# Contents

# Chapter 1

# Mathematical Preliminaries

We review in this unit some mathematical preliminaries that will be needed in the rest of the book. We start by defining some basic properies of sets, to see later issues related to the cardinality of finite and infinite sets.

## 1.1 Remembering sets and their properties

**Definition 1.1** *Power set*

*Given a set $A$, the power set of $A$, denoted as $2^A$, is defined as the set containing all possible subsets of $A$:*

$$2^A = \{B \mid B \subseteq A\}$$

*Note: For every set $A$, $A \in 2^A$ and $\emptyset \in 2^A$ ($\emptyset$ is the empty set).*

**Definition 1.2** *Proper subset*

*Given a set $A$, $B$ is a proper subset of $A$ iff $B \subseteq A \ \wedge \ B \neq A \ \wedge \ B \neq \emptyset$.*

*Notation: 1) $B \subseteq A$ : $B$ is a subset of $A$. 2) $B \subset A$ : $B$ is a proper subset of $A$.*

**Definition 1.3** *Partition*

*Given a set $A$ , $\Pi \subseteq 2^A$ is a partition of $A$ iff the following conditions are true:*

1. $\bigcup_{A_i \in \Pi} A_i = A$
2. $A_i \cap A_j = \emptyset \ \ \forall i \neq j$
3. $A_i \neq \emptyset \quad \forall A_i \in \Pi$

**Definition 1.4** *Relation*

*Given two sets $A$ and $B$, a relation $R$ from $A$ (initial set) to $B$ (final set) is a subset of the Cartesian product $A \times B$, i.e., $R \subseteq A \times B$.*

*Note: The Cartesian product of two sets $A$ and $B$, denoted by $A \times B$, is defined as:*

$$A \times B = \{(a, b) \mid a \in A,\ b \in B\}$$

*$A \times B$ is the set that includes all pairs such that the first component of the pair is a member of $A$ while the second component is a member of set $B$.*

**Definition 1.5** *Binary relation*

*Given a set $A$, $R$ is a binary relation on $A$ if $R$ is a subset of the Cartesian product of $A \times A$ ($R \subseteq A \times A$).*

**Definition 1.6** *Properties of Binary relations*

*Given a set $A$ and a relation $R$ on $A$, we say that:*

$R$ *is Reflexive iff* $(a, a) \in R\ \forall a \in A$.
$R$ *is Symmetric iff* $(a, b) \in R \Rightarrow (b, a) \in R$.
$R$ *is antisymmetric iff* $(a, b) \in R\ \wedge\ a \neq b \Rightarrow (b, a) \notin R$.
$R$ *is Transitive iff* $(a, b) \in R\ \wedge\ (b, c) \in R \Rightarrow (a, c) \in R$.

**Definition 1.7** *Equivalence relation*

*An Equivalence relation $R$ on a set $A$ is a relation that is reflexive, symmetric and transitive.*

**Definition 1.8** *Equivalence Class*

*Let $A$ be a set and $R$ an equivalence relation on $A$. Let $a$ be an element of $A$ ($a \in A$) . Then, the Equivalence class of $a$, denoted as $[a]$, is defined as:*

$$[a] = \{b \in A \mid (a, b) \in R\}$$

*Note: An aquivalence relation on a set $A$ defines a partition of the set where each element of the partition defines an equivalence class.*

**Definition 1.9** *Identity relation ($I$)*

*Given a set $A$ , the Identity relation $I$ is defined as $I = \{(a, a) \mid a \in A\}$.*

**Definition 1.10** *Inverse (or converse) relation* ($R^{-1}$)
*Given a set $A$ and a relation $R$ on $A$ the Inverse relation of $R$ is defined as:*

$$R^{-1} = \{(a, b) \mid (b, a) \in R\}$$

**Definition 1.11** *Power of a relation* ($R^n$)
*Given a set $A$ and a relation $R$ on $A$, the power of $R$ is defined as:*

*For $n = 1 : R^1 = R$.*
*For $n \leq 2 : (a, b) \in R^n$ iff $\exists\, x \in A \mid (a, x) \in R^{n-1} \ \wedge \ (x, b) \in R$.*

**Definition 1.12** *Closure of a relation*
*Given a set $A$ and a relation $R$, we define:*
*The reflexive closure of $R$ as $R \cup I$.*
*The symmetric closure of $R$ as $R \cup R^{-1}$.*
*The transitive closure of $R$ as $R^\infty$, where $R^\infty = \bigcup_{n=1}^{\infty} R^n$.*

**Definition 1.13** *Mapping*
*A mapping $f$ from a set $A$ to a set $B$, denoted $f : A \to B$ is a relation from $A$ to $B$ such that $\forall a \in A \ \exists! \ (a, b) \in f$.*

**Definition 1.14** *Function*
*We will use the term function as a synonym to mapping, and then the definition of function is the one given previously in 1.13. Usually functions are seen as input-output process writing $f(a) = b$ instead of $(a, b) \in f$.*

**Definition 1.15** *Domain of a function (also valid for an application)*
*Given the function $f : A \to B$. The Domain of $f$ ($Dom(f)$ ) is defined as:*

$$Dom(f) = \{a \in A \mid f(a) = b \text{ with } b \in B\}.$$

**Definition 1.16** *Range, Codomain or Image of a function.*
*Given the function $f : A \to B$ the range of $f$ ( $Rg(f)$ ) is defined as:*

$$Rg(f) = \{f(a) \in B \mid a \in A\}.$$

**Definition 1.17** *Injective function (one-to-one function)*
*A function $f : A \to B$ is injective iff $x \neq z \Rightarrow f(x) \neq f(z)$.*
*Note: $f$ is injective iff $f(x) = f(z) \Rightarrow x = z$.*

**Definition 1.18** *Onto or surjective function*
*A function $f : A \to B$ is surjective iff $Rg(f) = B$.*

**Definition 1.19** *Bijection*
*Given a function $f : A \to B$, $f$ is a bijection between $A$ and $B$ if it is both injective and onto.*

According to the definition given, in order to demonstrate that a given function is a bijection we should demostrate that it is both injective (one-to-one) and onto. To show that $f$ is injective, we write $f(x) = f(z)$ , solve the Equation , and if there is a single solution such that $x = z$ , then $f$ is injective. On the other hand, to show that a function is onto we write $f(x) = b$, solve in terms of $x$ , and if this is always solvable then function $f$ is onto.

**Example 1.20** *Let $I$ be the set of Odd numbers. Let $f : N \to I$ a function defined by $f(n) = 2n + 1$ . We will demonstrate that $f$ is a bijection between $N \to I$ .*
*a) We first show that is injective:*
*$f(x) = f(z) \Rightarrow 2x + 1 = 2z + 1 \Rightarrow 2x = 2z \Rightarrow x = z$ that is the unique solution.*
*b) we then check if the function is onto: $f(x) = b \Rightarrow 2x + 1 = b \Rightarrow 2x = b - 1 \Rightarrow x = (b - 1)/2$ as $b$ is odd, $(b - 1)$ is even, and then always $(b - 1)/2 \in N$.*

**Definition 1.21** *Internal operation*
*Let $A$ be a set, every application $f : A \times A \to A$ is an internal operation on $A$. It can be written as $A \times A \to A$.*
*Note: The concept of internal operation (binary) can be also extended to n-ary operations (Unitary, ternary, etc).*

**Definition 1.22** *Associative property*
*Let $A$ be a set and let $\bullet$ be an internal operation on $A$ $(\bullet : A \times A \to A)$ . The operation $\bullet$ is associative iff $(x \bullet y) \bullet z = x \bullet (y \bullet z) \forall x, y, z \in A$.*

**Definition 1.23** *Semigroup*
*A Semigroup is a pair $(A, \bullet)$, where $A$ is a set and $\bullet$ is an internal associative operation on this set.*

**Definition 1.24** *Neutral element $(e)$*
*Given an operation $\bullet$ over a set , we say that $e \in A$ is the neutral element for this operation iff $\forall a \in A a \bullet e = e \bullet a = a$.*

**Definition 1.25** *Monoid*
*A monoid is a semigroup with neutral element.*

**Definition 1.26** *Closure of a set for an operation*
*Let $A$ be a set and $\bullet$ an operation on $A$ , and let $B \subseteq A$ . $B$ is closed under the $\bullet$ operation iff $x \bullet y \in B \; \forall \; x, y \in B$.*

**Definition 1.27** *Strict closure of a set for an operation*
*Let $A$ be a set and $\bullet$ an operation on $A$. Let $B \subset A$ . The strict closure of $B$ for $\bullet$, denoted $B\bullet$ , is defined as:*

1. $x \in B \Rightarrow x \in B\bullet$
2. $x, y \in B\bullet \Rightarrow x \bullet y \in B\bullet$
3. *No other element is a member of $B\bullet$ .*

**Definition 1.28** *Extended Closure*
*Let the pair $(A, \bullet)$ be a monoid , and let $B \subset A$ . The closure set of $B$ for the operation $\bullet$ , noted as $B \bullet e = B$ , is defined as :*

$$B \bullet e = B \bullet \cup \{e\}$$

*where e is the neutral element of the monoid.*
*Note: Note that the closure for an operation is a property that can be true or false, while the closure is an operation and not properties.*

## 1.2 Finite and Infinite Sets

**Definition 1.29** *Equinumerous set*
*Let $A$ and $B$ be two sets. They are equinumerous if $\exists \; f : \; A \to B \mid f$ is bijection.*

**Definition 1.30** *Finite set*
*A set a is finite if it is the empty set or $\exists n \in N \mid \{1, 2, \ldots, n\}$ and $A$ are equinumerous.*

**Definition 1.31** *Cardinality of non-empty finite sets*
*The cardinality of a non-empty finite set $A$ is $n$, denoted $||A|| = n$, iff $A$ and $\{1, 2, \ldots, n\}$ are equinumerous.*

**Definition 1.32** *Cardinality of the empty set*
*The cardinality of the empty set is zero:*

$$||\emptyset|| = 0$$

.

