CHAPTER 3

RECURSION AND INDUCTION

Inductive and recursive definitions are the construction of finite objects from other finite objects, according to some given rules. Inductive definitions also provide us with a way of grasping infinite objects defined by recursive definitions: indeed, since only finite objects can be handled by computer science, such infinite objects are studied via sequences of finite approximations; usually, the finite approximations are also defined by an inductive definition.

Inductive proofs enable one to reason about inductively defined objects. Because computer science makes extensive use of such objects, this chapter is essential. For instance, recursive definitions constantly occur in data structures and in the conception of recursive programs (in functional languages such as LISP, but also in logic programming and PROLOG). The proofs of such recursive programs are then inductive proofs, as are the proofs of termination of iterative programs (sometimes called top–down induction).

However, the various induction principles are not stated in detail in textbooks (to our knowledge); this is why we cannot recommend any handbook for the present chapter.

In this chapter we review the two basic induction principles on the integers: the induction principle and the complete induction principle. We introduce the notion of definition of a set by induction and we show how to prove properties of sets defined by induction. As a special case, we introduce the concept of 'set of terms' which is a major tool in computer science. Finally, we present the concept of closure, which is a general way of looking at inductive definitions.

3.1 Reasoning by induction in $\mathbb{N}$

3.1.1 First induction principle

In $\mathbb{N}$, the first induction principle, also called the mathematical induction principle, is a most useful way of reasoning. We will use both terminologies 'proof by induction' and 'proof by mathematical induction' for proofs using this first induction principle.

Theorem 3.1 *Let $P(n)$ be a predicate (a property) depending on the integer n. If both the following conditions hold:*

(B) *$P(0)$ is true, and*
(I) *$\forall n \in \mathbb{N}$, the implication $\big(P(n) \Longrightarrow P(n+1)\big)$ is true,*

then $\forall n \in \mathbb{N}$, $P(n)$ is true.

(B) is called the *basis step* of the induction and (I) is called the *inductive step* (or sometimes 'going from n to $n+1$'). Here we give a direct proof of this result, but it is worth while noting that it can also be justified by using Proposition 3.11 and the inductive definition of $\mathbb{N}$ given in Example 3.9.

Proof. By contradiction. We consider the set

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

If X is non-empty, it has a least element n. By condition (B), $n \neq 0$. Thus $n-1$ is an integer and $n-1 \notin X$, namely, $P(n-1)$ is true. Using (I), we then obtain: $P(n)$ is true, which contradicts $n \in X$. Therefore, X is empty, and this proves the theorem. □

Z (I) does not assert that $P(n+1)$ or $P(n)$ hold, but only that **if** $P(n)$ is true, **then** $P(n+1)$ must be true. Only after proving (I) **and** (B) can we conclude that, for all $n \geq 0, P(n)$ is true. Usually the basis (B) is easy to prove, and the difficult part is the inductive step (I). However, one should not forget to prove the basis (B), otherwise one will obtain false results; for instance, we verify immediately that $\forall n \geq 0$, $(n > 10 \Longrightarrow n+1 > 10)$. It is none the less false that $\forall n \geq 0$, $n > 10$. (See also Exercise 3.6.)

Remark 3.2 We can prove a slightly more general form of Theorem 3.1 similarly. Let n_0 be an integer greater than or equal to 0, if both following conditions hold:

(B_{n_0}) $P(n_0)$ is true, and
(I_{n_0}) $\forall n \geq n_0$, the implication $\big(P(n) \Longrightarrow P(n+1)\big)$ is true,

then $\forall n \geq n_0$, $P(n)$ is true.

EXAMPLE 3.3 We wish to compute the sum $S_n = 1 + 2 + \cdots + n$. We note that $2S_1 = 2 = 1 \times 2$, $2S_2 = 2 + 4 = 2 \times 3$, $2S_3 = 2 + 4 + 6 = 3 \times 4$. We then conjecture that $\forall n > 0$, $2S_n = n(n+1)$. We prove this by induction. Let $P(n)$ be the property '$2S_n = n(n+1)$', we verify that

(B) $2S_1 = 1 \times 2$,
(I) Let $n \geq 1$. We assume $P(n)$. We have

$$2S_{n+1} = 2S_n + 2(n+1) = n(n+1) + 2(n+1) = (n+1)(n+2),$$

hence $P(n+1)$ is true.

We can then conclude that $\forall n \geq 1$, $P(n)$.

EXERCISE 3.1 Adopting the convention that $\forall r \in \mathbb{R}, r^0 = 1$, prove by induction that:

1. $\forall r \in \mathbb{R}, \forall n \in \mathbb{N}, \quad S_n = \displaystyle\sum_{i=0}^{n} r^i = \begin{cases} n+1 & \text{if } r = 1, \\ \dfrac{r^{n+1}-1}{r-1} & \text{if } r \neq 1. \end{cases}$
2. $\forall r \in \mathbb{R}, \forall n \in \mathbb{N}, \quad T_n = \displaystyle\sum_{i=0}^{n} ir^i = \begin{cases} n(n+1)/2 & \text{if } r = 1, \\ \dfrac{nr^{n+2} - (n+1)r^{n+1} + r}{(r-1)^2} & \text{if } r \neq 1. \end{cases}$ ♢

EXERCISE 3.2
1. Show that $\forall n \geq 1$, $S_n = 1^3 + 3^3 + \cdots + (2n-1)^3 = 2n^4 - n^2$.
2. Compute $T_n = \displaystyle\sum_{k=1}^{n} \frac{1}{4k^2 - 1}$ for all $n \geq 1$. ♢

EXERCISE 3.3 We consider the polynomial with real-valued coefficients

$$P(x) = \frac{1}{3}x^3 + ax^2 + bx.$$

1. Find a and b such that $\forall x \in \mathbb{R}$, $P(x+1) - P(x) = x^2$. We assume that this property holds in the remainder of the exercise.
2. Show that $\forall n \in \mathbb{N}$, $P(n)$ is an integer.
3. $\forall n \geq 0$, let $S_n = \displaystyle\sum_{k=0}^{n} k^2$. Show that

$$\forall n \geq 0, \quad S_n = P(n+1) = \frac{n(n+1)(2n+1)}{6}.$$ ♢

NOTATION We will write $p \mid n$ to denote the fact that p divides n, where p and n are integers.

EXERCISE 3.4 Let $n \geq 1$ and let $A \subseteq \{1, 2, \ldots, 2n\}$ be such that $|A| \geq n+1$. Show that there exist two distinct integers a and b in A such that $a \mid b$. ♢

EXERCISE 3.5 Let $\mathcal{R}$ be a binary relation on a set E. Let

$$\mathcal{R}^0 = Id_E \ , \ \mathcal{R}^{i+1} = \mathcal{R}.\mathcal{R}^i.$$

Show that $\forall i, j \geq 0$, $\mathcal{R}^{i+j} = \mathcal{R}^i.\mathcal{R}^j$. ♢

EXERCISE 3.6 We consider the properties $P(n)$: '$9 \mid 10^n - 1$' and $Q(n)$: '$9 \mid 10^n + 1$'.

