

MODULE 5



Prepared By Mr. EBIN PM, AP, IESCE

1

PUMPING LEMMA FOR CFLs

- Pumping Lemma for CFL is used to prove that a language is not context free.
- If A is a context free language, then A has a pumping length P such that any string S , where $|S| \geq P$ may be divided into 5 pieces $S = uvxyz$ such that the following conditions must be true
 - uv^ixy^iz is in A for every $i \geq 0$
 - $|vy| > 0$
 - $|vxy| \leq P$

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE

2

➤ To prove that a language is not CFL using Pumping Lemma, follow the steps: (Proof using contradiction)

- Assume that A is context free
- It has to have a pumping length P
- All strings longer than P can be pumped $|S| \geq P$
- Now, find a string S in A such that $|S| \geq P$
- Divide S into uvxyz
- Show that $uv^ixy^iz \notin A$ for some i
- Then consider the ways that S can be divided in to uvxyz
- Show that none of these can satisfy all the three pumping conditions at the same time
- S cannot be pumped = = Contradiction

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE 3

Eg: Show that $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context free.

Step 1 : Assume that L is context free

Step 2 : L must have a pumping length P

Step 3 : Now we take a string such that $S = a^P b^P c^P$

Step 4 : We divide S in to parts uvxyz

Let $P = 4$. (we can take any pumping length)

$$\text{So } S = a^4 b^4 c^4 = \text{aaaabbbbcccc}$$

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE 4

Case 1 - v and y each contains only one type of symbol.

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

a a a a b b b b c c c c
 u v x y z

$uv^i xy^i z$, suppose $i = 2$

So, $uv^2 xy^2 z = aaaaaabbbbcccc = a^6 b^4 c^5$, which is not in L

ie, **contradiction**

Hence proved

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

5

Case 2 - Either v or y has more than one kind of symbols.

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

a a a a b b b b c c c c
 u v x y z

$uv^i xy^i z$, suppose $i = 2$

So, $uv^2 xy^2 z = aaaaabbaabbbbcccc =$ This is not an acceptable pattern

ie, **contradiction**

Hence proved

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

6

CONTEXT- SENSITIVE GRAMMAR

- A grammar $G = (V, T, P, S)$ is said to be context sensitive , if all productions are of the form $\alpha \rightarrow \beta$ where $|\alpha| \leq |\beta|$ and β is not suppose to be empty.
- All rules in P are of the form $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$
- The strings α_1 and α_2 may be empty, but β must be non-empty
- Context-sensitive grammars are more powerful than context-free grammars because there are some languages that can be described by CSG but not by context-free grammars
- The languages generated by these grammars are recognized by a linear bounded automaton.

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 7

❖ Context-sensitive Language

➤ The language that can be defined by context-sensitive grammar is called CSL.

❖ Properties of CSL

- Union, intersection and concatenation of two context-sensitive languages is context-sensitive.
- Complement of a context-sensitive language is context-sensitive

Example

$$AB \rightarrow AbBc$$

$$A \rightarrow bcA$$

$$B \rightarrow b$$

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 8

Consider the following CSG. What is the language generated by this grammar?

$S \rightarrow abc/aAbc$

$Ab \rightarrow bA$

$Ac \rightarrow Bbcc$

$bB \rightarrow Bb$

$aB \rightarrow aa/aaA$

$S \rightarrow aAbc$

$\rightarrow abAc$

$\rightarrow abBbcc$

$\rightarrow aBbbcc$

$\rightarrow aaAbbcc$

$\rightarrow aabAbcc$

$\rightarrow aabbAcc$

$\rightarrow aabbBbcc$

$\rightarrow aabBbbcc$

$\rightarrow aaBbbcc$

$\rightarrow aabbbcc$

The language generated by this grammar is $\{a^n b^n c^n \mid n \geq 1\}$

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE

9

UNRESTRICTED GRAMMAR

- A grammar $G = (V, T, P, S)$ is called unrestricted, if all the productions are of the form $\alpha \rightarrow \beta$, where α is a string of terminals and nonterminals with at least one non-terminal and α cannot be null. β is a string of terminals and non-terminals.
- They generate the languages that are recognized by a **Turing machine**.

