22/10/24

UNIT - 4

CONTEXT FREE LANGUAGES AND TURING MACHINES

simplification of CFG:

steps to simplify CFG:

step-1: Eliminate the useless productions

step-2: Eliminate the null production (ϵ- production)

Eliminate the unit productions

step-1: Eliminations of useless productions

The productions and the non-terminal in the grammar that donot take part in the derivation of strings are called as useless production or symbol.

* To find the useless production in the grammar compute the generating and reachable symbols of grammar.

* The symbols that are not generating or not reachable are said to be useless symbols and the production that has the symbols are useless productions

Generating:

A symbol (non-terminal) 'A' is said to be generating if there is production of the form $S \rightarrow \alpha AB$ where A is the non-terminal which is able to derive a string (composed of terminal symbols)

Reachable:

The non-terminal symbols that are reachable from the start production either directly or indirectly are said to be reachable symbols.

Remove all the symbols from the grammar that are not generating or not reachable.

Problem:

1) consider the grammar $E \rightarrow cB/e$

$A \rightarrow a$

Remove the useless productions from the grammar.

$T = \{c, e, a\}$

$NT = \{E, B, A\}$

Generating:

$E \rightarrow e$

$A \rightarrow a$

B cannot derive string

Generating $= \{E, A\}$

Reachable:

E, B are reachable from start production

A is not reachable

Reachable $= \{E, B\}$

so, useful productions are

Generating and Reachable

$\{E, A\}$ and $\{E, B\}$

E is the useful as it is Generating and Reachable

∴ useless symbols are A & B

so remove productions with A & B

× $E \rightarrow cB$

$E \rightarrow e$ is the useful production.

× $A \rightarrow a$

2) $E \to AB \mid a$

$A \to ab \mid C$

$C \to aD \mid B$

$B \to aB$

$D \to dD$

Sol:

$T = \{a, b, d\}$

$NT = \{E, A, B, C, D\}$

Generating:

$E \to a$

$A \to ab$

C - cannot derive string

B - cannot derive string

D - cannot derive string

Generating = $\{E, A\}$

Reachable:

E, A, B, C, D are reachable ~~fro~~

A, B directly reachable from E

C, D indirectly reachable from A & B

Reachable = $\{E, A, B, C, D\}$

so, useful productions are

Generating and Reachable

$\{E, A\}$ and $\{E, A, B, C, D\}$

E, A is the useful as it is Generating and Reachable

∴ useless symbols are B, C, D

so remove productions with B, C, D

×$E \to AB$ ×$C \to B$

$E \to a$ ×$B \to aB$

$A \to ab$ ×$D \to dD$

×$A \to C$

×$C \to aD$

∴ $E \to a$

$A \to ab$ are the useful productions.

since there are two productions again find G & R

Generating:

$E \to a$

$A \to ab$

Generating = {E, A}

Reachable:

E is reachable from start production

A is not reachable

Reachable = {E}

so, useful productions are {E, A} and {E} = {E}

useless symbols are A, so remove production with A

$E \to a$ is the useful production

× $A \to ab$

---

3) $S \to aS \mid B$

$A \to aa$

$B \to bb$

T = {a, b}

NT = {S, A, B}

Generating:

$S \to bb$

$A \to aa$

$B \to bb$

Generating = {S, A, B}

Reachable:

S, B are reachable

A is not reachable

Reachable = {S, B}

so, useful productions are {S, A, B} and {S, B} = {S, B}

∴ useless symbols is A, so remove production with A.

$S \to aS$

$S \to B$

× $A \to aa$

$B \to bb$

∴ useful productions are

$S \to aS$

$S \to B$

$B \to bb$

Generating:

$S \rightarrow abb$

$B \rightarrow bb$

Generating = $\{S, B\}$

Reachable:

S, B are reachable

Reachable = $\{S, B\}$

So, useful productions are $\{S, B\}$ and $\{S, B\} = \{S, B\}$

$\therefore S \rightarrow aS / B$

$B \rightarrow bb$

are the useful productions.

---

4) $S \rightarrow aAa \mid aBC$

$A \rightarrow aS \mid bD$

$B \rightarrow aBa \mid b$

$C \rightarrow abb \mid DD$

$D \rightarrow aDa$

$T = \{a, b\}$

$NT = \{S, A, B, C, D\}$

Generating:

$S \rightarrow ababb$

$A \rightarrow aababb$

$B \rightarrow b$

$C \rightarrow abb$

Generating = $\{S, A, B, C\}$

Reachable:

S, A, B, C, D are reachable

Reachable = $\{S, A, B, C, D\}$

So, useful productions are $\{S, A, B, C\}$ and $\{S, A, B, C, D\}$

$= \{S, A, B, C\}$

$\therefore$ useless symbol is D, so, remove production with D

$S \rightarrow aAa / aBC$

$A \rightarrow aS / bD^{\times}$

$B \rightarrow aBa / b$

$C \rightarrow abb / {}^{\times}DD$

${}^{\times}D \rightarrow aDa$

∴ useful productions are

$S \rightarrow aAa \mid aBC$

$A \rightarrow aS$

$B \rightarrow aBa \mid b$

$C \rightarrow abb$

simplification or Elimination of null productions/ ϵ- productions:

1) Find the set of nullable symbols in grammar G.

2) A variable is said to be nullable if it as the production of form $\underset{\text{Non-terminal}}{A} \rightarrow \epsilon$

3) Replace a nullable variable in the production of the G with all possible combinations of non-terminal to generate new set of productions P'.

Example: Eliminate the null productions from the grammar given below.

$$S \rightarrow abB$$
$$B \rightarrow Bb \mid \epsilon$$

Nullable variable

$\therefore B \rightarrow \epsilon$

B is nullable variable

replacing B by ϵ

$$S \rightarrow ab\epsilon$$
$$\boxed{S \rightarrow ab}\ P'$$
$$B \rightarrow \epsilon b$$
$$\boxed{B \rightarrow b}$$

$$S \rightarrow abB \mid ab$$
$$B \rightarrow Bb \mid b$$

Note:

If the production is of the form $A \rightarrow \alpha_1, \alpha_2, \ldots \alpha_n$ were $\alpha_i \in NUT$ with $i = 1$ to n, Let α_1 derives ϵ and α_2 derives ϵ

$$\alpha_1 \rightarrow \epsilon$$
$$\alpha_2 \rightarrow \epsilon$$

then,

1) Generates the new set of production by replacing α_1 with ϵ.

2) Generates the new set of production by replacing α_2 with ϵ.

3) Generates the new set of productions by replacing α_1 & α_2 with ϵ

Eg:

1) Eliminate the null productions from following grammar.

$$S \rightarrow AB$$
$$A \rightarrow aAA/\epsilon$$
$$B \rightarrow bBB/\epsilon$$

sol:

(i) Nullable variable

$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$

$\therefore$ A and B are nullable variables

(ii) Generating a new set of production by A with ϵ

$S \rightarrow \epsilon B$	$S \rightarrow B$
$A \rightarrow a\epsilon A \mid a\epsilon\epsilon$	$A \rightarrow aA/a$
$B \rightarrow bBB \mid \cancel{b}\epsilon$	$B \rightarrow bBB/\epsilon$

$S \to AB/B$

$A \to aAA/aA/a$

$B \to bBB/\epsilon$

(iii) Generating a new set of production by B with ϵ

$S \to A\epsilon/\epsilon$	$S \to A/\epsilon$
$A \to aAA/aA/a$	$A \to aAA/aA/a$
$B \to b\epsilon B/b\epsilon\epsilon$	$B \to bB/b$

$S \to AB/B/A/\epsilon$

$A \to aAA/aA/a$

$B \to bBB/bB/b$

S is not present in RHS

$\therefore S \to \epsilon$ is directly eliminated

$S \to AB/B/A$

$A \to aAA/aA/a$

$B \to bBB/bB/b$

---

2) Eliminate the null production

$S \to ABC$

$A \to BC/a$

$B \to bAC/\epsilon$

$C \to CAB/\epsilon$

sol:

(i) Nullable variable

$B \to \epsilon$

$C \to \epsilon$

$\therefore$ B and C are nullable variable.

