221-1115
H

FAST

Tuesday — Automata — Date: 29/10/24

Design CFG (context free grammer)

Notes: Assignment question

Q1 $L = a^n b^m \mid 2n <= m <= 3n$

whether the given solution works for $n=0$

$L = \lambda, abb, \ldots, aabbbb, aabbbbb, \ldots$

if $n=1$, $m = 2$ or 3 →

if $n=2$, $m = 4,5,6$

if $n=3$, $m = 6,7,8,9$

$S = \lambda \mid aSabb \mid aSbbb$

Q2 Design CFG $L = \{ W \in (0,1)^* \mid W = W^R \}$

$L = 010, 0110, 1001 \ldots$

$S = 0S0 \mid 1S1 \mid 0 \mid 1 \mid \lambda$

Q3 Design CFG $a^n b^n c^m d^m e^p f^p g^q h^q$ → independent

$n, m, p, q > 0$

abcdefgh, aabb cccddd efgh

we have 4 indepedent variable

$S \rightarrow XYZW$ $\quad X = a^n b^n \quad Y = c^m d^m$

$X \rightarrow aXb \mid ab \quad Z = e^p f^p \quad W = g^q h^q$

$Y \rightarrow cYd \mid cd$

$Z \rightarrow eZf \mid ef$

$W \rightarrow gWh \mid gh$

Date: 29/10/24 Tuesday

Design CFG $a^n b^m e^p g^q h^a f^p c^m d^n$ → interleaved.

$S = gd \mid aXd \mid aSd$
$X = bc \mid bYc \mid bXc$
$Y = ef \mid eZf \mid eYf$
$Z = gh \mid gZh \mid gZh$

| sessional 2 till Q1 CFG

practice CFG, some examples given in slides

Normalized form:

(i) Chomsky NF
(ii) Greibach NF

these are standard form, production rules are given. we can convert CNF, grammer to GNF. Grammer should be in 1 of these forms.

1 Chomsky Normal Form

if λ should only come from 1st variable / starting Variable

(i) $S \rightarrow \lambda$
(ii) $A \rightarrow AB$
(iii) $A \rightarrow a$

if grammer is one of these rules then that is chomsky rule

Convert grammer into CNF

$S' \rightarrow S$ ← if S apears in right side of equation then augment S

Date:

Rules to convert grammer into CNF

(1) Augment the grammer $S' \to S$ $S^0 \to S$

(2) Remove null production → those who directly points to λ

(3) Remove unit production Chain Rule

(4) Remove useless variable

(5) Remove non-reachable variables

} remove redundancy

(6) CNF (finally) → (~~bring~~ grammer should follow 1 of those 3 rules)

NULL Production Rules

Exp consider CFG

$S \to a \mid Xb \mid aYa$

$X \to Y \mid \lambda$ X is nullable

$Y \to b \mid X$ Y is nullable.

$X \to \lambda$ is null production, $Y \to X$ is nullable

X, Y are nullable. variables.

→ now put λ instead of X, Y one time and 2nd time use X and Y as it is.

G' $S \to a \mid Xb \mid a \mid aYa \mid b \mid aa$

$X \to Y$

$Y \to b \mid X$

G' is after removing null production.

Date:

New examples: $S \rightarrow Xa$

$X \rightarrow aX \mid bX \mid \lambda$ X is nullable

X is nullable variable

$S \rightarrow Xa \mid a$

$X \rightarrow aX \mid bX \mid a \mid b$

$S \rightarrow XY$

$X \rightarrow Zb$ A, B, Z, W are nullable

$Y \rightarrow bW$

$Z \rightarrow AB$

$W \rightarrow Z$

$A \rightarrow aA \mid bA \mid \lambda$

$B \rightarrow Ba \mid Bb \mid \lambda$

$S \rightarrow XY$

$X \rightarrow Zb \mid b$

$Y \rightarrow bW \mid b$

$Z \rightarrow AB \mid A \mid B$

$W \rightarrow Z$

$A \rightarrow aA \mid bA \mid a \mid b$

$B \rightarrow Ba \mid Bb \mid a \mid b$

Date:

Unit Production Rules

$S \to A$ this is unit production.

Consider CFG $S \to A \mid bb$

$A \to B \mid b$

$B \to S \mid a$

$S \to bb$ $A \to b$ $B \to a$ are non-unit

$S \to A$ $A \to B$ $B \to S$ are unit production

$S \to A \mid bb$

$S \to b \mid B \mid bb$

$\hookrightarrow b \mid a \mid bb$ ✓

$A \to B \mid b$

$A \to a \mid S \mid b$

$\hookrightarrow a \mid bb \mid b$

$B \to S \mid a$

$B \to A \mid bb \mid a$

$\hookrightarrow B \mid b \mid bb \mid a$

$\hookrightarrow a \cdot b \mid bb \mid a$

$B \to S \mid a$

$\hookrightarrow b \mid a \mid bb$

$S \to b \mid a \mid bb$

$A \to a \mid bb \mid b$

$B \to b \mid bb \mid a$

all are same, to remove redundancy, we will keep 1.

Date:

How to go to Chomsky

(2) ~~Greibach Normal~~ Form. Grammatical Format

(1) $S \rightarrow \lambda$ CNF

(2) $A \rightarrow AB$

(3) $A \rightarrow a$

Ex: $S \rightarrow ABa$

$A \rightarrow aab$

$B \rightarrow AC$

(1) $S \rightarrow ABa$ $\quad$ $V_2 \rightarrow Ba$

$S \rightarrow AV_2 \rightarrow$ now in CNF ✓

$T_a \rightarrow a$ so

$V_2 \rightarrow BT_a \rightarrow$ CNF ✓

(2) $A \rightarrow aab$ $\quad$ $V_1 \rightarrow ab$

$A \rightarrow aV_1$ $\quad$ $A \rightarrow T_aV_1$ now in CNF

$T_b \rightarrow b$

$V_1 \rightarrow T_aT_b$ now in CNF

by adding new variable we can convert it into CNF.

$S \rightarrow AV_2$

$V_2 \rightarrow BT_a$

$T_a \rightarrow a$

$T_b \rightarrow b$

$V_1 \rightarrow T_aT_b$

Date:

Convert the following CFG to CNF

$S \rightarrow aSa \mid bSb \mid a \mid b \mid aa \mid bb$

$A \rightarrow a$

$B \rightarrow b$

$S \rightarrow ASA \mid BSB \mid A \mid B \mid AA \mid BB$

$C \rightarrow SA$

$D \rightarrow SB$

$S \rightarrow AC \mid BD \mid A \mid B \mid AA \mid BB$

Thursday Date: 31/10/24

Chomskys Normal Form: (CNF)

all production rules should be in one of these forms:

1- $S \rightarrow \lambda$ lambda from starting variable

2- $A \rightarrow AB$ right side pe 2 variable

3- $A \rightarrow a$ right side pe terminal

Greibach Normal Form: (GNF)

1. $S \rightarrow \lambda$
2. $A \rightarrow a A_1 A_2 A_3 A_4$
3. $A \rightarrow a$

Rules to convert grammer into CNF

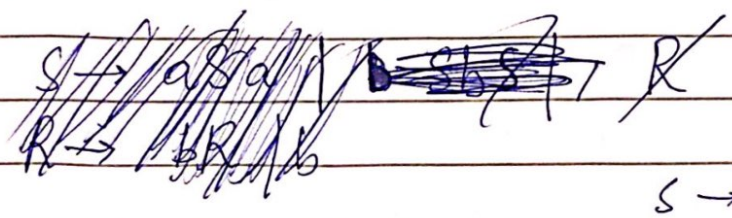
(i) augment grammar

(ii) remove null, useless, unit, non-reachable variables.

(iii) finally convert it into CNF

(1) $(a+b)^*$ $S \rightarrow a \mid L \mid \lambda \mid SS$ or $aS \mid bS \mid \lambda$

(2) $a^{(n)} b^m a^{(n)}$ $n>0,\ m>0$



aba

$S \rightarrow aSa \mid aXa$

$X \rightarrow bX \mid b$

$S \rightarrow aXa$

b

$\hookrightarrow aba$

aSa

$aaXaa$

$aabaa$

Theory of automata (TOA) Date: 7/11/24 Thursday

Normalized forms →

(1) Chomsky normal form (CNF)
(2) Greibach normal form (GNF)

CNF:

i) $S \rightarrow \lambda$
ii) $A \rightarrow AB$
iii) $A \rightarrow a$

GNF:

(i) $S \rightarrow \lambda$
(ii) $A \rightarrow aA_1A_2A_3....$
(iii) $A \rightarrow a$

CFG → CNF

① augment the grammar ($S' \rightarrow S$ to make starting var unique)
② Remove null words
③ Remove unit / chain rules
④ Remove useless vars / unreachable variable
⑤ conversion to CNF

to convert any grammar into GNF, it should be in CNF first.