1. Show that $\forall n \in \mathbb{N}$, $P(n) \Longrightarrow P(n+1)$ and $Q(n) \Longrightarrow Q(n+1)$.
2. Find the values of n for which $P(n)$ (resp. $Q(n)$) is true. ♢

EXERCISE 3.7 Find the error in the following proof by induction. Let $P(n)$ be the property 'in any group consisting of n individuals, all the people are of the same age'.

(B) $P(1)$ is clearly true.
(I) Let n be such that $P(n)$ is true. Let G be a group of $n+1$ individuals numbered from 1 to $n+1$. Let G_1 (resp. G_2) be the group consisting of the n first (resp. last) individuals in G. Since $P(n)$ is true, all the people of G_1 (resp. G_2) are of the same age. Moreover, individual number n is a member of both G_1 and G_2. Thus all the people of G are of the same age as individual number n, and this proves $P(n+1)$.

We hence deduce that $\forall n \geq 1$, $P(n)$. ♢

3.1.2 Second induction principle

In the first induction principle (see Theorem 3.1), the truth of $P(n+1)$ depends only upon that of $P(n)$, i.e. if proposition P is true at step n it is also true at step $(n+1)$. More complex cases may occur, where in order to establish that P is true at step $(n+1)$ we have to explicitly use the fact that P is true at steps $0, 1, \ldots, n-1, n$. In such a case, it is more convenient to use the second induction principle, which is stated as follows.

Theorem 3.4 *Let $P(n)$ be a property depending on the integer n. If the following proposition is verified:*

$$(\mathrm{I}') \qquad \forall n \in \mathbb{N}, \quad \Big(\big(\forall k < n, P(k)\big) \quad \Longrightarrow \quad P(n) \Big),$$

then $\forall n \in \mathbb{N}, P(n)$ is true.

This second induction principle is a consequence of Theorem 2.31 because the usual ordering on $\mathbb{N}$ is a well ordering (see Section 2.4).

REMARK 3.5
1. The fact that the second induction principle has no basis step may seem suspicious; in fact, the basis step is 'hidden' in (I′). Indeed, verifying (I′) implies proving that for $n = 0$ $\big(\forall k < 0, P(k)\big) \Longrightarrow P(0)$. But $\big(\forall k < 0,\ P(k)\big)$ is true because there is no negative integer $k < 0$, hence we must prove that $P(0)$ is true. Here we see a typical instance of reasoning with the empty set: $\big(\forall k < 0, P(k)\big)$ can be rewritten as $\big(k < 0 \Longrightarrow P(k)\big)$, or, since there is no negative integer $k < 0$ in $\mathbb{N}$, $\big(k \in \emptyset \Longrightarrow P(k)\big)$, which is true because $k \in \emptyset$ is always false; more generally, any 'empty' statement of the form $\big(\forall x \in \emptyset, P(x)\big)$ always holds.

2. As for the first induction principle, we may start from any integer n_0. We must then check that:

$$(\mathrm{I}'_{n_0}) \qquad \forall n \geq n_0, \quad \Big(\big(\forall k \in \{n_0, \ldots, n-1\}, P(k)\big) \quad \Longrightarrow \quad P(n)\Big)$$

and deduce $\forall n \geq n_0$, $P(n)$.
3. On $\mathbb{N}$, the two induction principles are equivalent (i.e. each can be shown to be valid from the other), but (see Section 2.4) only the second induction principle can be generalized to more general ordered sets.

Exercise 3.8 Verify that the two induction principles entail the same properties on $\mathbb{N}$, i.e. that, if $P(n)$ is a property depending on the integer n, P verifies (I′) if and only if P verifies (B) and (I). The two induction principles thus have the same power on $\mathbb{N}$; we will say that they are equivalent on $\mathbb{N}$. ♢

Example 3.6 The second principle is simpler to use when the property of the elements at step n involves simultaneously the property of the elements at steps $n-1, n-2, \ldots$, etc. For instance, we can quite easily show that any integer $n \geq 2$ can be written as a product of primes. Denote by $P(n)$ the property 'n can be written as a product of primes'; it suffices to verify (I'_2), see Remark 3.5 (2). Let $n \geq 2$. Assume $\forall k \in \{2, \ldots, n-1\}$, $P(k)$. Two cases can occur:
- n is a prime. Then n can clearly be written as a product of primes (a single prime is also considered as a product).
- n is not a prime. Then we can write $n = ab$, where a and b are two integers between 2 and $n-1$. $P(a)$ and $P(b)$ are true by hypothesis, and so we deduce that n can also be written as the product of the decompositions of a and b.

Exercise 3.9
1. Show that $\forall n \in \mathbb{N}$, $(n+1)^2 - (n+2)^2 - (n+3)^2 + (n+4)^2 = 4$.
2. Deduce that any integer m can be written as sums and differences of squares $1^2, 2^2, \ldots, n^2$ for an n, i.e.

$$\forall m \in \mathbb{N}, \exists n \in \mathbb{N}, \exists \varepsilon_1, \ldots, \varepsilon_n \in \{-1, 1\}, \ m = \varepsilon_1 1^2 + \varepsilon_2 2^2 + \cdots + \varepsilon_n n^2.$$

(Hint: first show the result for $m \in \{0, 1, 2, 3\}$.) ♢

Exercise 3.10 Let A^* be the free monoid on the alphabet A (see Definition 1.15). Show that $\forall u, v \in A^*$, $u \cdot v = v \cdot u \Longleftrightarrow \exists w \in A^*$, $\exists p, q \in \mathbb{N}$: $u = w^p$ and $v = w^q$. ♢

Exercise 3.11 Let A^* be the free monoid on the alphabet A (see Definition 1.15 and Example 2.4). A *language* is a subset of A^*. If L_1 and L_2 are two languages of A^*, we define their *concatenation* by:

$$L_1 \cdot L_2 = \{u \cdot v \ / \ u \in L_1 \, , \, v \in L_2\} \, .$$

Language concatenation is an associative operation with unit $\{\varepsilon\}$. We can then define the powers of language L as follows:

$$L^0 = \{\varepsilon\} \text{ and } \forall n \in \mathbb{N}, L^{n+1} = L^n \cdot L = L \cdot L^n.$$

Finally, the star of language L is the submonoid of A^* generated by L, i.e.

$$L^* = \bigcup_{n \in \mathbb{N}} L^n.$$

Let L and M be two languages on A^* such that $\varepsilon \notin L$. Show that in $\mathcal{P}(A^*)$, the equation $X = L \cdot X \cup M$ has as its unique solution the language $L^* \cdot M$. ♢

3.2 Inductive definitions and proofs by structural induction

In the present section, we introduce inductive definitions of sets and functions and proofs by induction on inductively defined structures.

3.2.1 Inductively defined sets

Quite often in computer science subsets are inductively (recursively) defined. In particular, many data structures may be so defined. Intuitively, the inductive definition of a subset X of a set explicitly gives some elements of the set X together with ways of constructing new elements of X from already known elements. Such a definition will hence have the following intuitive generic form:

(B) Some elements of the set X are explicitly given (*basis* of the recursive definition).
(I) The other elements of the set X are defined in terms of elements already in the set X (*inductive steps* of the recursive definition).