(ii) Generating a new set of production by B with ϵ

$S \to A\epsilon C$	$S \to AC$
$A \to \epsilon C \mid a$	$A \to C \mid a$
$B \to bAC$	$B \to bAC$
$C \to CA\epsilon \mid \epsilon$	$C \to CA \mid \epsilon$

$S \to ABC \mid AC$
$A \to BC \mid C \mid a$
$B \to bAC$
$C \to CAB \mid CA \mid \epsilon$

(iii) Generating a new set of production by C with ϵ

$S \to AB\epsilon \mid A\epsilon$	$S \to AB \mid A$
$A \to B\epsilon \mid \epsilon \mid a$	$A \to B \mid a \mid \epsilon$
$B \to bA\epsilon$	$B \to bA$
$C \to \epsilon AB \mid \epsilon A \mid \epsilon$	$C \to AB \mid A$

$S \to ABC \mid AC \mid AB \mid A$
$A \to BC \mid C \mid a \mid B \mid \epsilon$
$B \to bAC \mid bA$
$C \to CAB \mid CA \mid AB \mid A$

$A = \epsilon$

$S \to \epsilon BC \mid \epsilon C \mid \epsilon B \mid \epsilon$	$S \to BC \mid C \mid B \mid \epsilon$
$A \to BC \mid C \mid a \mid B$	$A \to BC \mid C \mid a \mid B$
$B \to b\epsilon C \mid b\epsilon$	$B \to bC \mid b$
$C \to C\epsilon B \mid C\epsilon \mid \epsilon B \mid \epsilon$	$C \to CB \mid C \mid B \mid \epsilon$

$S \rightarrow ABC|AC|AB|A|BC|B|C|\cancel{\epsilon}$

$A \rightarrow BC|B|C|a$

$B \rightarrow bAC|bA|bC|b$

$C \rightarrow CAB|CA|AB|A|CB|C|B|\epsilon$

$C = \epsilon$

$S \rightarrow AB\epsilon|A\epsilon|AB|A|B\epsilon|B|\epsilon$

$A \rightarrow B\epsilon|B|\epsilon|a$

$B \rightarrow bA\epsilon|bA|b\epsilon|b$

$C \rightarrow \epsilon AB|\epsilon A|AB|A|\epsilon B|\epsilon|B$

$S \rightarrow AB|A|B|\epsilon$

$A \rightarrow B|a|\epsilon$

$B \rightarrow bA|b$

$C \rightarrow AB|A|B|\epsilon$

Append with G

$S \rightarrow ABC|AC|AB|BC|A|B|C$

$A \rightarrow BC|B|C|a$

$B \rightarrow bAC|bA|bC|b$

$C \rightarrow CAB|CA|AB|A|CB|C|B$

25/10/24

Eliminating ~~null~~ unit production:

* only one N.T present in right side it is unit production. i.e. A
* start symbol la illa tha N.T, production oh iruka kudathu

1) $S \to 0A \mid 1B \mid 011$

$A \to 0S \mid 00$

$B \to 1 \mid A$

$C \to 01$

sol:

C is not in start state. therefore, replace the production of C

$I \to 0$

$X \to 1$

$S \to IA \mid XB \mid IXX$

$A \to IS \mid II$

$B \to X$

$I \to 0$

$X \to 1$

CFG to CNF

1) $S \to bA \mid aB$

$A \to bAA \mid aS \mid a$

$B \to aBB \mid bS \mid b$

sol:

$\times S \to bA$

$\times S \to aB$

$\times A \to bAA$

$\times A \to aS$

$\boxed{A \to a}$ ✓

$\times B \to aBB$

$\times B \to bS$

$\boxed{B \to b}$ ✓

CNF rule

$N.T \to NT\ NT$

$N.T \to T.$

① $S \to bA$

$S \to D_b A$

$D_b \to b$

② $S \to aB$

$S \to D_a B$

$D_a \to a$

③ $A \to bAA$

$A \to D_1 A$

$D_1 \to bA$

$D_1 \to D_b A$

④ $A \to aS$

$A \to D_a S$

⑥ $B \rightarrow aBB$

$B \rightarrow D_2 B$

$D_2 \rightarrow aB$

$D_2 \rightarrow D_a B$

⑦ $B \rightarrow bS$

$B \rightarrow D_b S$

$S \rightarrow D_b A \mid D_a B$

$A \rightarrow D_1 A \mid D_a S \mid a$

$B \rightarrow D_2 B \mid D_b S \mid b$

$D_1 \rightarrow D_b A$

$D_2 \rightarrow D_a B$

$D_a \rightarrow a$

$D_b \rightarrow b$

---

2) $S \rightarrow AAC$

$A \rightarrow aAb \mid \epsilon$

$C \rightarrow aC \mid a$

sol:

(i) Eliminate the ϵ- production

① $S \rightarrow AAC$ $[A \rightarrow \epsilon]$

$S \rightarrow AC$ $[A \rightarrow \epsilon]$

$S \rightarrow C$

② $A \rightarrow aAb$ $[A \rightarrow \epsilon]$

$A \rightarrow ab$

(ii) Eliminate unit production (replace it with production C.)

$S \rightarrow AAC \mid AC \mid C$

$A \rightarrow aAb \mid ab$

$C \rightarrow aC \mid a$

$S \rightarrow AAC \mid AC \mid aC \mid a$

$A \rightarrow aAb \mid ab$

$C \rightarrow aC \mid a$

(iii) Eliminate useless production

Generating:

$S \rightarrow a$

$A \rightarrow ab$

$C \rightarrow a$

Generating = $\{S, A, C\}$

Reachable:

S, A, C are reachable

Reachable = $\{S, A, C\}$

All are useful symbols

$\therefore$ no useless productions.

simplified CFG is

$$\boxed{\begin{array}{l} S \rightarrow AAC \mid AC \mid aC \mid a \\ A \rightarrow aAb \mid ab \\ C \rightarrow aC \mid a \end{array}}$$

$S \rightarrow AAC$

✓ $S \rightarrow AC$

$S \rightarrow aC$

✓ $S \rightarrow a$

$A \rightarrow aAb$

$A \rightarrow ab$

$C \rightarrow aC$

✓ $C \rightarrow a$

(you can use this 'C' also to replace 'a', no problem otherwise introduce a new term)

① $D_1 \rightarrow AA$

$S \rightarrow D_1 C$

③ $D_2 \rightarrow a$

$S \rightarrow D_2 C$

28/10/24

Normal forms of context Free Grammar:

1) Chomsky Normal Form (CNF)

2) Greibach Normal Form (GNF)

