Turing Machine

1) TM to accept all the strings containing substring aba.

2) TM to accept all the strings ending with abb.

Turing machine as language accepters

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ be a TM.
Then the language accepted by M is

$$L(M) = \{ w \in \Sigma^+ : q_0 w \vdash^* x_1 q_f x_2 \text{ for some } q_f \in F, x_1, x_2 \in \Gamma^* \}.$$

Turing machine as Transducers

A function f with Domain D is said to be Turing-computable or just computable if there exists some Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ such that

$$q_0 w \vdash^*_M q_f f(w), \quad q_f \in F$$

for all $w \in D$.

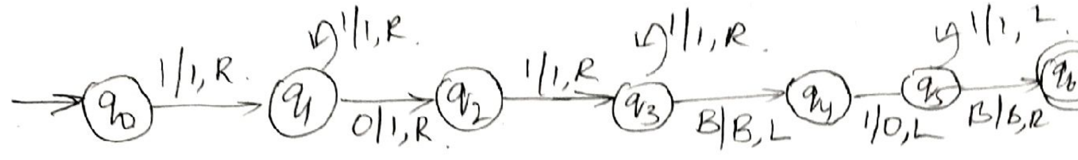
3) Build TM that accepts the language.
$L = \{ a^n b^{n+1} \}$

4) $L = \{ a^n b^{2n} \mid n \geq 0 \}$.

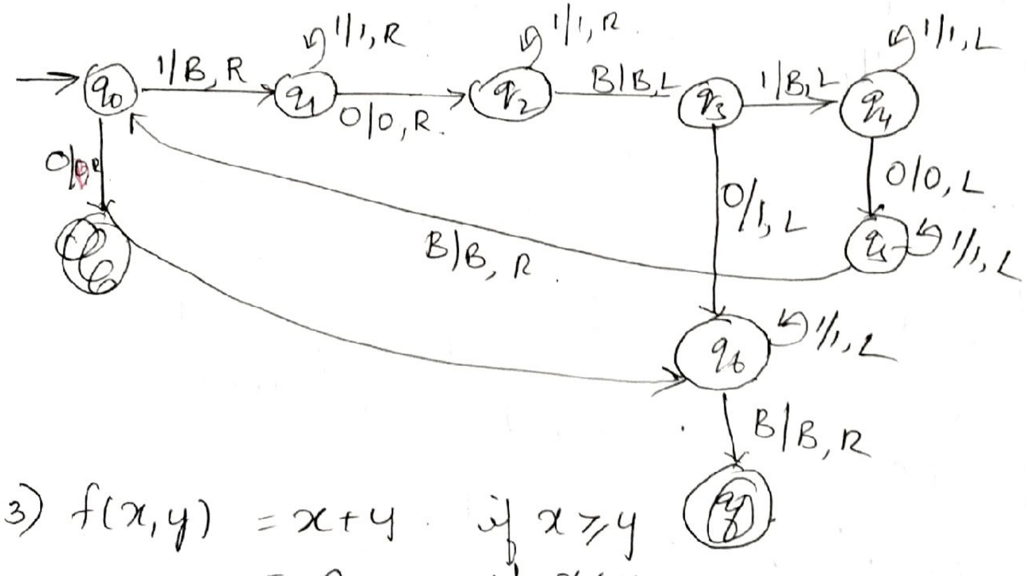
1) Given two positive integers x and y, design a TM that computes $x+y$.

$w(x) \in \{1^+\}$

$q_0 w(x) 0 w(y) \vdash^* q_f w(x+y) 0$.



2) Given two positive integer x and y where $x \geq y$. perform $x-y$



3) $f(x,y) = x+y$ if $x \geq y$
$= 0$. if $x < y$.

4) $f(x) = x \bmod 5$.

Recursive and Recursively Enumerable Languages.

A language L is said to be recursively enumerable if there exists a TM that accepts it. Says nothing about what happens for w not in L.

i.e $q_0 w \vdash^*_M x_1 q_f x_2$. for all $w \in L$.

A language L on Σ is said to be recursive if there exists a TM. M that accepts L and that halts on every w in Σ^+.

i.e language is recursive if there exists a membership algorithm for it.

Chomsky hierarchy

Exhibits relationship among families.

L_{RE} - Recursively Enumerable.