Formally, we have the following definition.

Definition 3.7 *Let E be a set. An inductive definition of a subset X of E consists of giving:*

- *a subset B of E and*
- *a set K of operations $\Phi\colon E^{a(\Phi)} \longrightarrow E$, where $a(\Phi) \in \mathbb{N}$ is the arity (or rank) of Φ.*

X is defined as the least set verifying the following assertions (B) *and* (I)*:*

(B) $B \subseteq X$.
(I) $\forall \Phi \in K, \forall x_1, \ldots, x_{a(\Phi)} \in X, \Phi(x_1, \ldots, x_{a(\Phi)}) \in X$.

The set thus defined is

$$X = \bigcap_{Y \in \mathcal{F}} Y \ ,$$

where $\mathcal{F} = \{Y \subseteq E \ / \ B \subseteq Y$, *and* Y *verifies* (I) *with* X *replaced by* $Y\}$.

Henceforth, we modify assertions (B) *and* (I) *slightly, and we denote an inductive definition by the form*

(B) $x \in X \quad (\forall x \in B)$,
(I) $x_1, \ldots, x_{a(\Phi)} \in X \Longrightarrow \Phi(x_1, \ldots, x_{a(\Phi)}) \in X \quad (\forall \Phi \in K)$.

Remark 3.8 The set $\mathcal{F}$ is non-empty because it contains E; indeed, E clearly verifies (B) ($B \subseteq E$) and (I). Moreover, if subsets of a set verify a condition then their intersection also verifies that condition. Indeed, let $\mathcal{Y}$ be a set of subsets Y of E verifying (B) and (I), and let $Z = \bigcap_{Y \in \mathcal{Y}} Y$. Since B is included in any set Y of $\mathcal{Y}$, B is also included in $Z = \bigcap_{Y \in \mathcal{Y}} Y$ and hence Z verifies (B); if $x_1, \ldots, x_{a(\Phi)} \in Z$, then for any $Y \in \mathcal{Y}$, $x_1, \ldots, x_{a(\Phi)} \in Y$, whence $\Phi(x_1, \ldots, x_{a(\Phi)}) \in Y$, and hence $\Phi(x_1, \ldots, x_{a(\Phi)}) \in Z$ and Z verifies (I). Thus $\bigcap_{Y \in \mathcal{F}} Y$, where $\mathcal{F}$ is the above-defined set of subsets of E, is indeed the least subset of E verifying the conditions (B) and (I).

Note that, in general, many sets verify these conditions. Consider, for instance, the conditions:

(B) $0 \in P$,
(I) $n \in P \Longrightarrow n + 2 \in P$.

There are infinitely many subsets of $\mathbb{N}$ verifying these properties : $\mathbb{N}$, $\mathbb{N} \setminus \{1\}$, $\mathbb{N} \setminus \{1, 3\}$, $\mathbb{N} \setminus \{1, 3, 5\}$, etc., are such subsets. The subset P defined by (B) and (I) is not among them because it consists of the set of even integers.

We consider now examples of inductive definitions.

Example 3.9

1. The subset X of $\mathbb{N}$ inductively defined by
   (B) $0 \in X$,
   (I) $n \in X \Longrightarrow n + 1 \in X$,
   is identical to $\mathbb{N}$. (B) and (I) thus constitute an inductive definition of $\mathbb{N}$.
2. The subset X of the free monoid A^* (see Definition 1.15) inductively defined by
   (B) $\varepsilon \in X$,
   (I) $u \in X \Longrightarrow \forall a \in A,\ u \cdot a \in X$,
   is identical to A^*. (B) and (I) thus constitute an inductive definition of A^*.

3. Let $A = \{(,)\}$ be the alphabet consisting of two parentheses (left and right). The set $D \subseteq A^*$ of strings of balanced parentheses, the so-called Dyck language, is defined by

(B) $\varepsilon \in D$,

(I) if x and y belong to D, then (x) and xy also belong to D.

4. Let E be the set of expressions all of whose subexpressions are included in parentheses and which are formed from identifiers in a set A and the two operators $+$ and $\times$. E is the subset of $\Big(A \cup \{+, \times, (\ ,\ )\}\Big)^*$ inductively defined by

(B) $A \subseteq E$,

(I) if e and f are in E then $(e + f)$ and $(e \times f)$ are also in E.

We note that in computer science syntactic definitions are almost always inductive. We often use the BNF (Backus–Naur Form) notation for describing them. For instance, the set E is defined by

$$E ::= A \mid (E + E) \mid (E \times E)\ ,$$

where the symbol '|' is read 'or'.

5. The set BT of labelled binary trees on the alphabet A is the subset of $\Big(A \cup \{\emptyset\,,\ (\ ,\ )\ ,\ ,\}\Big)^*$ inductively defined by

(B) $\emptyset \in BT$ (the empty tree),

(I) $l, r \in BT \Longrightarrow \forall a \in A,\ (a, l, r) \in BT$ (the tree with root a, left child l and right child r).

The set BT thus defined is a language on the alphabet $A \cup \{\emptyset\} \cup \{(\} \cup \{)\} \cup \{,\}$. In general, we use a very intuitive graphical representation of trees. To simplify, tree $(a, \emptyset, \emptyset)$ will simply be denoted by a. For instance, the trees a, (a, a, b) , $\big(a, \emptyset, (b, c, \emptyset)\big)$ and $\big(a, (a, b, c), d\big)$ can be drawn as in Figure 3.1

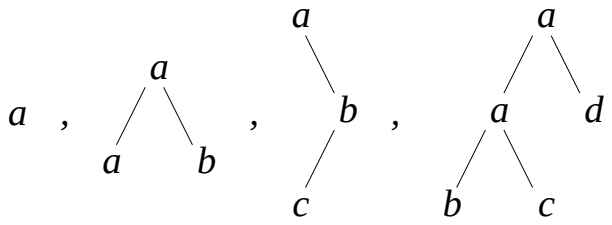

Figure 3.1

Binary trees are extensively used in algorithmics.

Exercise 3.12 Let A be an alphabet. We recursively define the sets $(BT_n)_{n\in\mathbb{N}}$ by

- $BT_0 = \{\emptyset\}$,
- $BT_{n+1} = BT_n \cup \{(a, l, r) \,/\, a \in A, l, r \in BT_n\}$.

Show that $X = \bigcup_{n\in\mathbb{N}} BT_n$ is the set BT of binary trees on alphabet A. ♢

The preceding exercise illustrates a more general phenomenon. Indeed, in most cases, the elements of an inductively defined set can be obtained from the basis by applying finitely many inductive steps. We have the following theorem.

Theorem 3.10 *If X is defined by the conditions (B) and (I), any element of X can be obtained from the basis by applying finitely many inductive steps.*

Proof. We define the sets

- $X_0 = B$,
- $X_{n+1} = X_n \cup \{\Phi(x_1, \ldots, x_{a(\Phi)}) \,/\, x_1, \ldots, x_{a(\Phi)} \in X_n \text{ and } \Phi \in K\}$.

