INTRODUCTION

Theory of Computability

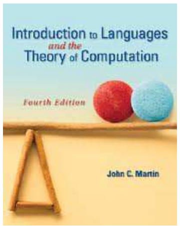
1. Course Objective

Providing student with formal language theory, grammars, regular grammar, regular expression, context-free grammar, and automata.

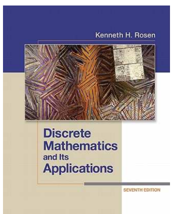
2. Syllabus

- Language and grammar
- Types of grammar
- Finite state automata.
- Types of Finite state automata.
- Regular language and regular expression.
- <u>Context- free grammar and context free language.</u>
- <u>Context- free grammar - Chomsky normal form</u>

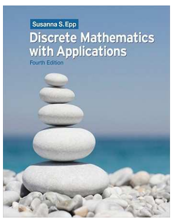
3. References:



Introduction to Languages and the Theory of Computation
4^{th} Edition
By John C. Martin
2011



Discrete Mathematics And Its Applications, Chapter 13
7^{th} Edition
By Kenneth H. Rosen
2012



Discrete Mathematics With Applications, Chapter 12
4^{th} Edition
By Susanna S. Epp
2011

LECTURE 1

Formal Language and Grammar

Introduction

Research in the automatic translation of one language to another has led to the concept of a formal language, which is specified by a well-defined set of rules of syntax called grammar. Rules of syntax are important not only in linguistics, the study of natural languages, but also in the study of programming languages

1. Formal Language:

A formal language over an alphabet is any set of strings of characters of the alphabet (denoted by Σ and read as Sigma).

- **Alphabet** Σ : a finite set of characters.
  Binary alphabet ={0, 1}
  Arabic alphabet ={أ ، ب، ت، ... ي}
  Greek alphabet = { $\alpha, \beta, \gamma, \ldots, \omega$ }
- **String over** Σ: a finite sequence of characters or the null string.
- ε: is the null string (read as epsilon), sometimes denoted by (λ read as lambda).
- **Length of a string over** Σ: The number of characters that made up the string. The null string having length 0.
  |computer|=8
  | ε |=0
- **Formal language over** Σ: a set of strings over the alphabet.
- Σ^n: the set of all strings over Σ that have length n, where n is nonnegative integer.
- Σ^+: the set of all strings over Σ that have length at least 1, (all string except for null string)
- Σ^* = the set of all strings over Σ including null string.

Example 1:

Let the alphabet Σ ={0, 1}, define a language **L1** over Σ to be the set of all strings that begin with 0 and have length at most three characters.

Solution:

L1 ={0, 00, 01, 000, 001, 010, 011}

Example 2:

Let $\sum$ ={x, y}, Find $\sum^0$,$\sum^1$,$\sum^2$,and $\sum^3$.

Solution:

$\sum^0$={ ε }

$\sum^1$={x, y}

$\sum^2$={xx, yy, xy, yx}

$\sum^3$={xxx, xxy, xyy, yyy, yyx, yxx, yxy, xyx }

Languages are sets. Thus, one way of constructing new languages from existing ones is to use set operations. For two languages L1 and L2 over the alphabet $\sum$, L1 ∪ L2, L1 ∩ L2,and L1 − L2 are also languages over $\sum$.

Also the string operation of **concatenation** can be used to construct new languages. If L1 and L2 are both languages over $\sum$, the concatenation of L1 and L2 is the language.

L1. L2 ={x.y | x ∈ L1 and y ∈ L2}

ε x = x ε =x
for every string x

For example, L1= {a, aa}, L2={ ε, b, ab}, then L1.L2={a, ab, aab, aa, aaab}.

The string operation of **Kleene star** is a unary operator applied to set of characters or symbol. Denoted by * and pronounced (clay knee)

If A is a set of characters, then A* is the set of all strings over A including ε .

For example, A= {0, 1}, then ={0, 1, 01, 10 ,11, 00, 101, 11010, 001101, ……..}.

Example 3:

Let L1 is the set of all strings consisting of an even number of a's {ε, aa, aaaa, aaaaaa,…}, and L2 ={b, bb, bbb}. Find L1L2, L1 ∪ L2.

