

## CHAPITRE (−1)

# *Ordres et induction*

Hugo SALOU MPI\*

Dernière mise à jour le 13 mai 2023

---

## Table des matières

<b>0</b>	<b>Motivation</b>	<b>2</b>
<b>1</b>	<b>Ordre</b>	<b>3</b>
1.1	Ordre produit . . . . .	4
1.2	Ordre lexicographique . . . . .	7
<b>2</b>	<b>Induction nommée</b>	<b>9</b>
<b>3</b>	<b>Synthèse du chapitre</b>	<b>15</b>

## 0 Motivation

```
1 let rec fact n =  
2   if n = 0 then 1  
3   else n * (fact (n-1));;
```

CODE 1 – Calcul de factorielle



LE PROGRAMME calcule la factorielle d'un nombre ? En développant, l'expression de `fact 3`, on a

$$\begin{aligned}(\text{fact } 3) &= 3 \times (\text{fact } 2) \\ &= 3 \times (2 \times (1 \times 1))\end{aligned}$$

On en déduit que ce programme calcule la factorielle car ce développement s'arrête à un certain point. Comment en être sûr ? En effet, avec  $n = -1$ , on obtient une `Stack Overflow Error` ; on n'a plus de mémoire. Pour en être sûr, il faut définir un *invariant*.

À faire : Ajouter 2<sup>ème</sup> exemple

Un autre exemple :

```
1 let mystere2 n m =  
2   let rec aux c b =  
3     if c = 0 and b = m then 0  
4     else if c = 0 then aux (b * m) (b + 1)  
5     else 1 + aux (c - 1) b  
6   in aux n 0;;
```

CODE 2 – Un programme mystère (2)

Ce programme a beaucoup plus de variables : les variables augmentent dans certains cas, puis diminuent. . . On peut représenter l'état des variables `b` et `c` dans une figure :

À faire : Figure à faire

FIGURE 1 – État des variables `b` et `c`

On en conclut que ce programme calcule la valeur de

$$n + m + 2m + \dots + m^2 = n + \frac{m^2 \times (m-1)}{2}.$$

Cherchons un invariant. On peut penser à  $b - m$  mais il ne diminue pas à chaque étape. Nous verrons quel est ce variant plus tard dans le chapitre.

Nouvel exemple : retourner une liste. Essayons de distinguer les différents cas possibles : si la liste est vide, on la renvoie ; sinon, on extrait un élément `x` et on note le reste de la liste `xs`, on retourne `xs` puis on concatène à droite `x`. On peut donc écrire

```
1 let rec rev l =  
2   match l with  
3   | [] -> []  
4   | x :: xs -> (rev xs) @ [x];;
```

CODE 3 – Inverser une liste

Cependant, le `@` est une opération lente en OCaml. En effet ce programme a une complexité en  $\mathcal{O}(n^2)$ , où  $n$  est la taille de la liste. Cette complexité est douteuse pour une opération aussi simple. On peut également se demander si cette opération se termine. Cela paraît très simple : la taille de la liste diminue mais nous n'avons pas le côté mathématique d'une liste. En effet, qu'est ce qu'une liste et la taille de cette liste ? On doit formaliser l'explication de pourquoi cet algorithme se termine.

Continuons avec un autre exemple :

```

1 let rec mystere3 m n =
2   if m = 0 then n + 1
3   else if n = 1 then mystere (n - 1) 1
4   else mystere (m - 1) (mystere m (n-1));;

```

CODE 4 – Un programme mystère (3)

Il s'agit de la fonction ACKERMANN. Sa complexité est très importante mais ce n'est pas le sujet de cette introduction. En effet, on a

$$\begin{aligned}
 A_{0,m} &= n + 1 \\
 A_{m,0} &= A_{n-1,1} \\
 A_{m,n} &= A_{m-1,A_{m,n-1}}
 \end{aligned}$$

Malgré ce que l'on peut penser, cette fonction se termine mais comment le prouver?

À faire : Exemple arbres binaires

On en conclut que, avec les outils de l'année passée, il est difficile de prouver que ces algorithmes se terminent rigoureusement.

## 1 Ordre

**Définition (Éléments minimaux) :** Lorsque  $(E, \preceq)$  est un espace ordonné, et  $A \subseteq E$  ("includ ou égal") est une partie de  $E$ , on appelle *élément minimal* de  $A$  un élément  $x \in A$  tel que

$$\forall y \in A, y \preceq x \implies y = x.$$

EXEMPLE :

La figure ci-dessous est un diagramme de HASSE : c'est un diagramme où les points représente les éléments de l'ensemble  $E$  et où les segments représentent une comparaison entre les deux éléments connectés : l'élément inférieur est représenté plus bas. Dans l'exemple ci-dessous, les éléments minimaux de  $A$  sont les points  $b$  et  $f$ .