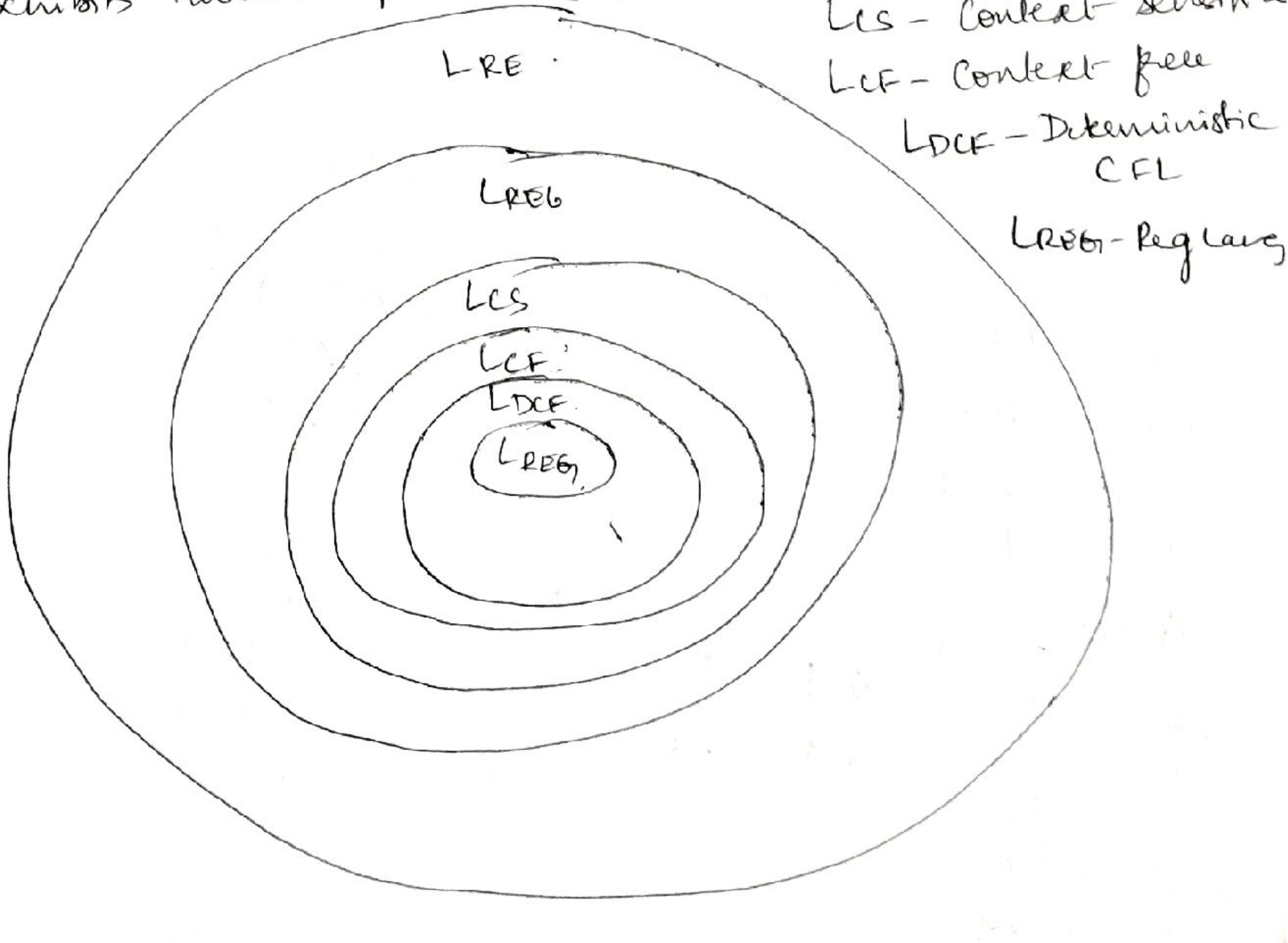
L_{REC} - Recursive lang.

L_{CS} - Context Sensitive

L_{CF} - Context free

L_{DCF} - Deterministic CFL

L_{REG} - Reg Lang



Context sensitive grammar

A grammar G is said to be context sensitive grammar if all productions are of the form

$\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$

$\alpha, \beta \in \{V \cup T\}^+$ with α containing at least one element in V.