Solution:

L1L2 = {b, bb, bbb, aab, aabb, aabbb, aaaab, aaaabb, aaaabbb, aaaaaab,aaaaaabb,……}

L1 ∪ L2 = {b, bb, bbb, ε, aa, aaaa, aaaaaa,… }

Example 4:

Let the alphabet $\sum$ ={a, b}, define the following language over $\sum$:

L1= $\{(ab)^n \mid n> 0\}$

L2= $\{a^nb^n \mid n> 0\}$

L3= $\{a^mb^n \mid n>= 0, m>0\}$

Solution:

L1={ab, abab, ababab,…}

L2={ab, aabb, aaabbb,……}

L3={a,ab, aa, aab, aabb, aabbb, aaaabbbbbb,……}

2. Grammar:

Languages can be specified in various ways:

1. List all the strings in the language.
2. Give some criteria that a string must satisfy to be in the language, just like set notation.
3. describe the language by using of a grammar.

Grammar is a generative system used to generate strings of language, and to determine whether a string is in a language or not. The grammar is denoted by G where G = (V, T, S, P).

grammar G = (V, T, S, P) consists of the following:

- Vocabulary **V** which is a set of symbols used to derive strings of the language.

- A subset **T** of V consisting of terminal symbols which is small letters. The other member of V is **N** which consists of capital letters called non-terminals.
  The characters of the strings generated by grammar G are belong to T but not N. while N is used to derive these strings from G.

- Start symbol **S** from V is a special member, which is the non-terminal that we always begin the derivation with.

- A finite set of productions rules P used to derive new string from given one. It is denoted by z0 → z1 (read as z0 yields z1) the production that specifies that z0 can be replaced by z1 within a string.

The language generated by grammar G is denoted by L(G).

Example 5:

Let G = (V,T,B, P) where V ={a, B, E},

N= {B, E}, T ={a}, B is the start symbol, and P ={B → a, B → aE, E → aB}. Find L(G).

Solution:

B → a

B → aE

→aaB →aaa

→aaaE

→aaaaB →aaaaa

Terminal symbols cannot be replaced by other symbols.

Non-terminals can be replaced by other symbols.

L(G)={ a, aaa, aaaaa,.....}

L(G)={a^n |n is odd number}

Example 6:

Describe the language L(G) that is generated from grammar G = (V,T,S,P) where V ={A, B, S, a, b}, T ={a, b}, and

P :{S → AB

A → ε|aA

B → ε|bB

}.

Solution:

S → AB → εε→ ε

S → AB

→AbB → εbε → b

→AbbB → εbbε → bb

→AbbbB → εbbbε → bbb

→aAB → aεε → a

→aaAB →aa εε → aa

→aaaAB →aaa εε → aaa

→aAbB → aεbε → ab

→aaAbB →aaεbε → aab

→aAbbB →aεbbε → abb

L(G)={$a^n b^m$ |n, m ≥ 0 }

Example 7:

Find L(G) for G = (V,T,S,P) where V= {S, N, Q, R, a}, T = {a} With Production rules P:

$S \to QNQ$

$QN \to QR$

$RN \to NNR$

$RQ \to NNQ$

$N \to a$

$Q \to \varepsilon$

Solution:

$S \to QNQ \to \varepsilon a \varepsilon \to a$

$\quad \hookrightarrow QRQ \to QNNQ \to aa$

$\quad\quad \hookrightarrow QRNQ \to QNNRQ \to QNNNNQ \to aaaa$

$L(G) = \{a^{2^n} \mid n \geq 0\}$

Example 8:

Suggest a grammar for the following languages:

1. L1 = {0, 011, 01111, 0111111,……}
2. L2 = {xyx, axyxb, aaxyxbb, aaaxyxbbb, …}
3. $L3 = \{ a^n b^m \mid n, m \geq 0\}$

Solution:

1. G1=(V, T, S, P) where V={S, A, 0, 1}, N={ S, A }, T={ 0, 1}

   P: { $S \to 0A$

   $\quad A \to \varepsilon \mid 11A$

   }

2. G2=(V, T, S, P) where V={ S, a, b, x, y}, N={ S }, T={ a, b, x, y}

   P: { $S \to aSb \mid xyx$ }

3. G3=(V, T, A, P) where V={ A, B, 0, 1}, N={ A }, T={ 0, 1}

   P: { $A \to \varepsilon \mid 1A \mid 0B$

   $\quad B \to \varepsilon \mid 0B$

   }

3. Homework:

HW 1:

What is the language L(G) that is generated by the grammar G = (V, T, A, P) where V={A, B, a, b, c}, N= {A, B}, T ={a, b, c}, and P is the following production rules:

A → ε |aAbB

B → cA.

HW 2:

Construct a grammar to generate the language represented by the following set:

L = { $(11)^n$ | n ≥ 0}