**Definition 1.33** *Infinite set*
*A set A is infinite iff it is not finite.*

**Definition 1.34** *Countably infinite set*
*A set A is countable infinite iff A and N are equinumerous.*

**Definition 1.35** *Countable and non countable sets*
*Let A be a set, we say that is countable in case it is finite or infinite countable. In any other case we say that it is uncountable.*

**Proposition 1.36** *Every subset of a countable set is countable.*

**Proof** We know that a countable set can be finite or countably infinite. On one hand, it is trivial to see that every subset of a finite set is also finite, and thus countable. On the other, a subset of an infinite countable set can be finite or infinite. The first case corresponds to a countable set. In the second case, we have a set $A$ that is countably infinite (through the bijection $g : N \to A$ ), and an infinite subset $B \subseteq A$ .

Given $D \subseteq A$ , and defining the minimum of a set $D$ , noted $min(D)$, as:

$$min(D) = g(min\{i \in N \mid g(i) \in D\})$$

Defining $f : N \to B$ as:

$$f(n) = min(B - \{f(m) \mid m < n\})$$

that for construction is a bijection implying that $B$ is countable.

**Proposition 1.37** *The union of any finite number of countably infinite sets is countably infinite.*

**Proof** We suppose that the sets are countably infinite and disjoints, as in the former case the demonstration is trivial while in the latter is reducible to disjoints sets. We a technique of interweaving the enumeration of several sets known as "dovetailing". In this way we establish a bijection between the union of the sets and the set of Natural numbers ( $f : A_0 \cup A_1 \cup \ldots \cup A_k \to N$) according to Fig. 1.1.

The expression of this function for $n$ sets is : $f_n(a_{ij}) = nj + i$

**Proposition 1.38** *The union of a countably infinite collection of countable sets is countable.*

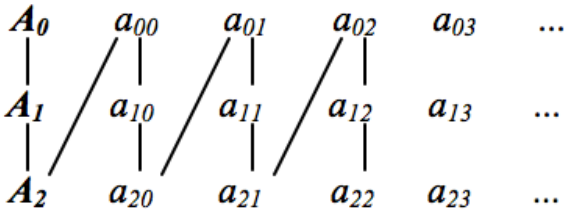


Figure 1.1: Basic dovetailing enumeration technique

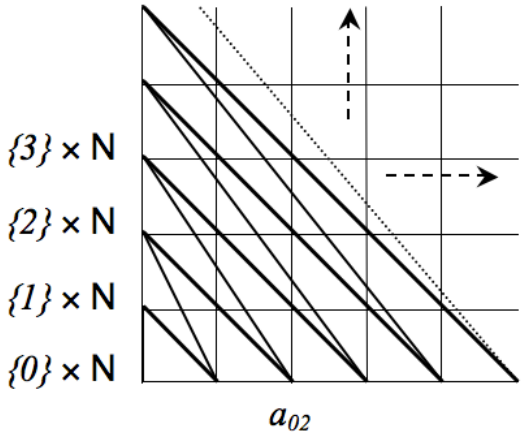


Figure 1.2: Dovetailing enumeration technique used to demonstrate proposition 1.38.

**Proof** Following the same ideas as in the previous demonstration, we construct a bijection between the elements of the union of sets and the set of natural numbers according to the diagram shown in Figure 1.2

The expression of $f$ in this case is: $f(a_{ij}) = \frac{(i+j)(i+j+1)}{2} + j$.

**Corollary 1.39** *The Cartesian product of two countably sets is countable.*

**Proposition 1.40** *The finite power of a countable set is countable.*

**Proof** We apply the induction technique:

a) $N^1 = N$, it is countable by definition.

b) Let's suppose that $N^n$ is countable.

c) $N^{n+1} = N^n \times N \Rightarrow N^{n+1}$ is countable, because of b) and corollary **1.38**.

Note: The induction technique will be seen later in this chapter. The proposition is also valid for $N^0$, as it is finite and thus countable.

**Example 1.41** *The set of odd numbers is countable as it is a subset of a countable set. Thus, we know that there should exist a bijection between this set and the set of Naturals numbers. A sample of this bijection is* $f(n) = 2n + 1$.

*The set of odd numbers larger than 100 is countable, as it is a subset of a countable set. The bijection with the naturals numbers in this case is* $f(i) = i + 100$.

*For the same reason, the set of numbers multiple of seven and larger than seventeen is also countable. A bijection with the naturals numbers is* $f(n) = 7n + 21$.

**Definition 1.42** *Cardinal of the set of Natural Numbers*

$||N|| = \aleph_0$ *(aleph zero).*

**Theorem 1.43** *Cantor's Theorem: The cardinal of the set of Real numbers is not equal to the cardinal of the set of Natural numbers:* $||N|| \neq ||R||$.

**Proof** : To show that $||N|| \neq ||R||$ instead of using the whole real line we will work only with the $(0, 1)$ interval, as this interval is equinumerous to the set of Real numbers $R$. The following function:

$$g(x) = tg\left[\left(x - \frac{1}{2}\right)\Pi\right]$$

establishes a bijection between $(0, 1)$ and $R$ .

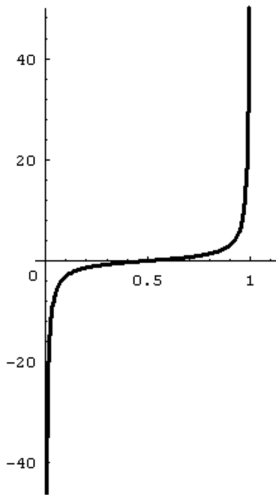


Figure 1.3: Tangent function

The demonstration that the interval $(0, 1)$ is not countable will be based on a contradiction that arises if we suppose the opposite.

Thus, let's assume that the interval $(0, 1)$ is equinumerous to $N$ . There shoud be a bijection $f : N \to (0, 1)$ between these two sets. Let's denote $r_i$ to $f(i)$, to have:

$$\begin{aligned} 0 &\to f(0) = r_0 \\ 1 &\to f(1) = r_1 \\ 2 &\to f(2) = r_2 \\ &\dots \end{aligned}$$

If the previous enumeration is holds, then we can assert that all real numbers in the interval $(0, 1)$ there will be included in the folowing list:

$$\begin{aligned} r_0 &= 0,\ d_{00}\ d_{01}\ d_{02}\ d_{03}\dots \\ r_1 &= 0,\ d_{10}\ d_{11}\ d_{12}\ d_{13}\dots \\ r_2 &= 0,\ d_{20}\ d_{21}\ d_{22}\ d_{23}\dots \\ r_3 &= 0,\ d_{30}\ d_{31}\ d_{32}\ d_{33}\dots \\ &\dots \end{aligned}$$

where each $d_{ij}$ (with $i, j \in N$ ) is the j-th digit in the complete decimal expression of the real number $r_i$ (the $i$-th real number of the list). For the

real numbers with two different decimal expressions (infinite zeros or nines to the right) we choose the expression with infinite number of nines.

Let $r = 0, d_0\ d_1\ d_2\ d_3 \ldots \in (0,1)$ where

$$\forall\ i \leq 0 \begin{cases} d_i = 4 \ \text{ if } \ d_{ii} \neq 4 \\ d_i = 5 \ \text{ if } \ d_{ii} = 4 \end{cases} \tag{1.1}$$

($d_{ii}$ is the $i$-th digit in the diagonal of the previous list).

Thus, $d_i \neq d_{ii}\ \forall\ i \leq 0 \Rightarrow r \neq r_i\ \forall\ i \leq 0 \Rightarrow r$ is not in the previous list.

But this contradicts the initial supposition, as we said that all real numbers where in the list. Then we conclude that $f$ is not a bijective function and then $R$ is no equinumerous to $N$.

**Definition 1.44** *Cardinal of R*

*The Cardinal of the set of Real numbers R is aleph one:* $||R|| = \aleph_1$.

**Definition 1.45** *Transfinite numbers*

$\aleph_i$ *( aleph i ) is the i-th transfinite number. The set* $\{\aleph_i \mid i \in N\}$ *is the set of all transfinite numbers.*

**Definition 1.46** *Cardinal of infinite sets*

*The cardinal of an infinite set A is* $\aleph_i$ *iff is equipotential to a set with known cardinal* $\aleph_i$*. We denote it as:* $||A|| = \aleph_i$*, with* $i \in N$.

**Example 1.47** *All three sets of the example 1.41 have cardinal* $\aleph_0$ *as bijection can be established between each of them and the set of naturals numbers.*

*The interval* $(0,1)$ *has cardinality* $\aleph_1$ *, as a bijection can be established between this set and the set of real numbers. A function for such bijection is* $f(x) = tg[(x - 0.5)\ \pi]$.

*The interval* $(0,2)$ *has cardinality* $\aleph_1$*, as there exists a bijection between this set and the interval* $(0,1)$*, which we know its cardinality (*$\aleph_1$*). A function to establish the bijection is* $f(x) = 2x$.

*Working with infinte sets can be sometimes counterintutive. The best way to understand it, could be to think the cardinality of infinite sets as a measure of its "density". The cardinality of N is known as the number Aleph zero (*$\aleph_0$*), the first transfinite number (the "less dense" infinite). Also* $||Q|| = \aleph_0$ *as* $a/b$ *can be considered like a different notation for* $(a,b) \in N \times N$.

Finally, two more results we need to know: $\aleph_1 - \aleph_0 = \aleph_1$ and $2^{\aleph_0} = \aleph_1$ .

## 1.3 Fundamental proof techniques

We are going to analyze three fundamental proof techniques: the induction principle, the pigeonhole principle and the diagonalization technique. We will show by contradiction the validity of the use of the three techniques.

**Induction principle**

Let $A$ a set of natural numbers for which:

a) $0 \in A$ ;

b) $\{0, 1, \ldots, n\} \subseteq A \Rightarrow n + 1 \in A \; \forall n \in N$ .

Then $A = N$.