Any CFG without ϵ production is context sensitive grammar. But every CSG is not CFG.

G is said to be CS iff it can be generated by a grammar in which every production has the form, $\alpha A \beta \rightarrow \alpha x \beta$, $\alpha, x, \beta \in (V \cup T)^+$ & $x \neq \phi$.

Ex1 $L = \{a^n b^n c^n \mid n \geq 1\}$.

(V1) $S \rightarrow ABCS_1 \mid ABC$

$S_1 \rightarrow ABCS_1 \mid ABC$

$BA \rightarrow AB$

$CA \rightarrow AC$

$CB \rightarrow BC$

$A \rightarrow a$

$aA \rightarrow aa$

$aB \rightarrow ab$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

(V2) $S \rightarrow aSBc \mid abc$

$cB \rightarrow Bc$

$bB \rightarrow bb$.

$S \rightarrow ABCS_1$

$S \rightarrow ABCABC$

$S \rightarrow ABACBC$

$S \rightarrow AABCBC$

$S \rightarrow AABBCC$

$S \rightarrow aABBCC$

$S \rightarrow aaBBCC$

$S \rightarrow aabBCC$

$S \rightarrow aabbCC$

$S \rightarrow aabbcC$

$S \rightarrow aabbcc$.

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

$S \to Aa$

$A \to aAc \mid B \mid b$

$B \to bBX \mid b$

$Xc \to cX$

$Xa \to aa$

$S \to Aa$

$S \to aAca [A \to aAc]$

$S \to abca$

To derive aabcca.

$S \to Aa$

$S \to aACa [A \to aAC]$

$S \to aaACCa [A \to aAC]$

$S \to aabcca [A \to b]$

$S \to aabcca$

3) $S \to aS'bX \mid abX$

$S' \to aS'bC \mid S'bC \mid S'C \mid C$

$Cb \to bC$

$CX \to Xc$

$X \to c$

$L = \{a^i b^j c^k \mid 1 \leq i \leq j \leq k\}$

To derive abbbccc

$S \to aS'bX$

$\to aS'bCbX [S' \to S'bC]$

$\to abCbCbX [S' \to bC]$

$\to abbCCbX [Cb \to bC]$

$\to abbCbCX$ — " —

$\to abbbCCX$ — " —

$\to abbbCXc$

$\to abbbXcc \to abbbccc$

Context Sensitive Grammar

$S \to abc \mid aAbc$

$Ab \to bA$

$Ac \to Bbcc$

$bB \to Bb$

$aB \to aa \mid aaA$

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

$S \to aAbc$

$\to abAc$

$\to abBbcc$

$\to aabbAcc$

$\to aabbBbccc$

$\to aabBbbccc$

$\to aaBbbbccc$

$\to aaabbbccc$

Unrestricted grammar

A grammar $G = (V, T, P, S)$ is said to be type 0 or unrestricted if all the productions are of the form $\alpha \to \beta$
$\alpha \in (V \cup T)^+$ and $\beta \in (V \cup T)^*$

1. Consider the following unrestricted grammar. Identify the language generated by this grammar.

$S \to S_1 B$
$S_1 \to a S_1 b$
$bB \to bbbB$
$aS_1 B \to aa$
$B \to \epsilon$

$S \to S_1 B$.
$S \to aS_1 bB$ $[S_1 \to aS_1 b]$
$S \to aaS_1 bbB$.
$S \xRightarrow{*} a^n S_1 b^n B$.
$S \to a^{n+1} b^{n+1} B$
$S \to a^{n+1} b^{n+1} bbB$
$S \to a^{n+1} b^{n+1} bbbbB$.

$\therefore L = \{a^{n+1} b^{n+k}, n \geq 1$
$K = -1, 1, 3 \ldots \}$.

2. $L = \{a^n b^n c^n, n \geq 0\}$

$S \to aBSc$
$S \to \epsilon$
$Ba \to aB$
$Bc \to bc$
$Bb \to bb$

$S \to aBSc$
$S \to aBaBScc$
$S \to aaBBScc$
$S \to aaBBcc$
$S \to aaBbcc$
$S \to aabbcc$

3) $L = \{a^n b^n c^n : n > 0\}$

$S \to ABCS$

$S \to T_c$

$CA \to AC$

$BA \to AB$.