Simplification of CFG (for CNF)

carry out in this order

1) Eliminate the null production

2) Eliminate the unit Production

3) Eliminate the useless Production

---

(5) $D_3 \rightarrow b$

$A \rightarrow D_2 A D_3$

$D_4 \rightarrow D_2 A$

$A \rightarrow D_4 D_3$

(6) $A \rightarrow D_2 D_3$

(7) $C \rightarrow D_2 C$

∴ CNF is

$S \rightarrow D_1 C / AC / D_2 C / a$

$A \rightarrow D_4 D_3 \mid D_2 D_3$

$C \rightarrow D_2 C \mid a$

$D_1 \rightarrow AA$

$D_2 \rightarrow a$

$D_3 \rightarrow b$

$D_4 \rightarrow D_2 A$

---

3) convert the CFG to CNF

$S \rightarrow AAA \mid B$

$A \rightarrow aA \mid B$

$B \rightarrow \epsilon$

(i) Eliminate null production

$B \to \epsilon$

$S \to AAA | \epsilon$

$A \to aA | \epsilon$

$$\boxed{\begin{array}{l} S \to AAA | B | \epsilon \\ A \to aA | B | \epsilon \end{array}}$$

(ii) Eliminate unit production

sub $A = \epsilon$

$S \to \epsilon AA | \epsilon\epsilon A | \epsilon\epsilon\epsilon | B$

$A \to a\epsilon | B$

$$\boxed{\begin{array}{l} S \to AAA | AA | A | B | \cancel{\epsilon} \\ A \to aA | a | B \end{array}}$$

no S to sub ϵ

(ii) Eliminate unit production

unit production $S \to A$ | $S \to aA | a | B$.

$S \to B$

$A \to B$

$S \to AAA | AA | aA | a | B$

$A \to aA | a | B$

(iii) Eliminate useless production

Generating:

$S \to a$

$A \to a$

Generating = $\{S, A\}$

Reachable:

S, A are reachable.

B is not reachable

Reachable - $\{S, A\}$

Useful productions are $\{S, A\}$ and $\{S, A\} = \{S, A\}$

useless symbol is B.

so remove production with B

simplified CFG is

$S \to AAA | AA | aA | a$

$A \to aA | a$

$S \to AAA$

✓ $S \to AA$

$S \to aA$

✓ $S \to a$

$A \to aA$

✓ $A \to a$

① $D_1 \to AA$

$S \to D_1 A$

③ $D_2 \to a$

$S \to D_2 A$

⑤ $A \to D_2 A$

∴ CNF is

$S \to D_1 A | AA | D_2 A | a$

$A \to D_2 A | a$

$D_1 \to AA$

$D_2 \to a$.

4) convert the CFG to CNF

$S \rightarrow 0A0 \mid 1B1 \mid BB$

$A \rightarrow C$

$B \rightarrow S \mid A$

$C \rightarrow S \mid \epsilon$

(i) Eliminate null production

$C = \epsilon$

$S \rightarrow 0A0 \mid 1B1 \mid BB$	$S \rightarrow 0A0 \mid 1B1 \mid BB$
$A \rightarrow \epsilon$	$A \rightarrow \epsilon$
$B \rightarrow S \mid A$	$B \rightarrow S \mid A$
$C \rightarrow S$	$C \rightarrow S$

Append with G

$S \rightarrow 0A0 \mid 1B1 \mid BB$

$A \rightarrow C \mid \epsilon$

$B \rightarrow S \mid A$

$C \rightarrow S$

$A = \epsilon$

$S \rightarrow 0\epsilon0 \mid 1B1 \mid BB$	$S \rightarrow 00 \mid 1B1 \mid BB$
$A \rightarrow C$	$A \rightarrow C$
$B \rightarrow S \mid \epsilon$	$B \rightarrow S \mid \epsilon$
$C \rightarrow S$	$C \rightarrow S$

Append with G

$S \rightarrow 0A0 \mid 1B1 \mid BB \mid 00$

$A \rightarrow C$

$B \rightarrow S \mid A \mid \epsilon$

$C \rightarrow S$

$B = \epsilon$

$S \rightarrow 0A0 \mid 1\epsilon1 \mid \epsilon B \mid \epsilon\epsilon \mid 00$	$S \rightarrow 0A0 \mid 11 \mid B \mid 00 \mid \epsilon$
$A \rightarrow C$	$A \rightarrow C$
$B \rightarrow S \mid A$	$B \rightarrow S \mid A$
$C \rightarrow S$	$C \rightarrow S$

$S \to 0A0|1B1|BB|B|11|00$

$A \to C$

$B \to S|A$

$C \to S$

(ii) Eliminate unit production

$S \to 0A0|1B1|BB|11|00$

$A \to 0A0|1B1|BB|11|00$

$B \to 0A0|1B1|BB|11|00$

$C \to 0A0|1B1|BB|11|00$

(iii) Eliminate useless Production

Generating = {S, A, B, C}

Reachable = {S, A, B}

usefull production = {S, A, B, C} ∩ {S, A, B} = {S, A, B}

∴ C is useless production, so remove production with C

$S \to 0A0|1B1|BB|11|00$

$A \to 0A0|1B1|BB|11|00$

$B \to 0A0|1B1|BB|11|00$

$S \to 0A0$

$S \to 1B1$

✓ $S \to BB$

$S \to 11$

$S \to 00$

(1) $D_1 \to 0$

$S \to D_1 A D_1$

$D_2 \to D_1 A$

$S \to D_2 D_1$

(2) $D_3 \to 1$

$S \to D_3 B D_3$

$D_4 \to D_3 B$

$S \to D_4 D_3$

(4) $S \to D_3 D_3$

(5) $S \to D_1 D_1$

∴ The CNF is

$S \to D_2 D_1 | D_4 D_3 | BB | D_3 D_3 | D_2 D_2$

$A \to D_2 D_1 | D_4 D_3 | BB | D_3 D_3 | D_2 D_2$

$B \to D_2 D_1 | D_4 D_3 | BB | D_3 D_3 | D_2 D_2$

Doubt not check it H.W sum

4/11/24

Greibach Normal Form (GNF)

The context free Grammar is reduced to GNF, if alpha production in G is of the form $\underset{N.T}{A} \rightarrow a\underset{(NUT)^*}{\alpha}$

steps to convert CFG to GNF:

step 1 - convert the CFG to CNF

step 2 - Rename the non-terminal in G to $A_1, A_2, \ldots A_n$

step 3 - Modify the production with new non-terminals

step 4 - Examine whether each production in G is of the form $A_i \rightarrow A_j x$ where $i > j$. Apply lemma 1 for these types of Production.

Lemma 1:

If there is a production $A \rightarrow \alpha_1 B \alpha_2$ ~~$B \rightarrow \alpha_2$~~ in G then the new grammar is formed by replacing B with its production.

step 5 - Repeat the process until every production in G is converted into the form $A_i \rightarrow A_j x$ with $i \leq j$. After converting if there is production in G $A_i \rightarrow A_j x$ with $i = j$ then Apply lemm 2.

Lemma 2:

If the production in G is of the form

$A \rightarrow A\alpha_1 | A\alpha_2 | \ldots | A\alpha_n \beta_1 | \beta_2 \quad | \beta_n$,

introducing a new non-terminal say 'x' and form a new set of productions as follows.

