# Foundations of Computer Science
# Logic, models, and computations

## Chapter: Recurrence and induction

Course CSC_41012_EP

of l'Ecole Polytechnique

Olivier Bournez

bournez@lix.polytechnique.fr

Version of August 20, 2024

# Recurrence and induction

## 1 Motivation

The *recursive definitions* are ubiquitous in computer science. There are present both in programming languages, but also in many concepts what we consider in computer science.

**Example 1 (Lists in JAVA)** *In JAVA, with*

```
class List {
  int content;
  List next;
  }
List lst;
```

*the class List is defined in a recursive (*inductive*) manner: by using in the definition of the class, the field "next' whose type is the class List itself.*

**Example 2 (Ranked trees)** *We have defined the ranked trees in the previous chapter by using the notion of graph. A natural alternative would be to describe the ranked trees through a recursive definition: A ranked tree is either empty, or reduced to a vertex (a root), or made of a vertex (a root) and a (ranked) list of ranked trees (its sons).*

In this chapter, we spent some time on the *recursive definitions* of sets and functions. This will be used to give some clean meaning to recursive definitions in next chapters.

We will furthermore define in this chapter how it is possible to do some proofs on structures defined inductively, by introducing the *proofs by (structural) induction*.

## 2 Reasoning by recurrence over $\mathbb{N}$

The *structural induction* is a generalization of the *proof by recurrence*: Let's come back first to this later to have clear ideas.

When reasoning on the integers, the *first principle of induction*, also called *principle of mathematical recurrence* is a reasoning mode particular useful.

**Theorem 1** *Let $P(n)$ be* predicate *(a property) depending on the integer $n$. If the the two following conditions are satisfied:*

($B$) *$P(0)$ is true;*

($I$) *$P(n)$ implies $P(n+1)$ for all $n$;*

*then for all integer $n$, $P(n)$ is true.*

**Proof**: The reasoning is done by contradiction. Consider

$$X = \{k \in \mathbb{N} | P(k) \text{ is false}\}.$$

If $X$ is non-empty, it admits some least element $n$. From condition, ($B$), $n \neq 0$, and hence $n-1$ is some integer, and $P(n-1)$ is true by definition of $X$. We then get a contradiction with the property ($I$) applied for integer $n-1$. □

To do a *proof by recurrence*, we prove a property for 0 (basis case), and we prove that the property is *hereditary*, or *inductive*: $P(n)$ implies $P(n+1)$ for all $n$.

The concept of *inductive proof* generalises this idea to other sets than the integers, namely to sets that are defined inductively.

**Exercise 1** *Consider $S_n = 1^3 + 3^3 + \cdots + (2n-1)^3$. Prove by recurrence that $S_n = 2n^4 - n^2$.*

**Exercise 2** *Prove by recurrence that $\sum_{k=1}^{n} \frac{1}{4k^2-1} = \frac{n}{2n+1}$.*

**Exercise 3** *(solution on page 203) The theorem above is sometimes called the "first principle of induction". Prove the "second principle of induction": Let $P(n)$ be a property depending on integer $n$. If the following property is satisfied: For all $n \in \mathbb{N}$, if assuming that for all $k < n$ the property $P(k)$, one can deduce $P(n)$, then for all $n \in \mathbb{N}$, the property $P(n)$ is true.*

**Exercise 4** *(solution on page 204) An alphabet $\Sigma$ is fixed. Recall that a language over $\Sigma$ is a subset of $\Sigma^*$. If $L_1$ and $L_2$ are two languages of $\Sigma^*$, their concatenation is defined by $L_1.L_2 = \{u.v | u \in L_1, v \in L_2\}$. The concatenation is an associative operation that admits $\{\epsilon\}$ as neutral element. One can then define the powers of a language $L$ in the following way: $L^0 = \{\epsilon\}$, and for all integer $n > 0$, $L^{n+1} = L^n.L = L.L^n$. The star of a language $L$ is defined by $L^* = \bigcup_{n\in\mathbb{N}} L^n$.*

*Let $L$ and $M$ two languages over $\Sigma$, with $\epsilon \notin L$. Prove that in $\mathscr{P}(\Sigma^*)$ (the languages over alphabet $\Sigma$), the equation $X = L.X \cup M$ admits for unique solution the language $L^*.M$.*

# 3 Inductive definitions

The inductive definitions aims at defining some subsets of a set $E$.