Exp Grammar: $S \rightarrow ACA$

$A \rightarrow aAa \mid B \mid C$
$B \rightarrow bB \mid b$
$C \rightarrow cC \mid \lambda$

1 remove null variables -

nullable variables → S, A, C

Date:

$S \rightarrow ACA \mid CA \mid AA \mid AC \mid A \mid C \mid \lambda$ (only S can give λ)

$A \rightarrow aAa \mid aa \mid B \mid C$

$B \rightarrow bB \mid b$

$C \rightarrow cC \mid c$

haftsa → ♡

Eliminate chain Rule:

Chain Rules → no need in B and C

↳ apply on S and A

$A \rightarrow aAa \mid aa \mid B \mid C$

↳ $bB \mid b$

$A \rightarrow aAa \mid aa \mid bB \mid b \mid cC \mid c$

$S \rightarrow ACA \mid CA \mid AA \mid AC \mid aAa \mid aa \mid bB \mid b \mid cC \mid c \mid \lambda$

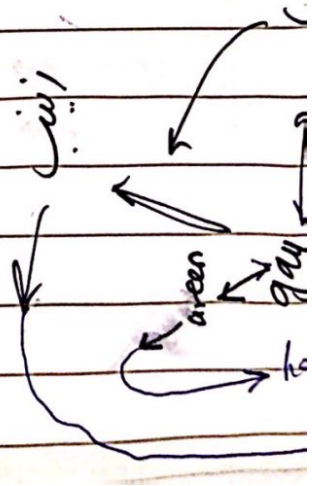
$S \rightarrow ACA \mid CA \mid AA \mid AC \mid aAa \mid aa \mid bB \mid b \mid cC \mid c \mid \lambda$

$A \rightarrow aAa \mid aa \mid bB \mid b \mid cC \mid c$

Remove useless variable

$S \rightarrow AC \mid \cancel{BS} \mid B$

$\cancel{A \rightarrow a}$



Date:

Remove Useless Variables

- ✓ $S \rightarrow AC \mid BS \mid B$
- ✓ $A \rightarrow aA \mid aF$
- * $B \rightarrow cF \mid b$
- ~~$C \rightarrow cC \mid D$~~
- ~~$D \rightarrow aD \mid BD \mid C$~~
- ✓ $E \rightarrow aA \mid BSA$
- * $F \rightarrow bB \mid b$

we have to identify whose is giving terminal directly. like B and F and then identify by putting value of B and F whose if giving termina

useful variables → B, F, S, A, E

useless → C, D

directly cut the production rules of useless variables.

$S \rightarrow BS \mid B$

$A \rightarrow aA \mid aF$

$B \rightarrow b$

$E \rightarrow aA \mid BSA$

$F \rightarrow bB \mid b$

remove those productions which has useless variables

Question pochny ky tareeqy.

→ Convert CFG to CNF

→ remove Chain rules / useless variables

Date:

Q: $S \to AA \mid a$
$A \to aA \mid B \mid \lambda$
$B \to bB$
$C \to SD \mid DB \mid a$
$D \to aD \mid DD$

convert the grammar into CNF

1. → Augment the grammar
$S' \to S \mid \lambda$
$S \to AA \mid a \mid A$
$A \to aA \mid B \mid \lambda$
$B \to bB$
$C \to SD \mid DB \mid a$
$D \to aD \mid DD$

2. Remove the null words
nullable variables → A, S, S'
~~$A \to \lambda \mid aA \mid bB$~~
~~$S \to AA$~~
$A \to aA \mid B \mid a$
$S \to AA \mid a \mid A$
$S' \to S \mid \lambda$
$C \to SD \mid DB \mid D \mid a$
$B \to bB$
$D \to aD \mid PD$

3. Remove chain / unit rules $\{S' \to S, S \to A, A \to B, C \to D\}$

$S' \to AA \mid a \mid \lambda \mid aA \mid bB$
$S \to AA \mid a \mid aA \mid bB$
$A \to aA \mid bB \mid a$
$B \to bB$
$C \to SD \mid DB \mid aD \mid DD \mid a$
$D \to aD \mid DD$

Date:

4/- Remove Useless Variable

Useful variable — A, S, S', C

Useless variables ⟶ B, D

$S' \rightarrow AA \mid aA \mid \cancel{BB} \mid a \mid \lambda$

$S \rightarrow AA \mid aA \mid a$

$A \rightarrow aA \mid a$

$C \rightarrow a$

5/ non-reachable variables ⟶ C, S

$S' \rightarrow AA \mid aA \mid a \mid \lambda$

$A \rightarrow aA \mid a$

~~$X \rightarrow a$~~

now

$S' \rightarrow AA \mid XA \mid a \mid \lambda$

$A \rightarrow XA \mid a$

$X \rightarrow a$

} CNF

Theory of automata Date: 12/11/24 Tuesday

GNF:

(i) $S \rightarrow \wedge$

(ii) $A \rightarrow aA_1A_2A_3$ ——

(iii) $A \rightarrow a$

convertion of grammar into GNF
first convert into CNF then GNF

(1) Removing of left recursion:

(1) $A \rightarrow Aa \mid b$ (A apne aap ko call kiye ja raha hai)

$A \rightarrow bZ \mid b$

$Z \rightarrow aZ \mid a$

(one time write that production rule that does not have recursion with new variable and 1 time as it is
→ same is with recursion past remove A and write Z 1 time and 1 time as it is)

(2) $A \rightarrow Aa \mid Ab \mid b \mid c$

$A \rightarrow bZ \mid cZ \mid b \mid c$

$Z \rightarrow aZ \mid a \mid bZ \mid b$

Date:

$A \rightarrow AB \mid BA \mid a \qquad B \rightarrow b \mid c$

$A \rightarrow BAZ \mid BA \mid aZ \mid a$
$Z \rightarrow BZ \mid B$
$B \rightarrow b \mid c$

Complete example of GNF

1 $S^1 \rightarrow {}^2AB \mid 1$ ✓
2 $A^2 \rightarrow {}^3AB \mid CB \mid a$ equal → left recurr
3 $B^3 \rightarrow {}^2AB \mid b$ → less → put value of A
4 $C \rightarrow AC \mid c$

grammar is already in CNF

→ assign sequence number to production Rules

Right side pe jo 1st variable hai uska sequence no should be greater than left side wala variable

$1S \rightarrow {}^2AB \qquad 2 > 1$ ✓

if value is less than left variable → put value of A.

equal value → left recursion ka case

Date:

S1: $S \rightarrow AB \mid \lambda$

$A \rightarrow CBR_1 \mid aR_1 \mid CB \mid a$

$^3B \rightarrow {}^2AB \mid b$

$^4C \rightarrow {}^2AC \mid c$

$R_1 \rightarrow BR_1 \mid B$

→ A has left recursion we solved that

S2:- $S \rightarrow AB \mid \lambda$

$A \rightarrow CBR_1 \mid aR_1 \mid CB \mid a$

$B \rightarrow CBR_1B \mid aR_1B \mid CBB \mid aB \mid b$

$C \rightarrow CBR_1C \mid aR_1C \mid CBC \mid aC \mid c$

$R_1 \rightarrow BR_1 \mid B$

→ now this has left recursion.

S3: $S \rightarrow AB \mid \lambda$, $A \rightarrow$ " , $B \rightarrow$ " } as it is

$C \rightarrow aR_1CR_2 \mid aCR_2 \mid cR_2 \mid aR_1C \mid aC \mid c$

$R_2 \rightarrow BR_1CR_2 \mid BCR_2 \mid BR_1C \mid BC$

left recursion now removed.

now C is in GNF

we put / use rules until 1 variable is in GNF

Arpan Zainab **Assignment #04**

FAST NU

Date:

Now look at B

$B \rightarrow CBR_1B \mid aR_1B \mid CBC \mid aB \mid b$

these 3 with ticks are in GNF

But value of C

$B \rightarrow (aR_1CR_2)BR_1B \mid (aCR_2)BR_1B \mid (CR_2)BR_1B \mid (aR_1C)BR_1B \mid (aC)BR_1B \mid (aC)BR_1B \mid cBR_1B \mid aR_1B \mid aB \mid b$ are as it is.