As in Exercise 3.12, we show by induction that $\forall n \in \mathbb{N}$, $X_n \subseteq X$, and we deduce that $X_\omega = \bigcup_{n\in\mathbb{N}} X_n \subseteq X$. The set of elements obtainable from the basis by applying finitely many inductive steps is exactly X_ω. We must now show that X_ω verifies (B) and (I). As $B = X_0 \subseteq X_\omega$, X_ω verifies (B). Let $\Phi \in K$ and let $x_1, \ldots, x_{a(\Phi)} \in X_\omega$. Each x_i belongs to a set $X_{n_i} \subseteq X_\omega$. Let $n = \sup\{n_1, \ldots, n_{a(\Phi)}\}$. Then $x_i \in X_n$, thus $\Phi(x_1, \ldots, x_{a(\Phi)}) \in X_{n+1} \subseteq X_\omega$, and X_ω verifies (I). □

3.2.2 Inductive prooofs

The induction principle is a generalization of the induction principle on the integers and is designed to prove the properties of inductively defined sets. The proof by induction exactly follows the inductive definition of the set; this is why it is also called *proof by structural induction*.

Proposition 3.11 *Let X be an inductively defined set (see Definition 3.7), and let $P(x)$ be a predicate expressing a property of the elements x of X. If the following conditions hold:*

(B″) *$P(x)$ is true for each $x \in B$, and*
(I″) *$\big(P(x_1), \ldots, P(x_{a(\Phi)})\big) \Longrightarrow P\big(\Phi(x_1, \ldots, x_{a(\Phi)})\big)$ for each $\Phi \in K$,*

then $P(x)$ is true for any x in X.

Verifying (B″) *and* (I″) *constitutes a proof by induction of property P on X.*

Proof. Let Y be the set of xs such that $P(x)$ is true. We have that $B \subseteq Y$ (by (B″)), and that Y verifies the inductive clauses (I) of the definition of X (by (I″)); hence $Y \supseteq X$ (see Definition 3.7). □

Remark 3.12 If we consider that the non-negative integers are defined as in Example 3.9, the first induction principle on the integers corresponds to the above definition. All the proofs by mathematical induction seen in Section 3.1 are hence examples of proofs by induction according to the present definition.

EXAMPLE 3.13 We show by induction that any string of the Dyck language has as many left parentheses as right parentheses (see Example 3.9). For x in D, we denote by $l(x)$ (resp. $r(x)$) the number of left (resp. right) parentheses in x. (The inductive definition of these functions is left to the reader.) Finally, let $P(x)$ be the property '$r(x) = l(x)$'. We prove by induction that $P(x)$ holds for any x in D.

(B) The only element of the basis is ε, and it satisfies P because

$$r(\varepsilon) = l(\varepsilon) = 0.$$

(I) Let $x, y \in D$ be such that $r(x) = l(x)$ and $r(y) = l(y)$ and let $z = xy$. We have that $r(z) = r(x)+r(y) = l(x)+l(y) = l(z)$, and so $P(z)$ is thus verified. The case where $z = (x)$ can be verified in the same way: $r(z) = r(x) + 1 = l(x) + 1 = l(z)$.

We deduce that $\forall x \in D$, $l(x) = r(x)$.

EXERCISE 3.13 Characterization of the Dyck language. We use the notations of Examples 3.9 and 3.13. Show that $D = L$, where

$$L = \{x \in A^* \,/\, l(x) = r(x) \text{ and } l(y) \geq r(y) \text{ for any prefix } y \text{ of } x\}. \qquad \diamondsuit$$

EXERCISE 3.14 Let BT be the set of binary trees and let h, n, f be the functions that give the height (see Example 3.24), the number of nodes (nodes are also called vertices) and the number of leaves of a tree respectively. Show that

1. $\forall x \in BT$, $n(x) \leq 2^{h(x)} - 1$,
2. $\forall x \in BT$, $f(x) \leq 2^{h(x)-1}$. $\diamondsuit$

EXERCISE 3.15 A binary tree is *strict* if it is non-empty and if it has no node with a single non-empty child. For instance, the trees of the Figure 3.5 (page 51) are strict, while the tree of the Figure 3.2 is non-strict.

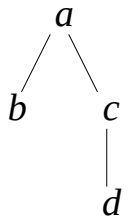

Figure 3.2

1. Give a definition of the set SBT of strict trees.
2. Show that $\forall x \in SBT$, $n(x) = 2f(x) - 1$. $\diamondsuit$

EXERCISE 3.16 A binary tree is said to be *balanced* if for each node in the tree, the difference between the heights of its left and right subtrees is at most 1. For instance, Figure 3.3 represents balanced trees of height 3, 4, and 5. (The labels of nodes are not represented.)

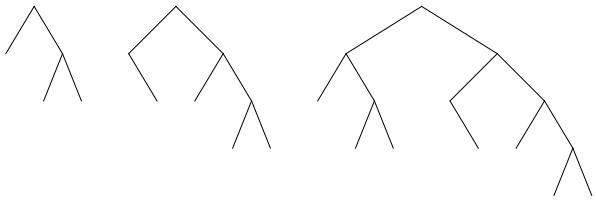

Figure 3.3

1. Give a definition of the set BBT of balanced binary trees.
2. We define $(u_n)_{n\in\mathbb{N}}$ by: $u_0 = 0, u_1 = 1$ and

$$\forall n \geq 0, \quad u_{n+2} = u_{n+1} + u_n + 1.$$

Show that $\forall x \in BBT$, $n(x) \geq u_{h(x)}$, where h and n are the functions that give the height and the number of nodes of a tree respectively. ♢

Exercise 3.17 Let A^* be the free monoid on alphabet A (see Definition 1.15, Example 2.4 and Exercise 3.11). The set Rat of rational languages is defined inductively by:
(B) $\emptyset \in Rat$ and $\forall a \in A$, $\{a\} \in Rat$,
(I_1) $L, M \in Rat \Longrightarrow L \cup M \in Rat$,
(I_2) $L, M \in Rat \Longrightarrow L \cdot M \in Rat$,
(I_3) $L \in Rat \Longrightarrow L^* \in Rat$.

1. The mirror image (or reverse) of language L is the set $\widetilde{L} = \{\widetilde{u} \,/\, u \in L\}$, where, if u is the string $u = a_1 a_2 \cdots a_n$, then $\widetilde{u}$ is the string $\widetilde{u} = a_n \cdots a_2 a_1$, see Exercise 3.18. Show that $L \in Rat \Longrightarrow \widetilde{L} \in Rat$.
2. We denote by $LF(L)$ the set of prefixes (left factors) of strings in the language L, i.e. $LF(L) = \{v \in A^* \,/\, \exists u \in L$ such that v is a prefix of $u\}$. Show that $L \in Rat \Longrightarrow LF(L) \in Rat$. ♢

3.3 Terms

In the present section we study a particular instance of definition by structural induction that is quite useful in computer science: the definition of terms. Many structures use terms in their representation.

3.3.1 Definition

Let $F = \{f_0, \ldots, f_n, \ldots\}$ be a set of operation symbols. With each symbol f is associated a finite arity (or rank) $a(f) \in \mathbb{N}$ representing the number of arguments of f. F_n denotes the set of arity n operation symbols.