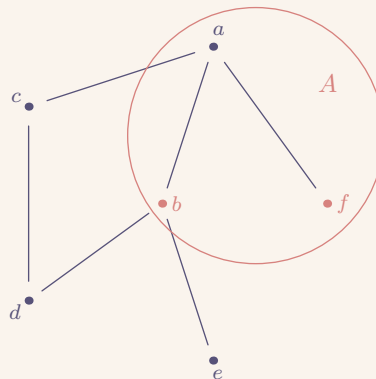


FIGURE 2 – Diagramme de HASSE

**Définition (Ordre bien fondé) :** Un ordre est *bien fondé* s'il n'existe pas de suite infiniment strictement croissante.

Exemple :

OUI	NON
$(\mathbb{N}, \leq)$	$(\mathbb{Z}, \leq)$
	$(\mathbb{R}, \leq)$
	$(\mathbb{R}^+, \leq) \text{ (1/2}^n \text{)}$
$(E, \subseteq) \text{ (si } E \text{ est fini)}$	$(E, \subseteq) \text{ (en général)}$

TABLE 1 – Exemples et non-exemples d'ordres bien fondés

**Propriété :** Une relation d'ordre  $\preccurlyeq$  sur un ensemble  $E$  bien fondé si et seulement si toute partie non vide de  $E$  admet un élément minimal.

*Preuve “ $\Leftarrow$ ”* Supposons que toute partie non vide de  $E$  admet un élément minimal. Supposons, de plus, qu'il existe une suite  $(x_n)_{n \in \mathbb{N}}$  infiniment strictement décroissante. Soit alors  $A = \{x_n \mid n \in \mathbb{N}\}$  qui admet un élément minimal ; soit  $n_0$  son indice. Or,  $x_{n_0+1} \preccurlyeq x_{n_0}$  ce qui est absurde.

*“ $\Rightarrow$ ”* Supposons que  $(E, \preccurlyeq)$  est un ensemble bien fondé. Supposons également qu'il existe un sous-ensemble  $A$  de  $E$  non vide n'admettant pas d'élément minimal. Comme  $A$  est non vide, on pose alors  $x_0 \in A$ . Et, comme  $A$  n'admet pas d'élément minimal, donc il existe  $x \in A$  tel que  $x \preccurlyeq x_0$ . Notons un tel  $x, x_1$ . En itérant ce procédé, on crée la suite  $(x_i)_{i \in \mathbb{N}}$  qui est infiniment strictement décroissante ; ce qui est absurde. □

**Théorème (Induction bien fondée) :** Soit  $(E, \preccurlyeq)$  un ensemble ordonné et bien fondé. Soit  $P$  une propriété sur les éléments de  $E$ . Si  $x \in E$ , on note  $E^{\preccurlyeq x} = \{y \in E \mid y \preccurlyeq x\}$ . Si  $\forall x \in E, (\forall y \in E^{\preccurlyeq x}, P(y)) \implies P(x)$ , alors  $\forall x \in E, P(x)$ .

REMARQUE :

Si  $(E, \preccurlyeq) = (\mathbb{N}, \leq)$ , alors le théorème précédent se traduit par :

$$\text{si } \forall n \in \mathbb{N}, (\forall p < n, P(p)) \implies P(n), \text{ alors } \forall n \in \mathbb{N}, P(n).$$

Ce résultat correspond à la “récurrence forte.” Décomposons ce “ $\forall$ ” : on extrait le cas où  $n = 0$

$$\text{si } P(0) \text{ et } \forall n \in \mathbb{N}^*, (\forall p < n, P(p)) \implies P(n), \text{ alors } \forall n \in \mathbb{N}, P(n).$$

On peut donc utiliser le principe de la récurrence pour tout ensemble ordonné bien fondé.

*Preuve :*  
Soit  $A = \{x \in E \mid P(x) \text{ n'est pas vrai}\}$ .

CAS 1  $A = \emptyset$ , alors OK.

