Theory of Computation

Prof. R Shivamallikarjun, Assistant Professor
Information Technology

CHAPTER-1

Introduction

Parul® University

DIGITAL LEARNING CONTENT

Introduction to Finite Automata

“A finite automaton has a finite set of states with which it accepts or rejects strings”.

- Initialization:
- Looking for "n"
- Recognized "n", looking for "a"
- Recognized "na", looking for "m"
- Recognized "nam", looking for "e"
- Recognized "name“

∑={a,b}

L3= set of all string where each string starts with a={a

Introduction to Finite Automata

Finite Automata(FA) is the simplest machine to recognize patterns.

A Finite Automata consists of the following :

Q : Finite set of states.

∑ : set of Input Symbols.

q : Initial state.

F : set of Final States.

δ : Transition Function.

Formal specification of machine is

{ Q, ∑, q, F, δ }.

Alphabet, Languages & Grammars

Alphabet: A set of letters or symbols.
For example: A….Z, 0….9.

$$\Sigma_{\Sigma} = \{a, b, c \dots. Z\}$$

String: Sequence of letters

- "bat" , "ball" , "House" , ….
- defined over an alphabet A….Z.

∑n

Alphabet, Languages & Grammars

Languages:

- Language containing a finite number of words.
- For example: a* , ab*.
- A language is any subset of Σ^*, $\Sigma=\{a,b\}$

  where $\Sigma^* = \{\lambda, a, b, aa, ab, bb, aaa, \ldots\}$

Grammars: A Grammar is a 4-tuple such that- G = (V, T, P, S)

Alphabet, Languages & Grammars

G = (V, T, P, S)

Where-

V = Finite non-empty set of non-terminal symbols

T = Finite set of terminal symbols

P = Finite non-empty set of production rules

S = Start symbol.

Consider a grammar: S → Aba, A□A b, A□a

S=abbbba

Parul® University

Productions and derivation

Production: Recursively performed to generate new symbol sequences. Denoted as by using arrow symbol □.

Derivation: A derivation proves that the string belongs to the grammar's language.

Chomsky hierarchy of languages

Grammars are divided of 4 types:

- •Type 0 known as Unrestricted Grammar.
- •Type 1 known as Context Sensitive Grammar.CSG
- •Type 2 known as Context Free Grammar.
- •Type 3 Regular Grammar.

Image source: GeeksForGeeks

Chomsky hierarchy of languages

Type 0: Unrestricted Grammar:

In Type 0 □ Include all formal grammars.

Known as the Recursively Enumerable languages.

Grammar Production in the form of

α □ β

Where

α is $(V + T)^* V (V + T)^*$

V : Variables

T : Terminals.

β is $(V + T)^*$.

For example,

Sb □ ba

A □ S

Here, Variables are S, A and Terminals a, b.

Chomsky hierarchy of languages

Type 1: Context Sensitive Grammar)

In Type 1□Context-sensitive languages.

Recognized by the Linear Bound Automata.

I. First of all Type 1 grammar should be Type 0.

II. Grammar Production in the form of

α □ β

|α| <= | β|

I.e. count of symbol in α is less than or equal to β

For Example,

S –> AB

AB –> abc

B –> b

Chomsky hierarchy of languages

Type 2: Context Free Grammar:

In Type-2□ Grammars generate the context-free languages.

Recognized by a Pushdown automata.

1. First of all it should be Type 1.
2. Left hand side of production can have only one variable.

|α|= 1.

There is no restriction on β.

For example,

S –> AB

A –> a

B –> b

Chomsky hierarchy of languages

Type 3: Regular Grammar

In Type-3 □ Grammars generate regular languages.

Type 3 is most restricted form of grammar.

Type 3 should be in the given form only :

V –> VT* / T*

(or)

V –> T*V /T*

For example,

S –> Aab

A□b

References

1. https://people.cs.clemson.edu/~goddard/texts/theoryOfComputation/1.pdf
2. https://en.wikipedia.org/wiki/Chomsky_hierarchy

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.i