**Remark 1** *This remark is for purists. It can be avoided in a first reading of this document.*

*We restrict in this document to the framework where one wants to define by induction some objects that correspond to subsets to an already known set $E$. We avoid this to avoid the subtleties and the paradoxes of the* set theory.

*The very attentive reader will observe that we will often consider the syntactic writing of some objects more than the objects themselves. Indeed, by doing so, we guarantee that we are living in a set $E = \Sigma^*$ for a certain alphabet $\Sigma$, and we avoid to have to worry about the existence of the set $E$ in the following reasoning's.*

*For example, to formalise completely the Example 1 above, we would try to define some syntactic representation of lists instead of lists.*

When one wants to define a set, or a subset of a set, one way to do it is by giving some *explicit definition*, that is by describing precisely which are its elements.

**Example 3** *The even integers can be defined by $P = \{n | \exists k \in \mathbb{N}\ n = 2 * k\}$.*

Unfortunately, this is not always as easy. It is often easier to define a set by some *inductive definition*. A typical example of *inductive definition* is a definition like this one:

**Example 4** *The even integers also correspond to the least set that contains $0$ and such that if $n$ is even, then $n+2$ is even.*

**Remark 2** *Observe that the set of integers $\mathbb{N}$ satisfies that $0$ is some integer, and that if $n$ is some integer, then so is $n+2$. It is hence necessary to say that this is*

*the least set with this property.*

## 3.1 General principle of an *inductive definition*

Intuitively, a subset $X$ is defined inductively if it can be defined from some explicitly given elements of $X$, and a mean to construct some new elements of $X$ from elements of $X$.

More generally, in an inductive definition,

- some elements of the set $X$ are explicitly given (that is to say, a set $B$ of elements $b$ of $X$). They correspond to the *base set* of the inductive definition;
- The other elements of the set $X$ are defined, as a function of elements that already belong to the set $X$, according to some rules: that is to say we are given a set of rules $R$ for the formation of new elements. This constitutes the *inductive steps* of the inductive definition.

One considers then the least set that contains $B$ and that is *stable* (one says also *closed*) by the rules of $R$.

## 3.2 Formalisation: First fix point theorem

Formally, all of this is justified by the following theorem.

**Definition 1 (Inductive definition)** *Let $E$ be a set. An* inductive definition *of a subset $X$ of $E$ consists of:*

- *a non-empty subset $B$ of $E$ (called the* base set*)*
- *and a set of* rules *$R$: each rule $r_i \in R$ is a function (possibly partial) $r_i$ from $E^{n_i} \to E$, for some integer $n_i \geq 1$.*

**Theorem 2 (Fix point theorem)** *To a* inductive definition *corresponds a least set that satisfies the following properties:*

($B$) *it contains $B$: $B \subset X$;*

($I$) *it is stable by the rules rules of $R$: for every rule $r_i \in R$, for every $x_1, \cdots, x_{n_i} \in X$, we have $r_i(x_1, \cdots, x_{n_i}) \in X$.*

*One says that this set is* inductively defined.

**Proof**: Let $\mathscr{F}$ be the set of subsets of $E$ satisfying ($B$) and ($I$). The set $\mathscr{F}$ is non empty as it contains at least one element: Indeed, the set $E$ satisfies the conditions ($B$) and ($I$) and hence $E \in \mathscr{F}$.

We can then consider $X$ defined as the intersection of all the elements of $\mathscr{F}$. Formally:

$$X = \bigcap_{Y \in \mathscr{F}} Y. \tag{1}$$

Since $B$ is included in each $Y \in \mathscr{F}$, $B$ is included in $X$. So $X$ satisfies the condition $(B)$.

The obtained set satisfies also $(I)$. Indeed, consider a rule $r_i \in R$, and some $x_1, \cdots, x_{n_i} \in X$. We have $x_1, \cdots, x_{n_i} \in Y$ for every $Y \in \mathscr{F}$. For every such $Y$, since $Y$ is stable by the rule $r_i$, we must hat $r(x_1, \cdots, x_{n_i}) \in Y$. Since this is true for every $Y \in \mathscr{F}$, we also have $r(x_1, \cdots, x_{n_i}) \in X$, which proves that $X$ is stable by the rule $r_i$.

