Parul® University

Theory Of Computation

Computer Science & Engineering

CHAPTER-1

INTRODUCTION

What is TOC ?

- **It is the mathematical study of computing machine and its capabilities.**
- **Theory of computation is a study of formal language and automata theory.**
- **TOC consist of language, Automata and Grammar**

Alphabet

- **The non empty finite set of symbols is called as an alphabet and it is denoted by ∑.**
    - **Example**
        - ∑ = {a.b.c………z}
        - ∑ = {0,1}
        - ∑ = {1}

String

- **Any finite sequence of symbol from alphabet ∑ , is called as string and is represented by w.**
  - **Example**
    - ∑ = {a,b}
      - w = a,ab,aa,bb
      - w= ~~ab,aab,abc~~
    - ∑ = {0,1}
      - w = 0,01,00,11,1
      - w = ~~102,2013~~

Length of a String

- **If w is any string over alphabet ∑ then the number of symbols involved in the sequence of string is called as length of the string and denoted by |w|.**

- **Example**
  - **∑ = {a,b}   W = a,ab,aa,bb   ,|w|=1,2,2,2**
  - **∑ = {0,1}   W = 0,01,00,11,1 |w| = 1,2,2,2**

- **Empty String**
- A string of length zero or string without any symbols is known as empty string and is denoted by €

  w = € , |w| = 0

  w. € = w = €.w

Substring

- **Let u,w be the two strings over same alphabet ∑ then u is said to be substring of w if u can be obtained from w.**
- **Any consecutive sequence of symbols over given string.**
  - If u is a substring of w then |u| <= |w|
  - Every string is substring to itself.
  - Empty string "€" is substring for every string.
  - **Example**
    - W = TOC
    - Zero length Substring= €
    - Two length substring= TO,OC

One length substring=T,O,C

Three length substring=TOC

Types of Substring

- **The substrings are classified into two types**
  1. **Trivial Substring or improper Substring**
  2. **Non-trivial or proper substring**

- **Trivial Substring or improper Substring**
  - If w is any string over alphabet ∑ then the substring w itself and € is called as trivial substring

- **Non-trivial Substring or proper Substring**
  - If w is any string over alphabet ∑ then any substring of w over w other than w itself and € is called non trivial substring.

Facts about Substring

- **If w is any string with distinct symbols and $|w| = n$**

  1. No of substrings = $\sum n + 1 = n(n+1)/2+1$
  2. No. of trivial string = 2
  3. No. of non trivial substring = $\sum n - 1$
  4. No. of non empty substring= $\sum n$
  5. No of substrings of distinct length = n+1
  6. No. of strings of length n generated over alphabet $\sum = |\sum|^n$

Prefix and Suffix

- **Prefix**
  - The sequence of starting or leading symbol is called as prefix.
- **Suffix**
  - The sequence of ending or trailing symbol is called as suffix.

- **Example**
- w=TOC , |w|=3
- Prefix : € , T , TO , TOC
- Suffix : TOC , OC , C , €

Facts about Prefix and Suffix

- **If w is any string of length ‘n’ then**

1. No. of prefix = No. of suffix = n+1
2. Trivial substrings are common for both prefix and suffix
3. Every prefix and suffix must be a substring but every substring need not be prefix or suffix.

Power of an alphabet

- **If ∑ is any alphabet then $\sum^k$ is the set of all the strings of length k.**
- Example : ∑ = {0,1}
  - $\sum^2$ = {00,01,10,11}
  - $\sum^3$ = {000,001,010,011,100,101,110,111}
  - $\sum^k$ = {all k-length stings}
- **+ve closure($\sum^+$)**
  - $\sum^+ = \{w \mid |w|>=1\}$
- **Kleen closure(∑*)**
  - $\sum^* = \{w \mid |w|>=0\}$
- $\Sigma^* = \sum^+ \cup €$

Language

- **The collection of strings from the alphabet ∑ is called language**

  - **Example** ∑ = {0,1}
  - L ={00,01,10,11}
  - L = { $(01)^n$ | n>=1 }
  - L = { $0^n 1^m$ |m>=1,n>=1}

