the fifth row

Row 1 - column-1 X First diagonal of the required cell

⋮

Row 4 - column-1 X Fourth diagonal

B X (S,C,A) and (A,S) X B

BS, BC, BA, AB, SB

↓ ↓

A S,C

classmate
Date
Page

Q. verify the String "baaba" is a member of following grammar or not?

With reference to table of productions Stated before.

Take the String baaba and check the last box of the table if Start symbol 'S' is present then the given String is a member.

Hence, "baaba" is a member

Q. which of the following Strings are members?

1) b

b	a	a	b	a
B	A,C	A,C	B	A,C
A,S	B	S,C	A,S	
ϕ	B	B		
ϕ	S,C,A			
S,C,A				

Read b and jump 1 length diagonal to find S. If S is present then b is a member.

b	a
B	

A,S

No a member

2) ba

Read ba and jump 2 length diagonal to find S. If S is present then ba is a member.

b	a
B	A,C
A,S	

Member

3) baa

Not a member

4) baab

Not a member

5) ab

Yes, a member

6) aaba

Member

Q. How many substrings are members of given grammar?

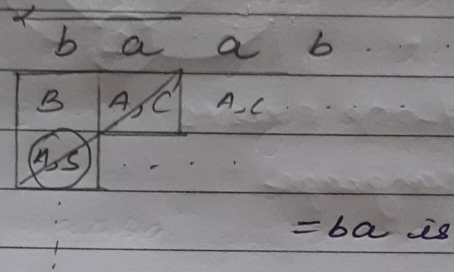
No. of boxes in the table that contain S.
= 5

Q. How many substrings are not-members of the given grammar?

⇒ 16 - 5 = 11

Q. How many different members of the given grammar

Go to the member box and diagonally traverse to read the substring.



= ba is one substring

Similarly other substrings.

⇒ ab, ba, aaba, baaba

∴ Total Number of different substrings = 4

NOTE:-

CYK algorithm is applicable only for Chomsky Normal Form (CNF).

H.W Q. $S \rightarrow AB$

$A \rightarrow BB / a$

$B \rightarrow AB / b$

How many different substrings of the string aabbb are members of given grammar?

ANS = 5

II Finiteness Problem -

Checking whether language generated by the given grammar is finite (or) not.

(Algorithm exist)

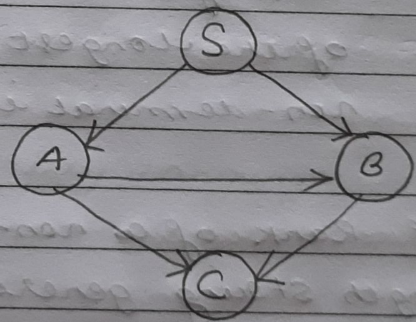
Algorithm -

1) Simplify the grammar
2) Construct CNF grammar
3) Construct CNF graph.
4) If Graph contains loop (or) cycle then the grammar generates infinite language otherwise finite language.

Example -

$S \to AB$
$A \to BC/a$
$B \to CC/b$
$C \to a$

CNF graph



No loop or cycle exists, hence finite

If finite then definitely regular.

Rank(S) = 3
Rank(A) = 2
Rank(B) = 1
Rank(C) = 0

classmate
Date
Page

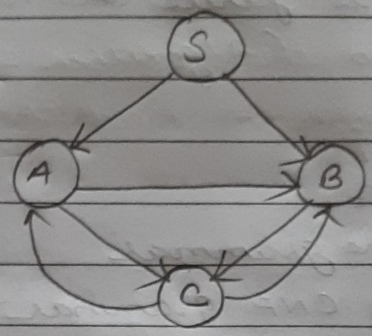
Example -

$S \to AB$

$A \to BC / a$

$B \to CC / a$

$C \to AB / a$



Cycle/Loop exists

Hence, infinite language

NOTE! -

If the language generated by a grammar is finite, then we can define Rank of every Non-terminal.

Rank of a non-terminal is the length of the longest path starting from that non-terminal in CNF graph.

If Rank of a non-terminal is r, then maximum length string generated by that non-terminal is 2^r.

Example -

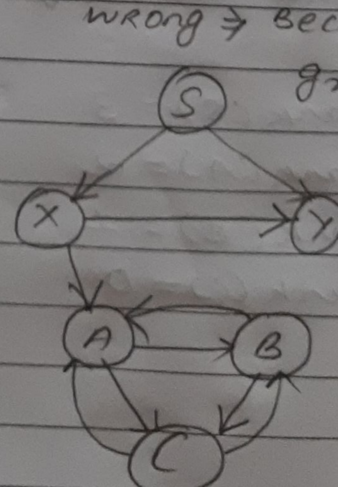
Wrong ⇒ Because grammar is not simplified

$S \to XY$ ϕ

$X \to AX / YY / b$

useless ϕ $\left\{ \begin{array}{l} A \to BC \\ B \to AC \\ C \to AB \end{array} \right.$

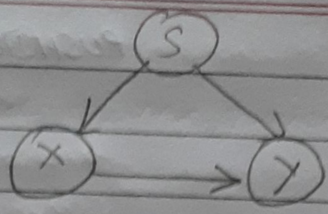
$Y \to b$



$\therefore$ $S \to XY$

$X \to YY/b$

$Y \to b$



No loop/cycle, hence finite

III Emptyness Problem

Checking if the given grammar generates empty language or not?

Algorithm -

1) Eliminate useless variables
2) If the starting symbol is useless then the grammar generates empty language; otherwise non-empty language.

Example -

(i) $S \to AB \to \phi$ $\quad\therefore$ Empty language.

$A \to aA \to \phi$

$B \to b$

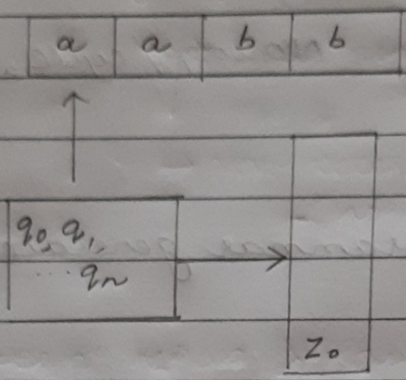
(ii) $S \to aSb / bSa / aS / Sb$

No string generated

$\therefore$ Empty language.

11-07-2021

PUSHDOWN AUTOMATA



Finite Automata + Stack = Pushdown Automata

F.A + Stack = PDA

NOTE :-

Finite Automata fails to accept language in which infinite Comparison exists

Finite Automata has finite memory

Examples -

(i) $L_1 = \{ a^n b^n \mid n \geq 1 \}$ = NotFA

(ii) $L_2 = \{ a^n b^n \mid 1 \leq n \leq 5 \}$ = FA

classmate
Date
Page

Formal Definition of PushDown Automata

$$PDA = (\underbrace{Q, \Sigma, q_0, \delta, F}_{FA}, Z_0, \Gamma)$$

Q : Finite set of States
Σ : Input alphabet
q_0 : Initial State
F : Set of final states
Z_0 : Initial stack element
Γ : Stack alphabet
δ : Transition function

$$\boxed{\delta = Q \times \Sigma \cup \{\epsilon\} \times \Gamma \rightarrow Q \times \Gamma^*}$$

Transition function

Q If the size of the stack in PDA is restricted to 10000 elements then the language accepted by that PDA is
a) Regular language only
b) Finite Language only
c) CFL but not regular
d) Regular but not CFL

PDA accepts
→ Regular
→ CFL

PDA
↓
Without Output
(Language Recogniser)

→ Deterministic Pushdown Automata (DPDA)
→ Non-deterministic Pushdown Automata (NPDA)

Power of NPDA > Power of PDA

Acceptance methods of PDA

- Empty stack
- Final state

Empty Stack

By reading complete String from left to Right end of the String Stack of PDA is empty then given String is accepted and irrelevent of final state.

Final state

By reading complete String from left to right end of the String if PDA enters into final state then the given String is accepted & irrelevent about empty Stack.

Q. Let n_1 is no. of languages accepted by PDA with empty stack. Let n_2 is no. of languages accepted by PDA with final state. Which of the following is true?