**Proof** Let $A \subseteq N$ and let conditions a) and b) be true. Suppose that $A \neq N$, then:

using condition a):

$A \neq N \Rightarrow N - A \neq \emptyset \Rightarrow \exists\, m \notin A \mid m = min(N - A) \Rightarrow m \neq 0$,

and using condition b):

$\Rightarrow \{0, 1, \ldots, m-1\} \subseteq A \Rightarrow m \in A \Rightarrow A = N$

The induction technique is used to demonstrate statements as: "For all natural numbers, property $P$ is true".

The principle is applied to the set $A = \{n \mid \text{P is true of } n\} = \{n|P(n)\}$ following the three next instructions:

Basis step (B.S.): We show that $P(0)$ is true (sometimes we can use $P(n)$ with $n > 0$ )

Induction Hypothesis (I.H.): Let's suppose that $\exists\ n \leq 0 \mid P(i)$ $\forall\ i = 0, 1, \ldots, n$

Induction step (I.S.): Demostrate using the I.H. that $P(n+1)$ is true.

Then because of the induction principle $A = N$ , i.e., $\forall n P(n)$ .

**Example 1.48** *Show that* $1 + 2 + \ldots + n = \frac{n^2+n}{2}$, $\forall\ n \leq 0$

*B.S.: Trivial for* $n = 0$

*I.H.: Let's suppose that* $\exists n \leq 0$ *such that* $1 + 2 + \ldots + m = \frac{m^2+m}{2}$ *with* $m \leq n$

*I.P.: We show the validity for* $n + 1$ *using the I.H.:*

$1 + 2 + \ldots + n + (n + 1) = (1 + 2 + \ldots + n) + (n + 1) =$

$= \frac{n^2+n}{2} + (n + 1) = \frac{n^2+n+2n+2}{2} = \frac{n^2+2n+1+n+1}{2} = \frac{(n+1)^2+(n+1)}{2}$

**Pigeonhole principle**

Let $A, B$ be finite sets, such that $||A|| > ||B|| > 0$, and let $f$ be a function from $A$ to $B$. Then the function $f$ is not injective.

**Proof** We will use the induction principle applied to the cardinal of $B$.

B.C.: Let $A, B$ be finite sets, such that $||A|| > ||B|| = 1 (i.e, B = \{b\})$ , and let $f$ be equal to $f : A \to B$.
$f : A \to B \Rightarrow \exists\ a_1, a_2 \in A\ \ | f(a_1) = f(a_2) = b, \text{with}\ \ a_1 \neq a_2 \Rightarrow f$ is not injective.

I.H.: If $A, B$ are finite sets, and $||A|| > ||B|| = n$ , with $n \geq 1$, and $f$ is a function $f : A \to B$, then $f$ is not injective. (We do not need to demonstrate this, we just suppose is valid)

I.S.: Let $A, B$ be finite sets, $||A|| > ||B|| = n+1$ , with $n \geq 1$, and $f$ is a function $f : A \to B$.

Let be $b \in B$ .

We know that $B - \{b\} \neq \emptyset$ because $||B|| \geq 2$.

Let's analyze the set $f^{-1}(b) = \{a \in A \mid f(a) = b\}$.

There are two possible cases:

$||f^{-1}(b)|| \geq 2$ and $||f^{-1}(b)|| \leq 1$ .

If $||f^{-1}(b)|| \geq 2 \Rightarrow \exists\ a_1, a_2 \in A\ \mid f(a_1) = f(a_2) = b$, with $a_1 \neq a_2$ $\Rightarrow f$ is not injective.

If $||f^{-1}(b)|| \leq 1$ then,

let $g$ be $g : A - f^{-1}(b) \to B - \{b\}\ \mid g(a) = f(a) \forall\ a \in A - f^{-1}(b)$ .

But as $||f^{-1}(b)|| \leq 1 \Rightarrow ||A - f^{-1}(b)|| \geq ||A|| - 1$,

and also $||A|| > ||B|| = n + 1 \Rightarrow ||A|| - 1 > ||B - \{b\}|| = n$

both conditions $\Rightarrow$

$||A - f^{-1}(b)|| > ||B - \{b\}|| = n$, using the I.H $\Rightarrow$

g is not injective $\Rightarrow$

$\exists a_1, a_2 \in A - f^{-1}(b) \mid g(a1) = g(a2) = c$, with $a_1 \neq a_2$ and $c \in B - \{b\} \Rightarrow f(a_1) = f(a_2) \Rightarrow f$ is not injective.

**Example 1.49** *Demonstrate that every proper subset of a finite set is not equinumerous to it.*

*Let $||A|| = n$ and let $B$ be a proper subset of $A \Rightarrow ||A|| > ||B|| > 0 \Rightarrow$*

*Using the Pigeonhole pronciple :*

*There is no injective function from $A$ to $B \Rightarrow$*

*there is no possible bijection between $A$ and $B \Rightarrow A$ and $B$ are not equinumerous.*

**Diagonalization principle**

Let $R$ be a binary relation defined on a set $(R \subseteq A \times A)$ .

Let $D$ be the diagonal set of $R$, defined as: $D = \{a \in A | (a, a) \notin R\}$ .

Let $R_a = \{b \in A \mid (a, b) \in R\}$ , then $D \neq R_a\ \forall a \in A$.

**Proof** $\forall a \in A$ we have that:
$(a,a) \in R \Rightarrow (a \in R_a) \wedge (a \notin D) \Rightarrow R_a \neq D$
$R_a \neq D \; \forall a \in A$
$(a,a) \notin R \Rightarrow (a \notin Ra) \wedge (a \in D) \Rightarrow Ra \neq D$

**Example 1.50** *Demonstrate that* $2N$ *is not countable.*
*Suppose that* $2^N$ *is countably infinite, then* $\exists \; f : N \to 2^N \mid f$ *is a bijection. Thus* $2^N$ *can be ordered as* $2^N = \{S_0, S_1, S_2, \ldots\}$ *, where* $S_i = f(i) \; \forall \; i \in N$
.

*a) Demonstration with implicit use of the principle:*
*Let* $D = \{n \in N \mid n \notin S_n\}$ *As* $D \subseteq N \Rightarrow D \in 2^N \Rightarrow \exists! K \in N | D = S_k$ *But, does* $k \in S_k$ *?*
*If* $k \in S_k \Rightarrow k \notin D \Rightarrow k \notin S_k \Rightarrow D \neq S_k \; \forall k \in N \Rightarrow D \notin Rg(f) \Rightarrow D = S_k$ *f is not a bijection* $\Rightarrow 2^N$ *is uncountable.*
*If* $k \notin S_k \Rightarrow k \in D \Rightarrow k \in S_k$
*b) Demonstration with explicit use of the principle:*
*Let* $R = \{(i,j) \mid j \in f(i)\}$ *on* $N$*.*
*Let* $D$ *be the diagonal set of* $R : D = \{n \in N \mid (n,n) \notin R\}$*.*
*As* $S_i = f(i)$ *we have that* $R_i = \{j \in N \mid (i,j) \in R\}$ *is* $S_i$*.*
*The diagonalization principle tell us that:*
$R^n \neq D \; \forall \; n \in N \Rightarrow D \notin Rg(f) \Rightarrow f$ *is not a bijective function* $\Rightarrow 2^N$ *is not countable.*

The previous example is known as Cantor's theorem named on the honour of George Cantor, even if in reality the proper Cantor's theorem is a generalization of the previous example that states: "For every set, the cardinal of its power set is larger than the cardinal of the set".

# Chapter 2

# Languages and Grammars

## 2.1 Languages

**Definition 2.1** *Alphabet*

*An alphabet is a finite non-empty set whose elements are symbols. We normally use the capital greek letter Sigma ($\Sigma$) to represent an alphabet, even if sometimes we may use latin capital letters as well.*

**Definition 2.2** *Strings*

*A string over an alphabet is a finite sequence of symbols from the same alphabet. Usually, we will use letters from the u and so on ($v, w, x, y, z$) to represent strings and also letters of the Greek alphabet ($\alpha, \beta, ...$).*

**Definition 2.3** *The empty string*

*The empty string is a string containing no symbol. We denote it by the greek letter epsilon $\epsilon$ .*

**Definition 2.4** $\Sigma^*, \Sigma^+$

$\Sigma^*$ *: set of all possible strings (including the empty string) over an alphabet* $\Sigma$ .

$\Sigma^+$ *: set of all possible strings, excluding the empty string over an alphabet* $\Sigma$ .

**Definition 2.5** *Length of a string ($|x|$)*

*The length of a string is defined as the number of symbols it contains.*

**Example 2.6** *Given the alphabet* $\Sigma = \{a, b\}$ :

$$|\epsilon| = 0$$
$$|a| = 1$$
$$|abab| = 4$$
$$\text{If } w = aba, \text{ then } |w| = 3$$

**Definition 2.7** *Ocurrences of a symbol*

*The symbol* $a \in \Sigma$ *ocurrs in the* $j$*-th position of chain* $w \in \Sigma^+$ *iff the* $j$*-th symbol of* $w$ *is* $a$ *. We use the notation* $w(j) = a$*, with* $0 < j \leq |w|$*. There is no symbol ocurrencies in the empty string. The number of times a symbol* $a$ *appears in a chain* $w \in \Sigma^*$ *is indicated by* $|w|_a$ *.*

Note: $|\epsilon|_a = 0 \; \forall a \in \Sigma$.