Let U be the set of all strings of symbols in $F \cup \{\text{`(', `)', `, '}\}$. Let F_i be the set of symbols of arity i.

Definition 3.14 *The set T of terms built on F is inductively defined by:*
(B) $B = F_0 \subseteq T$,
(I) $\forall f \in F_n$, $\Phi_f(t_1, \ldots, t_n) = f\big(t_1, \ldots, t_n\big)$ *for* $t_1, \ldots, t_n$ *in* T.

Z $\Phi_f(t_1, \ldots, t_n)$ represents the result of operation Φ applied to the n-tuple of terms $(t_1, \ldots, t_n)$, i.e. a *semantic* object, whilst $f\big(t_1, \ldots, t_n\big)$ represents a string of formal symbols constituting a term, i.e. a *syntactic* object.

A term may be represented as a tree; for instance, $f\big(t_1, \ldots, t_n\big)$ may be pictured as in Figure 3.4.

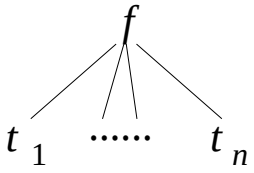

Figure 3.4

3.3.2 Interpretations of terms

Let V be an arbitrary set. With each element f of F_0 we associate an element $h(f)$ of V. With each element f of F_i with $i > 0$ we associate a mapping $h_f \colon V^i \longrightarrow V$.

Proposition 3.15 *There exists a unique function h^* from T to V such that:*
(B′) *If* $t \in F_0$, $h^*(t) = h(t)$.
(I′) *If* $t = f\big(t_1, \ldots, t_n\big)$, $h^*(t) = h_f(h^*(t_1), \ldots, h^*(t_n))$.

If t is a term, the element $h^*(t)$ of V will be called the *interpretation* of t by h^*.

Proof. By structural induction (or induction on the construction of terms). Let $P(t)$ be the property: 'there exists a unique $y = h^*(t)$ verifying (B′) and (I′)'.

(B) $P(t)$ is true if $t = f \in F_0$ because $y = h(t)$ by (B′).
(I) If $P(t_1), \ldots, P(t_n)$ are true, and if $t = f(t_1, \ldots, t_n)$, then $P(t)$ is true because
- on the one hand, there is a unique way of decomposing t in the form $f\big(t_1, \ldots, t_n\big)$: if $f\big(t_1, \ldots, t_n\big) = g\big(t'_1, \ldots, t'_p\big)$ then $f = g$, $n = p$, and $t_i = t'_i$, $\forall i = 1, \ldots, n$,
- on the other hand, by (I′) if $P(t_1), \ldots, P(t_n)$ are true then $P(t)$ must be true because $h^*(t)$ is entirely defined by

$$h^*(t) = h_f(h^*(t_1), \ldots, h^*(t_n)).$$

Another proof will be given in Section 3.4. □

EXAMPLE 3.16 Let $F_0 = \{a\}$, $F_1 = \{s\}$, $F = F_0 \bigcup F_1$. We have

$$T = \{a, s(a), s(s(a)), \ldots\}.$$

Let $V = \mathbb{N}$.

1. If $h_1(a) = 0$ and $h_{1s}(n) = n + 1$, then
$$h_1{}^*(s^n(a)) = h_1{}^*(\underbrace{s(s\ldots(s(a))\ldots)}_{n \text{ times}}) = n\,.$$
2. If $h_2(a) = 1$ and $h_{2s}(n) = 2n$, then
$$h_2{}^*(s^n(a)) = h_2{}^*(\underbrace{s(s\ldots(s(a))\ldots)}_{n \text{ times}}) = 2^n\,.$$
3. If $h_3(a) = 1$ and $h_{3s}(n) = n + 2$, then
$$h_3{}^*(s^n(a)) = h_3{}^*(\underbrace{s(s\ldots(s(a))\ldots)}_{n \text{ times}}) = 2n + 1\,.$$

Indeed, we verify by induction that:

1. $h_1(a) = 0$ and
$h_1{}^*(s^{n+1}(a)) = h_1{}^*(s(s^n(a))) = h_{1s}(h_1{}^*(s^n(a))) = h_{1s}(n) = n + 1$,
2. $h_2(a) = 1$ and
$h_2{}^*(s^{n+1}(a)) = h_2{}^*(s(s^n(a))) = 2 \times h_2{}^*(s^n(a)) = 2 \times 2 = 2^{n+1}$,
3. $h_3(a) = 1$ and
$h_3{}^*(s^{n+1}(a)) = h_3{}^*(s(s^n(a))) = h_3{}^*(s^n(a)) + 2 = 2n + 1 + 2 = 2(n + 1) + 1$.

Let E be an arbitrary set, and let X be the subset of E inductively defined by the conditions (B) and (I). Theorem 3.10 asserts that each element of X is obtained from the basis by applying a finite number of inductive steps. We refine this result by describing by a term how the element x is obtained.

With each element b of the basis B, we associate a nullary symbol denoted by $\overline{b}$. With each function Φ of K, we associate the arity $a(\Phi)$ symbol $\overline{\Phi}$. Let T be the set of all terms constructed with these symbols.

We consider the interpretation $h^* : T \to E$ defined by

- $h(\overline{b}) = b$,
- $h_{\overline{\Phi}}(x_1, \ldots, x_{a(\Phi)}) = \Phi(x_1, \ldots, x_{a(\Phi)})$.

Proposition 3.17 $X = \{h^*(t) \,/\, t \in T\}$.

Proof. For an element x of E, let $P(x)$ be the property: 'there exists a term t such that $x = h^*(t)$'. It is easy to see that P has properties (B″) and (I″) of Proposition 3.11, and thus $X \subseteq h^*(T)$.

For a term t of T, let $Q(t)$ be the property: '$h^*(t) \in X$'. Here also Proposition 3.11 enables us to conclude that $h^*(T) \subseteq X$. □

3.3.3 Unambiguous definitions

Definition 3.18 *An inductive definition of a set X is said to be unambiguous if the mapping h^* of Proposition* 3.17 *is injective, i.e. for any $x \in X$ there exists a unique term t such that $x = h^*(t)$.*

More intuitively, this means that there is a unique way of building up an element x of X.

EXAMPLE 3.19 The following definition of $\mathbb{N}^2$ is ambiguous:

(B) $(0,0) \in \mathbb{N}^2$,
(I_1) $(n,m) \in \mathbb{N}^2 \Longrightarrow (n+1,m) \in \mathbb{N}^2$,
(I_2) $(n,m) \in \mathbb{N}^2 \Longrightarrow (n,m+1) \in \mathbb{N}^2$.

Indeed, the pair $(1,1)$ can be obtained from $(0,0)$ by using the rule (I_1) first then the rule (I_2), or by using the rule (I_2) first then the rule (I_1).

More formally, we consider the terms built up from

- the arity 0 symbol $\overline{b}$ whose interpretation $h(\overline{b})$ is $(0,0)$,
- the unary symbols $\overline{f}$ and $\overline{g}$ whose interpretations are defined by
  (i) $h_{\overline{f}}(n,m) = (n+1,m)$,
  (ii) $h_{\overline{g}}(n,m) = (n,m+1)$.