√a) $n_1 = n_2$

b) $n_1 > n_2$

c) $n_1 < n_2$

d) Can't say

NOTE! —

1) Number of languages accepted by empty stack PDA and final state PDA method is same in NPDA

2) No. of languages accepted by final state is more than empty stack in DPDA

Default PDA is NPDA

L-39

Representations of PDA

1) Transition diagram
2) Transition notations

* PDAS are practically used in parsers
(Syntax analysis of compilers)

* PDA accepts CFL + Regular languages

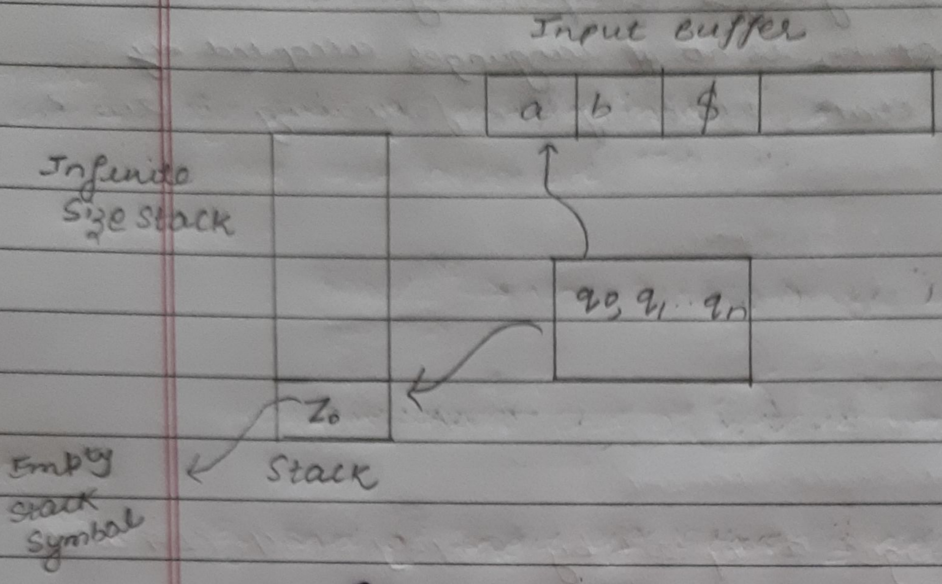
Drawbacks —

PDA fails to accept languages which requires more than 1 memory (Stack)
(More than one comparison exists)

Ex → $\{a^n b^n c^n \mid n \geq 1\}$ ⇒ Not CFL

L-40

12-07-21



Stack + Finite Automata = PDA

Construction of PDA

1) Construct PDA for $L = \{a^n b^n \mid n \geq 1\}$

$$Q \times \Sigma \cup \{\epsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$$

Logic = Push all as onto the stack.
For each b pop an a from the stack.
If at the end of all operations, stack is empty then the language is accepted

I/P symbol, Top of the stack, push

I $\quad \delta(q_0, a, Z_0) = (q_0, aZ_0)$

$Q \quad \Sigma \quad \Gamma \quad Q \quad \Gamma$

Push

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

Top of the stack

III $\delta(q_0, b, a) = (q_1, \epsilon)$

POP operation
ऐसे दिखायेंगे।

POP

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

V $\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$ } Accepted

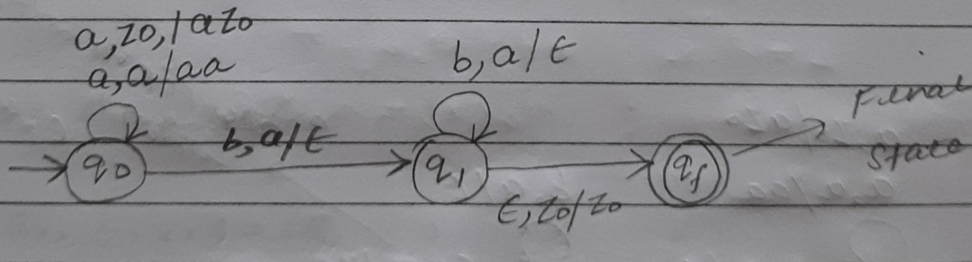
Empty stack method
Z_0 को भी POP करदो।

OR

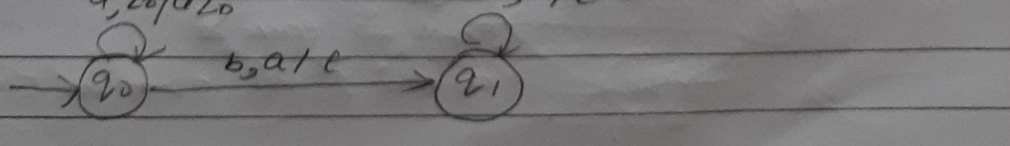
$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$

Final state method.

Transition diagram Notation of the above problem



OR



These are DPDA

Missing transitions are also Deterministic.

E.g q_1 पर a transition नहीं है।

Acceptance methods

Final State 1. By end of input if automata halts in final State then accepted.

Halts → Non-Final → Rejected

Halts without end of input → Rejected

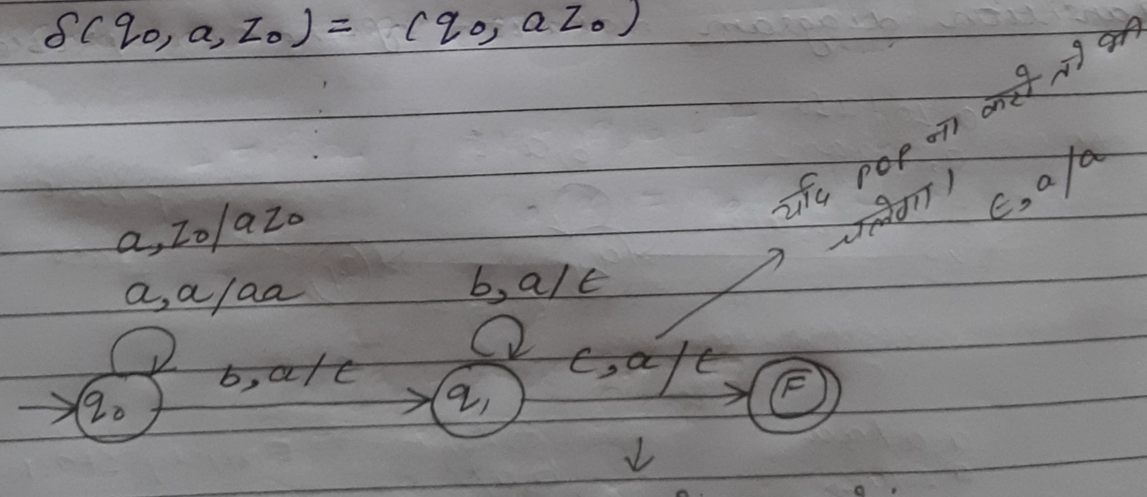
Empty Stack 2. By end of input if stack is not empty then rejected, otherwise accepted.

Q. Construct PDA for $L = \{ a^n b^m \mid n > m, \; n, m \geq 1 \}$

If For every b, a is popped & at the end a remains in the stack, accept

Transitions →

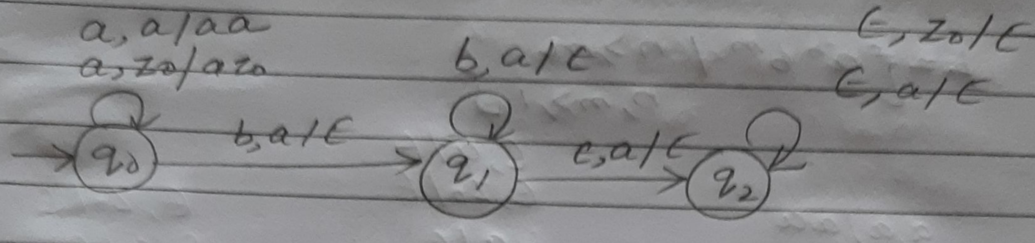
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$



End में stack में a बचे हैं तो n>m is Fullfilled

classmate
Date
Page

using empty stack.