$(aR_1CR_2)BC \mid (aCR_2)BC \mid (CR_2)BC \mid (aR_1C)BC \mid acBC \mid aCBB \mid cBC$

B is in GNF

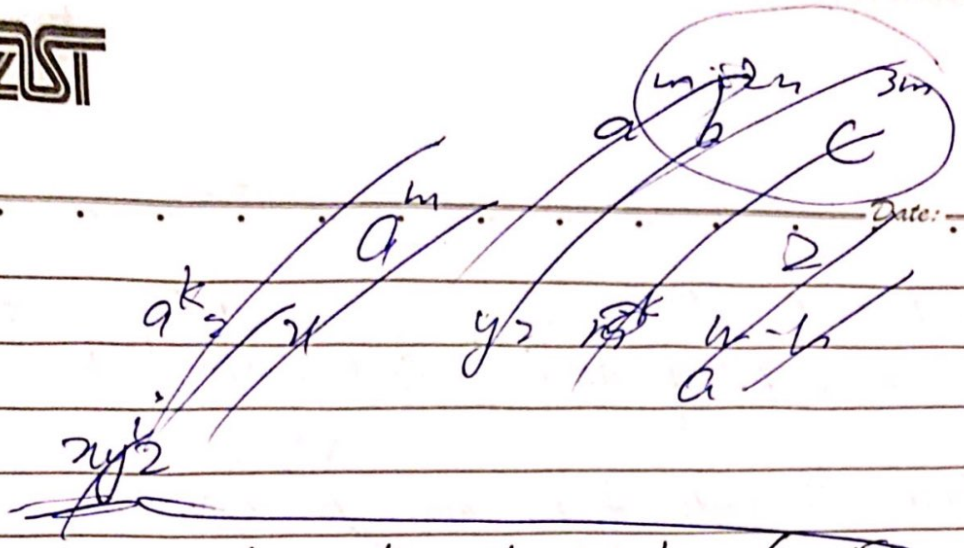
$A \rightarrow CBR_1 \mid aR_1 \mid CB \mid a$

$A \rightarrow aR_1 \mid a \mid (aR_1CR_2)BR_1 \mid aCR_2BR_1 \mid CR_2BR_1 \mid aR_1CBR_1 \mid aCBR_1 \mid cBR_1 \mid aR_1CR_2B \mid aCR_2B \mid CR_2B \mid aR_1CB \mid aCB \mid cB$

$S \rightarrow AB \mid \lambda$

$S \rightarrow (aR_1)B \mid aB \mid aR_1CR_2BR_1B \mid aCR_2BR_1B \mid CR_2BR_1B \mid aR_1CBR_1B \mid aCBR_1B \mid cBR_1B \mid aR_1CR_2BB \mid aCR_2BB \mid CR_2BB \mid aR_1CBB \mid acBB \mid cBB$

Now do it for R_1 and R_2



Convert it into GNF

1 $S \longrightarrow XA \mid BB$ ✓

2 $B \longrightarrow b \mid SB$ →

3 $X \longrightarrow b$

4 $A \longrightarrow a$

grammar is ~~already~~ in CNF

Now convert it into GNF

grammar is not in CNF

Augment the language.

$S' \longrightarrow S$

1 $S' \longrightarrow XA \mid BB$ ✓

2 $B \longrightarrow b \mid SB$ ×

3 $X \longrightarrow b$

4 $A \longrightarrow a$

S1:

$B \longrightarrow b \mid XAB \mid BBB$

$X \longrightarrow b$

$A \longrightarrow a$

Nimra Zainab Assignment #01

Date:

S2: $S \rightarrow XA \mid BB$ (1, 3, 2)

$B \rightarrow bR_1 \mid XABR_1 \mid b \mid XAB$ (2, 3, 3)

$X \rightarrow b$

$A \rightarrow a$

$R_1 \rightarrow BBR_1 \mid BB$

S3: $B \rightarrow bR_1 \mid b \mid XABR_1 \mid XAB$

$B \rightarrow bR_1 \mid b \mid bABR_1 \mid bAB$ (put value of X

now B is in GNF

Assignment #04

14/11/2024 ; Thursday

Date: 14/11/24 Thursday

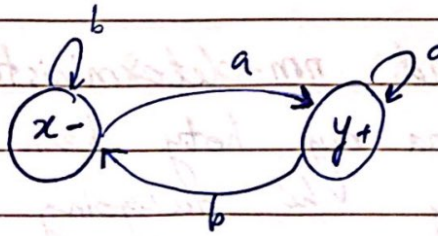
Topic: Theory of Automata.

Ch#14 'Pushdown Automata' (PDA)

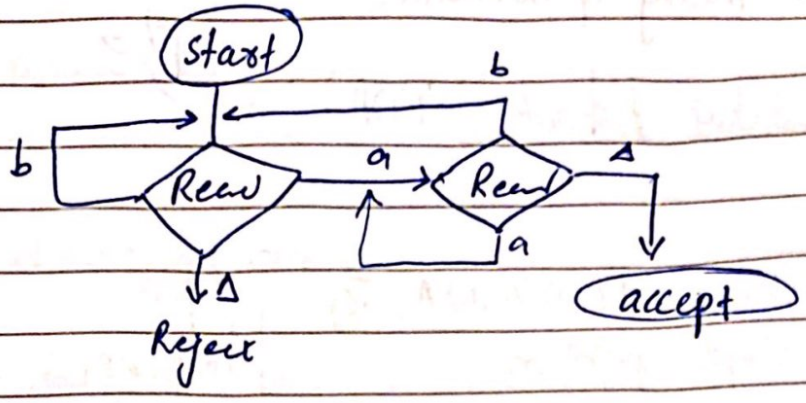
* 1st Part of the Course:
  - → Simple Machines: FA, DFA, NFA, TG ↑ for simple problems. } Regular languages

* 2nd Part of the Course:-
  - → Context Free Grammar (CFG)
    - ↓ Normalized Form: CNF, GNF
  - → Compiler Processing: Parsing + lexical Analysis.

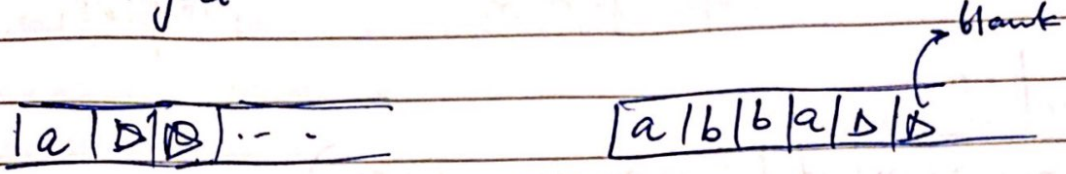
* 3rd Part of the Course:-
  - → PDA ⟹ Stack Memory
  - → Turing Machine ⟹ Tape Memory
  - → PM (Post Machine; uses Queue) → maybe in course
  - → Computable Functions



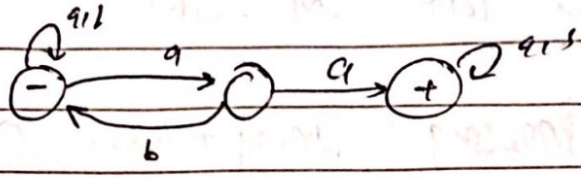
Date:

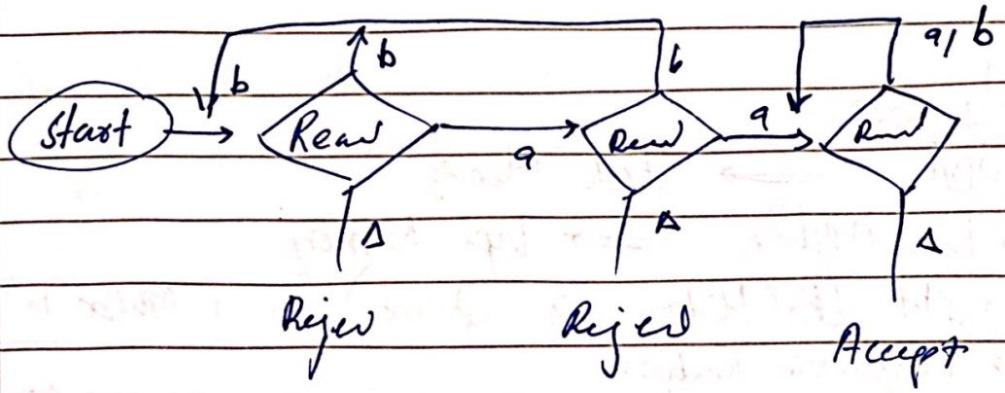


* Practice Practice Practice



Convert this FA to an PDA

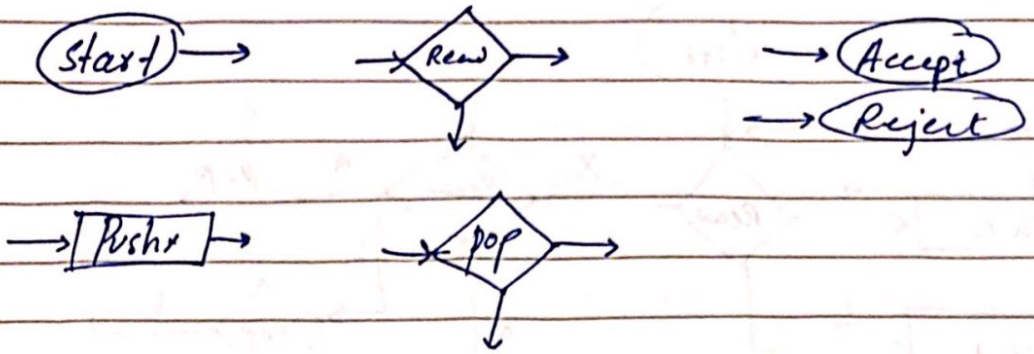




if can be deterministic and non-deterministic. usually non-deterministic bana rhy hoty hain hum. for deterministic → utny hi outgoing edges jitne sigma mein letters.