Then $(1,1) = h^*(\overline{f}\big(\overline{g}\big(\overline{b}\big))) = h^*(\overline{g}\big(\overline{f}\big(\overline{b}\big))$.

3.3.4 Inductively defined functions

In order to define a function on an inductively defined set unambiguously, it is convenient to use an inductive definition. Intuitively, we define the function on the elements of the basis directly, and then define new elements inductively, building them up from elements already defined.

Definition 3.20 *Let $X \subseteq E$ be an unambiguous inductively defined set (see Definitions* 3.7 *and* 3.18*), and let F be any set. The inductive definition of mapping ψ from X to F consists of*

(B) *specifying $\psi(x) \in F$ for each element $x \in B$,*
(I) *specifying the expression of $\psi\big(\Phi(x_1,\ldots,x_{a(\Phi)})\big)$ in terms of $x_1,\ldots,x_{a(\Phi)}$ and of $\psi(x_1),\ldots,\psi(x_{a(\Phi)})$ for each $\Phi \in K$. We will write*

$$\psi\big(\Phi(x_1,\ldots,x_{a(\Phi)})\big) = \psi_\Phi\big(x_1,\ldots,x_{a(\Phi)},\psi(x_1),\ldots,\psi(x_{a(\Phi)})\big)\ ,$$

where ψ_Φ is a mapping from $E^{a(\Phi)} \times F^{a(\Phi)}$ to F.

The definition is illustrated by the following examples.

EXAMPLE 3.21 The factorial function from $\mathbb{N}$ to $\mathbb{N}$ is defined inductively by

(B) $\text{Fact}(0) = 1$,
(I) $\text{Fact}(n+1) = (n+1) \times \text{Fact}(n)$.

Here we use the inductive definition of $\mathbb{N}$ given in Example 3.9. First, the factorial function for the unique element of the basis (0) is defined directly, and then the factorial applied to the new element $n+1$ is expressed in terms of n and $\text{Fact}(n)$.

Henceforth, we will also write inductive definitions of functions as follows:

$$\text{Fact}(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times \text{Fact}(n-1) & \text{otherwise.} \end{cases}$$

EXERCISE 3.18 Let A^* be the free monoid on the alphabet A (see Definition 1.15). The mirror image (or reverse) of a string $u = a_1 a_2 \cdots a_n$ is the string $\widetilde{u} = a_n \cdots a_2 a_1$. Give an inductive definition of the mirror image. ♢

EXERCISE 3.19 Let the lists L of letters from the alphabet A be defined inductively by:

(B) $\varepsilon \in L$,
(I) $\forall l \in L$, $\forall a \in A$, $(al) \in L$.

We define $g(x, y)$ on $L \times L$ by, $\forall a \in A$, $\forall l \in L$, $\forall y \in L$,

$$g(\varepsilon, y) = y \ ,$$
$$g((al), y) = g(l, (ay)) \ .$$

1. Let $Q(x)$ be the predicate '$\forall y$, $g(x, y)$ is defined'. Prove by induction on x that $Q(x)$ holds on L.
2. Compute $g((a_1), y)$, for $a_1 \in A$, $y \in L$.
3. Prove by induction on n (for $n \geq 1$) that $g\big((a_n(a_{n-1}(\ldots(a_1)\ldots))), y\big) = g\big(\varepsilon, (a_1(\ldots(a_{n-1}(a_n y))\ldots))\big)$.
4. Let $rev(x) = g(x, \varepsilon)$. Deduce from 3 that, for $a_1, \ldots, a_n \in A$,
$$rev\big((a_n(a_{n-1}(\ldots(a_1)\ldots)))\big) = (a_1(\ldots(a_{n-1}(a_n))\ldots)).$$ ♢

We now justify Definition 3.20 and explain why we have assumed the definition of the set X to be unambiguous.

Instead of defining a function ψ from X to F, we will define a function ψ' from T to F, where T is the set of terms whose interpretation is in X (see Proposition 3.17). ψ' is defined as follows:

- $\psi'(\overline{b}) = \psi(b)$,
- $\psi'(\overline{\Phi}(t_1, \ldots, t_{a(\Phi)})) = \psi_\Phi\big(h^*(t_1), \ldots, h^*(t_{a(\Phi)}), \psi'(t_1), \ldots, \psi'(t_{a(\Phi)})\big)$.

As in the proof of Proposition 3.15 we show that such a function exists and is unique.

If the inductive definition of X is unambiguous, then for each element x of X there exists a unique term t such that $h^*(t) = x$. Then let $\psi(x) = \psi'(t)$. ψ is thus indeed a mapping from X to F and it is easy to prove that ψ verifies the conditions (B) and (I) of Definition 3.20.

If the definition of X is ambiguous, then there exist several terms $t_1, \ldots, t_n$ whose interpretation is the same element x of X and, according to the chosen term, Definition 3.20 will give different values $\psi(t_1), \ldots, \psi(t_n)$ to $\psi(x)$. This is illustrated by the following example.

EXAMPLE 3.22 Let us consider the following inductive definition of ψ from $\mathbb{N}^2$ to $\mathbb{N}$, where the inductive definition of $\mathbb{N}^2$ is given in Example 3.19:

(B) $\psi(0,0) = 1$,
(I'_1) $\psi(n+1, m) = \psi(n, m)^2$,
(I'_2) $\psi(n, m+1) = 3 \times \psi(n, m)$.

The thus defined ψ is not a mapping because by using the rule (I'_1) first and then the rule (I'_2), we obtain $\psi(1,1) = \psi(0,1)^2 = (3 \times \psi(0,0))^2 = 3^2 = 9$, whilst by using the rule (I'_2) first and then the rule (I'_1), we obtain

$$\psi(1,1) = 3 \times \psi(1,0) = 3 \times \psi(0,0)^2 = 3.$$

More generally, we can consider that Definition 3.20 in fact defines a relation $\mathcal{R}$ from X to F by: $x \mathcal{R} y$ if and only if there exists a term t such that $x = h^*(t)$ and $y = \psi'(t)$. If h^* is injective then this relation is functional, as we just saw. We should, however, note that this is not the only case when $\mathcal{R}$ is functional. In fact $\mathcal{R}$ is functional if and only if

$$\forall t, t' \in T, \quad h^*(t) = h^*(t') \Longrightarrow \psi'(t) = \psi'(t').$$

EXAMPLE 3.23 We consider again the ambiguous definition of $\mathbb{N}^2$ given in Example 3.19 and we consider the inductive definition

(B) $g(0,0) = 1$,
(I_1) $g(n+1, m) = 2 \times g(n, m)$,
(I_2) $g(n, m+1) = 3 \times g(n, m)$.

Using Proposition 3.11, we easily show by induction that there exists a unique mapping g verifying these conditions and that this unique mapping is defined by $\forall (n, m) \in \mathbb{N}^2$, $g(n, m) = 2^n 3^m$.