CAS 2  $A \neq \emptyset$ . Soit alors  $x \in A$  un élément minimal de  $A$  (c.f. proposition d'avant). On a que  $\forall y \in E, y \preccurlyeq x, P(y)$  est vrai donc  $P(x)$  est vrai par hypothèse. Ce qui est absurde. □

## 1.1 Ordre produit

**Définition :** Soit  $(A, \preccurlyeq_A)$  et  $(B, \preccurlyeq_B)$  deux ensembles ordonnés on définit alors  $\preccurlyeq_x$  sur

$A \times B$  par

$$\forall (a, b), (a', b') \in A \times B, (a, b) \preceq_{\times} (a', b') \stackrel{\text{def}}{\iff} (a \preceq_A a' \text{ et } b \preceq_B b').$$

**Propriété :**  $\preceq_{\times}$  est une relation d'ordre.  $\square$

La proposition précédente est facilement vérifiée comme  $\preceq_A$  et  $\preceq_B$  sont, elles aussi, des relations d'ordre.

**REMARQUE (B) :**

Si  $\preceq_A$  et  $\preceq_B$  sont des ordres totaux,  $\preceq_{\times}$  ne l'est pas forcément.

**Propriété :** Soient  $(A, \preceq_A)$  et  $(B, \preceq_B)$  bien fondés, alors  $(A \times B, \preceq_{\times})$  l'est aussi.

*Preuve :*

Supposons alors que  $(A \times B, \preceq_{\times})$  ne soit pas bien fondée. Nous avons donc une suite infiniment strictement décroissante

$$(a_0, b_0) \succ_{\times} (a_1, b_1) \succ_{\times} (a_2, b_2) \succ_{\times} \dots$$

On a donc  $a_0 \succ_A a_1 \succ_A a_2 \succ \dots$ . Or,  $(A, \preceq_A)$  est bien fondée donc il existe  $n_0 \in \mathbb{N}$  tel que  $\forall i \geq n_0, a_i = a_{n_0}$ . On a donc  $\forall i \geq n_0, b_i \prec_B b_{i-1}$ . Considérons  $(b_i)_{i \geq n_0}$  est infiniment strictement décroissante dans  $(B, \preceq_B)$ , ce qui est absurde.  $\square$

**REMARQUE :**

On a défini une relation “produit,” on peut se demander si ces résultats s'appliquent aussi si la relation est “somme.” Ce n'est pas le cas : soit  $\preceq_+$  définie comme

$$(a, b) \preceq_+ (a', b') \stackrel{\text{def}}{\iff} a \preceq_A a' \text{ ou } b \preceq_B b'.$$

On peut démontrer que ce n'est pas une relation d'ordre.

**REMARQUE :**

Si  $(A, \preceq_A)$  est une relation d'ordre alors  $(A^{\mathbb{N}}, \preceq_{\times})$  a les mêmes propriétés.

**REMARQUE :**

Sur  $A^{\mathbb{N}}$ , on définit  $(u_n)_{n \in \mathbb{N}} \preceq_{\times} (v_n)_{n \in \mathbb{N}}$  si et seulement si

$$\forall i, u_i \preceq_A v_i.$$

L'ensemble ordonné  $(A^{\mathbb{N}}, \preceq_{\times})$  est-il bien fondé? La réponse est non.

Voici un contre-exemple : on pose  $A = \{0, 1\}$ .

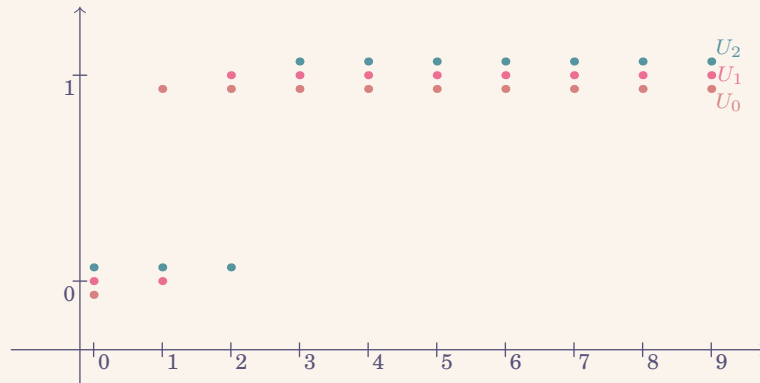


FIGURE 3 – Contre exemple :  $(A^{\mathbb{N}}, \preceq_{\times})$  est-il bien fondé ?