**Example 2.8** *Given* $\Sigma = \{a, b, c\}$ *and* $w \in \Sigma^*$ *with* $w = aba$ *, then:*

$$w(1) = a \;\; \& \;\; |w|_a = 2$$
$$w(2) = b \;\; \& \;\; |w|_b = 1$$
$$w(3) = a \;\; \& \;\; |w|_c = 0$$

**Definition 2.9** *Concatenation of strings*

*Two strings over the same alphabet (*$x, y \in \Sigma^+$*) can be combined to form a third string, written* $x \bullet y$ *or simply* $xy$*. Formally,* $xy$ *is the string that verifies the following two conditions:*

$$xy(j) = \begin{cases} x(j) \;\; \text{if } \; j \leq |x| \\ y(j - |x|) \;\; \text{if } \; j > |x| \end{cases} \qquad \text{with } 0 < j \leq |xy| \tag{2.1}$$

*Given a string* $x \in \Sigma^*$*,* $x\epsilon = \epsilon x = x$*.*

Note: Given three strings $x, y, z \in \Sigma^*$ , the following is true: $(xy)z = x(yz)$ . And thus, the pair ( $\Sigma^*, \bullet$) is a monoid, as concatenation is an associative internal operation with neutral element.

**Definition 2.10** *Substring*

*A string* $v \in \Sigma^*$ *is a substring of* $w \in \Sigma^*$ *iff* $\exists \; x, y \in \Sigma^* \mid w = xvy$*.*

Note: Every string is a substring of itself (taking $x = y = \epsilon$ in the definition). The empty string is a substring of every string (taking $x = w$ and $v = y = \epsilon$).

**Definition 2.11** *Suffix*
*Let $w, v \in \Sigma^*$ , $v$ is suffix of $w$ iff $x \in \Sigma^* \mid w = xv$.*

**Definition 2.12** *Prefix*
*Let $w, v \in \Sigma^*$ , $v$ is a prefix of $w$ iff $y \in \Sigma^* \mid w = vy$.*

**Example 2.13** *Let $v = ata$. If $w = qatar$ then $v$ is substring of $w$ as $x = q, y = r \; \exists \; \Sigma^* \mid w = xvy$*
*If $w = atalaya$ then $v$ is a prefix of $w$ as $x = laya \in \Sigma^* \mid w = vx$*
*If $w = data$ then $v$ is suffix of $w$ as $x = d \in \Sigma^* \mid w = xv$*

**Definition 2.14** *Power of a string ($w^n$)*

$$w^n = \begin{cases} \epsilon, & \text{if } n = 0 \\ w^{n-1}w, & \text{if } n > 0 \end{cases} \tag{2.2}$$

**Example 2.15** *Compute $(pa)^2$*

$$(pa)^2 = \begin{cases} \text{for } i = 1 : (pa)^2 = (pa)^1 \, pa \\ \text{for } \; i = 0 : (pa)1 = (pa)^0 \, pa \\ B.C. \; : (pa)^0 = 1 \end{cases} \tag{2.3}$$

$\Rightarrow (pa)^2 = ((\epsilon pa)pa) = papa.$

**Definition 2.16** *Reversal of a string ( $w^R$ ) (string spelled backwards)*

$$w^R \begin{cases} \text{if } \; |w| = 0 \Rightarrow w^R = w = \epsilon \\ |w| > 0 \;\; \text{and} \;\; w = ua \text{ with } u = \Sigma^* \text{ and } a \in \Sigma \Rightarrow w^R = au^R. \end{cases} \tag{2.4}$$

**Example 2.17** *Let $w = home$. Obtain $w^R$ .*
$(home)^R = e(hom)^R = em(ho)^R = emo(h)^R = emoh()^R = emoh = emoh$

**Proposition 2.18** *String reversal*
*For every string $x, w \in \Sigma^*$ we have that: $(wx)^R = x^R w^R$*

***Proof*** *: We will use the induction technique applied to the length of the string $x$.*

$$1^{st}.B.C. : |x| = 0 \Rightarrow x = \epsilon \Rightarrow (wx)^R = (w)^R = w^R = w^R = Rw^R = x^R\ w^R \tag{2.5}$$

$$2^{nd}.I.H. : |x| \leq n \Rightarrow (wx)^R = x^R w^R \tag{2.6}$$

$$3^{rd}.I.S. : Let |x| = n+1 \Rightarrow x = ua\ with\ u \in \Sigma^*,\ a \in \Sigma\ and\ |u| = n \tag{2.7}$$

$$(wx)^R = (w(ua))^R = ((wu)a)^R = a(wu)^R = au^R w^R = (ua)^R w^R = x^R w^R \tag{2.8}$$

$$\tag{2.9}$$

**Example 2.19** *Let $x, y \in \Sigma^*$, where $x = if$ and $y = then$.*
*$(xy)^R = y^R x^R$ as $(ifthen)^R = (then)^R\ (if)^R = nehtfi$*

**Definition 2.20** *Language*
*$L$ is a language over $\Sigma$ iff $L \in \Sigma^*$ .*

Note: A language is any set of strings over an alphabet $\Sigma$.

**Example 2.21**

$$\Sigma^*\ \ is\ a\ language \tag{2.10}$$

$$\{\epsilon\}\ \ is\ a\ language \tag{2.11}$$

$$\emptyset\ \ is\ a\ language \tag{2.12}$$

$$\{\{\epsilon\}\}\ \ is\ not\ a\ language\ (over\ the\ alphabet\ \Sigma) \tag{2.13}$$

$$\epsilon \neq \emptyset\ \ ; ||\{\epsilon\}|| = 1;\ \ ||\emptyset|| = 0 \tag{2.14}$$

Note: As we do not distinguish between symbols of an alphabet and strings of lenght 1 over the same alphabet, $\Sigma$ is also a language.

Since a language is simply a special kind of set, we can specify a finite language by listing all its strings.

**Example 2.22** *Let's define $\Sigma = \{a, b, c, d, f, r, z\}$. $L = \{aba, czr, d, f\}$ is a language over the alphabet $\Sigma$.*

However, most languages of interest are infinite, so that listing all the strings is not possible. Languages that might be considered $\{0, 01, 011, 0111, \}$, $\{w \in \{0,1\}^* |$ w has an equal number of 0's and 1's $\}$, and $\{w \in \Sigma^* : w = w^R\}$. Thus we shall specify infinite languages by the scheme :

$$L = \{w \in \Sigma^* : w \text{ has property } P\}$$

Strings that belong to the language verify property $P$ while those that do not belong do not verify.

**Example 2.23** *Given* $\Sigma = \{0, 1\}$*, consider the language* $L \in \Sigma^*$, $L = w \in \Sigma^* \mid |w|_0 = 2n + 1$ *with* $n \leq N$ .

**Proposition 2.24** *If* $\Sigma$ *is an alphabet, then* $\Sigma^*$ *is countably infinite.*

**Proof** We need to show that $||\Sigma^*|| = 0$ , and for that we should propose a biyection $f : \Sigma^* N$.

Given an arbitrary order of the alphabet $\Sigma = \{a_1, a_2, \ldots, a_n\}$, the members of $\Sigma^*$ can be enumerated in the following way:

1. Strings of length $k$ (there are $n^k$ such strings) are enumerated before all strings of length $k + 1$.
2. The $n^k$ strings of length $k$ are ordered lexicographically, that is: The string $a_{i_1} \ldots a_{i_k}$ precedes the string if
   $\exists\, m \in N, 0 \leq m \leq k - 1, \mid i_p = j_p$ for $p = 1, \ldots, m$ and $i_{m+1} < j_{m+1}$.
   And in this way all strings of the language can be ordered.

The expression for the bijection, given $\Sigma = \{a_1, a_2, ..., a_n\}$, for a string $w = a_{i_1} a_{i_2} \ldots a_{i_{|w|}}$ with $i_k \in \{1, ..., n\}$ is :

$$f : \Sigma^* \to N$$
$$f(w) = \Sigma_{j=1}^{|w|} n^{|w|-j}\ i_j$$

**Example 2.25** *Given* $\Sigma = \{a_1, a_2\}$

$$\begin{aligned}
|w| = 0\ &\epsilon \ldots\ldots\ldots\ldots 0\\
|w| = 1\ &a_1 \ldots\ldots\ldots\ldots 1\\
&a_2 \ldots\ldots\ldots\ldots 2\\
|w| = 2\ &a_1a_1 \ldots\ldots\ldots 3\\
&a_1a_2 \ldots\ldots\ldots 4\\
&a_2a_1 \ldots\ldots\ldots 5\\
&a_2a_2 \ldots\ldots\ldots 6\\
|w| = 3\ &a_1a_1a_1 \ldots\ldots 7\\
&\vdots
\end{aligned}$$

## 2.2 Language representation

A central issue in the theory of computation is the representation of languages by finite specifications.

Given an alphabet $\Sigma$, and given $\Sigma^*$ the set of all possible strings over the alphabet, any subset L of $\Sigma^*$ is a language over $\Sigma$ , and we call $2^{\Sigma^*}$ to the

set of all languages over $\Sigma$. We know the following about the cardinality of these sets:

$\Sigma$ is finite

$\Sigma^*$ is countably infinite

$L$ is countable (countably infinite or finite)

$2^{\Sigma^*}$ is uncountably infinite (as$2^{\aleph_0} = \aleph_1$)

Let $\Sigma_M$ be the alphabet of the natural language plus all mathematical symbols, that is , our mathematical alphabet, that we use in our mathematical language. Any description o representation (finite) of a language L over $\Sigma$ is not other than a string of $\Sigma_M^*$ .

We say that a string $r \in \Sigma_M^*$ is a representation of a language $L$ over $\Sigma$ if a relation exists that belongs to $\Sigma_M^* \times 2\Sigma^*$ such that the pair $(r, L) \in RS$ and for all $L' \neq L$ $(r, L') \notin RS$ . That is: $r \in \Sigma_M^*$ is a representation of a language $L$ over $\Sigma$ iff

$\exists\ RS\ \subset \Sigma_M^* \times 2^{\Sigma^*} \mid (r, L) \in RS\ \wedge\ \forall L' \neq L\ \ (r, L') \notin RS$

If r is a representation of a language L then L is representable. The set of all representable languages we will be denoted as L.REP . Two representations are equivalent if they represent the same language.

Given a representation system (RS) the set of language representations is defined for that RS. This set, denoted by REP, is always countable as is a subset of $\Sigma_M^*$ that is countable.