$X$ is the least set that satisfies the conditions $(B)$ and $(I)$, since it is by definition included in every other set that satisfies the conditions $(B)$ and $(I)$.

□

## 3.3 Various notations of an inductive definition

**Notation 1** *We often denote some* inductive definition *using the notation*

$(B)$ $x \in X$

*with a line like this one for every* $x \in B$

*(possibly one writes* $B \subset X$ *);*

$(I)$ $x_1, \cdots x_{n_i} \in X \Rightarrow r_i(x_1, \cdots, x_{n_i}) \in X$

*with such a rule for every rule* $r_i \in R$.

**Example 5** *According to this convention, the inductive definition of even integers (of the Example 4) is denoted by:*

$(B)$ $0 \in P$;

$(I)$ $n \in P \Rightarrow n+2 \in P$.

**Example 6** *Let* $\Sigma = \{(,)\}$ *be the alphabet made of the open parenthesis and of the closing parenthesis. The set* $D \subset \Sigma^*$ *of well founded parenthesising, called the Dyck language, is defined inductively by*

$(B)$ $\epsilon \in D$;

$(I)$ $x \in D \Rightarrow (x) \in D$;

$(I)$ $x, y \in D \Rightarrow xy \in D$.

**Notation 2** *One sometimes prefers to write some inductive definition as* deduction rules*:*

$$\frac{}{B \subset X} \qquad \frac{x_1 \in X \quad \ldots \quad x_{n_i} \in X}{r_i(x_1, \ldots, x_{n_i}) \in X}$$

*The principle of such a notation is that an horizontal line — means some* deduction rule. *What is written above the line is some* hypothesis. *What is written under the line is some conclusion. If what is above is empty this means that the conclusion is true without any hypothesis.*

**Notation 3** *We sometimes write also directly:*

$$\overline{b} \qquad \frac{x_1 \quad \ldots \quad x_{n_i}}{r_i(x_1,\ldots,\ x_{n_i})}$$

*for every* $b \in B$,
*or*

$$\overline{b \in B} \qquad \frac{x_1 \quad \ldots \quad x_{n_i}}{r_i(x_1,\ldots,\ x_{n_i})}$$

*or even:*

$$\overline{B} \qquad \frac{x_1 \quad \ldots \quad x_{n_i}}{r_i(x_1,\ldots,\ x_{n_i})}$$

# 4 Applications

## 4.1 A few examples

**Example 7** ($\mathbb{N}$) *The subset* $X$ *of* $\mathbb{N}$ *defined inductively by*

$$\overline{0} \qquad \frac{n}{n+1}$$

*is nothing but the whole set* $\mathbb{N}$ *of the integers.*

**Example 8** ($\Sigma^*$) *The subset* $X$ *of* $\Sigma^*$, *where* $\Sigma$ *is an alphabet, defined inductively by*

($B$) $\epsilon \in X$;

($I$) $w \in X \Rightarrow wa \in X$, *for every* $a \in \Sigma$;

*is nothing but the whole set* $\Sigma^*$.

**Example 9 (Language** $\{a^n bc^n\}$**)** *The language* $L$ *on the alphabet* $\Sigma = \{a, b, c\}$ *of words of the form* $a^n bc^n$, $n \in \mathbb{N}$, *is defined inductively by*

($B$) $b \in L$;

($I$) $w \in L \Rightarrow awc \in L$.

**Exercise 5** *(solution on page 204) Define inductively the set of well parenthesised expressions formed from identifiers taken in a set* $A$ *and using operators* $+$ *and* $\times$.

## 4.2 Labeled binary trees

Let's recall here the text of the course INF421 (version 2010-2011): "the notion of *binary tree* is rather different from the notion of *free tree* and *ranked tree*. A *binary tree* on a finite set of vertices is either empty, or the disjoint union of a vertex, called its root„ of a binary tree, called its *left sub-tree*, and of a binary tree, called its *right sub-tree*. It is useful to represent such a binary tree on the form of a triplet $A = (A_g, r, A_d)$."

We obtain immediately an *inductive definition* of *labeled binary trees* from this text.