$A \to \beta_1 | \beta_2 | \dots | \beta_n$

$A \to \beta_1 X | \beta_2 X | \dots | \beta_n X$

$X \to \alpha_1 | \alpha_2 | \dots | \alpha_n$

$X \to \alpha_1 X | \alpha_2 X | \dots | \alpha_n X$

Eg:

$S \to SaA | Sbb | aA | b$

$A = S$ $\alpha_1 = aA$ $\alpha_2 = bb$ $\beta_1 = aA$ $\beta_2 = b$

$S \to aA | b$

$S \to aAX | bX$

$X \to aA | bb$

$X \to aAX | bbX$

step-7: convert all the productions into the form $A \to a\alpha$

Problems

16M

1) convert the given CFG to GNF

$S \to AA | a$

$A \to SS | b$

sol:

step-1: CFG to CNF

Already in the form CNF

$S \to AA | a$

$A \to SS | b$

step-2: Rename non-terminals

S as A_1 and A as A_2

step-3: $A_1 \to A_2 A_2 | a$

$A_2 \to A_1 A_1 | b$

step-3: Modify the new non-terminals

step-4: check the productions in G

$A_1 \to A_2 A_2 \mid a \quad (i < j)$

$A_2 \to \underline{A_1} A_1 \mid b \quad (i > j)$ Apply lemma 1

step-5:

$A_2 \to A_2 A_2 A_1 \mid a A_1 \mid b \quad (i < j)$ (replace only 1st NT)

step-6:

$A_2 \to A_2 A_2 A_1 \mid aA_1 \mid b$ here $i = j$

Apply lemma 2

$A = A_2 \quad \alpha_1 = A_2 A_1 \quad \beta_1 = aA_1 \quad \beta_2 = b$

$A_2 \to aA_1 \mid b$

$A_2 \to aA_1 X \mid bX$

$X \to A_2 A_1$

$X \to A_2 A_1 X$

~~step-7:~~

$A_1 \to A_2 A_2 \mid a$

$A_2 \to aA_1 \mid b \mid aA_1 X \mid bX$

$X \to A_2 A_1 \mid A_2 A_1 X$

step-7: $A \to a\alpha$

$A_1 \to \underline{A_2} A_2 \mid a$

$A_1 \to aA_1 A_2 \mid bA_2 \mid aA_1 X A_2 \mid bXA_2 \mid a$

$A_2 \to aA_1 \mid b \mid aA_1 X \mid bX$

$X \to aA_1 A_1 \mid bA_1 \mid aA_1 X A_1 \mid bXA_1 \mid aA_1 A_1 X \mid bA_1 X \mid$
$aA_1 X A_1 X \mid bXA_1 X$

2) convert the CFG to GNF

$A \rightarrow aB / b$

$B \rightarrow c$

sol: It is already in the form GNF i.e starts with terminal symbol

step-1: CFG to GNF

3) convert the CFG to GNF

$A \rightarrow BaC$

$B \rightarrow b$

$C \rightarrow a$

step-1: convert CFG to CNF

Already the grammar is simplified

$A \rightarrow BCC$

$B \rightarrow b$

$C \rightarrow a$

$A \rightarrow BD$

$B \rightarrow b$

$C \rightarrow a$

$D \rightarrow CC$

step-2: Rename non-terminals

A as A_1, B as A_2, C as A_3, D as A_4

step-3: Modify the new non-terminals

$A_1 \rightarrow A_2 A_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

$A_4 \rightarrow A_3 A_3$

step-4: check the productions in G

$A_1 \rightarrow A_2 A_4 \quad (i < j)$

$A_4 \rightarrow A_3 A_3 \quad (i > j)$ Apply lemma 1 step-5:

step-6: No production have $i > j$

$A_4 \rightarrow aA_3$

step-7: $A \rightarrow a\alpha$

$A_1 \rightarrow A_2A_4 \rightarrow A_1 \rightarrow bA_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a$

$A_4 \rightarrow aA_3$

---

4) convert the CFG to GNF

$S \rightarrow AB$

$A \rightarrow BS | b$

$B \rightarrow SA | a$

step-1: CFG to CNF

Already in the form CNF

$S \rightarrow AB$

$A \rightarrow BS | b$

$B \rightarrow SA | a$

step-2: Rename non-terminals

S as A_1, A as A_2, B as A_3

step-3: Modify the new non-terminals

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 | b$

$A_3 \rightarrow A_1A_2 | a$

step-4: check the productions in G

$A_1 \rightarrow A_2A_3 \quad (i < j)$

$A_2 \rightarrow A_3A_1 | b \quad (i < j)$

$A_3 \rightarrow A_1A_2 | a \quad (i > j)$ Apply lemma 1

step-5:

$A_3 \rightarrow A_2A_3A_2 | a \quad (i > j)$ Apply lemma 1

$A_3 \rightarrow A_3A_1A_3A_2 | bA_3A_2 | a \quad (i \leq j)$

step-6:

$A_3 \rightarrow A_3A_1A_3A_2 \mid bA_3A_2 \mid a \quad (i = j)$

Apply lemma 2

$A = A_3, \ \alpha_1 = A_1A_3A_2, \ \beta_1 = bA_3A_2, \ \beta_2 = a$

$A_3 \rightarrow bA_3A_2 \mid a$

$A_3 \rightarrow bA_3A_2X \mid aX$

$X \rightarrow A_1A_3A_2$

$X \rightarrow A_1A_3A_2X$

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 \mid b$

$A_3 \rightarrow bA_3A_2 \mid a \mid bA_3A_2X \mid aX$

$X \rightarrow A_1A_3A_2 \mid A_1A_3A_2X$

step-7: $A \rightarrow a\alpha$

$A_1 \rightarrow A_3A_1A_3 \mid bA_3$

$A_1 \rightarrow bA_3A_2A_1A_3 \mid aA_1A_3 \mid bA_3A_2XA_1A_3 \mid aXA_1A_3 \mid bA_3$

$A_2 \rightarrow bA_3A_2A_1 \mid aA_1 \mid bA_3A_2XA_1 \mid aXA_1 \mid b$

$A_3 \rightarrow bA_3A_2 \mid a \mid bA_3A_2X \mid aX$

$X \rightarrow A_2A_3A_3A_2 \mid A_2A_3A_3A_2X$

$X \rightarrow A_3A_1A_3A_3A_2 \mid bA_3A_3A_2 \mid A_3A_1A_3A_3A_2X \mid bA_3A_3A_2X$

$X \rightarrow bA_3A_2A_1A_3A_3A_2 \mid aA_1A_3A_3A_2 \mid bA_3A_2XA_1A_3A_3A_2 \mid bA_3A_2A_1A_3A_3A_2X \mid aA_1A_3A_3A_2X \mid aXA_1A_3A_3A_2 \mid bA_3A_2XA_1A_3A_3A_2X \mid aXA_1A_3A_3A_2X$

6/11/24

Turing Machines

* Real computational model
* works like a human brain

The model designed to accept the phrase structured grammar (type 0) is a turing model, which is based on how a human brain approaches to solve a problem.

Phrase structured grammar or type 0 grammar is a grammar without any restriction.

Components of turing machines:

* Finite control Block with a read write head.
* A linear tape which is divided into cells, can hold one symbol in a cell with leftend block
* The read write head is bidirectional i.e, can move backward and forward.

Formal Definition Turing Machines:

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

where Q - finite set of states

Σ - set of input symbols

Γ - set of tape $(N \cup T \cup B)$

δ is a transition symbol which is defined as $\delta(q, a) = (P, b, D)$ which means that when the state 'q' reads 'a' it moves to the state 'p' replacing 'a' as 'b' in left or right or halt

D: L, R, H

q_0 - initial state of turing machine

B - Blank symbol

F - set of final states.