EXERCISE 3.20 Let $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. We give an inductive definition of the function 'modulo' defined on $\mathbb{N} \times \mathbb{N}^*$, that, when applied to the pair (n, m), gives the remainder of the Euclidean division of n by m

$$n \bmod m = \begin{cases} n & \text{if } n < m, \\ (n - m) \bmod m & \text{otherwise.} \end{cases}$$

Give the corresponding unambiguous inductive definition of $\mathbb{N} \times \mathbb{N}^*$. ♢

EXERCISE 3.21 Inductively define the gcd function on $X = \mathbb{N} \times \mathbb{N} \setminus \{(0,0)\}$ (i.e. the greatest common divisor). What is the corresponding unambiguous inductive definition of X? ♢

We now show some examples of inductively defined functions on sets other than $\mathbb{N}$.

EXAMPLE 3.24

1. The expressions of the set E (see Example 3.9) use an infix notation (in which the operator is placed between its arguments). We can also use a postfix notation without parentheses (in which the operator is placed after both its arguments). For instance, the postfix notation of expression

$$\Big(\big(a \times (b + c)\big) + d\Big)$$

is $abc+\times d+$. The transformation from the infix notation to the postfix notation is inductively defined by

(B) $\forall a \in A$, $\mathrm{Post}(a) = a$,

(I) $\forall e, f \in E$, $\mathrm{Post}\big((e + f)\big) = \mathrm{Post}(e)\,\mathrm{Post}(f)+$ and $\mathrm{Post}\big((e \times f)\big) = \mathrm{Post}(e)\,\mathrm{Post}(f)\times$.

2. The height of a binary tree is inductively defined by

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

(I) $\forall l, r \in BT, \forall a \in A$, $h\big((a, l, r)\big) = 1 + \max\big(h(l), h(r)\big)$.

A more elegant definition of this function is

$$h(x) = \begin{cases} 0 & \text{if } x = \emptyset, \\ 1 + \max\big(h(l), h(r)\big) & \text{if } x = (a, l, r). \end{cases}$$

3. The inorder traversal of a tree is the list of the labels of its nodes from left to right. We can notice that several trees may have the same inorder traversal. For instance, the two trees of Figure 3.5 have the same inorder traversal *bacad*. The inductive definition of the inorder traversal is

$$\mathrm{Inf}(x) = \begin{cases} \varepsilon & \text{if } x = \emptyset, \\ \mathrm{Inf}(l) \cdot a \cdot \mathrm{Inf}(r) & \text{if } x = (a, l, r). \end{cases}$$

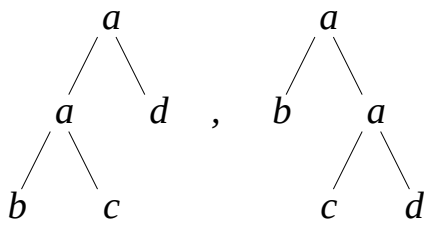

Figure 3.5

EXERCISE 3.22 Give inductive definitions of the functions n and l from BT to $\mathbb{N}$, defining respectively, the number of nodes and the number of leaves of a binary tree. For instance, if x is either tree in Figure 3.5, we have $n(x) = 5$ and $l(x) = 3$. ♢

EXERCISE 3.23 Define the preorder traversal of a binary tree. The preorder traversals of the trees of Figure 3.5 are *aabcd* and *abacd*. ♢

Note that inductive definitions are appropriate as definitions of certain algorithms: sorting algorithms, algorithms on trees such as binary search, insertion, traversal, etc.

3.4 Closure operations

In the proof of Theorem 3.10, we showed that the set X inductively defined by

(B) $B \subseteq X$, and
(I) $\forall \Phi \in K, \forall x_1, \ldots, x_{a(\Phi)} \in X, \Phi(x_1, \ldots, x_{a(\Phi)}) \in X$,

is the union of the sets X_n with $X_0 = B$ and $X_{n+1} = X_n \cup \{\Phi(x_1, \ldots, x_{a(\Phi)}) / x_1, \ldots, x_{a(\Phi)} \in X_n \text{ and } \Phi \in K\}$. We see that the subset (I) of the inductive definition of X is used in order to build a new set X_{n+1} from an already known set X_n. Indeed, it suffices to define X_{n+1} (or even $X_{n+1} - X_n$) from X_n and, when no new element can be added, the inductive definition is completed.

More generally, we will assume that from any given set E we can build a new set $C(E)$. We will study the properties that C should have in order to give an inductive definition which will be completed whenever C can add no new element to E. With this standpoint we will generalize the results of Section 3.2.

Let U be any set and let $C : \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ be a *monotone* mapping, i.e. a mapping verifying $\forall E, E' \subseteq U$, $E \subseteq E' \Longrightarrow C(E) \subseteq C(E')$.

A subset E of U is said to be *C-closed* if $C(E) \subseteq E$.

Proposition 3.25 *Let I be any set of indices. Let E_i be a C-closed set, for any $i \in I$. Then $\bigcap_{i \in I} E_i$ is C-closed.*

Proof. Let $E = \bigcap_{i \in I} E_i$. Since $E \subseteq E_i$ and E_i is C-closed, $C(E) \subseteq C(E_i) \subseteq E_i$ hence $C(E) \subseteq \bigcap_{i \in I} E_i = E$. □

If E is any subset of U, the intersection of all the C-closed subsets of U containing E is a C-closed subset of U containing E, denoted by $\hat{C}(E)$.

Proposition 3.26
- *If E' is a C-closed subset containing E, then $\hat{C}(E) \subseteq E'$,*
- $E \subseteq \hat{C}(E)$,
- $\hat{C}(\hat{C}(E)) = \hat{C}(E)$,
- $E \subseteq E' \Longrightarrow \hat{C}(E) \subseteq \hat{C}(E')$.

Proof. The first two points are clear by the definition of $\hat{C}(E)$.

- On the one hand, we have that $E \subseteq \hat{C}(E) \subseteq \hat{C}(\hat{C}(E))$. And, on the other hand, since $\hat{C}(E)$ is a C-closed subset containing $\hat{C}(E)$, then $\hat{C}(\hat{C}(E)) \subseteq \hat{C}(E)$.
- If $E \subseteq E'$, then $E \subseteq \hat{C}(E')$ which is a C-closed subset containing E. We thus have that $\hat{C}(E) \subseteq \hat{C}(E')$. □

The next proposition is a generalization of the induction principles and may be called the *universal induction principle*.

Proposition 3.27 *Let $P \subseteq U$ be such that $C(P) \subseteq P$. Then*

$$\forall E\ , \quad E \subseteq P \implies \hat{C}(E) \subseteq P.$$

Proof. If $C(P) \subseteq P$, then P is C-closed. So if $E \subseteq P$, then $\hat{C}(E) \subseteq P$. □

Example 3.28 Let X be a subset of a set U; assume that X is inductively defined by (B) and (I) (see Definition 3.7). Define $C\colon \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ by

$$C(Y) = \{\Phi(y_1, \ldots, y_{a(\Phi)}) \,/\, \Phi \in K, y_1, \ldots, y_{a(\Phi)} \in Y\}.$$

Then $X = \hat{C}(B)$.