Example

$S \rightarrow ACaB$

$Bc \rightarrow acB$

$CB \rightarrow DB$

$aD \rightarrow Db$

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE

10

CHOMSKY HIERARCHY

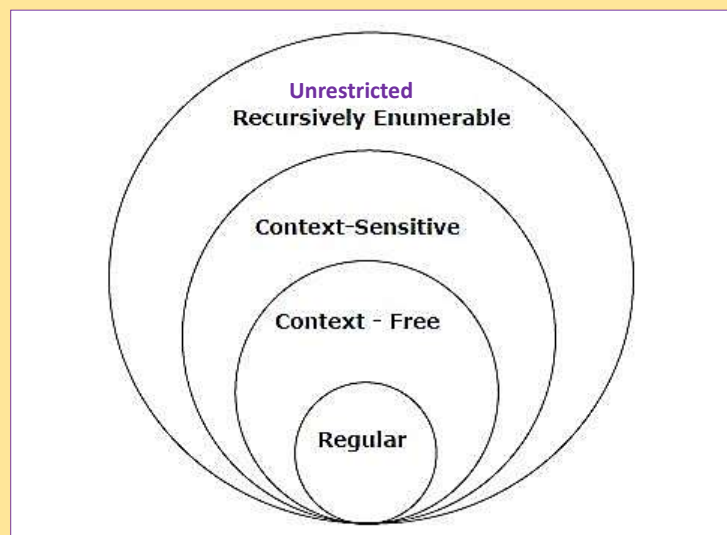
➤ According to **Noam Chomsky**, there are four types of grammars – Type 0, Type 1, Type 2, and Type 3. The following table shows how they differ from each other

Grammar Type	Grammar Accepted	Language Accepted	Automaton
Type 0	Unrestricted grammar	Recursively enumerable language	Turing Machine
Type 1	Context-sensitive grammar	Context-sensitive language	Linear-bounded automaton
Type 2	Context-free grammar	Context-free language	Pushdown automaton
Type 3	Regular grammar	Regular language	Finite state automaton

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 11

The following illustration shows the scope of each type of grammar



Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 12

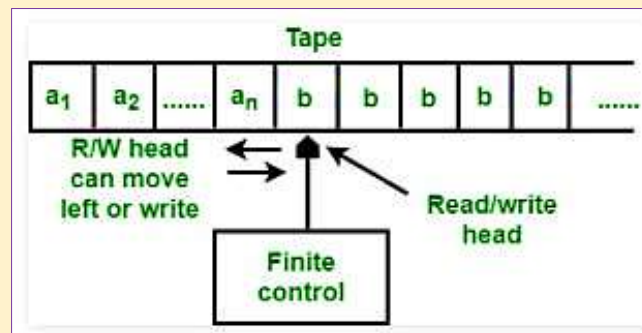
TURING MACHINE (TM)

- Turing machine was invented in 1936 by **Alan Turing**.
- It is an accepting device which accepts **Recursive Enumerable Language** generated by **type 0** grammar.
- A Turing Machine consists of a tape of infinite length on which read and writes operation can be performed.
- The tape consists of infinite cells on which each cell either contains input symbol or a **special symbol** called **blank**.
- It also consists of a head pointer which points to cell currently being read and it can move in both directions.

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 13

Turing Machine



Tape alphabets : $\Sigma = \{0,1,a,b,x,z_0\}$

Blank symbol ($_$) is a special symbol which is not in Σ

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 14

❖ Operations on the Tape

- Read / Scan the symbol which is pointed by the tape head
- Update / Write a symbol which is pointed by the tape head
- Move the tape head one step left
- Move the tape head one step right

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 15

❖ Features of the Turing machine:

- It has an **external memory** which remembers arbitrary long sequence of input.
- It has **unlimited memory capability**.
- The model has a facility by which the input at left or right on the tape can be read easily.
- The machine can produce a certain output based on its input. Sometimes it may be required that the same input has to be used to generate the output. So in this machine, the distinction between input and output has been removed. Thus a common set of alphabets can be used for the Turing machine.

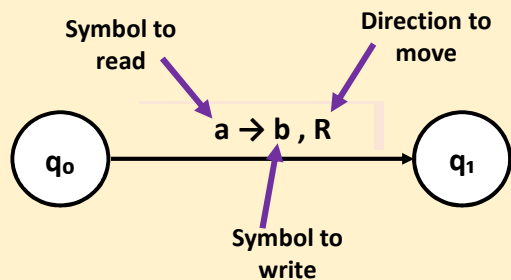
Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 16

❖ Operation Rule – 1

At each step of computation

- Read the current symbol
- Write (update) the same cell
- Move exactly one cell either LEFT or RIGHT



Transition Diagram

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 17

❖ Operation Rule – 2

- Turing Machine has a control portion, which controls the Turing Machine and is similar to FSM or PDA. It is deterministic also
- TM has an initial state
- TM has two final states

Accept state

Reject state

When the computation Halt (stop) ?