Transition states (Empty Stack)

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ — a's Push

↓ I/P ↓ Top of stack ↓ New modified stack

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

$\delta(q_0, b, a) = (q_1, \epsilon)$ — b's POP

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

$\delta(q_1, \epsilon, a) = (q_2, \epsilon)$ — Extra as are popped

↓ I/P समाप्त ↳ Stack Non Empty

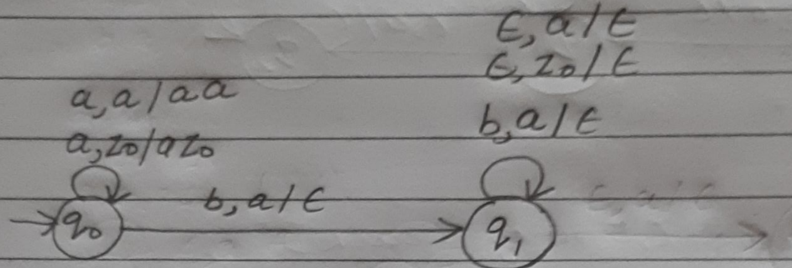
$\delta(q_2, \epsilon, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, Z_0) = (q_2, \epsilon)$ → Accept

classmate
Date
Page

Q. Construct PDA for

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



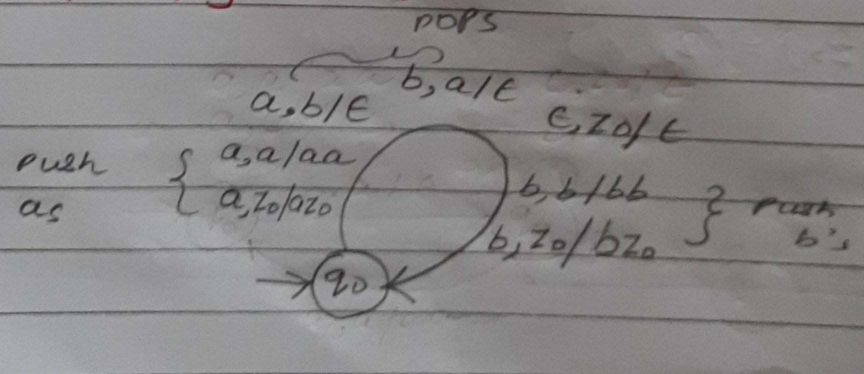
[Emptystack Method]

$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$
$$\delta(q_0, a, a) = (q_0, aa)$$
$$\delta(q_0, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, a) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$$

* Q. Construct PDA for

$$L = \{ x \mid x \in \{a, b\}^*,\ n_a(x) = n_b(x) \}$$

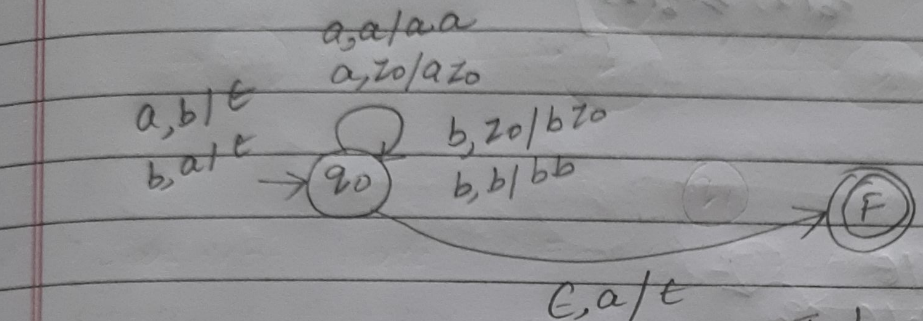
(By Empty stack)



Logic → Same Symbols = Push
(I/P & Stack)

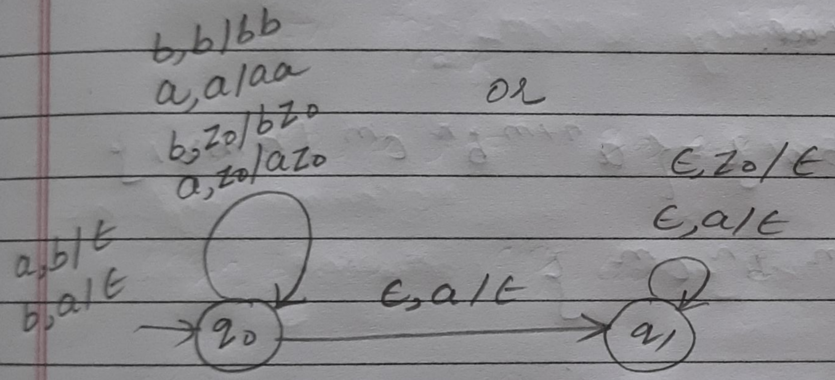
Different Symbols = POP

Q. Construct PDA for $L = \{ x \mid x \in \{a,b\}^*, \; n_a(x) > n_b(x) \}$



अंत मे a रहे गया ।

or



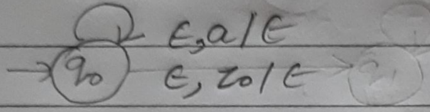
L-42

13-07-21

Q. Construct PDA for the following language

$$L = \{ x \mid x \in \{a,b\}^*,\ n_a(x) \geq n_b(x) \}$$

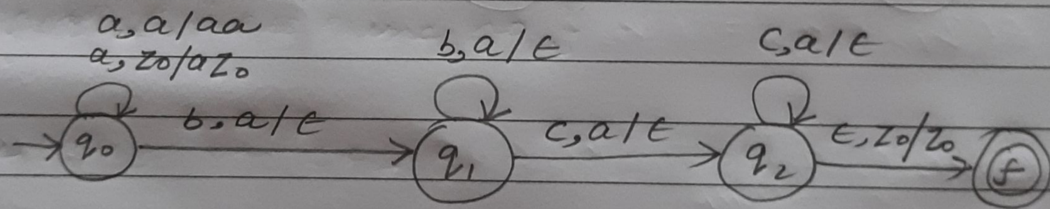
a,b/ε
b,a/ε
b,b/bb
a,a/aa
b,Z0/bZ0
a,Z0/aZ0



Empty stack method.

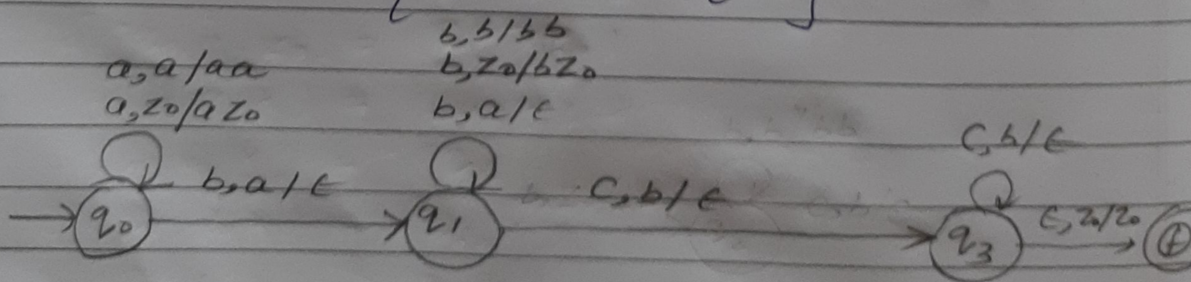
Q. Construct PDA for $L = \{ a^{n+m} b^n c^m \mid n,m \geq 1 \}$

$$\{ a^n a^m b^n c^m \}$$



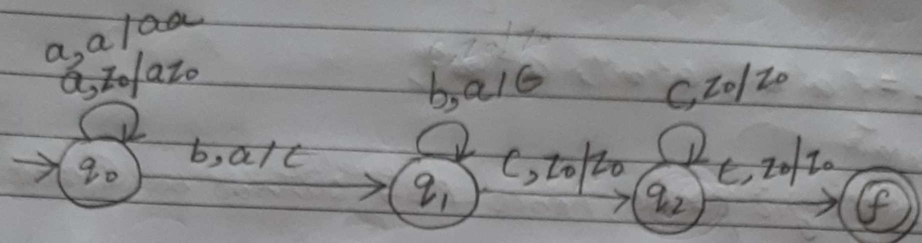
Q. Construct PDA for $L = \{ a^n b^{n+m} c^m \mid n,m \geq 1 \}$