Arpan Zainab Assignment #01

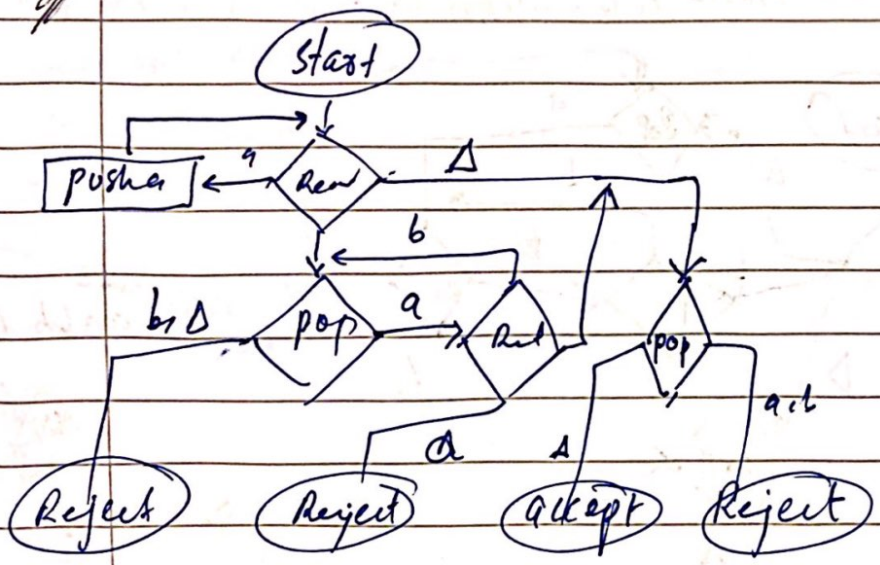
Date:



output characters → Tahoe ~~#~~ T

The language accepted or recognized by a pushdown automaton is the set of word whose paths ends in

Eg. | a | a | a | b | b | b | Δ | ... $L = a^n b^n$



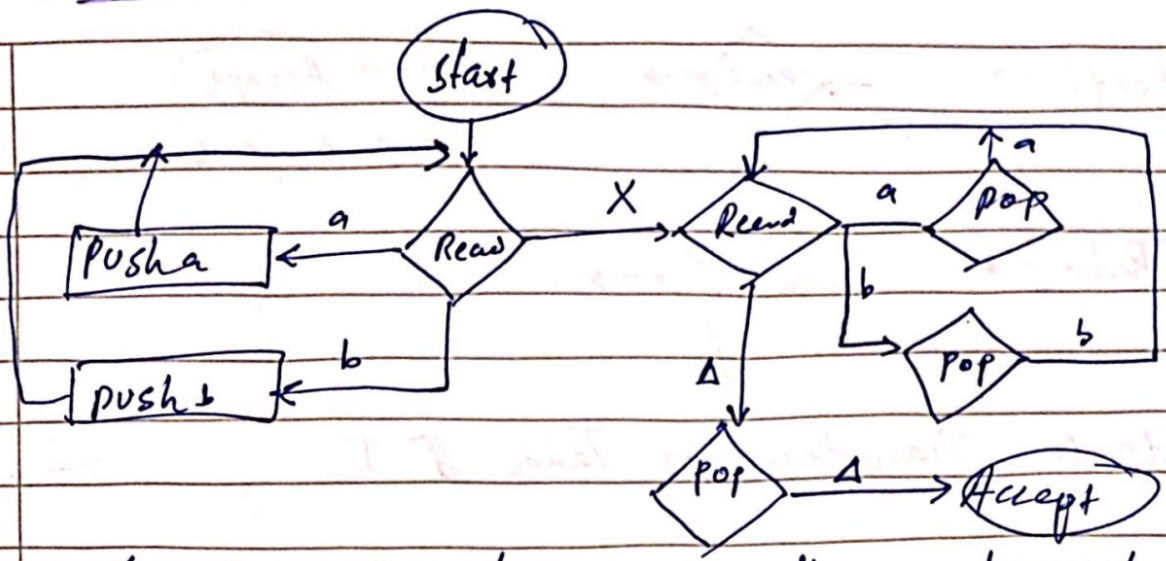
when read a push on stack, when read b pop a.

If we read aaabb → machine will crash

introduction to computing theory → Book.

ODD Palindrome :→ deterministic

Date:

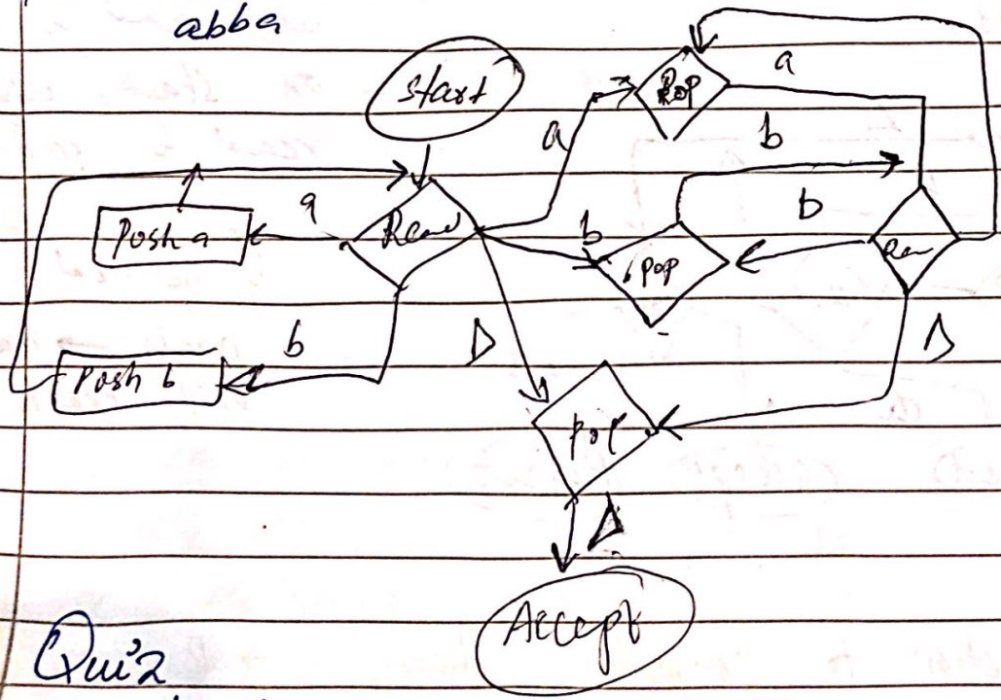


X → & marker to represent the centre of PDA.
this is deterministic PDA → there should be

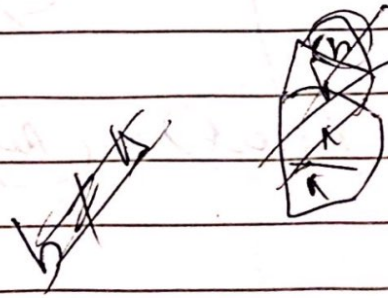
abb X bba ✓ odd palindrome

Even Palindrom → non-deterministic → if there exist 1 path that can take you from start state to end state, that is valid

abba



aabb Δ



Quiz
on tuesday → CNF & GNF

Theory of automata Date: 19/11/24 Tuesday

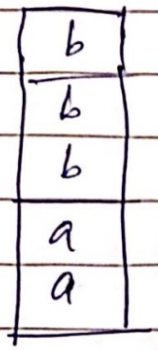
Odd palindrome: $\Sigma = \{a, b\}$

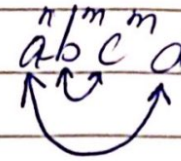
aba, ↳ deterministic.