$CB \to BC$

$CT_c \to T_c c$

$T_c \to T_B$

$BT_B \to T_B b$

$T_B \to T_A$

$AT_A \to T_A a$

$T_A \to \epsilon$.

$S \to ABCS$
$\to ABCABCS$
$\to ABACBCS$
$\to AABCBCS$
$\to AABBCCS$
$\to AABBCCT_c$
$\to AABBCT_c c$
$\to AABBT_c cc$
$\to AABBT_B cc$
$\to AABT_B bcc$
$\to AAT_B bbcc$
$\to AAT_A bbcc$
$\to AT_A abbcc$
$\to T_A aabbcc$
$\to aabbcc$.

4) $L = \{ww : w \in a, b\}$.

$S \to S'Z$

$S' \to aS'A \mid bS'B \mid \epsilon$

$AZ \to XZ$ $\quad BZ \to YZ$ $\quad Z \to \epsilon$.

$AX \to XA$ $\quad AY \to YA$

$BX \to XB$. $\quad BY \to YB$.

$aX \to aa$. $\quad aY \to ab$.

$bX \to ba$. $\quad bY \to bb$.

5) $L = \{a(aa)^n \mid n \geq 0\}$.

$S \to AS \mid aT$

$Aa \to aaaA$

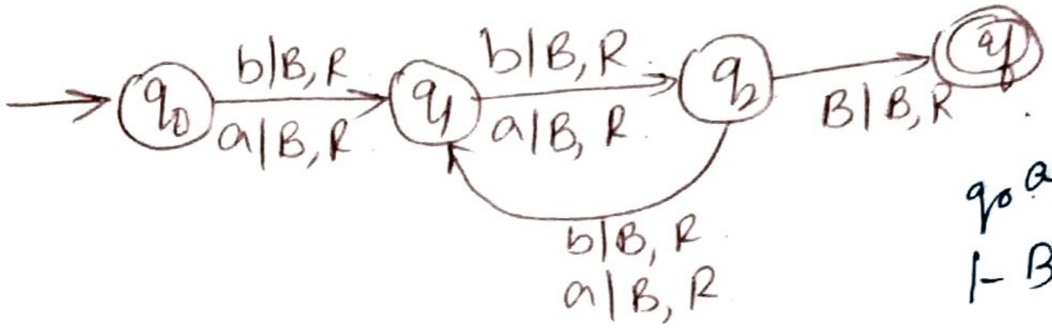
$AT \to T$

$T \to \epsilon$.