$$\{ a^n b^n b^m c^m \}$$

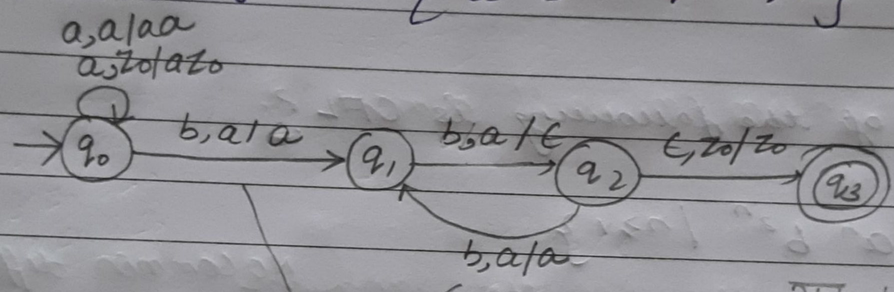


9 transitions

Q. Construct PDA for $L = \{ a^n b^n c^m \mid n, m \geq 1 \}$



Q. Contruct PDA for $L = \{ a^n b^{2n} \mid n \geq 1 \}$

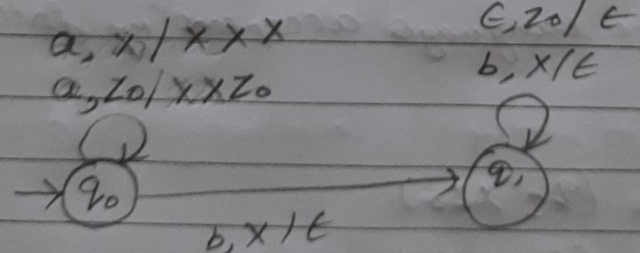


एक b को ignore करो
दूसरे b पर pop करदो।

1 ignore करो

Q. Construct PDA for $L = \{ a^n b^{2n} \mid n \geq 1 \}$

हर एक a के लिये 2 a
push करदो।



Q. Construct PDA for $L = \{a^n b^{n^2} \mid n \geq 1\}$

$L = \{a^1 b^1, a^2 b^4, a^3 b^9, \ldots\}$

No logic exist, hence PDA cannot be Created.

PDA Not possible
(NON-CFL)

Q. Which of the following is CFL?

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

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

c) $L = \{a^p b^p \mid p \text{ is Prime no}\}$

No common difference btn these infinite series

✓ d) None.

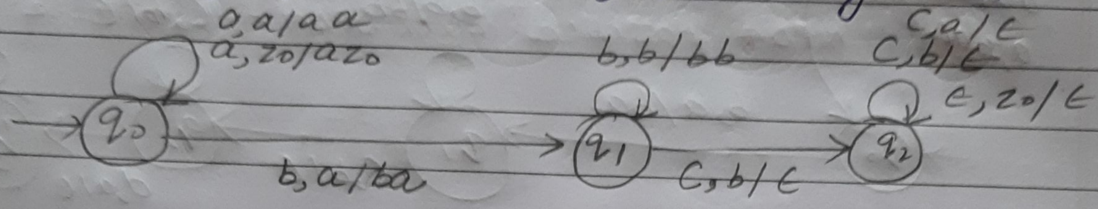
Q. Identify the language accepted by following PDA?

$\delta(q_0, a, z_0) = (q_0, xz_0)$
$\delta(q_0, a, x) = (q_0, xx)$
$\delta(q_0, b, x) = (q_1, \epsilon)$
$\delta(q_1, b, x) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, x) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, z_0) = (q, \epsilon)$.

classmate
Date
Page

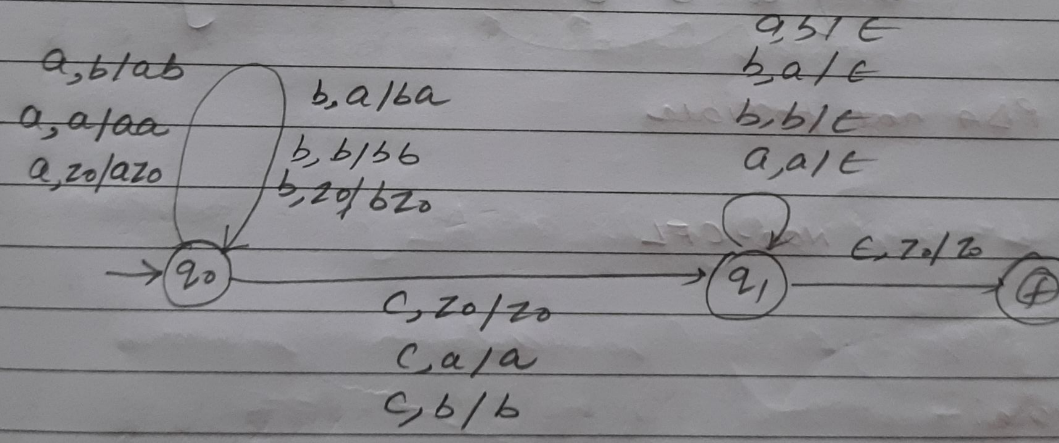
Language ⇒ $\{ a^n b^m , n \geq m \}$
$n, m \geq 1$

Q. Identify language accepted by following PDA.



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

Q. Identify language accepted by following PDA?

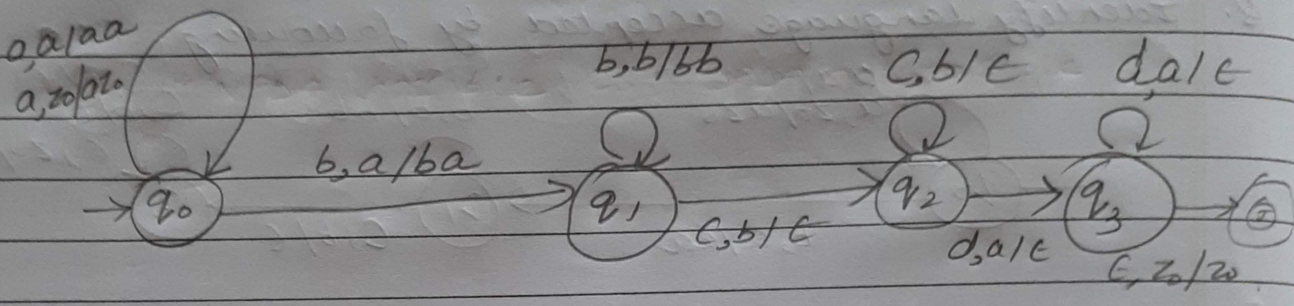


$$L = \{ w_1 c w_2 \mid |w_1| = |w_2| \quad w_1, w_2 \in (a+b)^* \}$$

(Push) (Skip) (POP)

classmate
Date
Page

Q. Identify language accepted by following PDA?



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

Q. Construct PDA for the language
$L = \{ a^n b^m c^n d^m \mid n, m \geq 1 \}$

PDA not possible

Non-CFL

Note:-

$L_1 = \{ a^n b^m c^m d^n \mid n, m \geq 1 \}$ CFL

$L_2 = \{ a^n b^m c^n d^m \mid n, m \geq 1 \}$ Non-CFL

Q. Which of the following is CFL?

(a) $L = \{a^n b^m c^n d^m \mid n, m \geq 1\}$ 2 stacks required.

(b) $L = \{a^n b^n c^n \mid n \geq 1\}$ CSL क्योंकि 2 comparisons

(c) $L = \{a^n b^n c^n d^n \mid n \geq 1\}$ 2 comparisons.

✓ (d) None

Q. Which of the following is Non-CFL?

(a) $L = \{a^n b^n c^m d^m \mid n, m \geq 1\}$ CFL

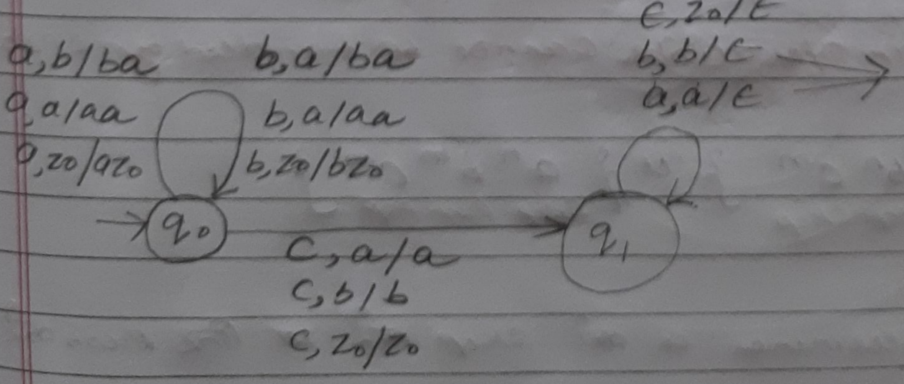
(b) $L = \{a^n b^m c^m d^n \mid n, m \geq 1\}$ CFL

✓ (c) $L = \{a^n b^n c^{2n} \mid n \geq 1\}$ 2 comparisons
2 stacks

(d) None

Q. Construct PDA for $L = \{ w c w^R \mid w \in \{a, b\}^* \}$

*



Note :-
क्योंकि w^R
है तो same symbol
पर ही pop करेंगे।

Q. Construct PDA for the language.

$$L = \{ ww^R \mid w \in \{a,b\}^* \}$$

NPDA possible but DPDA not possible.