**Example 10 (Labeled binary trees)** *The set $AB$ of labeled binary trees on the set $A$ is the subset of $\Sigma^*$, where $\Sigma$ is the alphabet $\Sigma = A \cup \{\emptyset, (, ), ,, \}$, defined inductively by*

($B$) $\emptyset \in AB$;

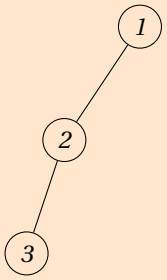
($I$) $g, d \in AB \Rightarrow (g, a, d) \in AB$, *for every* $a \in A$.

**Remark 3** *In the expression above, $(g, a, d)$ denotes the concatenation of the word of length 1 (, of word $g$, of word , of length 1, of word $a$, of word , of length 1, of word $d$ and of mot ) of length 1. All these words are indeed words over the alphabet $\Sigma$ that contains all the required symbols.*

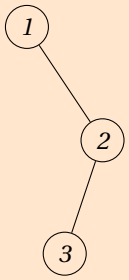
**Remark 4** *$g, d \in AB \Rightarrow (g, a, d) \in AB$, for every $a \in A$ denotes the fact that one repeats, for every $a \in A$, the rule $g, d \in AB \Rightarrow (g, a, d) \in AB$. This is actually not a rule, but a family of rules: one for each element $a$ of $A$.*

**Remark 5** *Be careful: a* binary tree *is not a* ranked tree *such that all the nodes are of arity at most* 2.

**Example 11** *For example, the* labeled binary tree




*and the* labeled binary tree




*are not the same, since the first corresponds to the word* $(((\emptyset,3,\emptyset),2,\emptyset),1,\emptyset)$ *and the second to the word* $(\emptyset,1,((\emptyset,3,\emptyset),2,\emptyset))$. *However, if these trees are considered as ranked trees, they are the same.*

**Exercise 6** *(solution on page 204) Let $A$ be an alphabet. One defines recursively the sequence of sets $(AB_n)_{n\in\mathbb{N}}$ by*

- $AB_0 = \{\emptyset\}$.
- $AB_{n+1} = AB_n \cup \{(a,g,d) | a \in A, g,d \in AB_n\}$

*Prove that $X = \bigcup_{n\in\mathbb{N}} AB_n$ corresponds also to the set $AB$ of labeled binary trees for the set $A$.*

## 4.3 Arithmetic expressions

One can define the well formed arithmetic expression on the alphabet $\Sigma_{exp}$ of example 1.1. Recall that we have defined the alphabet

$$\Sigma_{exp} = \{0,1,2,\cdots,9,+,-,*,/,(,)\}.$$

Let's start by defining what is a number in radix 10. Usually, one doesn't write an integer in radix 10 by starting by a 0 (except for 0). For example, 000192 is not authorized. By opposition, 192 is a valid expression.

We obtain the following inductive definition.

**Example 12** *The set $\mathcal{N}$ of non-null integers written in radix* 10 *is the subset of $\Sigma_{exp}{}^*$, defined inductively by*

(*B*) $a \in \mathcal{N}$ *for each* $a \in \{1,2,\ldots,9\}$;

(*I*) $g \in \mathcal{N} \Rightarrow ga \in \mathcal{N}$, *for each* $a \in \{0,1,2,\ldots,9\}$.

One can then defined the arithmetic expression in the following way:

**Example 13** *The set $Arith$ of* arithmetic!expressions *is the subset of $\Sigma_{exp}{}^*$, defined inductively by*

(*B*) $0 \in Arith$;

(*B*) $\mathcal{N} \subset Arith$;

(*I*) $g, d \in Arith \Rightarrow g + d \in Arith$;

(*I*) $g, d \in Arith \Rightarrow g * d \in Arith$;

(*I*) $g, d \in Arith \Rightarrow g / d \in Arith$;

(*I*) $g, d \in Arith \Rightarrow g - d \in Arith$;

(*I*) $g \in Arith \Rightarrow (g) \in Arith$;

For example, we have $(1 + 2 * 4 + 4 * (3 + 2)) \in Arith$ that corresponds to some valid expression. By opposition, $+1 - /2($ is not in $Arith$.

## 4.4 Terms

Les *terms* are particular labeled ranked trees. They play an essential role in many structures in computer science.

