CHAPTER 3
LANGUAGES AND GRAMMAR

Words in a language can be combined in various ways. The grammar of a language tells us whether a combination of words is a valid sentence. The syntax, of a **natural language** (body of words and methods of combining words used and understood by a considerable community), that is, a spoken language, such as English is extremely complicated. Research in the automatic translation of one language to another has led to the concept of **formal language**, which is specified by a well defined set of rules or syntax.

Sentences of a formal language are described using a grammar which an algebraic system is describing the process by which sentences of a language can be constructed.

Alphabet

Let Σ denote a non-empty set of symbols. Such a set is called an **alphabet**. The elements of an alphabet are called letters. E.g. The alphabet of English Language Σ = {a, b, c,.,z}, the alphabet of binary numbers Σ = {0, 1}

Word or String

A word or string on the set Σ is a finite sequence of elements. For example, if Σ = {a, b}, then $u = aba$ and $v = aaabb$ are strings on Σ. The strings named u and v indicate that they have specific values *abab* and *aaabb*. a^i is used to denote a sequence of i a's. i.e a^2 for *aa*, a^3 for *aaa*. The length of a string u, denoted by |u| is the number of symbols in the string. The empty sequence of letter, denoted by λ is also considered to be a string on Σ, called the empty string and $|\lambda|$=0. λ is not a symbol o f the alphabet, but rather denotation for the empty word.

The set of all strings (including λ) on the alphabet set A is denoted by $\Sigma *$ and the set of non-empty strings by $\Sigma^+ = \Sigma * - \{\lambda\}$. $\Sigma *$ on the set Σ = {a, b} is $\Sigma * = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab....\}$.

Operation on Strings

The basic operation for strings is **concatenation** operations. Let u and v be two strings in $\Sigma*$. The concatenation of two strings *u* and *v* is the string obtained by writing down the letters of v to the right end of u. for example, for the above strings *u* and *v*, $uv = abaaaabb$.

Some of the basic properties of concatenation are:

If *u* is a string, then u^n stands for the string obtained by repeating *u n* times.

As a special case $u^0 = \lambda$.

Property 1: Concatenation on a set $\Sigma*$ is associative since for each *u, v, w* in $\Sigma*$,

u(vw) = (uv)w.

Property 2: *Identity element*. The empty string is an identity element for the operation i.e. $\lambda u = u\lambda = u$ for all *u* in $\Sigma*$.

Property 3. $\Sigma*$ has left and right cancellation. For *u, v, w* in $\Sigma*$,

wu = *wv* implies *u*=*v* (left cancellation).

uw = *vw* implies *u* = *v* (right cancellation).

Property 4. For *u, v* in $\Sigma*$, we have |*uv*|=|*u*| + |*v*| i.e. the length of concatenation of two strings is the sum of the individual length.

Substrings

Consider any string $u = a_1a_2a_3..........a_n$ on an alphabet Σ. Any sequence $w = a_ja_{j+1}.......a_k$ is called a *substring* of *u*. In particular, the substring $w = a_ja_{j+1}.......a_k$, beginning with the first letter of *u*, called an initial segment of *u*. For example, if $u = xyzx$ then the substrings of *u* are $\lambda, x, y, z, xy, zx, yzx, xyzx$ and the initial segments of u are $\lambda, x, xy, xyz, xyzx$.

Prefix and Suffix

Given two words *x* and *y* over an alphabet Σ, we say that *x* equals *y*, written *x* = *y*, if they have the same length and the same symbols at the same positions. We say that x is a prefix of y if there is a word *z* over Σ such that *xz* = *y*. Intuitively, there is partial equality from the left. For example, if x = *in* and *y* = *in dian*, then *x* is a prefix of *y*, since *xz* = *y*, where *z* = *dian*.

On the other hand, if *x* = *can* and *y* =*scan* then *x* is not a prefix of *y* and *y* is not a prefix of *x*.

It is said that *x* is a proper prefix of *y* if $x \neq \lambda$, $x \neq y$ and *x* is a prefix of *y*. clearly λ is always a prefix of any word *x* including itself and a word *x* is always a prefix of itself.
Similarly, a suffix and proper suffix are defined. For example, if *x* = *happy* and *y* = *un happy*, then *x* is a suffix of *y* and is, in fact, a proper suffix of *y*.

Languages

Dictionaries define the word language informally as a system suitable for the expression of certain ideas, facts, or concepts including a set of symbols and rules for their manipulation. Such languages are often called **natural languages**. The syntax of natural languages is extremely complicated. A **formal language** consists of a set of symbols and well-defined set of rules or syntax by which these symbols can be combined into entities called sentences. Formal languages are used to model natural languages and to communicate with computers. They have proved to be of tremendous help in the design of modern programming languages, compilers, operating systems and other software.
A language **L** over an alphabet Σ is a subset of the set Σ^***.** A string in a language is called a sentence of **L**. For example, if $\Sigma = \{a, b\}$**,** then $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab......\}$
The sets:

i) $L_1 = \{a, aa, ab, bb\}$ Which is a subset of Σ^* is a language on Σ**.**

ii) $L_2 = \{a^m b^m : m > 0\}$ is a language on Σ**.** The strings *ab, aabb* and *aaabbb* are in the language but the string *abb* is not in L$_2$.

iii) $L_3 = \{a^m b a^n : m \geq 0, n \geq 0\}$ is a language on Σ where all strings consist of exactly one b in the form *aba, a^{2}ba,….*

The languages L$_1$ has a finite number of sentences, it is called a **finite language**. The language L$_2$ and L$_3$ are **infinite**. Since languages are defined as sets of strings, all set operations can be applied to languages. For example, let L$_1$ and L$_2$ be two languages, $L_1 \cup L_2$ is also a language that contains the sentences in either L$_1$ or L$_2$. Let L$_1$ be the C programming language and L$_2$ be the C++ programming language. Then $L_1 \cap L_2$ will be the set of all statements that are valid both in C and C++. The complement of a language is defined with respect to Σ^***,** that is the Complement of L is $L' = \Sigma^* - L$. The concatenation of two languages L_1 and L_2 is the set of all strings obtained by concatenating any element of L_1 with any element of L_2; $L_1L_2 = \{xy : x \in L_1, y \in L_2\}$.