NOTE:-

DPDA fails when:-

abb bba

Confusion whether to push or pop

(cannot determine)

Go for NPDA

$$L = \{a^n b^{2n}\} \cup \{a^n b^{3n}\} \Rightarrow \text{NPDA}$$

$\delta(q_0, a, z_0) = (q_0, aaz_0)$ or $(q_0, aaaz_0)$

Confusion 2 a push करें या फिर 3 a.

Confusion in operations cannot be handled by DPDA

$$L = \{\underbrace{a^n}_{\text{push}} \underbrace{b^n}_{\text{pop}} \underbrace{c^m}_{\text{skip}}\} \cup \{\underbrace{a^n}_{\text{skip}} \underbrace{b^m}_{\text{push}} \underbrace{c^m}_{\text{pop}}\}$$

Cannot determine a को skip करें या फिर push करें

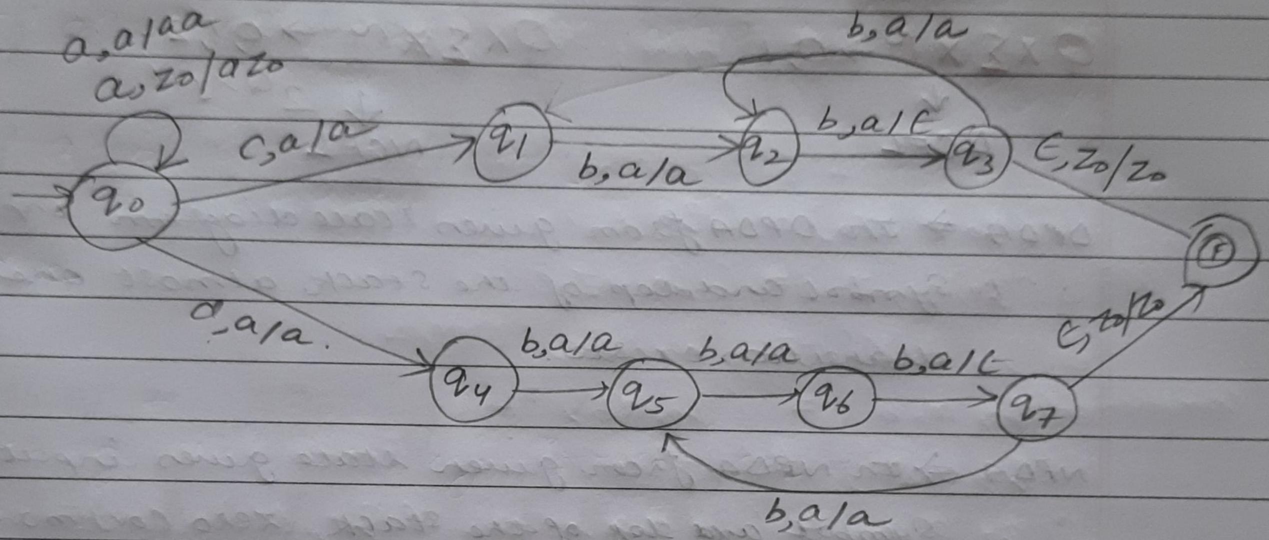
DPDA not possible

Different δ's

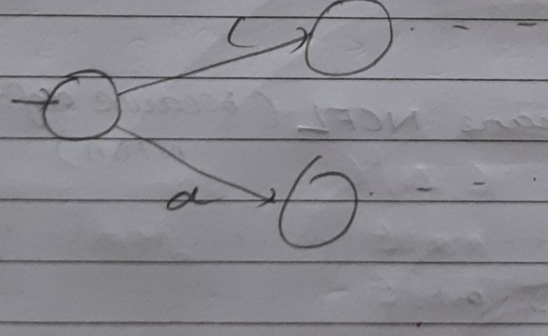
$L = \{ a^n c b^{2n} \} \cup \{ a^n d b^{3n} \}$

Push skip alternate skip/pop — Push skip 2 skip 1 pop.

DPDA POSSIBLE



$L = \{ c a^n b^{2n} \} \cup \{ d a^n b^{3n} \}$



DPDA possible

classmate
Date
Page

1-07-21

PDA

↙ ↘

DPDA NPDA

$Q \times \Sigma \times \Gamma \to 0,1$ $Q \times \Sigma * \Gamma \to 0,1,2 \ldots$

↓ $2^{Q \times \Gamma^*}$

DCFL ↓

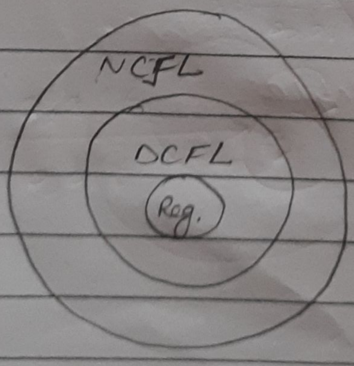
NCFL

DPDA ⇒ In DPDA from given state diagram input symbol and top of the stack atmost one transition exists.

NPDA ⇒ In NPDA from given state given input symbol and top of the stack, zero (or) more transitions exist.

* Every DPDA is NPDA, but ~~not~~ every NPDA need not be a DPDA

* By Default CFL means NCFL (because default PDA is NPDA)



Q. which of the following is CFL but not DCFL?

a) $L = \{ w\#w^R \mid w \in \{a,b\}^* \}$ Deterministic

b) $L = \{ a^n c b^{2n} \} \cup \{ a^n d b^{3n} \}$ deterministic

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

Push Pop Push Pop

✓ d) None.

Q. which of the following is DCFL?

(a) $L = \{ a^n b^{2n} \} \cup \{ a^n b^{3n} \}$ confusion how many bs to be ignored at a time

(b) $L = \{ ww^R \mid w \in \{a,b\}^* \}$ Non deterministic

(c) $L = \{ a^n b^m c^k \mid n = m \text{ (or) } m = k \}$ Non-deterministic

✓ (d) None.

Q. which of the following is DCFL?

(a) $L = \{ a^p b^q c^r \mid p < q < r \}$ 2 comparisons

(b) $L = \{ a^p b^q c^r \mid p < q \text{ (or) } (q < r) \}$ NPDA possible

✓ (c) $L = \{ a^p b^q c^r \mid q = p + r \}$

Push a Pop b Push P, POP C (remaining)

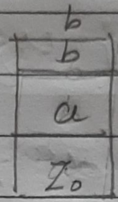
(d) none

classmate
Date
Page

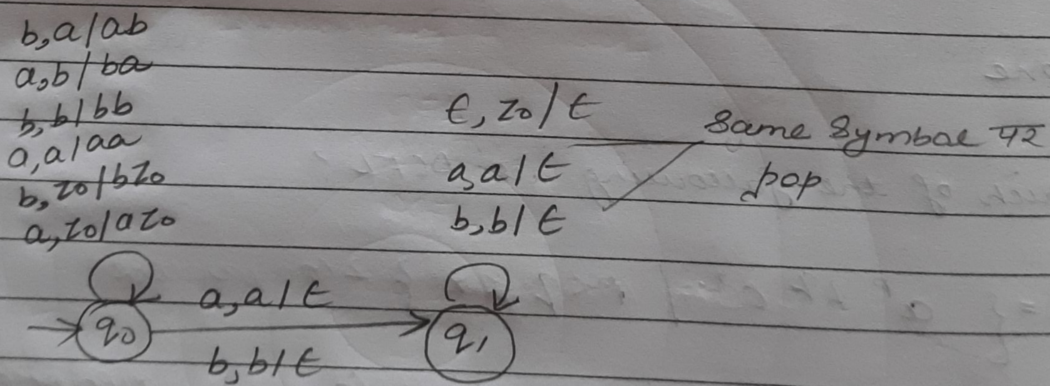
For every non-deterministic nature
complexity = $O(2^n)$