## 2.3 Cardinality, Representation and Languages

A representation is no other than a string over an alphabet $\Sigma_M$, that represents a single language, that is , if two representable languages are different their representation should be as well, and for this reason: $||L.REP|| \leq ||REP||$ .

We know that $||\Sigma_M^*|| = \aleph_0$, and as the set of representations belongs to $\Sigma_M^*$ follows that $||REP|| \leq \aleph_0$ .

On the other hand, it is trivial to see that there are an infinite number of finite languages ( L.finite ), and as every finite language is representable (at least by listing all its members), then $\aleph_0 \leq ||L.REP|| \leq ||REP||$ .

Putting all together: $\aleph_0 \leq ||L.REP|| \leq ||REP|| \leq \aleph_0 \Rightarrow ||L.REP|| = ||REP|| = \aleph_0$, and then, a countably infinite number ( $\aleph_0$ ) of representations exists and also for representable languages.

As by definition $L \subseteq \Sigma^*$, the set of all languages over an alphabet $\Sigma$ is $2^{\Sigma^*}$ . From what follows, as $||\Sigma^*|| = \aleph_0$ that $||2^{\Sigma^*}|| = \aleph_1$ . (There is a non countable infinite number ( $\aleph_1$ ) of languages.)

If from the total of languages ( $2^{\Sigma^*}$ ) we substract those that are representable (L.REP) we obtain the set of non-representable languages (L.NOREP), and as $\aleph_1 - \aleph_0 = \aleph_1$ we can ensure that there is an uncountably infinite number ( $\aleph_1$ ) of non-representable languages.

## 2.4 Language Representation

The standard form of defining an infinite language (also valid for the finite case) is by giving a property that the string of that language should verify: $L = \{w \in \Sigma^* | P(w)\}$.

Apart from the usual ways of stating the property $P$, for the case of formal languages there are two additional ways of defining a language: a recognizing device, and a generating device.

**Definition 2.26** *Conclusive Algorithm*

*Finite sequence of precise and finite instructions (unambiguas) such that given an input/question/problem, returns (compute) always , in finite time an output/answer/solution.*

**Definition 2.27** *Language Recognizing Device*

*A conclusive algorithm designed for a language L that can answer correctly the question:*

*Does the string w belong to L ?.*

**Definition 2.28** *Language generating device*

*A set of conclusive algorithms created for a Language L such that produce all and only the strings of L is a called a generator device of L.*

**Example 2.29** *Let* $L_1 = \{w \in \{a,b\}^* \mid w$ *does not contain the substring* $bbb\}$ .

A recognizing device of $L_1$ could be the following:

We keep reading the symbols of the input string from left to right and (after initialization of a counter to zero).

1. A counter is resetted every time we read an 'a'.
2. Add one to the counter every time a 'b' is encountered.
3. Stop and answer 'NO' if the counter reach three.
4. Stop and answer 'YES' if the string is read completely and the counter has not reached three.

A generating device of $L_1$ could be the following:

For producing a member of $L_1$ first we write nothing or we write 'b' or 'bb'; then we repeat, zero or more times the operation of writing 'a', or 'ab' or 'abb'.

Contrary to the recognizing devices, the generating devices are not algorithms, as they can be ambiguous (free choice of options). We will study recognizing devices that we call Automata (or Automaton in singular), while the generators will be the Grammars.

## 2.5 Grammars

We will see next the formal definiton of a grammar and the classification of grammars and languages.

**Definition 2.30** *Grammar*

*A grammar is a quadruple $G = (N, T, P, S)$ where:*

*$N$ is an alphabet, the nonterminal alphabet, formed by nonterminal symbols.*

*$T$ is an alphabet, the terminal alphabet, formed by terminal symbols.*

$$\begin{aligned} &N \cap T = \emptyset \\ &N \cup T = V \\ &S \in N \text{ are the axioma} \\ &P \subset (V^+ \times V^*) \text{ and it is finite} \end{aligned} \tag{2.15}$$

*The elements of $P$, $(\alpha, \beta) \in P$ , are the production rules and will be denoted as $\alpha \to \beta$. Then, $P$ is finite such as:*

$$P = \{\alpha \to \beta, \text{ with } \alpha \in V^+ \wedge \ \beta \in V^*\}.$$

*In general, the symbols of the nonterminal alphabet $(N)$ are represented by capital letters form the begining of the alphabet, while those belonging to the terminal $(T)$ are represented with standard letters from the beginning of the alphabet.*

**Definition 2.31** *To produce directly*

*Let $G = (N, T, P, S)$ be a grammar . Given $x \in V^+$ and $y \in V^*$, we say that $x$ produces directly $y$, noted as $x \Rightarrow y$ , if $\exists u, v \in V^*$, such that:*

$$a) x = uzv \tag{2.16}$$

$$b) y = u\beta v \tag{2.17}$$

$$c) \exists (z \to \beta) with z \in V^+ \text{ and } \beta \in V^* \tag{2.18}$$

**Example 2.32** *Let $aBaCaab$ be a string. If $(aB \to b) \in P$, then $aBaCaab \Rightarrow baCaab$ If $(C \to BaD) \in P$, then $aBaCaab \Rightarrow aBaBaDaab$*

**Definition 2.33** *To produce in n steps Let $G$ be a grammar $G = (N, T, P, S)$, and given $x, y \in V^*$ .*

*We say that $x$ produces $y$ in zero steps, denoted as $x \Rightarrow^0 y$, iff $x = y$ .*

*We say that $x$ produces $y$ in a single step, denoted $x \Rightarrow^1 y$, iff $x \Rightarrow y$ .*

*We say that $x$ produces $y$ in $n$ steps, with $n > 1$ , denoted $x \Rightarrow^n y$, iff:* $\exists z_1, z_2, \ldots, z_{n-1} \in V^+ \mid x \Rightarrow h \Rightarrow z_1, z_1 \Rightarrow z_2, \ldots, z_{n-1} \Rightarrow y$

**Definition 2.34** *To produce in at least one step We say that $x$ produces in at least one step $y$ , denoted $x \Rightarrow^+ y$, iff $\exists n > 0 \mid x \Rightarrow^n y$ .*

**Definition 2.35** *To produce We say that $x$ produces $y$ , denoted by $x \Rightarrow^* y$, iff $\exists n \leq 0 \mid x \Rightarrow^n y$ .*

As a summary: $\Rightarrow$ is a binary relation between strings over $V^*$
$\Rightarrow^+$ is the transitive closure.
$\Rightarrow^*$ is the reflexive and transitive closure.

**Definition 2.36** *Derivation, Length of a derivation Given $G = (N, T, P, S)$, a derivation in $G$ is a finite sequence of firect productions of the form:*

*$w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n$ where $w_0 = S$ and $w_i \in V^*$ with $0 \leq i \leq n$*

*The length of a derivation is $n$ $(n \in N)$.*

**Definition 2.37** *Sentential form*

*Let $G = (N, T, P, S)$ be a grammar. $\alpha$ is a sentencial form of $G$ iff $S \Rightarrow^* \alpha$ with $\alpha \in V^*$*

A sentential form of a grammar G is any sequence of grammar symbols (terminals or nonterminals) derived in 0 or more steps from the start symbol of $G$.

**Definition 2.38** *String generated by a grammar*

*String $y$ is generated by grammar $G = (N, T, P, S)$ iff $(S \Rightarrow^* y)(y \in T^*)$*

That is, $y$ is a sentential form containing only terminal symbols.

**Definition 2.39** *Language generated by a grammar Let $G = (N, T, P, S)$ , we define the language generated by $G$, denoted $L(G)$ , as:*

$L(G) = \{y \in T^* \mid S \Rightarrow^* y\}$

The set of all strings generated by the grammar.

**Example 2.40** *Let $G = (N, T, P, S)$ be a grammar such as:*

$$N = \{S\}$$
$$T = \{a, b\}$$
$$P = \{S \to aSb, S \to ab\}$$

*Applying the first production rule ($S \to aSb$) $n-1$ times, followed by the application of the second rule ( $S \to ab$ ), we obtain:*

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow \ldots \Rightarrow a^{n-1}Sb^{n-1} \Rightarrow a^nb^n$

*In ths expression (a derivation) all chains of symbols produced are sentential forms.*

*We obtain that $L(G) = \{w \in a, b^* \mid w = a^nb^n with n \geq 1\}$.*

**Example 2.41** *Given $G = (N, T, P, S)$ with:*

$$N = \{A, B\}$$
$$T = \{a, b\}$$
$$P = \{A \to AaA, A \to b\}$$
$$S = A$$

$$A \longrightarrow b \in L(G)$$
$$\uparrow$$
$$A \Rightarrow b$$

$$\begin{array}{lllllllll} A & \Rightarrow & AaA \Rightarrow & AaAaA & \Rightarrow & baAaA & \Rightarrow & baAab & \Rightarrow \quad babab \quad \in L(G) \\ \uparrow & & \uparrow & \uparrow & & \uparrow & & \uparrow & \\ A \to AaA & & A \to AaA & A \to b & & A \to b & & A \to b & \end{array}$$

$L(G) = \{b, bab, babab, \ldots\} = \{w \in \{a, b\}^* \mid w = b(ab)^n with n \geq 0\}$.

**Definition 2.42** *Equivalent Grammars ($G_1 \equiv G_2$)*

*Given two grammars $G_1$ and $G_2$ , they are equivalents iff $L(G_1) = L(G_2)$*

**Example 2.43** *Let* $G_1 = (N_1, T_1, P_1, S_1)$ *and* $G_2 = (N_2, T_2, P_2, S_2)$ *be two grammars with:*

$$\begin{array}{lll} N_1 = \{A\} & & N_2 = \{A\} \\ T_1 = \{a, b\} & & T_2 = \{a, b\} \\ P_1 = \{A \to a, A \to Aa\} & P_2 = \{A \to a, A \to aa, A \to Aaa\} & \\ S_1 = A & & S_2 = A \end{array} \quad (2.19)$$

*Show that they are equivalent grammars.*

$L(G_1) = L(G_2) = \{a, aa, aaa, \ldots\} = \{w \in \{a, b\}^* \mid w = a^n \ \ with \ \ n \leq 1\}$, *and thus,* $G_1$ *and* $G_2$ *are equivalent.*

## 2.6 Grammar Classification

The classification of grammars is done as a function of the production rules they include, that is, according to the members of $P$. We will start by defining the different types of production rules.