Let $F = \{f_0, f_1, \cdots, f_n, \cdots\}$ be a set of symbols, called *function symbols*. To each such symbol $f$ is associated some integer $a(f) \in F$, that is called its *arity*, and this represents the number of arguments of function symbol $f$. one writes $F_i$ for the subset of symbols of functions of arity $i$. The function symbols of arity 0 are called *constants*.

Let $\Sigma$ the alphabet $\Sigma = F \cup \{(, ), ,\}$ constituted of $F$, of the opening parenthesis, the closing parenthesis, and of comma.

**Definition 2 (Terms over $F$)** *The set $T$ of terms built over $F$ is the subset of $\Sigma^*$ defined inductively by:*

(*B*) $F_0 \subset T$
*(that is to say: the constants are some terms)*

(*I*) $t_1, t_2, \cdots, t_n \in T \Rightarrow f(t_1, t_2, \cdots, t_n) \in T$
*for every integer n, for every symbol $f \in F_n$ of arity n.*

**Remark 6** *In the definition above, we are indeed talking about words over the alphabet $\Sigma$: $f(t_1, t_2, \cdots, t_n)$ denotes the word whose first letter is $f$, the second (, the following the ones of $t_1$, etc.*

**Example 14** *For example, we can fix $F = \{0, 1, f, g\}$, with 0 and 1 of arity 0 (these are constants), $f$ of arity 2 and $g$ of arity 1. Then $f(0, g(1))$ is a term over $F$. $f(g(g(0)), f(1,0))$ is a term over $F$. $f(1)$ is not a term over $F$.*

The terms over $F$ correspond to particular ranked labeled trees: The nodes are labeled by symbols of functions from $F$, and a node labeled by a symbol of arity $k$ has exactly $k$ sons.

# 5 Proofs by induction

We will need regularly to prove some properties on a the elements of a set $X$ defined inductively. This turns out to be possible by using what is called a *proof!by (structural) induction*, sometimes also called *proof!by (structural) induction*, which generalises the principle of the proof by recurrence.

**Theorem 3 (Proof by induction)** *Let $X \subset E$ be a set defined inductively from a base set $B$ and some rules $R$. Let $\mathscr{P}$ be a* predicate *expressing some property of an element $x \in E$: That is to say a property $\mathscr{P}(x)$ that is either true or false in a given element $x \in E$.*

*If the following conditions are satisfied:*

- $(B)$ *$\mathscr{P}(x)$ is satisfied for every element $x \in B$;*
- $(I)$ *$\mathscr{P}$ is* hereditary*, that is to say stable by the rules of $R$: Formally, for every rue $r_i \in R$, for every $x_1, \cdots, x_{n_i} \in E$, $\mathscr{P}(x_1), \cdots, \mathscr{P}(x_{n_i})$ true implies $\mathscr{P}(x)$ true in $x = r_i(x_1, \cdots, x_{n_i})$.*

*Then $\mathscr{P}(x)$ is true for every element $x \in X$.*

**Proof**: Consider the set $Y$ of elements $x \in E$ that satisfy the property $\mathscr{P}(x)$. $Y$ contains $B$ by the property $(B)$. $Y$ is stable by the rules of $R$ by the property $(I)$. The set $X$, that corresponds to the least set that contains $B$ and that is stable by the rules of $R$, is hence included in $Y$. □

**Remark 7** *The proof by induction indeed generalises the proof by recurrence; Indeed, $\mathbb{N}$ is defined inductively as in Example 7. A proof by induction on this inductive definition of $\mathbb{N}$ corresponds to a proof by recurrence, that is to say to hypotheses of Theorem 1.*

**Example 15** *To prove by induction that all the words of the language defined inductively in Example 9 have as many a's as c's, it is sufficient to observe that this is true for the word reduced to a b: Indeed, this is 0 times the letter a and the letter c; and that if this holds for the word $w$, then the word $awc$ has as many times the letter a than the letter c, namely exactly one more than in word $w$.*

**Exercise 7** *(solution on page 204) We consider the subset $ABS$ of strict binary trees defined as the subset of language $AB$ (of labeled by $A$ binary trees) defined inductively by:*

($B$) $(\emptyset, a, \emptyset) \in ABS$, *for every* $a \in A$.

($I$) $g, d \in ABS \Rightarrow (g, a, d) \in ABS$, *for every* $a \in A$.