TM to accept language

1) $L = \{w, \ |w| \text{ is even}\} \ w \in \{a,b\}^*$

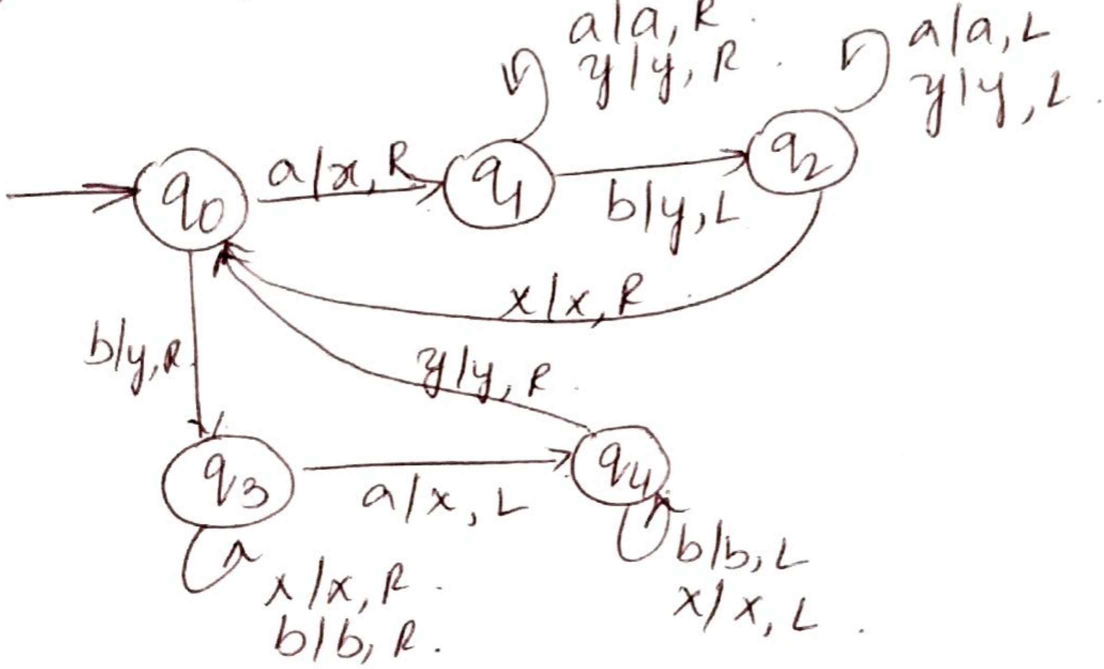
$|w| > 0$



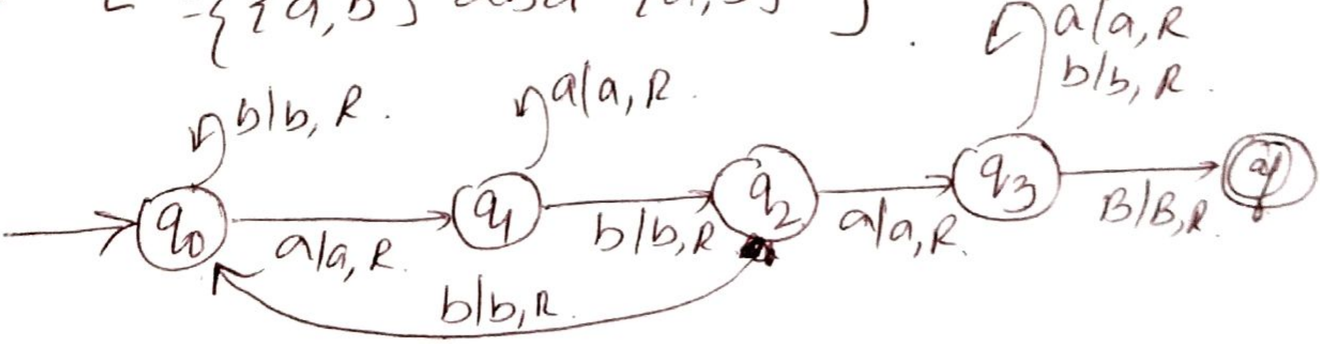
$q_0 abab$

$\vdash B q_1 bab \vdash BB q_2$

2) $L = \{w : n_a(w) = n_b(w)\}$



3) $L = \{\{a,b\}^* aba \{a,b\}^*\}$



Chomsky Hierarchy

Noam chomsky founder of FL classifies grammar into following categories.

Type 0 grammar (Phase structure grammar).
Type 1 grammar (Context sensitive grammar).
Type 2 grammar (Context-Free Grammar)
Type 3 grammar (Regular grammar).

Type 0 A grammar $G = (V, T, P, S)$ is said to ~~be~~ be type 0 or unrestricted if all the productions are of the form.

$\alpha \rightarrow \beta$ where.

$\alpha \in (V \cup T)^+$ and $\beta \in (V \cup T)^*$.

Example

$S \rightarrow aAb \mid \epsilon$

$aA \rightarrow bAA$

$bA \rightarrow a$.

Type 1 : A grammar $G = (V, T, P, S)$ is said to be context sensitive if all the productions are of the form $\alpha \rightarrow \beta$ as in type 0 grammar, But there is restriction on length of β.

$|\beta| \geq |\alpha|$, α and $\beta \in (V \cup T)^+$

ϵ cannot appear on left or right hand side of any production.

Linear bounded Automata can be constructed to recognize the language.

<u>Ex</u> : $S \to aAb$.
$aA \to bAA$.
$bA \to aa$.

<u>Type 2</u> : A Grammar $G = (V, T, P, S)$ is said to be type 2 grammar or context free grammar if all the productions are of the form $A \to \alpha$. where $\alpha \in (V \cup T)^*$ and A is non-terminal

$S \to aB \mid bA \mid \epsilon$

$A \to aA \mid b$.

$B \to bB \mid a \mid \epsilon$.

<u>Type 3</u> (Regular grammar).
The grammar $G = (V, T, P, S)$ is said to be type 3 or regular iff the grammar is right-linear or left-linear.