Chapitre I : Introduction aux Langages

1- **Définition d'alphabet :** ensemble fini non vide de lettres ou de symboles

2- **Définition d'un mot :** on appelle mot sur un alphabet X une suite finie d'éléments de X.

Exemple : soit l'alphabet X={0,1}, les suites 1,01,101 sont des mots construits sur X.

- On appelle mot inverse ou miroir d'un mot w, noté w^R ou w~, le mot obtenu en inversant les lettres de w. si $w=a_1\ldots a_n$ $w^R= a_n\ldots a_1$
- On appelle mot vide un mot particulier note ε ou λ.
- L'ensemble de tous les mots formés à partir d'un alphabet X avec le mot vide est noté $X^*=X^0+X^1+X^2+\ldots$ Où $X^0=\{ \varepsilon \}$
- **Opérations sur les mots :**
  1- **Concaténation :** soient x et y deux mots de X* , la concaténation notée x•y est définie par :
  Si $x=a_1\ldots a_m$ et $y=b_1\ldots b_n$ alors $x\bullet y= a_1\ldots a_m b_1\ldots b_n$
  La concaténation est associative et non commutative.
  2- **Fonction Longueur :** la fonction f définie de X* vers N est notée | | donne la longueur d'un mot.
  Remarque :
  - $|\varepsilon| = 0$
  - X* munie de la concaténation et le mot vide (X*, • , ε) est un monoïde engendré par X.
  - La fonction longueur est un morphisme de monoïde de (X*, • , ε) vers (N,+,0)

3- **Définition d'un langage** : soit X un ensemble d'alphabet, on appelle un langage sur X, un ensemble de mots sur X ou un sous-ensemble de X*.

Exemple :

1- Soit X={0,1} L={01,11,101} est un langage sur X.
2- Soit X={a,+,*,(,)} ; L est le langage formé des expressions arithmétiques avec parenthèses sur a. ((a+a)*a) est mot de L .

Remarque : le monoïde X* engendré par X est infini alors qu'un langage peut être fini ou infini.

4- **Opérations sur les langages** : soit X un alphabet et **L** et **L'** des langages sur X : par analogie aux ensembles, nous définissons les opérations suivantes sur les langages :

a) L'union : $L+L' = L\cup L' = \{ w \in X^* / w\in L \text{ ou } w\in L' \}$
b) L'intersection : $L\cap L' = \{ w \in X^* / w\in L \text{ et } w\in L' \}$
c) Le produit ou la concaténation : $L\times L' = \{ xy \in X^* / x\in L \text{ et } y \in L' \}$
d) La puissance : $L^i = L \times L^{i-1}$ avec $i \geq 2$
e) Les fermetures :
Fermeture transitive et réflexive $L^* = L^0 \cup L^1 \cup L^2 \dots = \cup L^i$ tel que $i\geq 0$ et $L^0 =\{ \varepsilon \}$
Fermeture transitive : $L^+ = L^1 \cup L^2\cup L^3 \dots = \cup L^i$ tel que $i\geq 1$
f) Miroir ou inverse : $L^R = \{ w^R / w \in L\}$ tel que si $w = x_1x_2..x_n$
alors $w^R= x_n\dots x_2x_1$

Propriétés des opérations :

1- $L^+ = L \bullet L^* = L^* \bullet L$
2- $L^* = (L^*)^*$
3- $(L \bullet L')^R = L'^R \bullet L^R$
4- Le produit des langages est associatif, non commutatif, distributif par rapport à l'union et non distributif par rapport à l'intersection :
  - $L \bullet(L_1\bullet L_2) = (L \bullet L_1) \bullet L_2)$
  - $L \bullet L_1 \neq L_1\bullet L$
  - $L \bullet(L_1+ L_2) = L \bullet L_1 + L\bullet L_2$
  - $L \bullet(L_1\cap L_2) \neq L \bullet L_1 \cap L \bullet L_2$

Un langage est toujours utilisé de 2 manières :

1- **Emission de mots** : cet aspect est lié à la génération des mots du langage en respectant la grammaire qui permet d'établir des règles précises à suivre. En informatique, cette opération est assurée par l'utilisateur.