- **If ∑ is any alphabet then Σ^* is called as universal language**

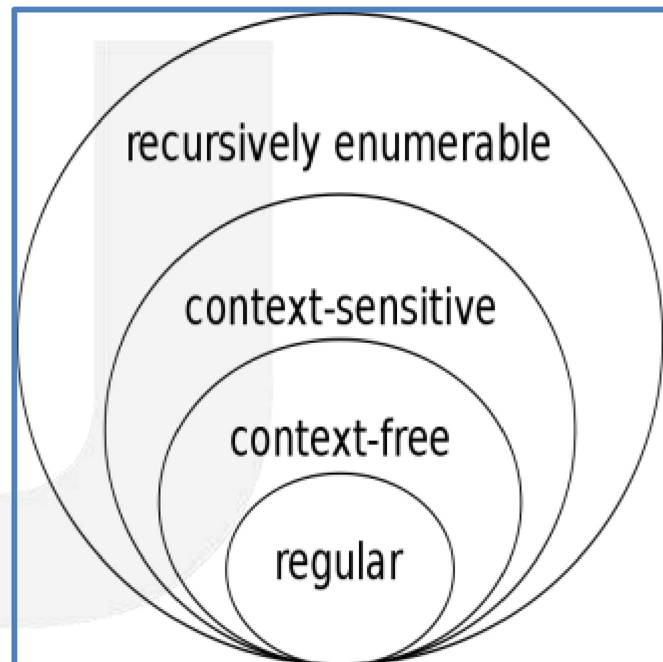
Formal Language

- **The collection of strings where we can put some restriction in the formation of string is called as formal language.**
  - **Example** $\sum = \{0,1\}$
  - $L = \{00,01,10,11\}$
  - $L = \{ (01)^n \mid n>=1 \}$
  - $L = \{ 0^n 1^m \mid m>=1, n>=1\}$

Chomsky classification of Formal Language

- **According to Chomsky the formal languages are classified as**

    1. Type 0 or Recursive Enumerable Languages
    2. Type 1 or Context sensitive languages
    3. Type 2 or Context free languages
    4. Type 3 or Context Regular languages



Types of Language

- **Empty Languages**

The Language that does not contain any string even empty string is called as empty language and is denoted by ϕ.

- **Non Empty Languages**

The Language that contain at least one string is called as non empty language .

- **Finite Languages**

The Language which contains finite number of strings and length of each string is finite is called as finite language .

EX : $L = \{ 0^n 1^n \mid 1<=n<=1\}$

- **Infinite Languages**

The Language which contains infinite number of strings and length of each string is finite is called as infinite language .

EX : $L = \{ 0^n 1^n \mid n>=1\}$

Automata

- **The mathematical system that can represent the formal language is called as Automata i.e. The mathematical representation of formal language is called as an automata.**
- **Types of Automata**
    1. Finite Automata(FA)
    2. Push Down Automata(PDA)
    3. Linear Bound Automata(LBA)
    4. Turing Machine(LBA)

Expressive Power of an Automata

- **The number of languages accepted by an automata is called as Expressive Power of an Automata.**

    1. E(FA)=1
    2. E(PDA)=2
    3. E(LBA)=3
    4. E(TM)=4

Grammar

- **The collection of rules which are used to generate the string is called grammar.**
- **Grammar G is a collection of 4 tuples {V,T,P,S}**

G= {V,T,P,S} ,Where

V= set of all Nonterminal symbol/variable

P= setoff all productions

T= set of all terminal symbols

S= starting symbol

- **Example**

1. $A \longrightarrow XYZ$ (r1) V={A,X,Y,Z}

   $X \longrightarrow a$ (r2) T={a,b,c}

   $Y \longrightarrow b$ (r3) P={r1,r2,r3,r4}

   $Z \longrightarrow c$ (r4) S=A

Chomsky classification of Grammar

- **According to Chomsky the Grammar is classified as**

1. Type 0 or Recursive Enumerable Grammar
2. Type 1 or Context sensitive Grammar
3. Type 2 or Context free Grammar
4. Type 3 or Context Regular Grammar