- HALT & Accept
- HALT & Reject
- Loop (Keep on computation and fail to stop)

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 18

❖ Formal definition of Turing machine

A Turing machine can be defined as a collection of 7 components:

- Q** : the finite set of states
- Σ** : the finite set of input symbols
- T** : the tape symbol
- q_0** : the initial state
- F** : a set of final states (Accept & Reject state)
- B** : a blank symbol used as a end marker for input
- δ** : a transition or mapping function.

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 19

❖ Transition Function (δ)

$$Q \times \Sigma \rightarrow T \times (R/L) \times Q$$

❖ Production rule of TM

$$\delta (q_0, a) \rightarrow (q_1, \gamma, R)$$

- Any computations that can be done on existing digital computer, can also be done by Turing Machine
- If we are having an algorithm of a problem, then that can also be solved by Turing Machine.

Prepared By Mr. EBIN PM, AP, IESCE

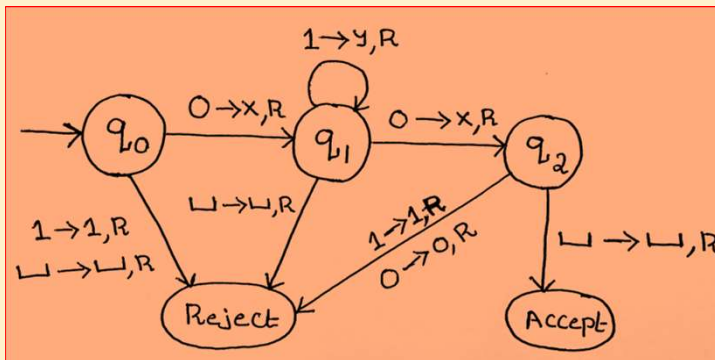
EDULINE 20

❖ Turing Thesis

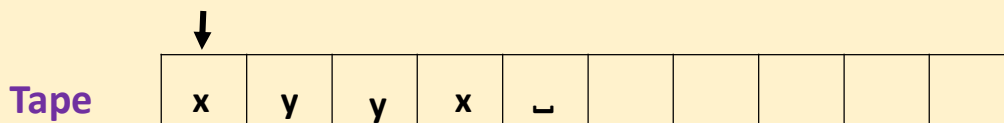
It states that, any computation that can be carried out by mechanical means can be performed by some Turing Machine.

- The language that acceptable by the Turing Machine is called **Recursively Enumerable Language**.

Q: Design a Turing Machine which recognize the language $L=01^*0$



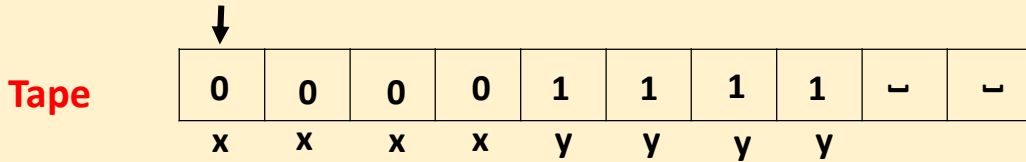
Consider 0110



Q: Design a Turing Machine which recognize the language $L=0^n1^n$

Algorithm

- Change 0 to x
- Move right to first "1" (If none: Reject)
- Change 1 to y
- Move left to leftmost "0"
- Repeat the above steps until no more 0's
- Make sure no more 1's remain

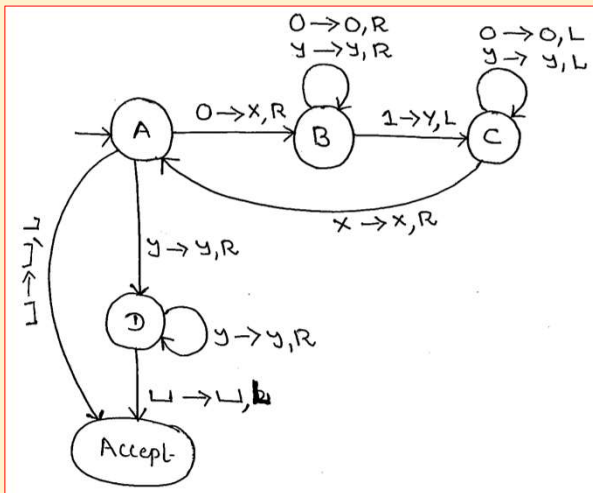


Tape

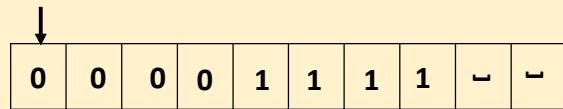
Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 23