Instantaneous Description (ID):

The ID of the turing machine is given as (α_1, q, α_2) where α_1, α_2 are input symbols and q is a state.

Let the ID of turing machine be

$$(\underbrace{x_1 x_2 \dots x_{i-1}}_{\alpha_1} \underbrace{q}_{q} \underbrace{x_i x_{i+1} \dots x_n}_{\alpha_2})$$

① If $\delta(q, x_i) = (P, y, L)$ then

$ID = (x_1 x_2 \dots P x_{i-1} y x_{i+1} \dots x_n)$

② If $\delta(q, x_i) = (P, y, R)$

$ID = (x_1 x_2 \dots x_{i-1} y P x_{i+1} \dots x_n)$

Eg:

if $ID = (aaba\, q_1\, baabb)$ — L, R

for the transition

(i) $\delta(q_1, b) = (q_2, X, L)$ find ID

$ID = (aab q_2 a X aabb)$

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

$ID = (aaba Y q_2 aabb)$

Language of Turing Machine:

The language accepted by the turing machine is defined as $L(M) = \{ w \mid w \in \Sigma^* \ \& \ q_0 w \vdash^* \alpha_1 P \alpha_2$ for some P in $F \}$

Note: when a turing Machine reads an input string it enters into three states,

1) It reads a string, enters into the final state and accept the string.

2) It reads a string, enters into the non-accepting state and rejects the string.

3) Enter into an infinite state.

8/11/24

Problems

1) Design a turing machine for language $L = \{a^n b^n / n > 0\}$

sol:

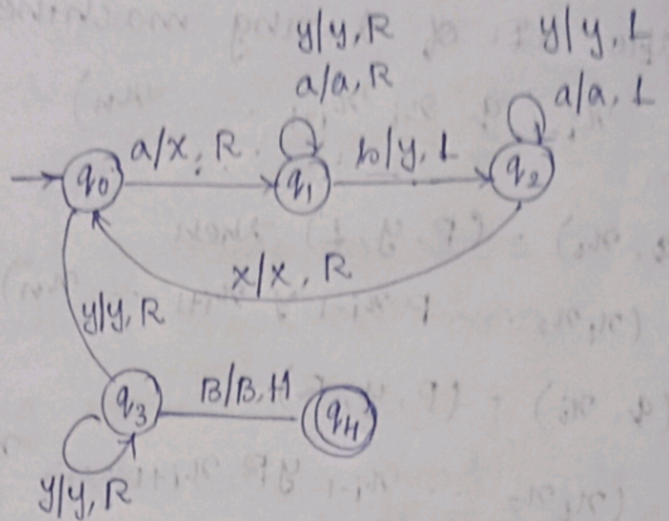
$L = \{ab, aabb, aaabbb \ldots\}$

a a b b B B

x a y

x a

X X Y Y



The components of turing machines are.

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

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

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

$\Gamma = \{a, b, x, Y, B\}$

$q_0 = q_0$

$B = B$

$F = \{q_4\}$

δ is defined as

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

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

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

$\delta(q_1, b) = (q_1, Y, L)$

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

$\delta(q_2, Y) = (q_2, Y, L)$

$\delta(q_2, X) = (q_0, X, R)$

$\delta(q_0, y) = (q_3, y, R)$

$\delta(q_3, y) = (q_3, y, R)$

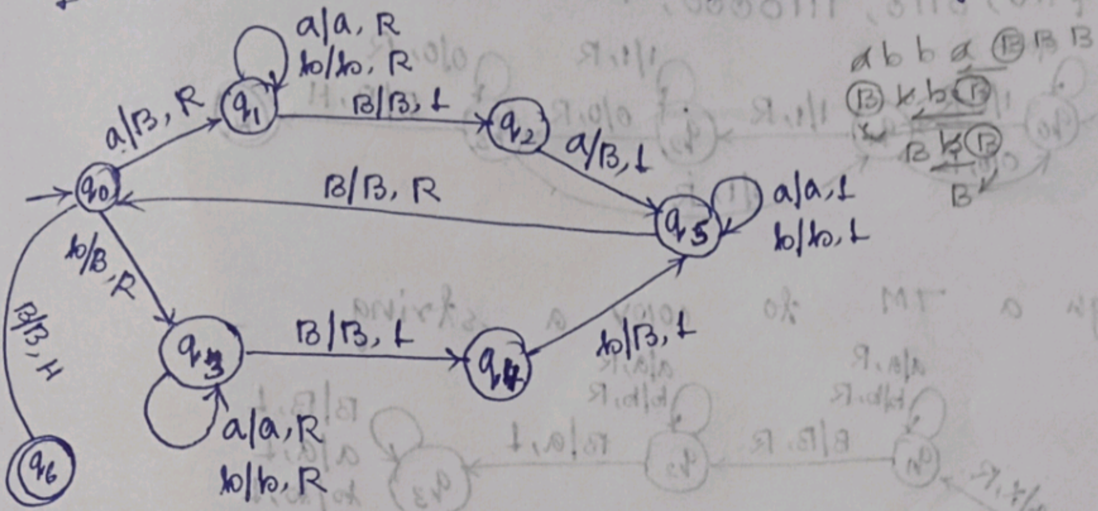
$\delta(q_3, B) = (q_4, B, H)$

11/11/24

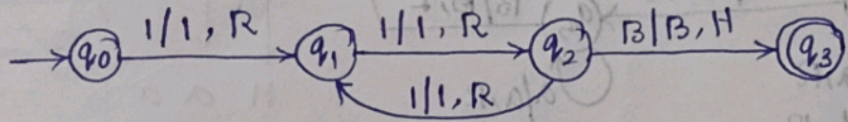
2) Design a Turing Machines $L = \{a^n b^n c^n \mid n > 0\}$

3) Design a TM $L = \{ww^R \mid w \in \{a, b\}^+\}$

sol: $L = \{aa, bb, abba, baab, aaaa, bbbb, \ldots\}$



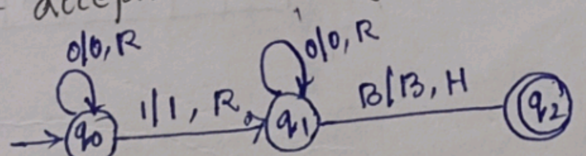
4) Design a TM that accepts strings with even no. of 1's {0, 1}



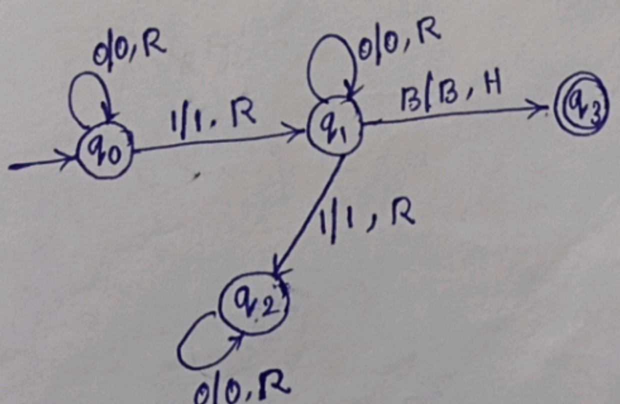
5) TM - accepts strings with atleast one 1 over {0, 1}

6) TM - accepts exactly one 1 over {0, 1}

5)



6)

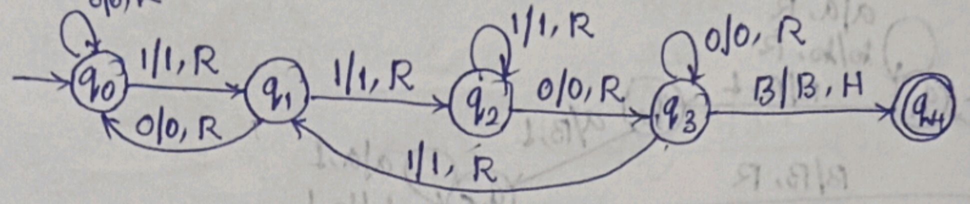


3) ~~Enters into an~~ infinite - ~~state~~.