★ Q. Construct NPDA for $L = \{ WW^R \mid W \in \{a,b\}^* \}$

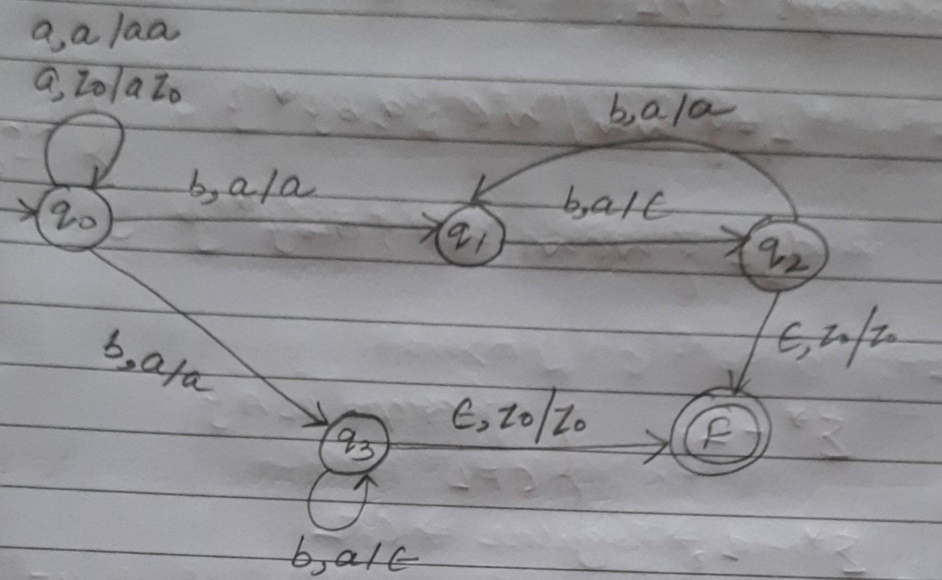
abb bba
w w^R



$\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, a, b) = (q_0, ab)$
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, a) = (q_0, aa)$ or (q_1, ϵ)
$\delta(q_0, b, b) = (q_0, bb)$ or (q_1, ϵ)
$\left.\begin{array}{l} \delta_1(q_1, a, a) = (q_1, \epsilon) \\ \delta_1(q_1, b, b) = (q_1, \epsilon) \end{array}\right\} W^R$
$\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$



Q. Construct PDA for $L = \{a^n b^n \mid n \geq 1\} \cup \{a^n b^{2n} \mid n \geq 1\}$



Q. Construct PDA for $L = \{wcw \mid w \in \{a,b\}^*\}$

Non CFL

PDA not possible

abab c abab

Q. Construct PDA for $L = \{ww \mid w \in \{a,b\}^*\}$

$L = \{\epsilon, aa, abab, baba, abaaba\}$

Non-CFL

PDA not possible.

classmate
Date
Page

NOTE :-

The following languages non-CFL but their complement is always CFL.

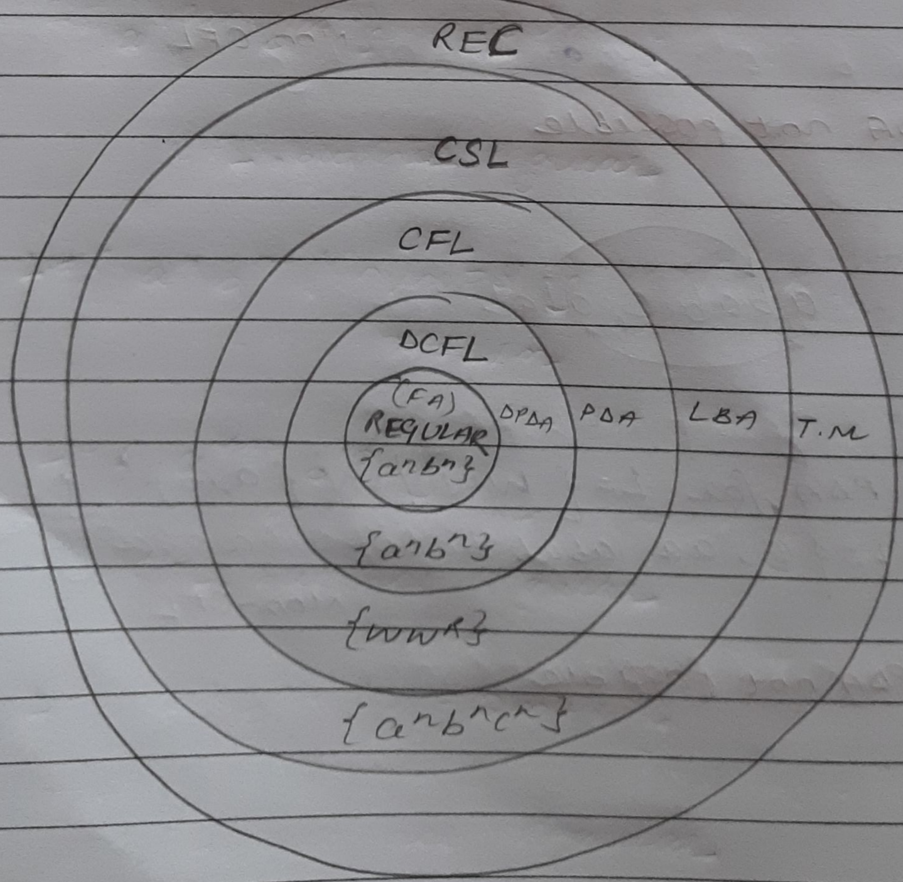
$$L_1 = \{ ww \mid w \in \{a,b\}^* \}$$
$$L_2 = \{ wcw \mid w \in \{a,b\}^* \}$$

$\left.\begin{matrix} L_1 \\ L_2 \end{matrix}\right\}$ Non-CFL.

Complements $\left[\begin{matrix} \Sigma^* - L_1 \\ \Sigma^* - L_2 \end{matrix}\right\}$ CFL $\{CFG = PDA\}$

L-45

*



Chomsky Hierarchy

CFL Detection

Q. Which of the following is

a) CFL and Regular
b) CFL but not Regular
c) Non CFL

Points to Remember

1) Every finite Language is regular and CFL.
2) Every infinite language which requires more than 1 stack is non-CFL (more than 1 comparison exists)
3) All palindrome languages are CFL.
4) If language formed over one symbol there is no difference between CFL and regular

   [A.P = Regular] [Non-AP = Non-Regular].
   ↓ Common difference   Non-CFL

5) Union of two CFL is always CFL
6) If, any infinite language having one comparison but not having stack property then non-CFL [not satisfying LIFO]

   E.g $\{ww \mid w \in \{a,b\}^*\}$

1) $L = \{a^n b^n c^n \mid n \leq 10\}$

Finite $\Rightarrow$ Regular $\Rightarrow$ CFL

2) $L = \{a^n b^m c^m \mid n \neq m\}$

2 comparisons, Non-CFL

3) $L = \left\{ a^n b^m c^n \;\middle|\; \begin{array}{l} n > m \\ n, m \leq 1000 \end{array} \right\}$