*Prove that*

- *an element of $ABS$ is always non-empty and without a vertex with only one non-empty son.*
- *that in a strict binary tree, the number of vertices $n$ satisfies $n = 2f - 1$, where $f$ is the number of leaves.*

**Exercise 8** *Prove that any word of the Dyck language has as many open parenthesis than closing parentheses.*

**Exercise 9** *Prove that any arithmetic expression, that is to say any word of language $Arith$, has as many open parenthesis than closing parentheses.*

**Exercise 10** *A binary tree is said to be* balanced *if for every vertex of the tree, the difference between the height of its right subtree and the height of its left subtree value either $-1$, $0$ or $1$ (i.e. at most one in absolute value).*

- *Provide an inductive definition of the set $AVL$ of balance binary trees.*
- *Define the sequence $(u_n)_{n\in\mathbb{N}}$ by $u_0 = 0$, $u_1 = 1$, and for all $n \geq 2$, $u_{n+2} = u_{n+1} + u_n + 1$.*

  *Prove that for every $x \in AVL$, $n \geq u_{h+1}$ where $h$ and $n$ are respectively the height and the number of vertices of a tree.*

# 6 Derivations

## 6.1 Explicit expression of the elements: Second fix point theorem

We have seen up to now several examples of sets $X$ defined inductively. The existence of each set $X$ follows from Theorem 2, and actually from the Equation (1) used in its proof.

This is a *bottom-up definition* of $X$, as Equation (1) defines $X$ from super-sets of $X$. This has the clear advantage to show easily the existence of sets defined inductively, a fact that we used abundantly up to now.

However, this has the default that it does not say what the elements of obtained sets $X$ exactly are.

It is actually also possible to define each set $X$ defined inductively from a *bottom-up definition*. One obtains then an explicit definition of the elements of $X$, with in addition a way to describe them explicitly.

This is what states the following result:

**Theorem 4 (Explicit definition of a set defined inductively)** *Every set $X$ defined inductively from the base set $B$ and from the rules $R$ can also be written*

$$X = \bigcup_{n\in\mathbb{N}} X_n,$$

*where $(X_n)_{n\in\mathbb{N}}$ is the family of subsets of $E$ defined by recurrence by*

- $X_0 = B$
- $X_{n+1} = X_n \cup \{r_i(x_1,\cdots,x_{n_i}) | x_1,\cdots,x_{n_i} \in X_n \text{ and } r_i \in R\}$.

In other words, every element of $X$ is obtained by starting from elements of $B$ and by applying a finite number of times the rules of $R$ to obtain new elements.

**Proof**: It is sufficient to prove that this set is the least set that contains $B$ and that is stable by the rules of $R$.

First since $X_0 = B$, $B$ is indeed in the union of the $X_n$. Second, if one takes some rule $r_i \in R$, and some elements $x_1,\cdots,x_{n_i}$ in the union of the $X_n$, by definition each $x_j$ is in $X_{k_j}$ for some integer $k_j$. Since the sets $X_i$ are increasing (i.e. $X_i \subset X_{i+k}$ for all $k$, which can be proved easily by recurrence over $k$), all the $x_1,\cdots,x_{n_i}$ are in $X_{n_0}$ for $n_0 = \max(k_1,\cdots,k_{n_i})$. We obtain immediately that $r(x_1,\cdots,x_{n_i})$ is in $X_{n_0+1}$, which proves that it in the union of the $X_n$.

Finally, this is the least set, since every set that contains $B$ and that is stable by the rules of $R$ must contain each of the $X_n$. This is proved by recurrence over $n$. This is true at the rank $n = 0$, as such a set must contain $X_0$ since it contains $B$. Suppose the hypothesis at rank $n$, that is to say $X$ contains $X_n$. Since the elements of $X_{n+1}$ are obtained from elements of $X_n \subset X$ by applying some rule $r_i \in R$, $X$ contains each of these elements. □

## 6.2 Derivation trees

The *bottom-up* definition of $X$ from the previous theorem invite to attempt to keep the trace on how each element is obtained, starting from $X$ and by applying the rules of $R$.