Example 3.29 Let $U = \mathbb{N}$, $C(E) = \{n+1 \,/\, n \in E\}$. Then

$$E' = \hat{C}(E) = \{n + m \,/\, n \in E, m \in \mathbb{N}\}.$$

Indeed, $C(E') = \{n + m + 1 \,/\, n \in E', m \in \mathbb{N}\} \subseteq E'$. Assume there is a C-closed subset E'' containing E and strictly included in E'. Let k be the least integer of E' that is not in E'', i.e. $k \in E', \quad k \notin E''$ and $(k = 0$ or $k - 1 \in E'')$.

- If $k = 0$ then, since $k = n + m$ with $n \in E$, $0 \in E$, and hence $0 \in E''$, a contradiction.
- Otherwise, $k - 1 \in E'' \Longrightarrow k = (k-1) + 1 \in C(E'') \subseteq E''$, a contradiction.

We deduce: $\hat{C}(E) = \{m \,/\, m \geq \inf(E)\} = \hat{C}(\{\inf(E)\})$. Let P be such that $n \in P \Longrightarrow n + 1 \in P$. Then $C(P) \subseteq P$, and hence $\inf(E) \in P \Longrightarrow \hat{C}(E) \subseteq P$.

If $\inf(E) = 0$ then $\hat{C}(E) = \mathbb{N}$, and we again find the induction principle on the integers.

EXERCISE 3.24 Let $U = \mathbb{N}$ and $C(E) = \{n + m \,/\, n \in E, m \in E\}$. Let $k\mathbb{N} = \{kn \,/\, n \in \mathbb{N}\}$. Show that if $E \subseteq k\mathbb{N}$ then $\hat{C}(E) \subseteq k\mathbb{N}$. ♢

Let $C: \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ be such that $E \subseteq E' \Longrightarrow C(E) \subseteq C(E')$. C is *finitary* if it also verifies: $\forall E, \forall e \in C(E)$, there exists a *finite* subset F of E such that $e \in C(F)$.

Let $E \subseteq U$. Consider the monotone increasing (for inclusion) sequence

$$\begin{aligned} E_0 &= E \\ E_1 &= E_0 \cup C(E_0) \\ &\vdots \\ E_{i+1} &= E_i \cup C(E_i) \\ &\vdots \\ \text{and} \qquad \hat{E} &= \bigcup_{i \leq 0} E_i \ . \end{aligned}$$

Proposition 3.30 *$\hat{E} \subseteq \hat{C}(E)$. If C is finitary, $\hat{E} = \hat{C}(E)$.*

Proof. Let $E' = \hat{C}(E)$. We show by induction on the integers that $\forall i \geq 0$, $E_i \subseteq E'$.

- $E_0 = E \subseteq E'$.
- We assume $E_i \subseteq E'$. Then $C(E_i) \subseteq C(E') \subseteq E'$ and $E_{i+1} = E_i \cup C(E_i) \subseteq E'$. Since $\forall i \geq 0$, $E_i \subseteq E'$, we have $\hat{E} = \bigcup_{i \geq 0} E_i \subseteq E'$.

We show that if C is finitary then $\hat{E}$ is C-closed, and we will therefore deduce that $E' \subseteq \hat{E}$. Let $e \in C(\hat{E})$. Because C is finitary, there exists a finite subset $F = \{x_1, \ldots, x_p\}$ of $\hat{E}$ such that $e \in C(F)$. Since $x_j \in \bigcup_{i \geq 0} E_i$, there exists i_j such that $x_j \in E_{i_j}$; let $k = \max\{i_j \,/\, j = 1, \ldots, p\}$. We thus have that $F \subseteq E_k$ and $e \in C(F) \subseteq C(E_k) \subseteq E_{k+1} \subseteq \hat{E}$. Hence $C(\hat{E}) \subseteq \hat{E}$. □

EXAMPLE 3.31 The mapping C that inductively defines a set X (see Example 3.28) is finitary, whence Theorem 3.10.

EXAMPLE 3.32 The mapping C from $\mathcal{P}(\mathbb{R})$ to itself, which is defined by $y \in C(X)$ if and only if there exists $Y \subseteq X$ such that $y = \inf Y$, is not finitary. Indeed, let $X = \{1/n \,/\, n \in \mathbb{N}, n > 0\}$. We thus have $0 \in C(X)$. But for all finite subsets F of X, $0 \notin C(F)$ because the greatest lower bound of any finite subset of X is of the form $1/n$ for some $n > 0$.

EXERCISE 3.25 Let E be a vector space on $\mathbb{R}$. For $a, b \in E$, let

$$[a, b] = \{\lambda a + \mu b \,/\, \lambda \geq 0, \mu \geq 0, \text{ and } \lambda + \mu = 1\}$$

be the closed segment subtended by a and b. Let $C : \mathcal{P}(E) \to \mathcal{P}(E)$ be defined by

$$C(X) = \bigcup_{a,b \in X} [a, b].$$

What usual name is given to $C(X)$?

1. Is C monotone increasing and finitary?
2. Given $a \in \hat{C}(A)$ and $b \in \hat{C}(B)$, show that $[a, b] \subseteq \hat{C}(A \cup B)$.
3. Deduce that $\bigcup_{F \in \mathrm{fin}(X)} \hat{C}(F)$ is C-closed, where $\mathrm{fin}(X)$ is the set of finite subsets of X.
4. Is $\hat{C}$ monotone increasing and finitary?
5. Can you generalize (4) to any set transformation which is monotone increasing and finitary? ♢

We can apply the closure operations in order to define the terms. Let U, F and F_i be as in Section 3.3. Let $C: \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ be defined by

$$C(E) = \bigcup_{i>0} \{f(\sigma_1, \ldots, \sigma_i) \,/\, \sigma_j \in E, f \in F_i\}.$$

Then C is finitary and the set T of terms built on F is identical to $\hat{C}(F_0)$.

EXERCISE 3.26 Let $C' = C(E) \cup F_0$. Show that $\hat{C}(F_0) = \hat{C}'(\emptyset)$. ♢

EXERCISE 3.27 Let $\hat{E} = \hat{C}(F_0)$. Show that $T = \hat{E}$. ♢

EXERCISE 3.28 Show that there exists a unique function h^* verifying conditions (B′) and (I′) of Proposition 3.15. ♢

EXERCISE 3.29 Let $U = \mathbb{N}$ and let $C: \mathcal{P}(\mathbb{N}) \longrightarrow \mathcal{P}(\mathbb{N})$ be defined by

$$C(X) = \begin{cases} \{x + 1 \,/\, x \in X\} & \text{if } X \text{ is finite,} \\ \{x + 1 \,/\, x \in X\} \cup \{0\} & \text{if } X \text{ is infinite.} \end{cases}$$

1. Show that the limit of the sequence

$$\begin{aligned} E_0 &= \{1\} \\ E_1 &= E_0 \cup C(E_0) \\ &\vdots \\ E_{i+1} &= E_i \cup C(E_i) \\ &\vdots \end{aligned}$$

is equal to $\mathbb{N} \setminus \{0\}$.
2. Show that $\hat{C}(\{1\}) = \mathbb{N}$.
3. Explain this result. ♢