2- **Réception des mots :** est la reconnaissance des mots d'un langage. Des machines formelles sont utilisées pour cela, elles sont appelées automates.

Chapter I: Introduction to Languages

1- **Definition of alphabet:** finite non-empty set of letters or symbols

2- **Definition of a word:** a word on an alphabet X is a finite sequence of elements of X.

Example: let the alphabet X={0,1}, the sequences 1,01,101 are words constructed on X.

- The inverse or mirror word of a word w, denoted w^R or w~, is the word obtained by reversing the letters of w. if $w=a_1 \dots a_n$ $w^R = a_n \dots a_1$
- A particular word denoted ε or λ is called an empty word.
- The set of all words formed from an alphabet X with the empty word is denoted $X^*=X^0+X^1+X^2+\dots$ Where $X^0=\{\varepsilon\}$
- **Operations on words:**
  1- **Concatenation:** let x and y be two words of X*, the concatenation noted x •y is defined by:
  If $x=a_1 \dots a_m$ and $y=b_1 \dots b_n$ then $x \bullet y= a_1 \dots a_m b_1 \dots b_n$
  Concatenation is associative, not commutative.
  2- **Length function:** the function f defined from X* to N is denoted | | gives the length of a word.
  Noticed :
  - $|\varepsilon| = 0$
  - X* equipped with concatenation and the empty word (X*, •, ε) is a monoid generated by X.
  - The length function is a monoid morphism from (X*, •, ε) to (N,+,0)

3- **Definition of a language** : let X be a set of alphabets, we call a language on X, a set of words on X or a subset of X*.

Example :

1- Let X={0,1} L={01,11,101} be a language on X.
2- Let X={a,+,*,(,)} ; L is the language formed by arithmetic expressions with parentheses on a. ((a+a)*a) is a word in L .

Note: the monoid X* generated by X is infinite while a language can be finite or infinite.

4- **Operations on languages** : let X be an alphabet and **L** and **L'** be languages on X: by analogy to sets, we define the following operations on languages:

a) The union: $L+L' = L \cup L' = \{ w \in X^* / w \in L \text{ or } w \in L' \}$
b) The intersection: $L \cap L' = \{ w \in X^* / w \in L \text{ and } w \in L' \}$

c) The product or concatenation: $L \times L' = \{ xy \in X^* / x \in L \text{ and } y \in L' \}$
d) The power: $L^i = L \times L^{i-1}$ with $i \geq 2$
e) The closures:
Transitive and reflexive closure $L^* = L^0 \cup L^1 \cup L^2 \ldots. = \cup L^i$ such that $i \geq 0$ and $L^0 = \{ \varepsilon \}$
Transitive closure: $L^+ = L^1 \cup L^2 \cup L^3 \ldots = \cup L^i$ such that $i \geq 1$
f) Mirror or inverse: $L^R = \{ w^R / w \in L\}$ such that if $w = x_1 x_2 .. x_n$
then $w^R = x_n \ldots x_2 x_1$

Properties of operations:

1- $L^+ = L \bullet L^* = L^* \bullet L$
2- $L^* = (L^*)^*$
3- $(L \bullet L')^R = L'^R \bullet L^R$
4- The product of languages is associative, noncommutative, distributive with respect to the union and nondistributive with respect to the intersection:
   - $L \bullet (L_1 \bullet L_2) = (L \bullet L_1) \bullet L_2)$
   - $L \bullet L_1 \neq L_1 \bullet L$
   - $L \bullet (L_1 + L_2) = L \bullet L_1 + L \bullet L_2$
   - $L \bullet (L_1 \cap L_2) \neq L \bullet L_1 \cap L \bullet L_2$

A language is always used in 2 ways:

1- **Word emission**: this aspect is linked to the generation of words of the language while respecting the grammar which allows establishing precise rules to follow. In computer science, this operation is ensured by the user.

2- **Word reception:** is the recognition of words of a language. Formal machines are used for this, they are called automata.