**Example 16** *The word* $1+2+3$ *corresponds to some arithmetic expression. Here is a proof.*

$$\dfrac{\dfrac{1 \in \mathcal{N} \qquad 2 \in \mathcal{N}}{1+2 \in Arith} \qquad 3 \in Arith}{1+2+3 \in Arith}$$

*This is not the only one possible. Indeed, we can also write:*

$$\dfrac{1 \in Arith \qquad \dfrac{2 \in \mathcal{N} \qquad 3 \in \mathcal{N}}{2+3 \in Arith}}{1+2+3 \in Arith}$$

To encode each trace, the notion of term, on a set $F$ of well-chosen symbols appears naturally: One considers that each element $b$ of the base set $B$ is symbol of arity 0. To each rule $r_i \in R$ is associated some symbol of arity $n_i$. A term $t$ on this set of symbols is called a *derivation.* derivation

To each derivation $t$ is associated some element $h(t)$ as expected: To $t$ of arity 0, is associated the corresponding element $b$ of $B$. Otherwise $t$ is of the form $r_i(t_1, \cdots, t_{n_i})$, for some rule $r_i \in R$ end for some terms $t_1, \cdots, t_{n_i}$. To such a $t$ is associated the result of the application of the rule $r_i$ to elements $h(t_1), \cdots, h(t_{n_i})$.

**Example 17** *For arithmetic expressions denote by the symbol + of arity* 2 *the rule* $g, d \in Arith \Rightarrow g+d \in Arith$*;*

*The first proof of Example 16 corresponds to derivation* $+(+(1,2),3)$*. The second to derivation* $+(1,+(2,3))$*. The image by the function h of these derivations is the word* $1+2+3$*.*

We can then reformulate the previous theorem in the following way.

**Proposition 1** *Let $X$ be a set defined inductively from the base set $B$ and from the rules of $R$. Let $D$ be the set of the derivations corresponding to $B$ and to $R$. Then*

$$X = \{h(t) | t \in D\}.$$

In other words, $X$ is precisely the set of the elements of $E$ that have a derivation.

We see in the previous example, that an element of $X$ may have several derivations.

**Definition 3** *An inductive definition of $X$ is said* non ambiguous *if the previous function $h$ is injective.*

Intuitively, this means that there exists a unique way to build every element of $X$.

**Example 18** *The following definition of* $\mathbb{N}^2$ *is ambiguous:*

$(B)$ $(0,0) \in \mathbb{N}^2$;

$(I)$ $(n,m) \in \mathbb{N}^2 \Rightarrow (n+1,m) \in \mathbb{N}^2$;

$(I)$ $(n,m) \in \mathbb{N}^2 \Rightarrow (n,m+1) \in \mathbb{N}^2$.

*Indeed, one can obtain for example* $(1,1)$ *by starting from* $(0,0)$ *and by applying the second rule, and then the third, but also by applying the third rule, and then the second.*

**Example 19** *The definition of* $Arith$ *of example 13 is ambiguous since* $1+2+3$ *has several derivations.*

**Example 20** *This problem is intrinsic to arithmetic expressions, since when we write* $1+2+3$*, we do not precise if we are talking about the result of the addition of* $1$ *to* $2+3$ *or of* $3$ *to* $1+2$*, the idea being that since addition is associative, this is not important.*

**Example 21** *To avoid this potential problem, let's define the set* $Arith'$ *of the parenthesised arithmetic expressions as the subset of* $\Sigma_{exp}{}^*$*, defined inductively by*

$(B)$ $0 \in Arith$;

$(B)$ $\mathcal{N} \subset Arith'$;

$(I)$ $g,d \in Arith' \Rightarrow (g+d) \in Arith'$;

$(I)$ $g,d \in Arith' \Rightarrow (g*d) \in Arith'$;

$(I)$ $g,d \in Arith' \Rightarrow (g/d) \in Arith'$;

$(I)$ $g,d \in Arith' \Rightarrow (g-d) \in Arith'$;

$(I)$ $g \in Arith' \Rightarrow (g) \in Arith'$;

*This times,* $1+2+3$ *is not a word of* $Arith'$*. By opposition* $(1+(2+3)) \in Arith'$ *and* $((1+2)+3) \in Arith'$.