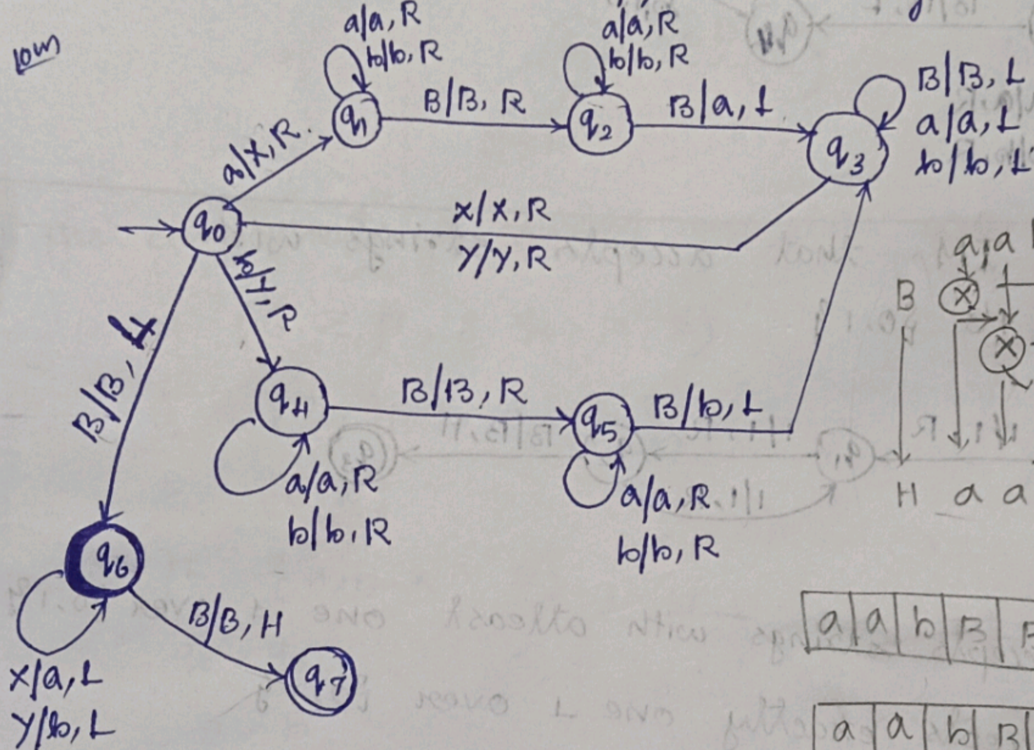
12/11/24

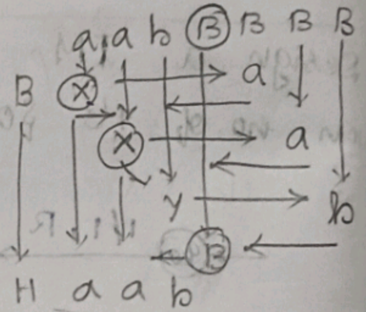
7) TM - accepts 110 as a substring

L = {110, 0110, 1110000, 110000, 110111, ...}



8) Design a TM to copy a string.





a	a	b	B	B	B	B

a	a	b	B	a	a	B

12/11/24

TOC

Recursively Enumerable Language:

The ~~strings~~ language accepted by the turing machine is called Recursively Enumerable language.

The turing machine viewed as a function of positive to positive integer. Each integer in TM is represented as zero's.

Eg: The number 3 in TM is represented as 000.

computable Function:

A function 'f' is defined from N to N is represented as $f: N \rightarrow N$ is said to be a computable function of K arguments.

If there exist a TM 'M' that halts on the tape consisting of 0^m for some m as integer then it is represented as $f(i_1, i_2, i_3, \ldots i_K) = m$

Total Recursive Function:

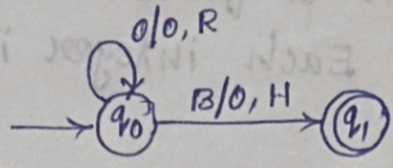
If the function $f(i_1, i_2, i_3, \ldots i_K) = m$ is defined for all items $i_1, i_2, \ldots i_K$ then f is a total recursive function.

Partial Recursive Function:

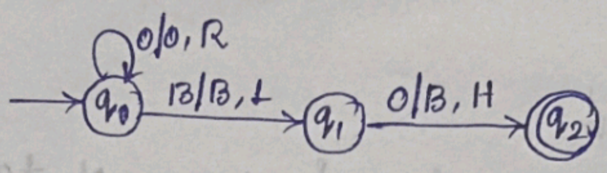
If a function $f(i_1, i_2, \ldots i_K) = m$ is not defined for $i_1, i_2, \ldots i_K$ then it is partial recursive function. They are computated by the TM that may or may not halt.

Problem:

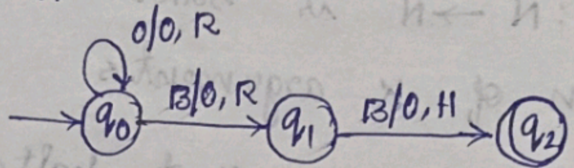
1) Let f be a recursive function with $f(n) = n+1$ where n is a positive integer. Design a TM that computes $f(n)$.



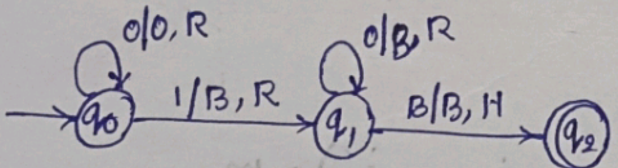
2) Design a TM that computes $f(n) = n-1$