For example, {if, else}{ then, ()} = {ifthen, if(),elsethen,else()}.

Powers of a Language

L^n is defined as L concatenated with itself n times with special cases:

$L^0 = \{\lambda\}, L' = L$, if $L = \{a^n b^n : n \geq 0\}$ then $L^2 = \{a^n b^n a^m b^m : n \geq 0, m \geq 0\}$. Note that *n* and *m* in the above example are unrelated, the string *aabbaaabbb* is in L^2.

Product is similar but **different from Cartesian product**. For example, let $L_1 = \{a, ab\}$ and $L_2 = \{b, bb\}$, then $L_1L_2 = \{ab, abb, abbb\}$ contains only three words, whereas $L_1 \times L_2 = \{(a,b),(a,bb),(ab,b),(ab,bb)\}$ contains four pairs of words. Concatenation satisfies a number of properties.

It is associative, that is, for all languages A,B and C, A(BC) = (AB)C = ABC.

It has an identify, since $\{\lambda\}A = A\{\lambda\} = A$ for all languages A.

The **star** or **closure** of L, denoted by L*, by $L^* = \bigcup_{i=0}^{\infty} L^i$. L* consists of all words formed by concatenating a finite number of possibly zero, words from L. Then L* also is called the **Kleene Closure** of L.

The plus or positive closure of L, denoted by L^+, is defined by $L^+ = \bigcup_{i=1}^{\infty} L^i$. L^+ consist of all words formed by concatenating a finite number, never, zero, words from L. The star and plus are related since $L^+ \subseteq L^*$, for all languages L. Moreover, $L^* = \{\lambda\} \cup L^+$ by definition. But $L^+ \neq L^* - \{\lambda\}$, since λ is in L^+ if λ is in L. For example let $L = \{\lambda, a\}$. Then $L^+ = \{a_i : i \geq 0\}$: hence $L^* = L^+$ and $L^+ \neq L^* - \{\lambda\}$.

Example:

Let $L_1 = \{x, xy, x^2\}$ and $L_2 = \{y^2, xyz\}$ be a language of . Find:

i) L_1L_2

ii) $L_2{}^2$.

$L_1L_2 \quad = \{x, xy, x^2\}\ \{y^2, xyz\} = \{xy^2, x^2yz, xy^3, xyxyz, x^2y^2, x^3yz\}$

Exercise:

Let $\Sigma = \{a, b, c\}$ find L* and L^+ where:

i) $L = \{b^2\}$

ii) $L = \{a, b\}$.

Regular Expressions

Regular expressions are useful for representing certain sets of strings in an **algebraic fashion**. The formal recursive definition is follows:

Let $\sum$ be a given set of alphabets. Then

1. λ and ϕ are regular expressions.
2. Each letter a in Σ is a regular expression.
3. The union of two regular expressions R_1 and R_2 written as $R_1 \cup R_2$, is also a regular expression.
4. The concatenation of two regular expressions R_1 and R_2, written as $R_1 R_2$, is also a regular expression.
5. The closure of a regular expression R, written as R^*, is also a regular expression.
6. If R is a regular expression, then (R) is also a regular expression.

All regular expressions over $\sum$ are precisely those obtained recursively by the application of the rules 1-6 once or several times. The parenthesis influence the order of evaluation of a regular expression and in the absence of parenthesis, the hierarchy of operations is:

i) Closure.

ii) Concatenation and union which is similar to that followed for arithmetic expressions exponentiation, multiplication and addition.

Example:

Let $\sum = \{a, b\}$ be an alphabet. Then ϕ, λ, a, b are four regular expressions over $\sum$ by rules 1, and 2. By rule 3, $a \cup b$ is a regular expression over $\sum$.

Each regular expression represents a set specified by the following rules:

- ϕ represents the empty set i.e. set with no strings.
- λ Represents the set $\{\lambda\}$, which is the set containing empty string.
- x represents the set $\{x\}$ containing the string with one symbol x
- $x_1 \cup x_2$ represents the set $\{x_1 \cup x_2\}$
- $x_1 x_2$ represents the set $\{x_1 x_2\}$
- x^* denotes the set $\{\lambda, x, xx, xxx, ...\}$

Example

Describe the following sets by regular expression

a) $\{01,10\}$

b) $\{101\}$

c) $\{\lambda,1,11,111,...\}$

d) $\{1,11,1111,....\}$

Solution

a) As $\{01,10\}$ is the union of $\{01\}$ and $\{10\}$. Hence $\{01,10\}$ is represented by $01 \cup 10$

b) 101is the concatenation of 1, 0 and 1. Hence $\{101\}$ is represented by 101

c) $\{\lambda,1,11,111,...\}$ is represented by 1^*

d) $\{1,11,1111,....\}$ is represented by $1(1)^*$

Exercise

Express each of the following sets using a regular expression

a) The set of strings of one or more $0s$ followed by a 1

b) The set of all strings of $0s$ and $1s$ ending in 00