*The interest of this writing is that we now have some* non-ambiguous *rules.*

# 7 Functions defined inductively

We will need sometimes to defined functions on some sets $X$ defined inductively. This can be done easily when $X$ admits some *non-ambiguous definition.*

**Theorem 5 (Inductively defined function)** *Let* $X \subset E$ *be a set defined inductively in a non-ambiguous way from a the base set* $B$ *and from rules* $R$*. Let* $Y$ *be a set.*

*For an application* $f$ *from* $X$ *to* $Y$ *is perfectly defined, it suffices that the following are given:*

($B$) *the value of $f(x)$ for each of the elements $x \in B$;*

($I$) *for reach rule $r_i \in R$, the value of $f(x)$ for $x = r_i(x_1, \cdots, x_{n_i})$ as a function of the values $x_1, \ldots, x_{n_i}$, $f(x_1), \ldots,$ and $f(x_{n_i})$.*

In other words, informally, if one knows how to "program recursively", that is to say "describe in a recursive way the function" then the function is perfectly defined on the inductive set $X$.

**Proof**: The statement above means that there exists a unique application $f$ from $X$ to $Y$ that satisfies these constraints. It suffices to prove that for every $x \in X$, the value of $f$ in $x$ is defined in a unique way. This is proved easily by induction: This is true for the elements $x \in B$. If this is true in $x_1, \cdots, x_{n_i}$, this is true in $x = r_i(x_1, \cdots, x_{n_i})$: The definition of $X$ being non-ambiguous, $x$ can obtained only by the rule $r_i$ from $x_1, \cdots, x_{n_i}$. Its value is hence perfectly defined by the constraint for the rule $r_i$. □

**Example 22** *The factorial function $Fact$ from $\mathbb{N}$ into $\mathbb{N}$ is defined inductively by*

($B$) $Fact(0) = 1$;

($I$) $Fact(n+1) = (n+1) * Fact(n)$.

**Example 23** *The height $h$ of a labeled binary tree is defined inductively by*

($B$) $h(\emptyset) = 0$;

($I$) $h((g,a,d)) = 1 + \max(h(g), h(d))$.

**Example 24** *The value $v$ of an arithmetic expression of $Arith'$ is defined inductively by ($v$ is a function that goes from words to the rational numbers)*

($B$) $v(0) = 0$;

($B$) $v(x) = h(x)$ *pour* $x \in \mathcal{N}$;

($I$) $v((g+d)) = v(g) + v(d)$;

($I$) $v((g*d)) = v(g) * v(d)$;

($I$) $v((g/d)) = v(g)/v(d)$, *if* $v(d) \neq 0$;

($I$) $v((g-d)) = v(g) - v(d)$;

($I$) $v((g)) = v(g)$;

*where $h$ is the function that, to a word of $\mathcal{N}$ maps its value as a rational number: $h$ is defined inductively by*

($B$) $h(a) = a$ *for each* $a \in \{1,2,\ldots,9\}$;

$(I)$ $h(ga) = 10 * h(g) + a$ *for each* $a \in \{0,1,2,\ldots,9\}$.

We observe that all these definitions are essentially only the translation on how their can be programmed in some recursive way. The use of non-ambiguous definitions avoids any ambiguity on the evaluation.

**Remark 8** *For the arithmetic expressions,* $1+2*3 \in Arith$ *is also ambiguous. A computer program that would take as input a word of* $Arith$ *and supposed to return the value of the expression would have to manage the priorities, and understand that* $1+2*3$ *is not the result of the multiplication of* $1+2$ *by* $3$*. By using the definition of* $Arith'$*, we avoid completely this difficulty, since the expressions are encoding explicitly how their must be evaluated, and an inductive definition becomes possible. At first sight, the value of an expression of* $Arith$ *cannot be defined simply inductively, if only because the value of* $x+y*z$ *is not obtained directly from the one of* $x+y$ *and from* $z$ *only.*

# 8 Bibliographic notes

**Suggested readings** To go further on the notions of this chapter, we suggest to read [Arnold & Guessarian, 2005]. For a more general presentation of inductive definitions, and of fix point theorems, we suggest to read [Dowek, 2008].

**Bibliography** This chapter has been written by using [Dowek, 2008] as well as [Arnold & Guessarian, 2005].

# Index

# Bibliography

[Arnold & Guessarian, 2005] Arnold, A. & Guessarian, I. (2005). *Mathématiques pour l'informatique.* Ediscience International.

[Dowek, 2008] Dowek, G. (2008). *Les démonstrations et les algorithmes.* Polycopié du cours de l'Ecole Polytechnique.