Transition Diagram



Tape



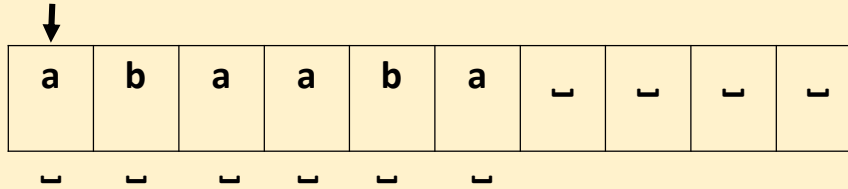
Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 24

Q: Design a Turing Machine which accepts even palindromes over the alphabet $\Sigma = \{a, b\}$

Eg: abaaba

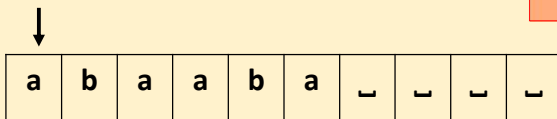
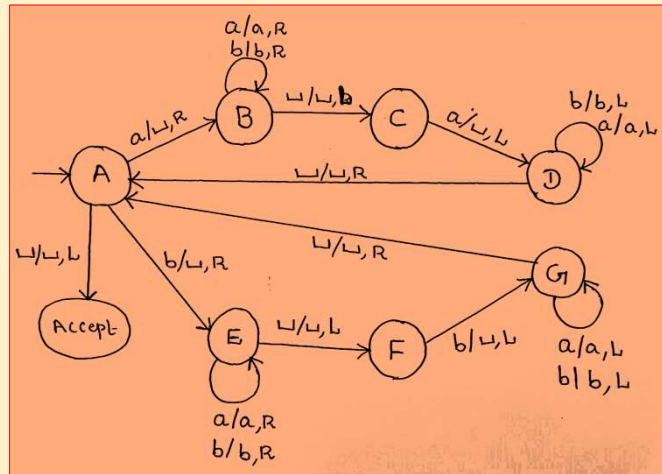
Tape



Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 25

Transition Diagram



Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 26

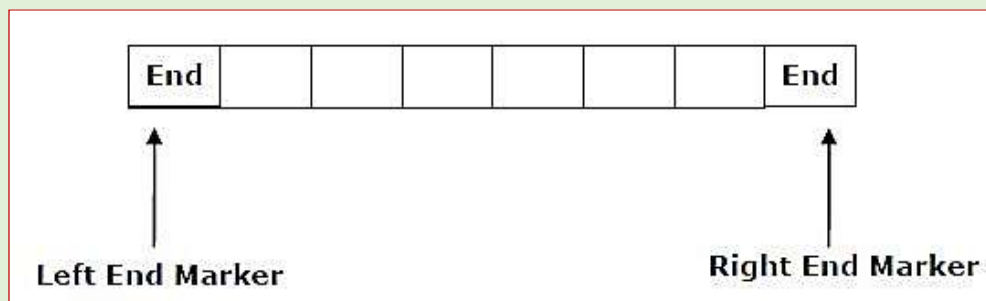
LINEAR BOUNDED AUTOMATA (LBA)

- It behaves as a Turing machine but the storage space of tape is restricted only to the length of the input string.
- It is less powerful than a Turing machine but more powerful than push down automata.
- The computation is restricted to the constant bounded area.
- The input alphabet contains two special symbols which serve as **left end** markers and **right end** markers which mean the transitions neither move to the left of the left end marker nor to the right of the right end marker of the tape.

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 27

- A deterministic linear bounded automaton is always context-sensitive and the linear bounded automaton with empty language is undecidable.



Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 28

➤ A linear bounded automaton can be defined as an 8-tuple

$(Q, T, \Sigma, q_0, ML, MR, \delta, F)$ where

Q is a finite set of states

T is the tape alphabet

Σ is the input alphabet

q_0 is the initial state

ML is the left end marker

MR is the right end marker where **$MR \neq ML$**

δ is a transition function which maps each pair (state, tape symbol) to (state, tape symbol, Constant 'c') where c can be 0 or +1 or -1

F is the set of final states

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 29

❖ Linear Bounded Automata (LBA) – Applications

➤ For implementation of genetic programming.

➤ For constructing syntactic parse trees for semantic analysis of the compiler.

❖ Expressive Power of various Automata

➤ The Expressive Power of any machine can be determined from the class or set of Languages accepted by that particular type of Machine. Here is the increasing sequence of expressive power of machines :

$FA < DPDA < PDA < LBA < TM$

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE 30

- As we can observe that FA is less powerful than any other machine.
- It is important to note that DFA and NFA are of same power because every NFA can be converted into DFA and every DFA can be converted into NFA .
- The Turing Machine i.e. TM is more powerful than any other machine.