3) $f(n) = n+2$



4) $f(x, y) = x$, x and y are +ve integers



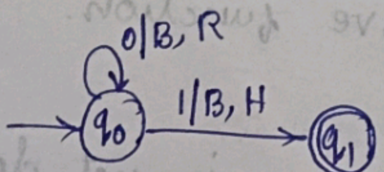
$f(2, 3) = 2$

0	0	1	0	0	0	B	B

⇓

0	0	B	B	B	B	B	B

5) $f(x, y) = y$



$f(2, 3) = 3$

0	0	1	0	0	0	B	B	B

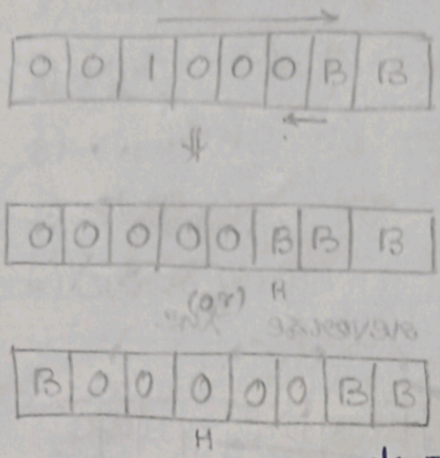
⇓

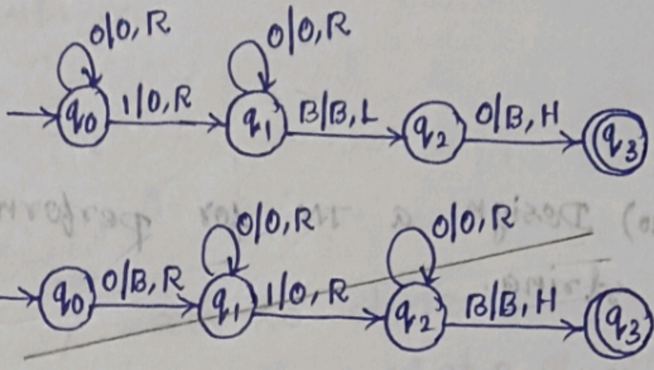
B	B	B	0	0	0	B	B	B

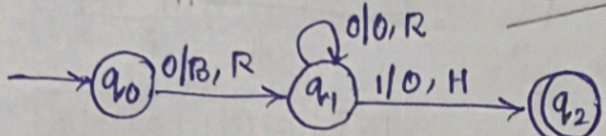
13|11|24

6) Design a TM compute $f(m,n) = m+n$

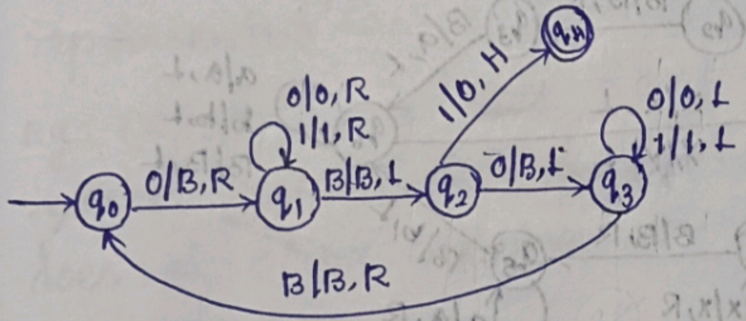
$f(2,3) = 5$

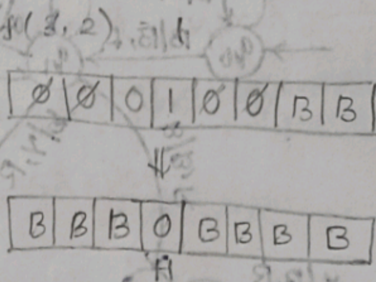




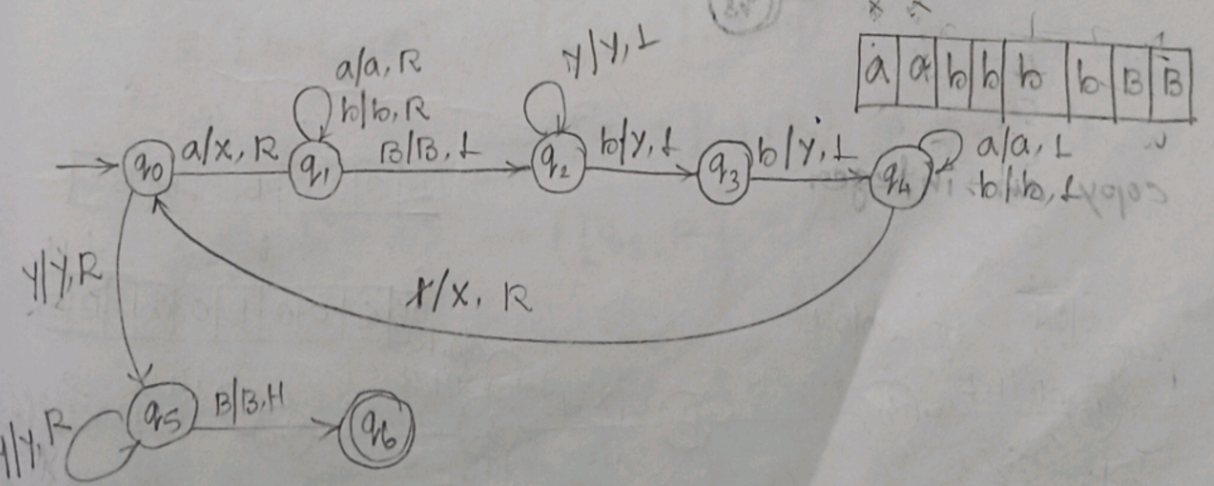


7) $f(m,n) = m-n$, $m \geq n$





8) TM for $L = \{a^n b^{2n} \mid n > 0\}$



9) TM for $L = \{a^{2n} b^n \mid n > 0\}$

a	a	a	a	b	b	B

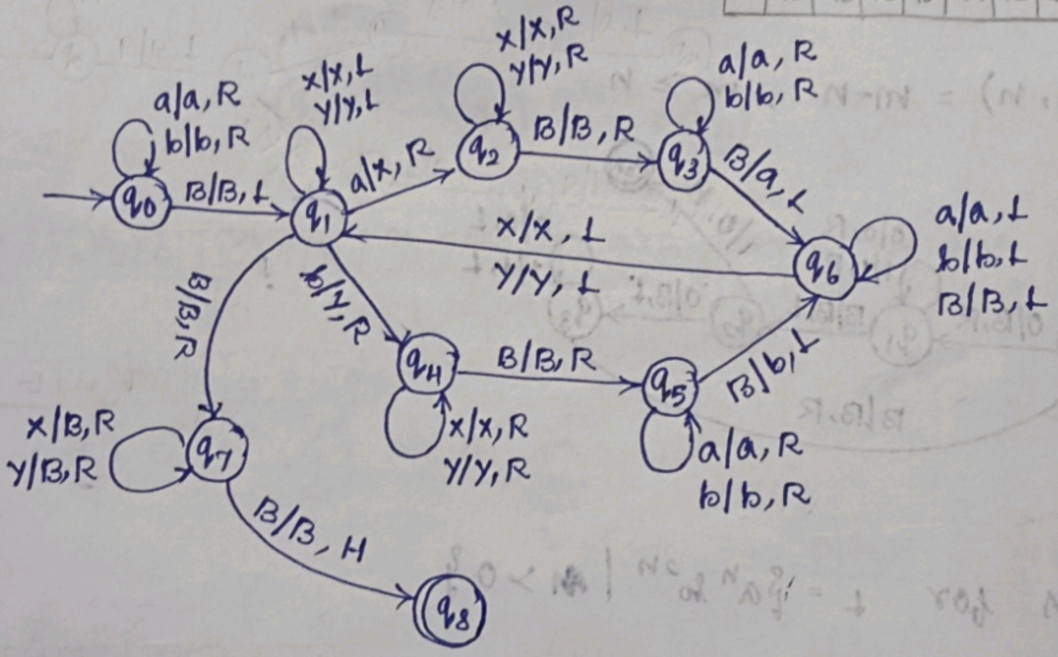
10) Design a TM for perform reverse the string.