Finite $\Rightarrow$ Regular $\Rightarrow$ CFL

4) $L = \{a^n b^m \mid n - m = 4\}$

$n = 4 + m$

$\{a^{4+m} b^m\}$

CFL, but not regular

5) $L = \{a^n b^m \mid n/m = 4\}$

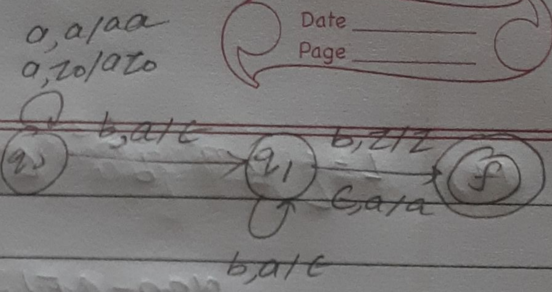
$n = 4m$

$L = \{a^{4m} b^m\}$

CFL, but not Regular

6) $L = \{a^n b^m \mid n = 2m + 1\}$

CFL, but not Regular



7) $L=\{a^n b^m \mid n \neq m\}$

CFL, but not regular

8) $L=\{a^n b^m \mid n \neq 2m\}$

CFL, but not regular

9) $L=\{a^n b^m \mid n = m^2\}$

Non-CFL

10) $L=\{a^{n!} b^{n!} \mid n \geq 1\}$

Non-CFL

11) $L=\{a^n b^m \mid n \leq m^2$

Non-CFL

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

CFL (DCFL) but not regular

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

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

Non-CFL

classmate
Date
Page

14) $L = \{ a^{m^2} b^{n^3} c^{k^3} \mid n, m, k \geq 1 \}$

Non-CFL

15) $L = \{ a^{3n} b^{5k} c^{25l} \mid n, k, l \geq 1 \}$

regular ⇒ CFL

16) $L = \{ a^i b^j c^k \mid j = i + k \}$

CFL, but not regular

17) $L = \{ a^i b^j c^k \mid i > j \text{ (or) } j < k \}$ NPDA

CFL ∪ CFL = CFL

CFL, but not regular

18) $L = \{ a^i b^j c^k \mid i > j > k \}$

2 comparisons

Non-CFL

19) $L = \{ a^i b^j c^k \mid j = \max(i, k) \}$

2 comparisons also FIFO not followed

Non-CFL

20) $L = \{ a^i b^j c^k \mid j = i^2 + k^2 \}$

Non-CFL

21) $L = \{ a^i b^j c^k d^l \mid i = l \text{ \& } j = k \}$

CFL, but not regular

22) $L = \{ a^i b^j c^k d^l \mid i = k \text{ and } j = l \}$

Non-CFL

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

CFL $\cup$ CFL = CFL

CFL, but not regular

24) $L = \{ a^i b^j c^k d^l \mid i = 2k \text{ or } j \neq 5l \}$

CFL $\cup$ CFL = CFL

CFL, but not regular

25) $L = \{ a^i b^j c^k d^l \mid i + j = k + l \}$

CFL, but not regular

26) $L = \{ a^i b^j c^k d^l \mid i = 4l \text{ and } j = 3k \}$

CFL, but not regular

27) $L = \{ a^i b^j \mid (i+j) \bmod 5 = 0 \}$

length % 5

Regular $\Rightarrow$ CFL

28) $L = \{ a^{2^n} \mid n \geq 1 \}$

No common difference

Hence, non Regular $\Rightarrow$ Non-CFL

29) $L = \{ a^{n^2} \mid n \geq 1 \}$

Non Regular and non-CFL

30) $L = \{ 1^{2n+1} \mid n \geq 1 \}$

Regular and CFL

31) $L = \{ a^p \mid p \text{ is Prime no.} \}$

Non-Regular and Non-CFL

32) $L = \{ a^k \mid k \text{ is odd no.} \}$

Regular $\Rightarrow$ CFL

33) $L = \{ wxw \mid w \in \{a,b\}^* \}$

Non-CFL

34) $L = \{ wxw \mid w, x \in \{a,b\}^* \}$

Regular $\Rightarrow$ CFL

35) $L = \{ ww^R x \mid w, x \in \{a,b\}^+ \}$

CFL, but not regular

36) $L = \{ \Sigma^* - \{ ww \mid w \in \{a,b\}^* \} \}$

CFL, but not regular

37) $L = \{ ww^R w \mid w \in \{a, b\}^* \}$

2 comparisons ⇒ Non-CFL

38) $L = \{ ww^R ww^R \mid w \in \{a, b\}^* \}$

More than 1 comparisons ⇒ Non-CFL

39) $L = \{ wwww \mid w \in \{a, b\} \}$

Finite हो गया है।

Regular ⇒ CFL

40) $L = \{ x \mid x \in \{a, b, c\}^*$
$n_a(x) = n_b(x) = n_c(x) \}$

Multiple Comparisons ⇒ Non-CFL

41) $L = \{ x \mid x \in \{a, b\}^*$
$n_b(x) = n_a(x) + n_c(x) \}$

CFL

42) $L = \{ x \mid x \in \{a, b, c\}^*$
$n_a(x) = n_b^2(x) + n_c^2(x) \}$

Non-CFL

43) $L = \{ x \mid x \in \{a, b\}^*$
$n_a(x) \bmod 5 = 0$ and $n_b(a) \bmod 4 = 0 \}$

Reg ∩ Reg = Reg

Regular ⇒ CFL

44) $L = \{ a^n b^{2n} c^{3n} / n \geq \}$

Regular $\Rightarrow$ CFL

45) $L = \{ a^n b^n c a^m b^m / n, m \geq 0 \}$

CFL

46) $L = \{ a^n b^m c^k / n \neq m \text{ or } m \neq k \}$

CFL $\cup$ CFL = CFL

CFL

47) Set of all odd length palindromes strings of Hindi alphabet

CFL

48) Set of all even length palindrome string of Japanese alphabet

CFL

49) Set of all balanced Paranthesis

CFL

50) set of all lexical errors produced by compiler -

Regular $\Rightarrow$ CFL

classmate
Date
Page

Q. Consider the following languages:-

$L_1 = \{ a^p \mid p \text{ is a prime number} \}$

$L_2 = \{ a^n b^m c^{2m} \mid n \geq 0, m \geq 0 \}$

$L_3 = \{ a^n b^n c^{2n} \mid n \geq 0 \}$

$L_4 = \{ a^n b^n \mid n \geq 1 \}$

Which of the following are correct?

I) L_1 is context-free but not regular
II) L_2 is not context-free
III) L_3 is not context-free but Recursive
IV) L_4 is determiniristic context-Free.

a) I, II & IV only
b) II and III only
c) I and IV only
✓ d) III and IV only

classmate
Date
Page

Q. Let $L_1 = \{ 0^{n+m}\, 1^{n}\, 0^{m} \mid n, m \geq 0 \}$ CFL

push pop pop

$1^{n+m} > 0^{m}$

$L_2 = \{ 0^{n+m}\, 1^{n+m}\, 0^{m} \mid n, m \geq 0 \}$ = Non-CFL

2 comparison

$L_3 = \{ 0^{n+m}\, 1^{n+m}\, 0^{n+m} \mid n, m \geq 0 \}$ Non-CFL

2 comparison

Which of these languages are NOT context free?

a) L_1 only
b) L_3 only
c) L_1 and L_2
✓ d) L_2 and L_3

L-46

15-07-21

Q. Which of the following is non-CFL?

a) Set of all palindrome strings of Hindi language
b) set of all Palindromes strings of English language.
c) Set of all odd length Palindrome strings of Telugu language
✓ d) None

Q. Which of the following set are context free & which or not?

a) $\{a^n b^m c^k \mid n,m,k \geq 1 \text{ and } (2n = 3k) \text{ or } (5k = 7m)\}$

$\text{CFL} \text{ or } \text{CFL}$
$(\cup)$
$= \text{CFL}$

CFL

b) $\{a^n b^m c^k \mid n,m,k \geq 1 \text{ and } (2n = 3k) \text{ and } (5k = 7m)\}$

$\text{CFL} \cap \text{CFL}$

Non-CFL

c) $\{a^n b^m c^k \mid n,m,k \geq 1 \text{ and } (n \neq 3k) \text{ and } (n \neq 5k)\}$

Non-CFL

d) $\{a^n b^m c^k \mid n,m,k \geq 1 \text{ and } (n \neq 3k) \text{ or } (n \neq 5k)\}$

CFL

e) $\{a^n b^m c^k \mid n,m,k \geq 1 \text{ and } n+k = m\}$

CFL (also DCFL)

f) $\{a^i b^j c^k d^l \mid i,j,k,l \geq 1,\ i = j \cdot k = l\}$

CFL

g) $\{a^i b^j c^k d^l \mid i,j,k,l \geq 1,\ i = k \cdot j = l)\}$

Non-CFL

h) $\{a^i b^j c^k d^l \mid i,j,k,l \geq 1,\ i=l,\ j=k\}$

CFL (also DCFL)

Q. Say whether the following sets are
(i) Regular
(ii) Context Free but not Regular
(iii) Not Context Free

a) $\{x \in \{a,b,c\}^* \mid \#a(x) = \#b(x) = \#c(x)\}$

Not CFL

b) $\{a^j \mid j \text{ is a power of } 2\}$

Not CFL No common difference.

c) $\{x \in \{0,1\}^* \mid x$ represents a power of 2 in binary$\}$

Regular

d) $L(a^* b^* c^*)$

Regular

e) The set of all balanced strings of parantheses of 3 types,
( ) [ ] { }

CFL but not Regular

classmate
Date
Page

(f) $\{a^n b^m \mid n \neq m\}$

CFL but not regular

(g) $\{a^n b^m c^k d^l \mid 2n = 3k \text{ or } 5m = 7l\}$

CFL but not regular

(h) $\{a^n b^m c^k d^l \mid 2n = 3k \text{ and } 5m = 7l\}$

Non-CFL

(i) $\{a^n b^m c^k d^l \mid 2n = 3m \text{ and } 5k = 7l\}$

CFL but not Reg.