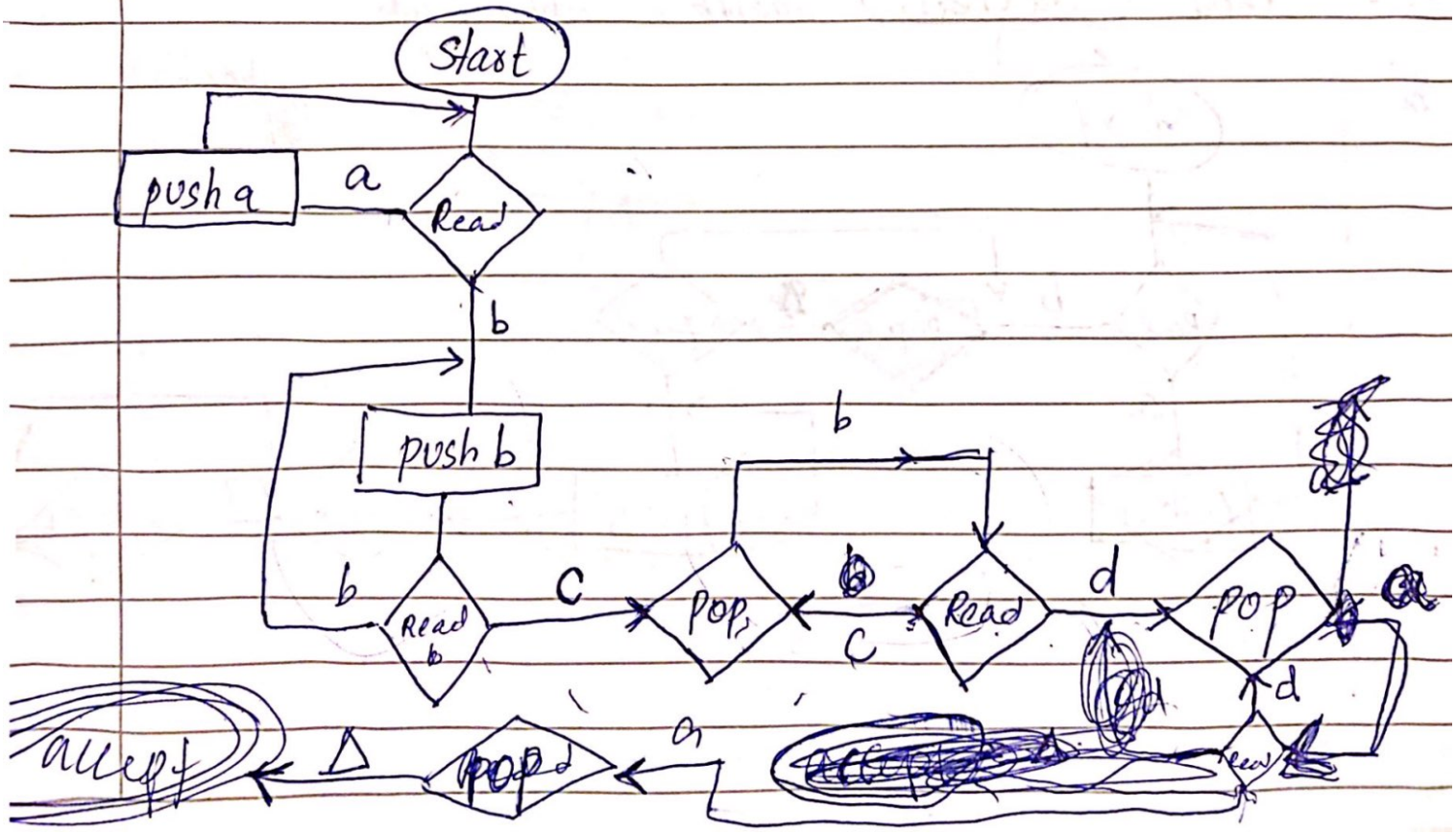
aabaa → ~~right~~ left pe push, right pe pop.

Design PDA for $a^n b^m c^m d^n$ $n > 0$

→ if question has design PDA → use only 1 stack
if question has design 2PDA, → use 2 stacks
3PDA → 3 stack
n-PDA → hamari marzi ka

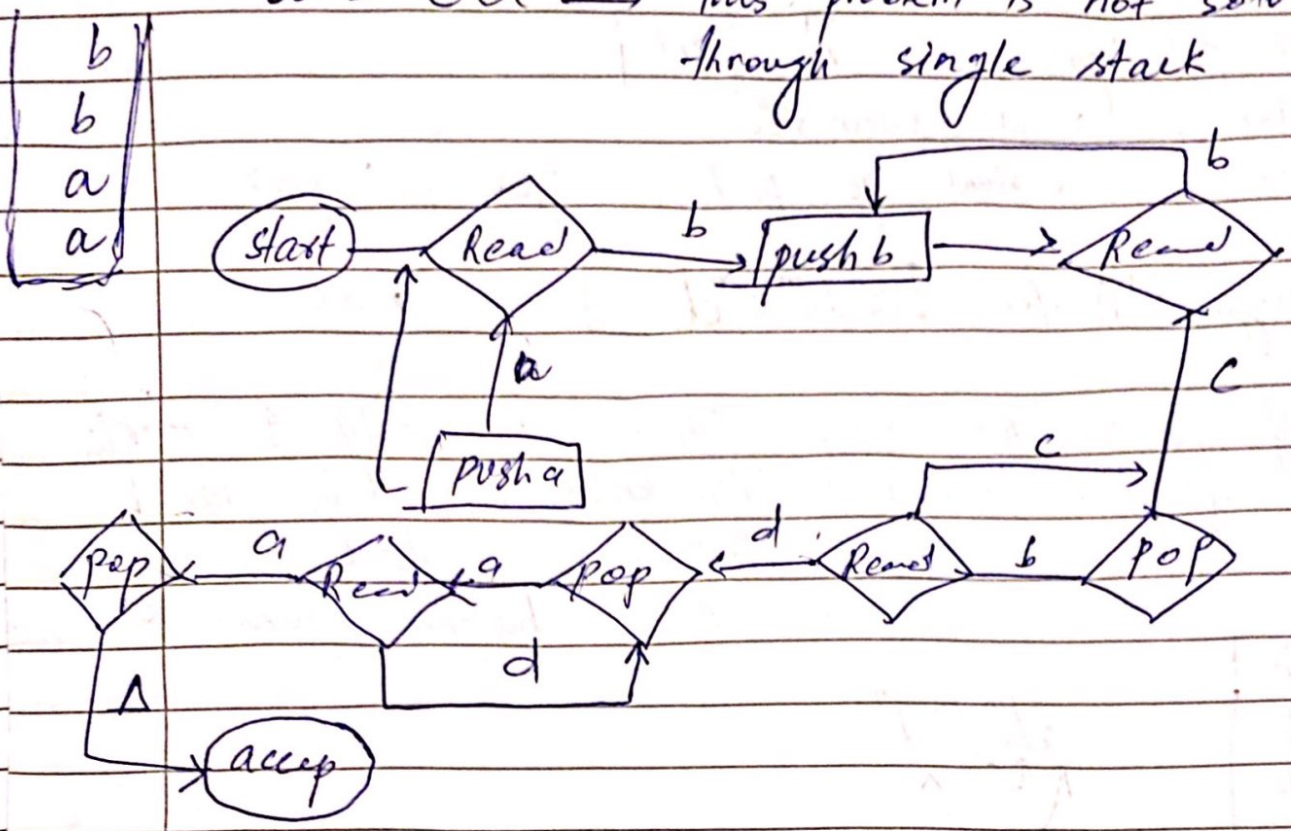






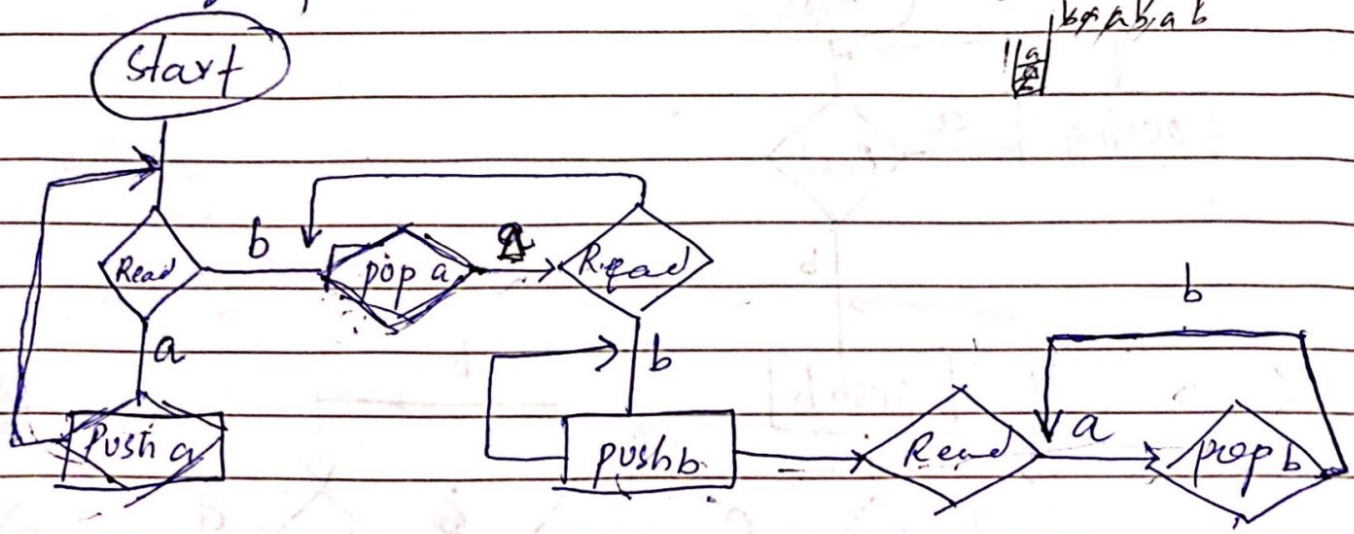
Date:

$a^n b^m c^n d^m$ → this problem is not solvable through single stack



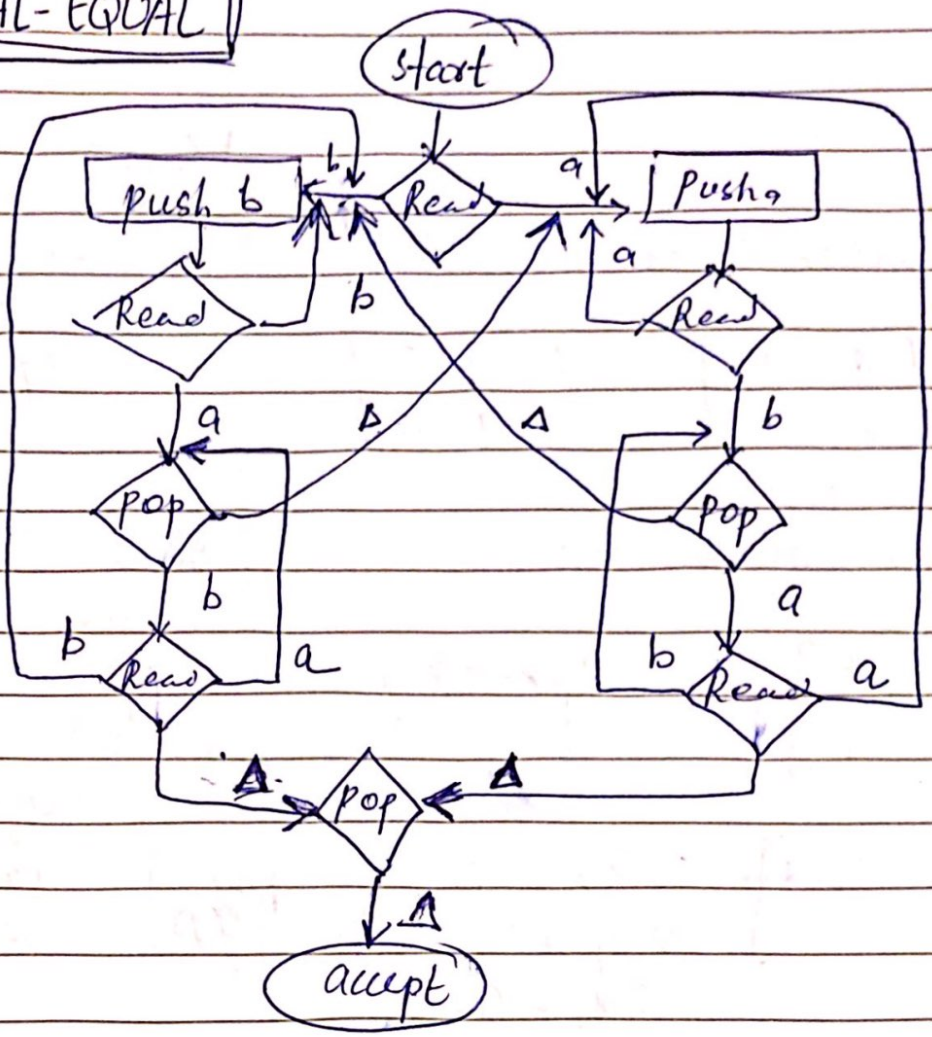
Design a PDA for EQUAL - EQUAL

abba, baabab, aaabbb, abbaabab



Date:

EQUAL-EQUAL



if language is in CNF you can directly convert it into PDA.

① Start ② Push ③ POP ④ Read ⑤ Accept

CFG = PDA ⟶ next class sy pehle kar ky ana hai

~~abab~~ abab ~~abababab~~

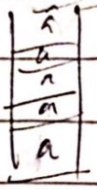
Automata C203B → office no. Date: 21/11/24 Thursday

Turing Machines → non-deterministic

Pda for $a^n b^n$

PDA for $a^n b^n a^n$

aaabbbaaa.



→ read 1 a pop 2
↓
push 2 a m Stack

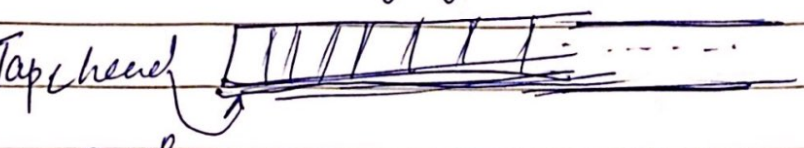
$a^3 b^2 a^4$
↓
This will also be accepted by PDA

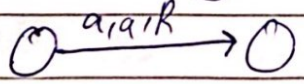
- Turing Machines.

e.g. $a^n b^n a^n$, prime number, WW → (double word)

non-context free languages → these are solved by turing

Tapehead





(a,a,R) → (input, output, direction)

* Memory will be in form of Tape.

abbbaa →

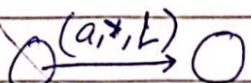
a	b	b	b	a	a	Δ	Δ	Δ	—

↑ Tapehead

you can access any location through tapehead.

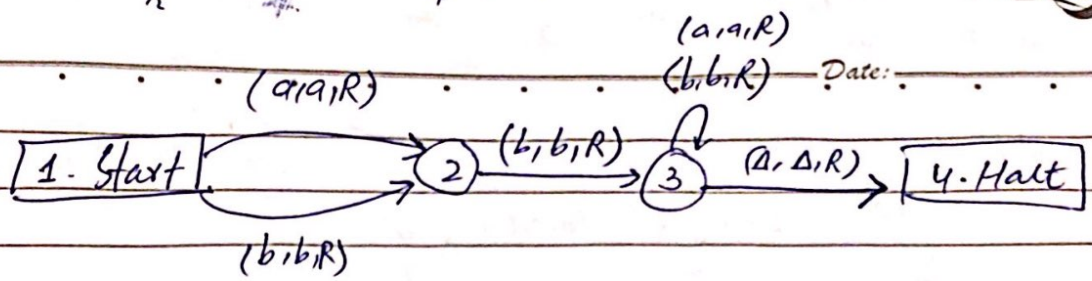
FA → Accept TM → Halt

Transition is shown by this →



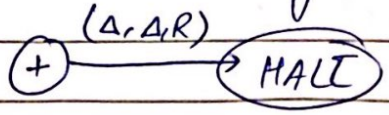
(a,*,L) → read a, print * move left.

Date:

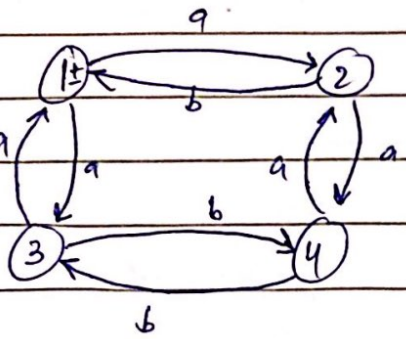


this turing machine is accepting all words where 2nd letter is b. aabb → word be accepted
↳ will give error index out of bound

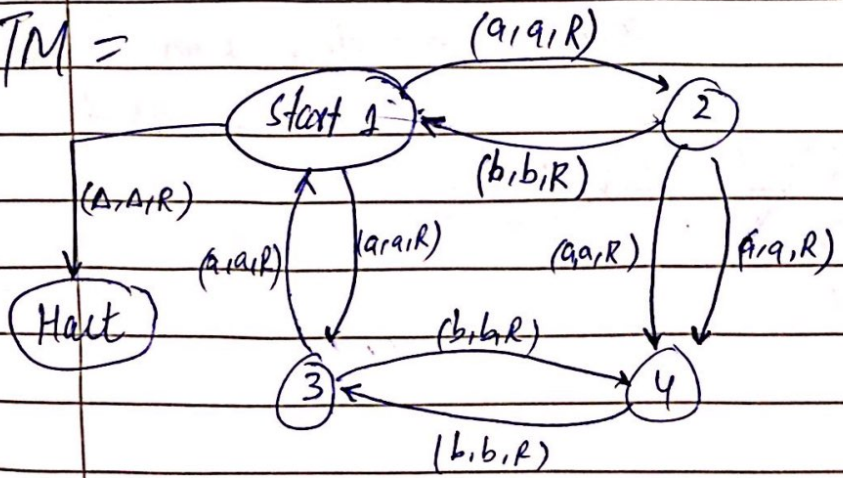
FA to TM:
FA's start state is Turing Machine's start state