B	a	b	B	B	B	B	B

⇓

B	B	B	B	b	a	B	B

14/11/24

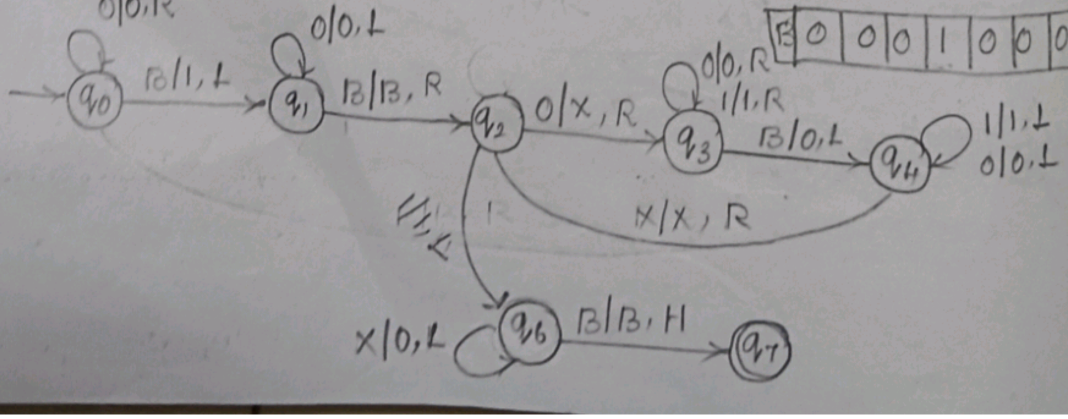


11) copy an integer

B	0	0	0	B	B	B	B

⇓

B	0	0	0	1	0	0	0	B	B



16/11/24

construction of Turing Machine:

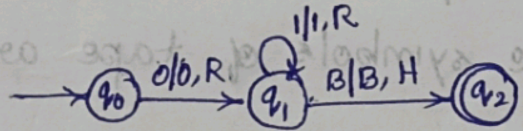
A turing machine components (tools) used for construction are

1) storage in finite control
2) Multi track turing machinge
3) checking of symbols
4) shifting over
5) subroutinges.

storage in Finite control:

The finite control is used to store some finite information about the state of the TM. Instead of representing the transition of a state in single symbol, here it is represented as a pair of ~~symbols~~ elements.

Eg: Design a TM that takes the first input symbol, records it and checks that the symbol does not occur elsewhere in the string



$$\delta([q_0, B], 0) \rightarrow ([q_1, 0], 0, R)$$

$$\delta([q_1, 0], 1) \rightarrow ([q_1, 0], 1, R)$$

$$\delta([q_1, B], B) \rightarrow ([q_2, B], B, H)$$

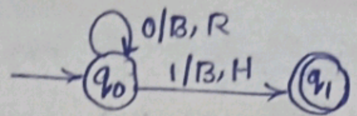
2) Design a TM to compute $f(x, y) = y$ using storage in finite control.

$f(2, 3) = 3$

0	0	1	0	0	0	B	B

↓

B	B	B	0	0	0	B	B



$\delta([q_0, B], 0) \rightarrow ([q_0, B], B, R)$

$\delta([q_0, B], 1) \rightarrow ([q_1, 1], B, H)$

Multi track turing Machine:

Design a turing machine to check whether the input string is prime or not using multi track turing Machine. consider the number of track is 3.

1	¢	0	1	0	0	$	B
2	B	0	0	1	0	B	B
3	B	0	1	0	0	B	B

step-1: Write the number to be checked in the 1st track of the tape in binary form. It should be start with ¢ and end with $.

step-2: The allowable symbols of tape are ¢, 0, 1, $, B

step-3: To check whether the input is prime or not. Place the binary value of 2 in the 2nd track. copy the first track to the 3rd track.

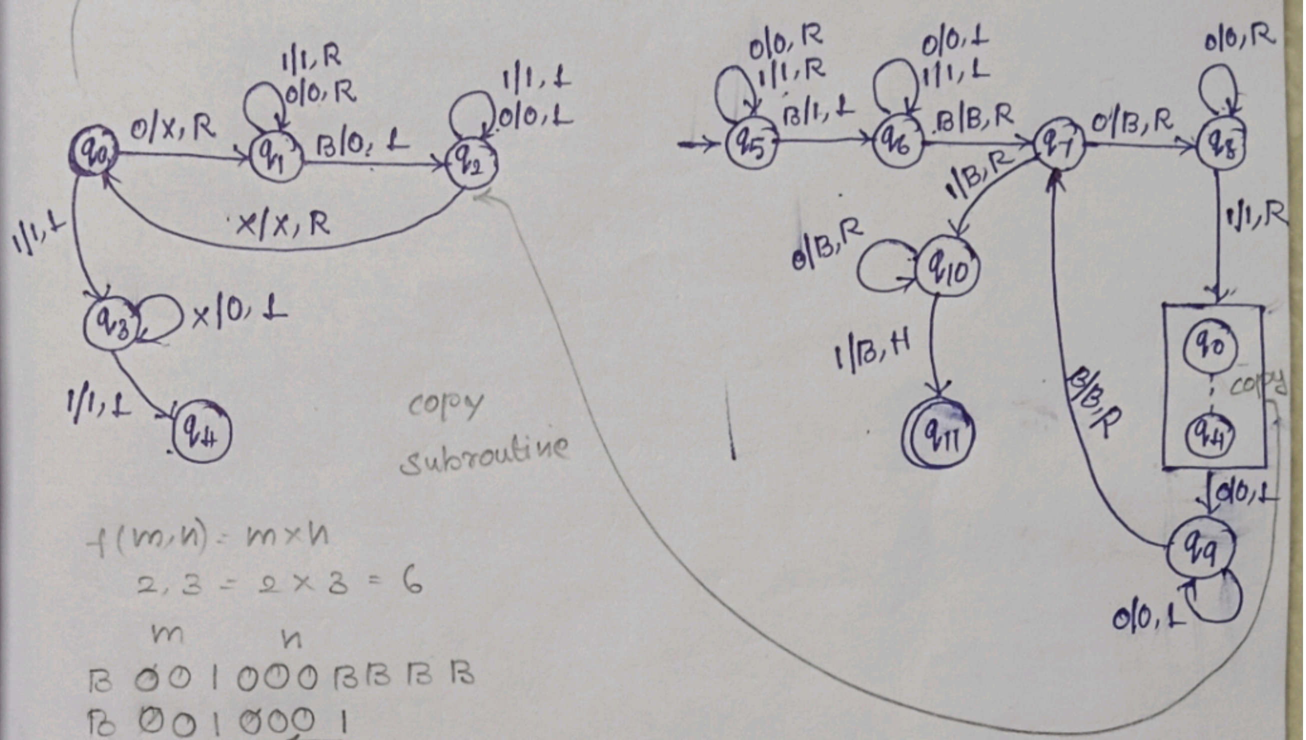
step-4: subtract the 2nd track from the 3rd track repeatedly until the 3rd track becomes zero or less than the 2nd track.

step-5: If the 3rd track becomes zero then declare it is not prime.

step-6: when the 3rd track is not equal to zero then increasement the 2nd track by one and repeat the process of subtraction until n-1.

18/11/24

Multiplication using subroutine



$f(m,n) = m \times n$

2, 3 = 2 × 3 = 6

m n

B Ø0 1 000 BBBB

B Ø0 1 000 1

B BØ 1 X 00 1 0

X Ø 1 0 0

X 1 0 0 0

1 B 0 1 Ø 0 0

B 1 X Ø 0 1 0 0 0 0

X Ø 1 0 0 0 0 0

X 1 0 0 0 0 0 0

B 1 0 0 0 1

BBBBB

H