**Definition 2.44** *Type-0 rule ( Unrestricted )*

*Given a grammar* $G = (N, T, P, S)$ *a rule belonging to* $P$ *is of type* 0 *iff is of the form:*

$\alpha \to \beta$ *with* $(\alpha \in V^+) \wedge (\beta \in V^*)$ .

*Nota: Every production rule is of type-0.*

**Definition 2.45** *Type-1 rule (Context-sensitive)*

*Given a grammar* $G = (N, T, P, S)$ *a rule belonging to* $P$ *is of type-1 iff is of the form:* $A\alpha\beta \to \alpha\gamma\beta$ *with* $(\alpha, \beta V^*) \wedge (\gamma \in V^+) \wedge (A \in N)$ .

**Definition 2.46** *Type-2 rule (Context free)*

*Given a grammar* $G = (N, T, P, S)$, *a rule belonging to* $P$ *is of type-2 iff is of the form:* $A \to \alpha$ *with* $(A \in N) \wedge (\alpha \in V^+)$ .

**Definition 2.47** *Left regular Rule*

*Given a grammar* $G = (N, T, P, S)$ *a rule belonging to* $P$ *is left regular iff is of the form* $A \to aB$ *with* $(A, B \in N) \wedge (a \in T)$ .

**Definition 2.48** *Right regular rule*

*Given a grammar* $G = (N, T, P, S)$ *a rule belonging to* $P$ *is right regular iff is of the form* $A \to Ba$ *with* $(A, B \in N) \wedge (a \in T)$ .

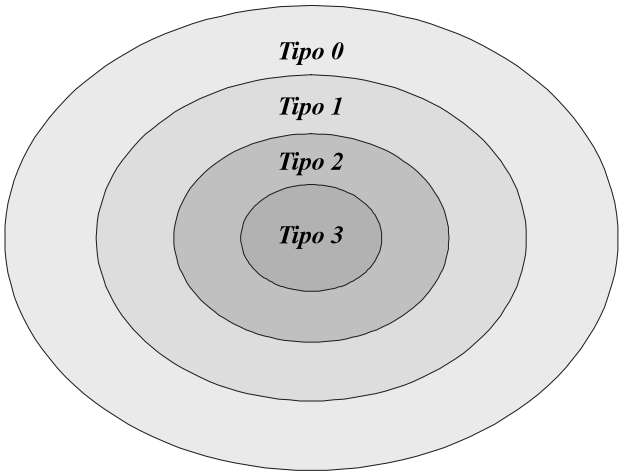


Figure 2.1: Relationship between production rules.

**Definition 2.49** *Terminal regular rule Given a grammar* $G = (N, T, P, S)$ *a rule belonging to* $P$ *is terminal regular iff is of the form* $A \to a$ *with* $(A \in N) \wedge (a \in T)$.

**Definition 2.50** *Type-3 rule (Regular)*
*Given a grammar* $G = (N, T, P, S)$ *a rule belonging to* $P$ *is type-3 iff is right regular or left regular, or terminal regular.*

According to these definitions we can write and display the relationship between the production rules:
$Type - 3 \subset Type - 2 \subset Type - 1 \subset Type - 0$

**Example 2.51** *Given* $G = (N, T, P, S)$ *with* $N = \{S, B\}$ *and* $T = \{c, d\}$ . *Let´s analyze the type of the following rules that are members of* $P$ *:*

$S \to c$ *is a terminal regular and so is type-3.*
$S \to cB4$ *is left regular and so is type-3.*
$B \to cc$ *is type-2 and is not type-3.*
$B \to SS$ *is type-2 and is not type-3.*
$S \to cBSccd$ *is type-2 and is not type-3.*
$BB \to cB$ *is type-1 and is not type-2.*
$cdBSScB \to cdBcBBdScB$ *is type-1 and is not type-2.* .
$BBB \to SSS$ *is type-0 and is not type-1.*
$cB \to Bc$ *is type-0 and is not type-1.*
$c \to d <$ *is type-0 and is not type-1.*
$c \to \epsilon$ *is type-0 and is not type-1.*

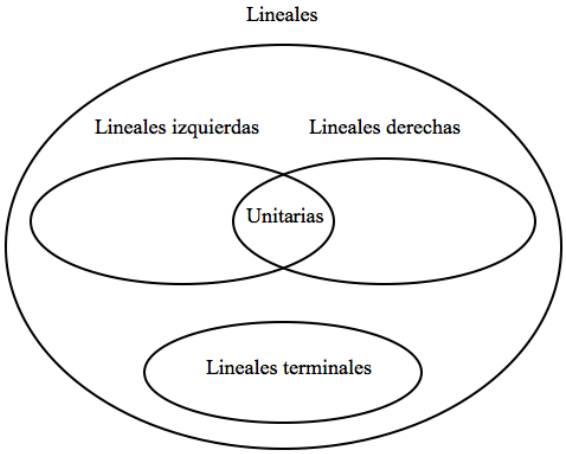


Figure 2.2: Relationship between linear production rules.

**Definition 2.52** *Terminal linear rule*
*Given a grammar* $G = (N, T, P, S)$ *we say that the rule is terminal linear iff is of the form:*
$A \to \alpha$ *with* $(A \in N) \wedge (\alpha \in T^+)$ .

**Definition 2.53** *Linear rule*
*Given a grammar* $G = (N, T, P, S)$ *we say that the rule is linear if is terminal linear or iff is of the form:* $A \to \alpha B \beta$ *with* $(A, B \in N) \wedge (\alpha, \beta \in T^*)$

**Definition 2.54** *Left linear rule*
*Given a grammar* $G = (N, T, P, S)$ *we say that the rule is left linear iff is of the form:* $A \to \alpha B$ *with* $(A, B \in N) \wedge (\alpha \in T^*)$ .

**Definition 2.55** *Right linear rule*
*Given a grammar* $G = (N, T, P, S)$ *we say that the rule is right linear iff is of the form:* $A \to B\alpha$ *with* $(A, B \in N) \wedge (\alpha \in T^*)$ .

**Definition 2.56** *Unitary rule*
*Given a grammar* $G = (N, T, P, S)$ *a rule is a Unitary rule iff is of the form:* $A \to B$ *with* $A, B \in N$.
*Note: A rule can be linear without being left linear or right linear.*

The relationships between linear rules are shown in Fig. 2.2 .

**Example 2.57** *Let* $G$ *be a grammar* $G = (N, T, P, S)$ *with* $N = \{S, B\}$ *and* $T = \{c, d\}$. *Let´s analyze the type of the following rules belonging to* $P$ *:*

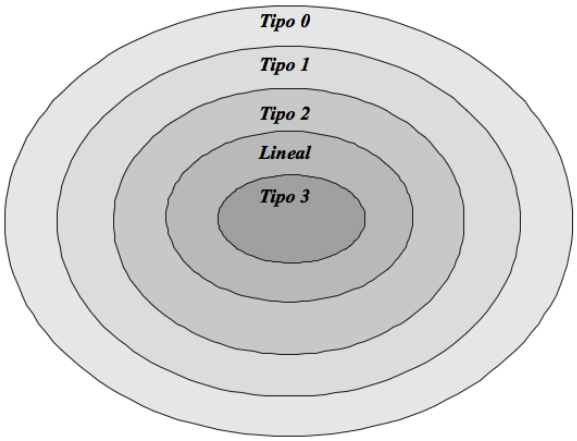


Figure 2.3: Relationship between production rules including the linear ones.

*$S \to B$ is unitary, and linear.*
*$S \to S$ is unitary, and linear.*
*$B \to c$ is terminal linear.*
*$S \to cdd$ is terminal linear.*
*$B \to cS$ is left linear.*
*$B \to dddB$ is left linear.*
*$S \to Sc$ is right linear.*
*$S \to Bcdcc$ is right linear.*
*$B \to cBd$ is linear but not left or right linear.*
*$S \to cdcBc$ is linear but not left or right linear.*

Let´s see the relationship between the rules putting all cases together (a diagram is also shwon in Fig. 2.3):

Type-3 $\subset$ Linear $\subset$ Type-2 $\subset$ Type-1 $\subset$ Type-0.

Let´s see now the classification of grammars according to the types of rules we have just defined.

**Definition 2.58** *Unrestricted Grammar (type-0) (UG)*
*A grammar is unrestricted iff all its rule are of type-0.*
*Note: Every grammar is of type-0.*

**Definition 2.59** *Context Sensitive Grammar (type-1) (CSG)*
*A grammar is context sensitive iff all its rules are of type-1.*

**Definition 2.60** *Context Free Grammar (type-2) (CFG)*
*A grammar is context free iff all its rules are of type-2.*

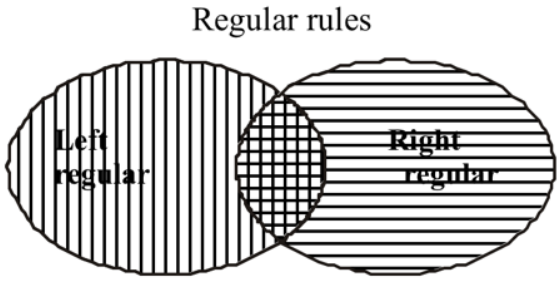


Figure 2.4: Relationship between regular grammars.

Figure 2.5: Chomsky Hierarchy for grammars.