(j) $\{a^n b^m c^k d^l \mid 2n = 3l \text{ and } 5k = 7m\}$

CFL but not Reg.

(k) $\{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i > j \text{ and } j > k\}$

CFL ∩ CFL

Non-CFL

(l) $\{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } (i > j \text{ or } j > k)\}$

CFL but not regular

(m) $\{x \in \{a, b\}^* \mid \#a(x) > \#b(x)\}$

CFL, but not regular

n) $\{a^m b^n \mid m,n \geq 0,\ 5m+3n = 24\}$

Regular

o) $\{a^m b^n \mid m,n \geq 0,\ 5m-3n = 24\}$

CFL

Q. Which of the following languages over $\{a,b,c\}$ is accepted by DPDA?

✓ a) $\{wcw^R \mid w \in \{a,b\}^*\}$

b) $\{ww^R \mid w \in \{a,b,c\}^*\}$

c) $\{a^n b^n c^n \mid n \geq 0\}$

d) $\{w \mid w$ is a palindrome over $\{a,b,c\}\}$ CFL but not DCFL

Q. Consider the languages

$L_1 = \{0^i 1^j \mid i \neq j\}$

$L_2 = \{0^i 1^j \mid i = j\}$

$L_3 = \{0^i 1^j \mid 2j+1\}$

$L_4 = \{0^i 1^j \mid i \neq 2j\}$

Which of the above languages are/is Context Free?

ALL

classmate
Date
Page

$n<13$ or $n>13$

Q. $L = \{ a^n b^n \mid n \geq 0, n \neq 13 \}$ is

Reg $\cup$ DCFL = DCFL

✓ a) DCFL

b) CFL but not DCFL

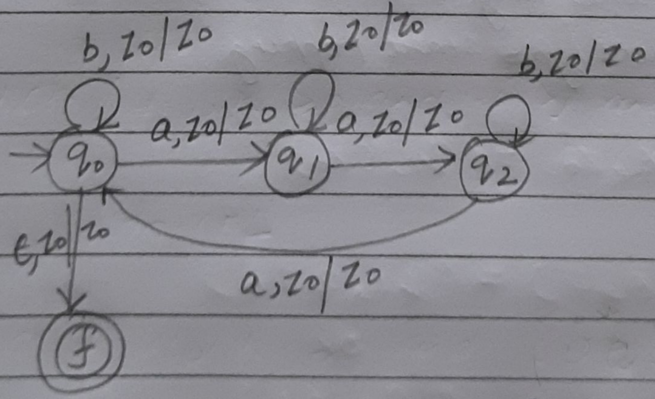
c) Non CFL

d) Regular

L-47

Q.
$$\delta(q_0, a, Z_0) = (q_1, Z_0)$$
$$\delta(q_0, b, Z_0) = (q_0, Z_0)$$
$$\delta(q_1, a, Z_0) = (q_2, Z_0)$$
$$\delta(q_1, b, Z_0) = (q_1, Z_0)$$
$$\delta(q_2, a, Z_0) = (q_0, Z_0)$$
$$\delta(q_2, b, Z_0) = (q_2, Z_0)$$
$$\delta(q_0, \epsilon, Z_0) = (q_F, Z_0)$$

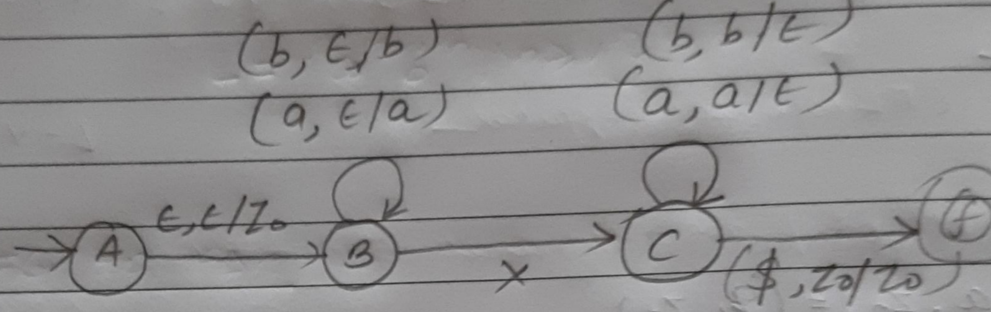
Language accepted by the given PDA ?



Number of as divisible by 3.

classmate
Date
Page

Q. The following machine is designed with PDA acceptance by final state mechanism to accept the language L where all strings of L are odd length palindromes.



What are the transition at X to accept L?

a) (a, a/ε), (b, b/b)
b) (a, a/ε), (b, b/ε)
c) (a, ε/ε), (b, ε/ε) Skip
d) None of these.

X ⇒ ignore any a or b to reach C where you start popping

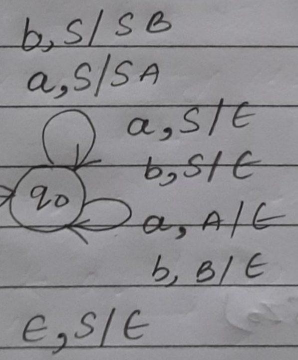
classmate
Date
Page

Q. What is the language accepted by following transitions of PDA which uses acceptance by empty stack.

$\delta(q_0, a, S) = (q_0, SA)$
$\delta(q_0, b, S) = (q_0, SB)$
$\delta(q_0, a, S) = (q_0, \epsilon)$
$\delta(q_0, b, S) = (q_0, \epsilon)$
$\delta(q_0, \epsilon, S) = (q_0, \epsilon)$
$\delta(q_0, a, A) = (q_0, \epsilon)$
$\delta(q_0, b, B) = (q_0, \epsilon)$

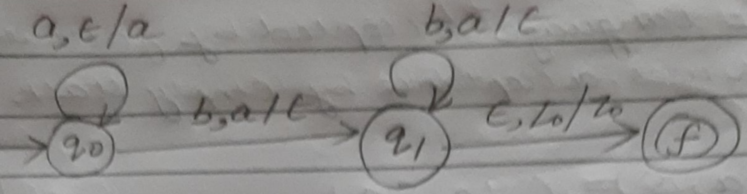
✓a) set of all palindromes
b) set of all odd length palindromes.
c) set of all even length palindromes
d) set of equal no. of as & bs

$S \rightarrow aSA / bSB / a / b / \epsilon$
$A \rightarrow a$
$B \rightarrow b$



$S \rightarrow asa / bsb / a / b / \epsilon$

Q.



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

CFG to PDA

1) $\delta(q_0, \epsilon, Z_0) = (q_0, SZ_0)$ For any Grammar $A \to \alpha$

Replace

2) $\delta(q_1, \epsilon, A) = (q_0, \alpha)$

$\delta(q_1, a, a) = (q_1, \epsilon)$ LL(1) Parsing

pop

3) $\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$

Example :-

$S \to aSb / ab$

1) $\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

2) $\delta(q_1, \epsilon, S) = (q_1, aSb)$

or

(q_1, ab)

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

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

3) $\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$

classmate

Date

Page

* CFG = PDA algorithm always results in NPDA only (DPDA may or may not exist).

LL(1) algorithm ⇒ DPDA

↓

Table makes it Deterministic

Q. Construct PDA for
$$\begin{cases} S \rightarrow aAB \\ A \rightarrow aA/b \\ B \rightarrow bB/b \end{cases}$$

1) $\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

2) $\left.\begin{array}{l} \delta(q_1, \epsilon, S) = (q_1, aAB) \\ \delta(q_1, \epsilon, A) = (q_1, aA) \text{ or } (q_1, b) \\ \delta(q_1, \epsilon, B) = (q_1, bB) \text{ or } (q, b) \end{array}\right\}$ Non. terminals replace

$\left.\begin{array}{l} \delta(q_1, a, a) = (q_1, \epsilon) \\ \delta(q_1, b, b) = (q_1, \epsilon) \end{array}\right\}$ terminals pop

3) $\delta(q_1, \epsilon, z_0) = (q_f, z_0)$