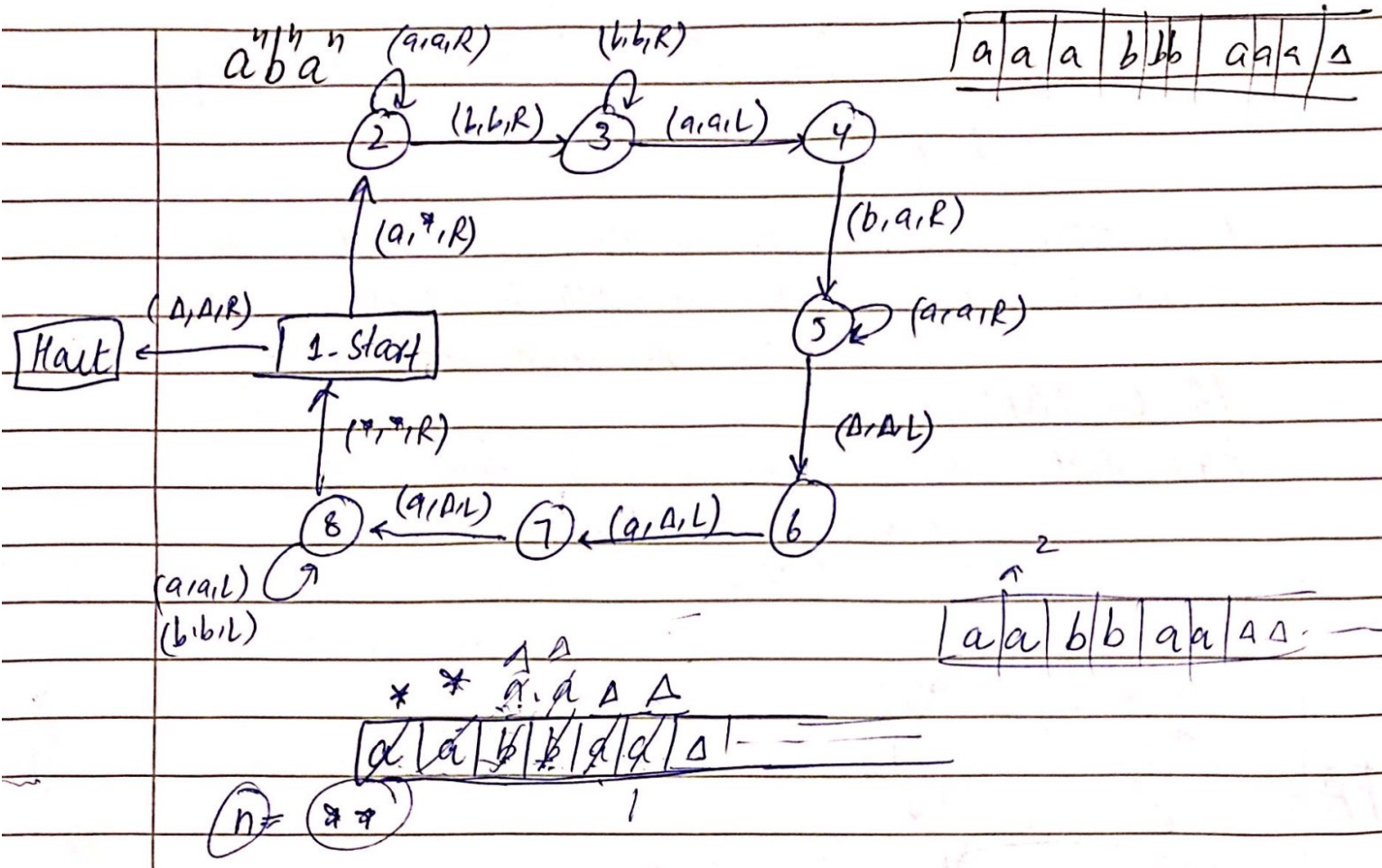
FA =



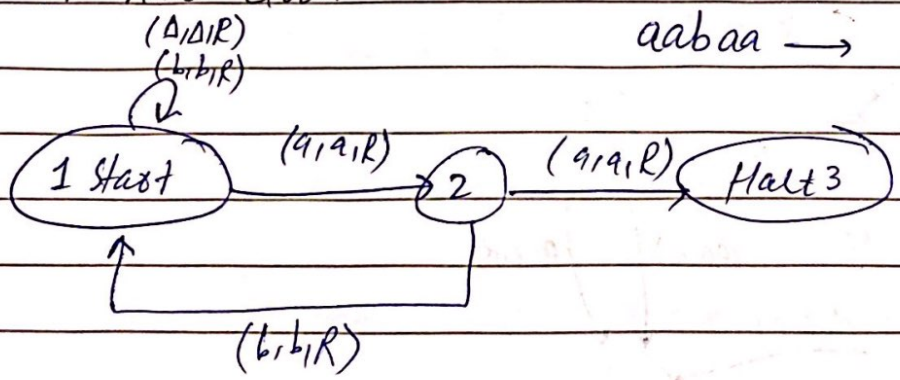
TM =



Date:



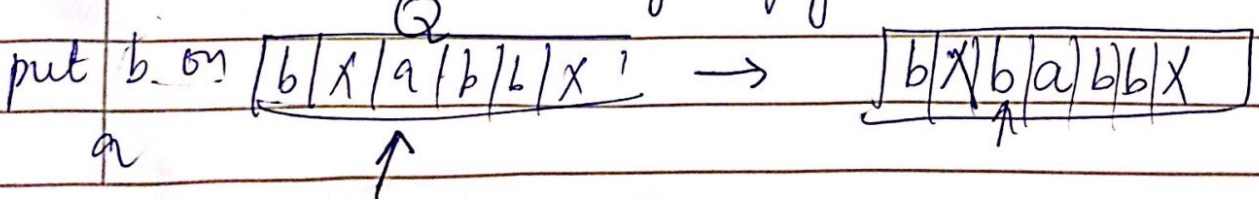
Must have double aa



aabaa → after 1 aa it stops taking input

Insert character:

use a marker 'Q' where you want to insert letter and move baqi string agay forwa



aabb

Date:

DELETE any character from Tape:

| b | c | a | b | b | c |

input delete a

↑

| b | c | b | b | c |

output

b

b c Δ

| b | c | a | b | b | c | Δ |

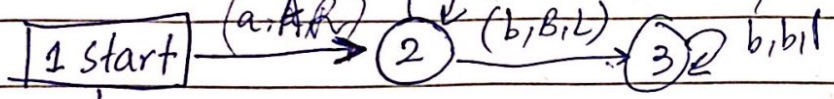
Question → Design Turing machine for any language

Design TM for $a^n b^n$

aaa bbb

(a,A,R)

| a | a | a | b | b | b | Δ |



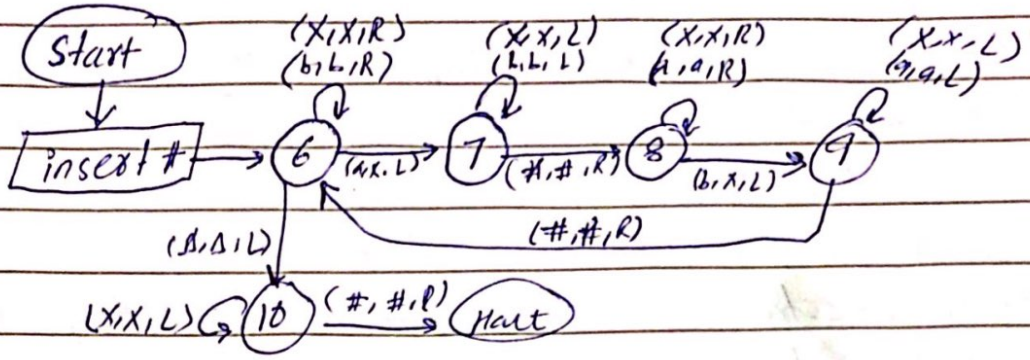
(Δ,Δ,R)

Halt

Book → exercise questions solve, Past Papers.