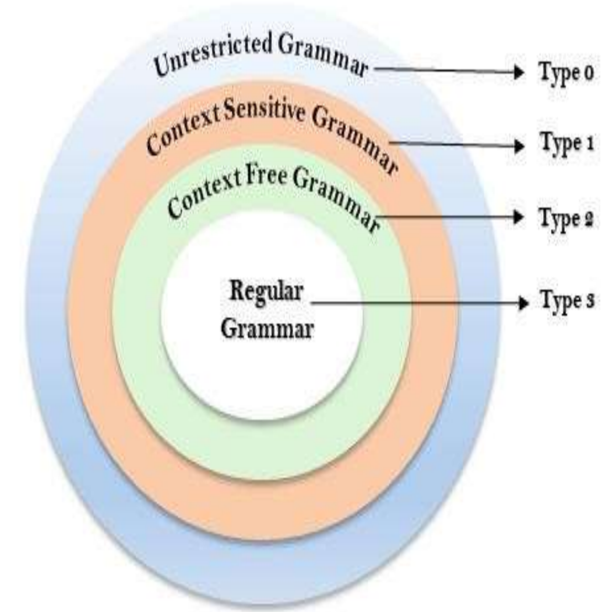


Fig: Chomsky Hierarchy

Type 0 or Unrestricted Grammar

- **A grammar G is said to be Recursive Enumerable or Unrestricted grammar if every production is of the form**

$$A \longrightarrow B$$

Where,

$A \in (V+T)^{+}$ AND $B \in (V+T)^{*}$

EXAMPLE

1. S ⟶ AaB
   Aa ⟶ Bb\Aa\€
   bBb ⟶ aa\bb\ €

2. S ⟶ Aa\bB
   aAb ⟶ bb
   Ba ⟶ aAb\Baa\b

Type 1 or Context Sensitive Grammar

- **A grammar G is said to be Type 1 or Context sensitive grammar(CSG) if every production is of the form**

A → B

Where,

|A|<=|B| , B≠ €   AND   A,B∈$(V+T)^+$

EXAMPLE

1. S → aSAc\abc
   cA → Ac
   bA → bb

2. S → Aa\Bb
   aA → aAB\ bB
   B → aBb\aa

Type 2 or Context free Grammar

- **A grammar G is said to be Type 2 or Context Free grammar(CFG) if every production is of the form**

$$A \longrightarrow B$$

Where,

$A \in V$ AND $B \in (V+T)^*$

EXAMPLE

1. $S \longrightarrow Aab$
   $A \longrightarrow Aa \backslash b$
   $A \longrightarrow bB \backslash a$

2. $S \longrightarrow aSb \backslash €$

Type 3 or Regular Grammar

- **A grammar G is said to be Type 3 or Regular grammar if every production is of the form**

$$A \longrightarrow xB/x \quad \text{or} \quad A \longrightarrow Bx/x$$

Where,

$A, B \in V$ AND $x \in T^*$

EXAMPLE

1. $S \longrightarrow 10S\backslash 01$
2. $S \longrightarrow 001S\backslash 11$

Other classification of Grammar

- **Recursive grammar**
  - The grammar g is said to be recursive if atleast one production contain the same variable at both left hand side and right hand side of production.
  - **Example** **S $\longrightarrow$ aSb|€**
- **Non Recursive grammar**
  - The grammar g is said to be recursive if noproduction contain the same variable at both left hand side and right hand side of production.
  - **Example** **S $\longrightarrow$ ab|€**

Derivation

- **The process of deriving a string is called as derivation.**
- **The geometrical representation of derivation is called Derivation tree or parse tree.**
- **Steps involved in derivation is called sentential form**

- **Example**

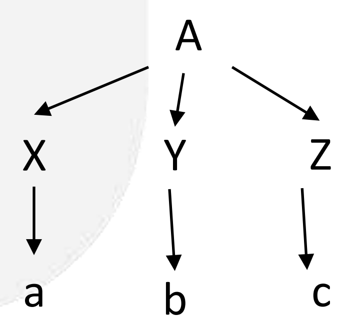
1\.

$A \rightarrow XYZ$
$X \rightarrow a$
$Y \rightarrow b$
$Z \rightarrow c$

Derivation

$A \rightarrow XYZ$
$A \rightarrow aYZ$
$A \rightarrow abZ$
$A \rightarrow abc$

Parse Tree



DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in