**Definition 2.61** *Regular Grammars (type-3) (RG)*

*A grammar is regular iff is left regular (LRG) or right regular (RRG).*

*A grammar is left regular (LRG) iff all its rules are left regular or terminal regular.*

*A grammar es right regular (RRD) iff all its rules are right regular o terminal regular.*

Note: $G = (\{A, B\}, \{a\}, \{A \to aB, A \to Ba\}, A)$ is not a regular grammar. $G = (\{A, B\}, \{a\}, \{A \to a\}, A)$ is both right regular and left regular.

The Chomsky Hierarchy for grammars establishes the following relationships:

$$\text{Type-3} \subset \text{Type-2} \subset \text{Type-1} \subset \text{Type-0}$$

**Example 2.62** *Let $G$ be a grammar $G = (N, T, P, S)$ with $N = \{S, A, B\}$ and $T = \{a, b, c\}$ :*

*If $P = \{S \to a, S \to aS\}$ then $G$ is left regular , and so is regular (type-3).*

*If $P = \{S \to b, B \to Bc\}$ then $G$ is right regular, and so is regular (type-3).*

*Si $P = \{S \to c, S \to aA, S \to bB, A \to aA, A \to a, B \to bB\}$ then $G$ is left regular, and so is regular (type-3).*

*If $P = \{S \to c, A \to a, B \to b\}$ then $G$ is both right and left regular, and so is regular (type-3).*

*If $P = \{S \to a, S \to SS\}$ then $G$ is context free (type-2) and is not regular (type-3).*

*If $P = \{A \to ccc\}$ then $G$ is context free (type-2) and is not regular (type-3).*

*If $P = \{S \to A, A \to a\}$ then $G$ is context free (type-2) and is not regular (type-3).*

*If $P = \{S \to aaScc, S \to b\}$ then $G$ is context free (type-2) and is not regular (type-3).*

*If $P = \{S \to a, S \to SS, SS \to SSS\}$ then $G$ is context sensitive (type-1) and is not context free (type-2).*

*If $P = \{S \to b, aBAc \to aBAcc\}$ then $G$ is context sensitive (type-1) and is not context free (type-2).*

*If $P = \{A \to aAb, aA \to aaA, aA \to aa, Ab \to Abb, aAb \to aabb\}$ then $G$ is context sensitive (type-1) and is not context free (type-2).*

*If $P = \{S \to a, S \to SS, SS \to S\}$ then $G$ is unrestricted (type-0) and is not context sensitive (type-1).*

*If $P = \{S \to a, a \to b\}$ then $G$ is unrestricted (type-0) and is not context sensitive (type-1).*

*If $P = \{S \to a, a \to aa, a \to b, b \to c\}$ then $G$ is unrestricted (type-0) and is not context sensitive (type-1).*

*If $P = \{S \to a, aB \to Ba, SSS \to BBB\}$ then $G$ is unrestricted (type-0) and is not context sensitive (type-1).*

**Definition 2.63** *Linear Grammar (LG)*

*We say that a grammar is linear iff all its rules are linear.*

**Definition 2.64** *Left Linear Grammar (LLG)*

*We say that a grammar is left linear iff each of its rules are left linear or terminal linear.*

**Definition 2.65** *Right Linear Grammar (RLG)*

*We say that a grammar is right linear iff each of its rules are right linear or terminal linear.*

Note: A grammar can be linear without being Left Linear or Right Linear.

Taking into account the linear grammars, we have that:

type-3 $\subset$ Linear $\subset$ type-2 $\subset$ type-1 $\subset$ type-0

A graphic representation of the relationship between types of grammars including linear and regular ones is shown in Fig. XX

**Example 2.66** *Let $G$ be a grammar, $G = (N, T, P, S)$, with $N = \{S, A, B\}$ and $T = \{a, b, c\}$ :*

*If $P = \{S \to aaa, S \to bbb\}$ then $G$ is left linear and right linear, and so is linear.*

*If $P = \{S \to a, S \to aaS\}$ then $G$ is left linear and thus linear but not right linear.*

*If $P = \{S \to a, S \to aS, S \to aaS\}$ then $G$ is left linear and thus linear but not right linear..*

*If $P = \{S \to a, S \to Saaaa, S \to Saa\}$ then $G$ is right linear and thus linear, but not left linear.*

*If $P = \{S \to a, S \to Sa\}$ then $G$ is right linear and thus linear, but not left linear.*

*If $P = \{S \to bbc, S \to Bbbc, B \to bbcS\}$ then $G$ is linear, but is not left linear nor right linear.*

*If $P = \{S \to aab, S \to aaSb\}$ then $G$ is linear, but is not left linear nor right linear.*

**Definition 2.67** *Epsilon rule*

*Given a grammar $G = (N, T, P, S)$, we say that a rule that belongs to $P$ is an epsilon rule iff is of the form: $A \to \epsilon$ , with $A \in N$.*

**Definition 2.68** *$\epsilon$ Regular Grammar ($\epsilon$-RG)*

*We say that a grammar is $\epsilon$-regular iff is $\epsilon$ left regular or $\epsilon$ right regular.*

**Definition 2.69** *$\epsilon$ Left Regular Grammar ($\epsilon$-RG)*

*We say that a grammar is $\epsilon$ left regular ($G\epsilon RI$) iff each of its rule are left regular or terminal regular or epsilon rule.*

**Definition 2.70** *$\epsilon$ Right Regular Grammar ($G\epsilon RR$)*

*A grammar is $\epsilon$ right regular ($G\epsilon RR$) iff each of its rules is rght regular, terminal regular or epsilon rule.*

**Definition 2.71** *$\epsilon$ Context free grammar ($\epsilon$-CFG) A grammar is an $\epsilon$-Context Free grammar iff each of its rules is context free or epsilon rule.*

**Example 2.72** *Given a grammar $G = (N, T, P, S)$ with $N = \{S, A, B\}$ and $T = \{a, b, c\}$ :*

*If $P = \{S \to a, S \to aS, A \to \epsilon\}$ then $G$ is $\epsilon$ left regular and so is a regular grammar.*

*If $P = \{S \to b, B \to Bc, B \to \epsilon\}$ then $G$ is $\epsilon$ right regular and so is a regular grammar.*

*If $P = \{S \to c, S \to aA, S \to bB, A \to aA, A \to a, B \to bB, S \to \epsilon\}$ then $G$ is $\epsilon$ left regular , and so is a regular grammar.*

*If $P = \{S \to c, A \to a, B \to b, S \to \epsilon\}$ then $G$ is $\epsilon$ left regular and so is a regular grammar.*

*If* $P = \{S \to a, S \to SS, S \to \epsilon\}$ *then* $G$ *is* $\epsilon$ *context free and thus is not regular.*

*If* $P = \{A \to cccA \to \epsilon\}$ *then* $G$ *is* $\epsilon$ *context free and thus is not regular.*

*If* $P = \{S \to A, A \to a, B \to \epsilon\}$ *then* $G$ $\epsilon$ *context free and thus is not regular.*

*If* $P = \{S \to aaScc, S \to b, S \to \epsilon\}$ *then* $G$ $\epsilon$ *context free and thus is not regular.*

*If* $P = \{S \to a, S \to aS, B \to Bc, A \to \epsilon\}$ *then* $G$ $\epsilon$ *context free and thus is not regular.*

**Proposition 2.73** *Given an* $\epsilon$ *right regular grammar* $G$*, then exists* $G'$ *left regular grammar such that* $L(G') = L(G) - \{\epsilon\}$ .

**Proposition 2.74** *Given a left regular grammar* $G$*, then exists* $G'$ $\epsilon$ *left regular grammar such that* $L(G') = L(G) - \{\epsilon\}$ .

**Proposition 2.75** *Given an* $\epsilon$ *context free grammar, then exists a context free grammar grammar* $G'$ *such that* $L(G') = L(G) - \{\epsilon\}$ .

**Proposition 2.76** *Given* $G$ *a context free grammar, then exists* $G'$ $\epsilon-$*context free grammar such that* $L(G') = L(G) \cup \{\epsilon\}$ .

## 2.7 Notation

Even if we have already define the concept of a grammar, we specify next the complete notation system that will be used in relationship to grammars:

$T$ Terminal alphabet.
Notation for elements of $T$:
- Letters of the beginning of the alphabet (subindexes permitted): $a, b, c, \ldots$
- Operators: $+, -, *, /$
- Special characters: $@, (, ), [, ], \&, \#$
- Digits: $0, 1, \ldots, 9$
- Words underlined: <u>if</u>, <u>then</u>, ...

$N$: Non-terminal alphabet.
Notation of elements from N:
- Capital letter from the beginnign of the alphabet (subindexes allowed): $A, B, C, \ldots$

- Words in underlined capital letters: DIGITS, EXPRESSION, ...
- Words between angular brackets: ⟨ digits ⟩, ⟨ EXPRESSION ⟩, ...

It should be verified that $N \cup T = \emptyset$.
$V = N \cup T$.

- The elements of $V$ are denoted by capital letters from the final of the alphabet $(\ldots, X, Y, Z)$.
- The elements of $T^*$ are denoted from letters of the final of the alphabet $(t, u, v, \ldots)$.
- The elements of $V^*$ are denoted with greek letters from the beginning of the alphabet (subindexes accepted), with the exception of $\epsilon, \alpha, \beta, \ldots)$. ($\epsilon$ indicates the empty string).
- $S$ indicates the Axioma, with $S \in N$ (subindexes permitted).
- The vertical bar symbol "| " is used to abreviate the notation of rules, in a way such that:

$$A \to \alpha_1 \mid \alpha_2 \mid \ \ldots \ \mid \alpha_n \equiv \begin{cases} A \to \alpha_1 \\ A \to \alpha_2 \\ \ldots \\ A \to \alpha_n \end{cases} \tag{2.20}$$

## 2.8 Language Classification

**Definition 2.77** *Type-i Language (L.i)*
*A language $L$ is of type $i$, with $i \in \{0, 1, 2, 3\}$, if exists a type-i grammar $G$ such that $L(G) = L - \{\epsilon\}$ .*

Note: Languages are also denoted according to the name of the corresponding grammar type.