26/11/24 (online) (Automata) Date:

Turing machine for language EQUAL

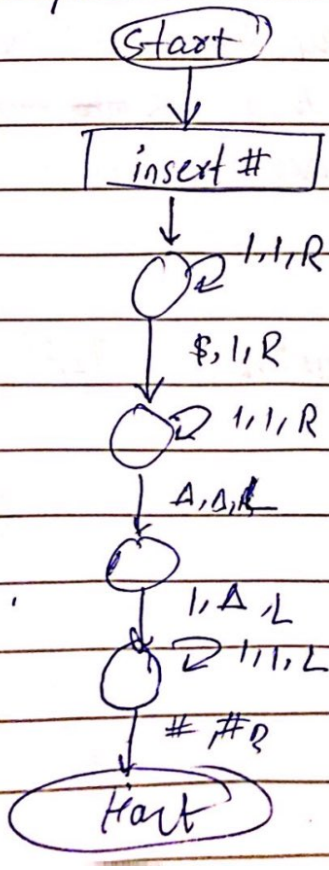


input → aabb →

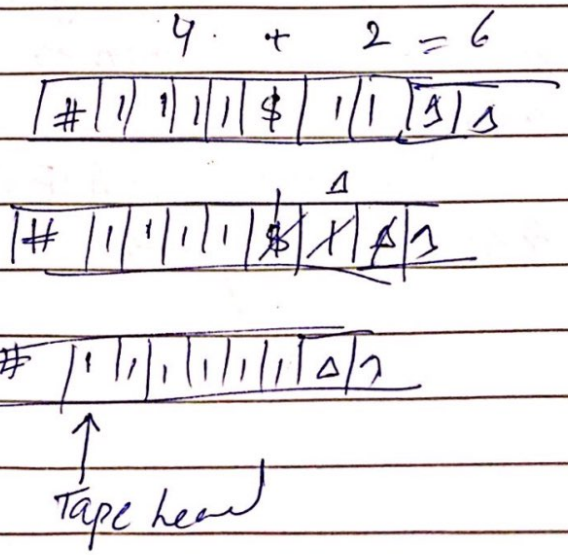
Turing machine → computing device
↳ define computable functions

Computable function:
Unary input
$ → separator

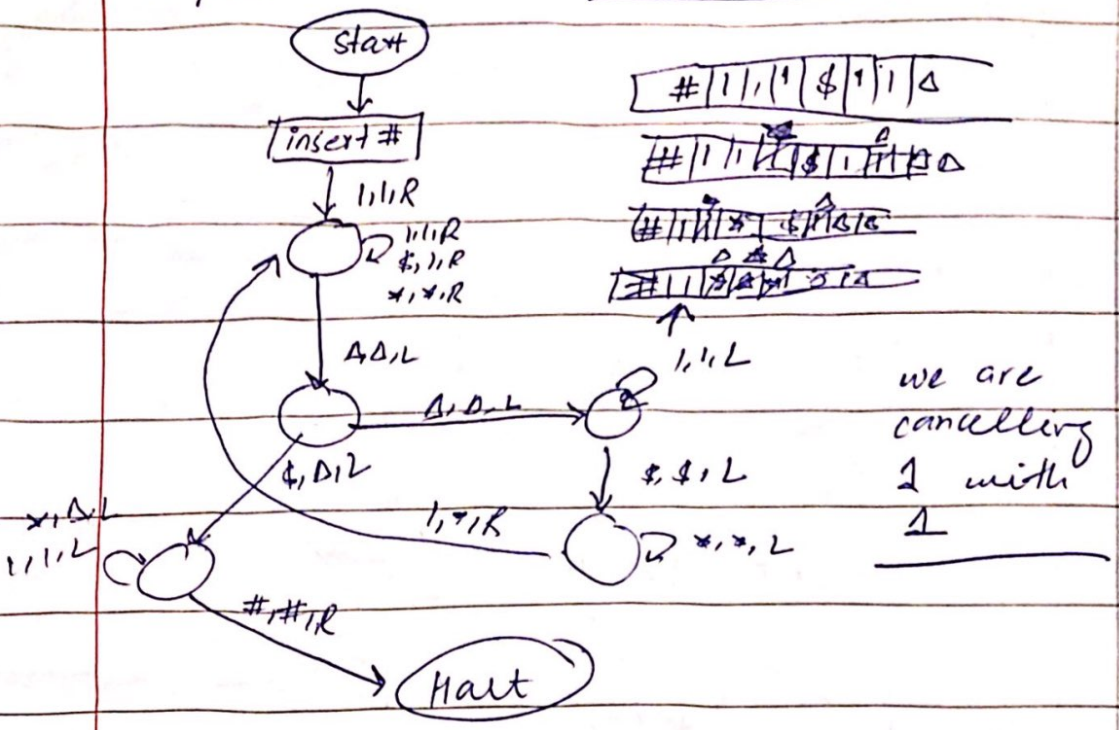
Computable function for addition



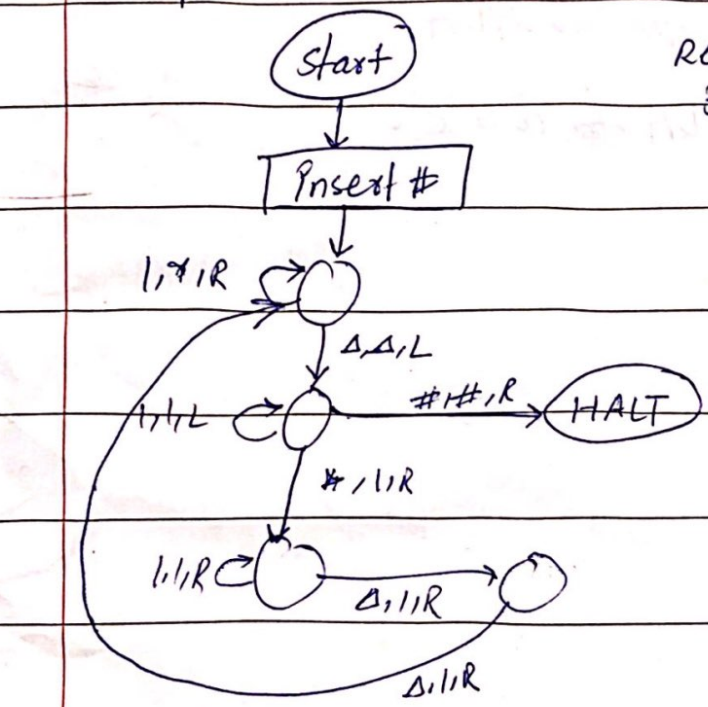
4 + 2 = 6



Computable function for **Subtraction**

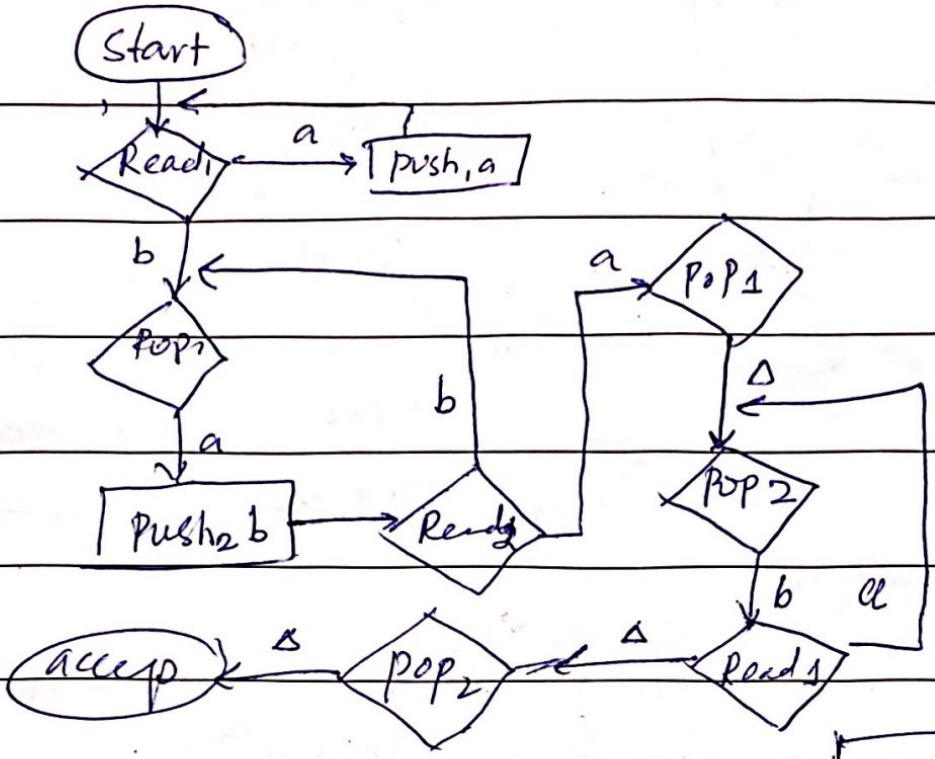


Computable function $f(x) = 3x$



2- PDA

Language $a^n b^n a^n$ eg → aaabbbaaa



see book for 2PDA

Course completed

Quiz

HW → Design 2 PDA for $a^n b^m c^n d^m$