The Chomsky hierarchy for languages establishes the following relationships:
$L.3 \subset L.2 \subset L.1 \subset L.0$

**Definition 2.78** *Linear Languages (Linear.L)*
*We say that a language $L$ is linear if there is a linear grammar $G$ such that $L(G) = L - \{\epsilon\}$.*

Note: If $G$ is a right linear grammar or a left linear grammar then $L(G)$ is a regular language.

Taking into account linear languages together with those associated to the Chomsky hierarchy, we have that:

$L.3 \subset L.linear \subset L.2 \subset L.1 \subset L.0$

As it can be deduced from the previous definitions, a given language can be generated by grammars of different types (see Examples 2.62, 2.66 and 2.68).

**Example 2.79** *Let be* $\Sigma = \{a, b, c\}$.

$$L = \{w \in \Sigma* \mid w = a^n \text{ with } n \geq 0\} \in L.3.$$

$$L = \{w \in \Sigma* \mid w = a^n b^n \text{ with } n \geq 0\} \in (L.linear - L.3).$$

$$L = \{w \in \Sigma* \mid w = a^n b^n c a^m b^m \text{ with } n \geq 0, m \geq 0\} \in (L.2 - L.linear).$$

$$L = \{w \in \Sigma* \mid w = a^n b^n c^n \text{ with } n \geq 0\} \in (L.1 - L.2).$$

$$L = \{w \in \Sigma* \mid w \notin L(g(f(w)))\} \in (L.0 - L.1),$$

*where:* $f : \Sigma* \to N$
$g : N \to \{G \text{ context sensitive} \mid T_G = \Sigma\}$
*are bijections (that can be computed) .*

## 2.9 Basic questions about Languages

Given two grammars $G_1$ and $G_2$ of the same type, we can ask the following questions:

Equivalence: $G_1 \equiv G_2$ ?
Inclussion: $L(G_1) \subset L(G_2)$ ?
Belonging: $x \in L(G_1)$ ?
Emptiness: $L(G_1) = \emptyset$ ?
Finiteness: $||L(G_1)|| \in N$ ?
Regularity: $L(G_1) \in L.3$ ?

Table 2.1 shows whether exists a conclusive algorithm to answer those questions according to grammar type

Table 2.1: Existence of conclusive algorithm to answer relevant questions about grammars according to their types

| | Unrestricted | Context sensitive | Context free | Linear | Regular |
|---|---|---|---|---|---|
| Equivalence | No | No | No | No | Yes |
| Inclusion | No | No | No | No | Yes |
| pertenence | No | Yes | Yes | Yes | Yes |
| Emptiness | No | No | Yes | Yes | Yes |
| Finiteness | No | No | No | No | Trivial |

## 2.10 Operations over languages

As languages are sets of strings the operations define over sets can be applied to languages, and then we suppose known the operations of union, intersection, difference and complement ($(\bar{L} = \Sigma^* - L)$). Besides these operations, we define others that are specific for languages that are all based on the concatenation of languages, that in turn is based in the concatenation of strings.

**Definition 2.80** *Concatenation of languages*

*If $L_1, L_2$ are languages over $\Sigma$, we define the language $L_1$ concatenated with $L_2$, denoted $L_1 \cdot L_2$ or simply $L_1L_2$, as: $L_1L_2 = \{w \in \Sigma^* \mid w = xy, \text{ with } x \in L_1 \text{ and } y \in L_2\}$.*

Note:
$L\emptyset = \emptyset L = \emptyset \ \ \forall L \subseteq \Sigma^*$
$L\{\epsilon\} = \epsilon L = L \ \ \forall L \subseteq \Sigma^*$
$L_1(L_2L_3) = (L_1L_2)L_3 \ \ \forall L_1, L_2, L_3 \subseteq \Sigma^*$
$L_1(L_2 \cup L_3) = L_1L_2 \cup L_1L_3 \ \ \forall L_1, L_2, L_3 \subseteq \Sigma^*$
$(L_1 \cup L_2)L_3 = L_1L_3 \cup L_2L_3 \ \ \forall L_1, L_2, L_3 \subseteq \Sigma^*$

**Example 2.81** *Consider $\Sigma = \{0, 1\}$ and $L_1, L_2 \subseteq \Sigma^*$ :*
*a) If $L_1 = \{010, 00\}$ and $L_2 = \{111, 10\}$ then*
$L_1L_2 = \{010111, 01010, 00111, 0010\}$
$L_2L_1 = \{111010, 11100, 10010, 1000\}$

*b) If $L_1 = \{w \in \Sigma^* \mid w = 01^n \text{ with } n \leq 0\}$ and*
*$L_2 = \{w \in \Sigma^* \mid |w|_0 = 2n + 1 \text{ with } n \leq 0\}$ then*
$L_1L_2 = \{w \in \Sigma^* \mid w(1) = 0 \wedge |w|_0 = 2n \text{ with } n \leq 1\}$
$L_2L_1 = \{w \in \Sigma^* \mid |w|_0 = 2n \text{ with } n \leq 1\}$

**Definition 2.82** *Power of a language* $(L^n)$

*Given a language $L$ over $\Sigma$, we define the n-th power of $L$, denoted $L^n$, as:*

$$L^n = \begin{cases} \epsilon & \text{if } n = 0 \\ LL^{n-1} & \text{if } n > 0 \end{cases} \tag{2.21}$$

**Example 2.83** *Given $\Sigma = \{a, b\}$ and $L \in \Sigma^*$ with $L = \{aa, b\}$*

$L^3 = LL^2 = LLL^1 = LLLL^0 = LLL\epsilon = LLL = \{aa, b\}\{aa, b\}\{aa, b\} = \{aaaaaa, aaaab, aabaa, aabb, baaaa, baab, bbaa, bbb\}$

**Definition 2.84** *Closure or Kleene star*

*Given $L \in \Sigma^*$ we define the closure or Kleene star of language $L$, denoted $L^*$, as:*

$L^* = L^{\cdot e}$, *where $L^{\cdot e}$ is the extended closure for the set $L$ for the concatenation of strings in the monoid $(\Sigma^*, \cdot)$ .*

Note: $L* = \{w \in \Sigma^* \mid w = w_1 w_2 \dots w_k \text{ with } w_1, w_2, \dots, w_k \in L \text{ and } k \leq 0\}$ . $L* = \bigcup_{n \leq 0} L^n$

According to the definition we have: $\epsilon \in L^* \forall L \subseteq \Sigma^*$ .

**Example 2.85** $\emptyset^* = \{\epsilon\}$

$\{\epsilon\}* = \{\epsilon\}$

*If $L = \{01, 1, 100\}$ we have that:*

$110001110011 \in L^*$

$100001 \notin L^*$

$\epsilon \in L^*$

*If $L = \{\epsilon, a\}$ we have that:*

$L^* = \{\epsilon, a, aa, aaa, \dots\}$

*Given $\Sigma = \{a, b\}$ and $L\Sigma^*$ with $L = \{bbb\}$*

$L^* = \{w \in a, b^* \mid w = (bbb)^n \text{ with } n \leq 0\}$

**Definition 2.86** *Strict closure*

*Given $L \in \Sigma^*$ we define the strict closure of language $L$, denoted $L^+$, as:*

$L^+ = L$ ,

*where $L^+$ is the strict closure for the set $L$ for the operation of concatenation of strings.*

Note: $L^+ = \{w \in \Sigma^* \mid w = w_1 w_2 \dots w_k \text{ with } w_1, w_2, \dots, w_k \in L \text{ and } k \leq 1\}$.

$L^+ = \cup L^n$

$L^+ = LL^*$

$If \epsilon \; in L$ then $L^+ = L^*$.

$If \epsilon \in L$ then $L^+ = L^* - \epsilon$.

**Example 2.87** $\epsilon^+ = \emptyset$
$\{\epsilon\}^+ = \{\epsilon\}$
*If* $L = \{01, 1, 100\}$ *we have that:*
$110001110011 \in L^+$
$100001 \in L^+$
$\epsilon \in L+$
*IF* $L = \{\epsilon, a\}$ *we have that:*
$L^+ = \{\epsilon, a, aa, aaa, \ldots\}$
*Given* $\Sigma = \{a, b\}$ *and* $L \in \Sigma^*$ *with* $L = \{bbb\}$
$L^+ = \{w \in a, b^* \in | \in w = (bbb)^n \text{ with } n \leq 1\}$

**Definition 2.88** *Inverse of a language*
*Given* $L \in \Sigma^*$*, we define the inverse of* $L$*, denoted* $L^R$*, as* $L^R = \{x^R \mid x \in L\}$ .

**Example 2.89** *Given* $\Sigma = \{a, b\}$ *and* $L \in \Sigma^*$ *with* $L = \{w \in \{a, b\}^* \mid w = a^n b^n \text{ with } n \leq 0\}$
$L^R = \{w \in \{a, b\}^* \mid w = b^n a^n \text{ with } n \leq 0\}$

## 2.11 Closure for the different types of languages

The following table shows for every type of language whether it is closed for everyone of the operations bettwen languages.

The fact that $L.0$ is not closed for the complement operation, together with the types of languages previously defined, determined the foolowing relationships:

$L.finite \subset L.3 \subset L.lineal \subset L.2 \subset L.1 \subset L.0 \subset L.REP \subset 2^{\Sigma^*}$

Table 2.2: Table.

| | Unrestricted | Context sensitive | Context free | Linear | Regular |
|---|---|---|---|---|---|
| Union | Yes | Yes | Yes | Yes | Yes |
| Intersection | Yes | Yes | No | No | Yes |
| Intersection with L.3 | Yes | Yes | Yes | Yes | Yes |
| Complement | No | Yes | No | No | Yes |
| Concatenation | Yes | Yes | Yes | No | Yes |
| Power | Yes | Yes | Yes | No | Yes |
| Closure | Yes | Yes | Yes | No | Yes |
| Strict closure | Yes | Yes | Yes | No | Yes |
| Inverse | Yes | Yes | Yes | Yes | Yes |