On considère la suite  $U_0$  qui a pour tout  $n \in \mathbb{N}$  la valeur de 1. Puis, on considère la suite  $U_1$  qui a, pour  $n = 0$ , la valeur de 0 puis pour les autres valeurs de  $n$ , la valeur de 1. Ensuite, on considère la suite  $U_2$  qui, pour  $n = 0, 1$ , la valeur de 0 puis, pour les autres valeurs de  $n$ , la valeur de 1. En itérant ce procédé, on crée une suite de suite  $(U_n)_{n \in \mathbb{N}}$  infiniment strictement décroissante :

$$U_0 \succ_{\times} U_1 \succ_{\times} U_2 \succ_{\times} \dots$$

On considère le programme suivant :

```
1 let rec pgcd a b =
2   if a = b then a
3   else if a > b then pgcd (a-b) b
4   else pgcd a (b-a);;
```

CODE 5 – Calcul du PGCD

Étudions ce programme. Ce programme se termine si et seulement si  $a = 0$  et  $b = 0$  où si  $a > 0$  et  $b > 0$ . Prouvons-le rigoureusement. On choisit comme variant  $(a, b)$  vivant dans l'ensemble ordonné  $(\mathbb{N}^* \times \mathbb{N}^*, \preceq_{\times})$  où  $\preceq_{\times}$  est la relation d'ordre produit. À faire : Recopier une partie du cours ici. On a donc bien une décroissance stricte de la valeur de l'expression  $(a, b)$  valeurs dans un espace bien fondé. D'où terminaison.

Démontrons maintenant la correction, c'est-à-dire, démontrons que

$$\forall (a, b) \in \mathbb{N}^* \times \mathbb{N}^*, (\text{pgcd } a \ b) = a \wedge b.$$

Pour cela, on procède par induction sur  $((\mathbb{N}^*)^2, \preceq_{\times})$  pour démontrer la proposition

$$P(a, b) = (\text{pgcd } a \ b) = a \wedge b.$$

— Soit  $(a, b) = (1, 1)$ . On a

$$(\text{pgcd } a \ b)_{(\text{code})} = a = a \wedge b.$$

— Soit  $(a, b) \neq (1, 1) \in (\mathbb{N}^*)^2$  tel que pour tout  $(c, d) \in (\mathbb{N}^*)^2$  tel que  $(c, d) \prec_{\times} (a, b)$  on ait  $P(c, d)$ . Montrons donc  $P(a, b)$ .

— Si  $a = b$  :

$$(\text{pgcd } a \ b)_{(\text{code})} = a = a \wedge b.$$

— Si  $a > b$  :

$$(\text{pgcd } a \ b)_{(\text{code})} = (\text{pgcd } (a - b) \ b)$$

$$\stackrel{(\text{hypothèse})}{=} (a - b) \wedge b$$

$$\stackrel{(\text{maths})}{=} a \wedge b.$$

## 1.2 Ordre lexicographique

**Définition :** Soit  $(A, \preccurlyeq_A)$  et  $(B, \preccurlyeq_B)$  deux ensembles ordonnés, on définit alors sur  $A \times B$  l'ordre

$$(a, b) \preccurlyeq_\ell (a', b') \iff (a \prec_A a') \text{ ou } (a = a' \text{ et } b \preccurlyeq_B b').$$

**EXEMPLE :**

Dans  $(\mathbb{N}^2, \preccurlyeq_\times)$ , on cherche les éléments  $(x, y) \in \mathbb{N}^2$  tels que  $(x, y) \preccurlyeq_\ell (3, 4)$  :

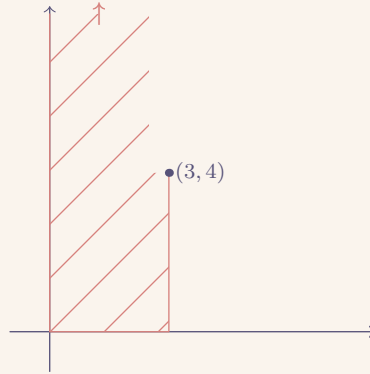


FIGURE 4 – Ordre lexicographique sur  $\mathbb{N}^2$

**Propriété :**  $\preccurlyeq_\ell$  est une relation d'équivalence.

*Preuve :*  
À faire

□

**Propriété :** Si  $\preccurlyeq_A$  est totale et  $\preccurlyeq_B$  est totale alors  $\preccurlyeq_\ell$  est totale.

*Preuve :*  
Soit  $(a, b)$  et  $(c, d) \in (A \times B)$ .

- Si  $a \prec_A c$ , alors  $(a, b) \prec_\ell (c, d)$ .
- Si  $a = c$ ,
  - si  $b \preccurlyeq_B d$  alors  $(a, b) \preccurlyeq_\ell (c, d)$ .
  - sinon  $(b \succcurlyeq_B d)$  alors  $(a, b) \succ_\ell (c, d)$ .
- si  $a \succcurlyeq_A c$  alors  $(c, d) \prec_\ell (a, b)$ .

□

**Propriété :** Si  $(A, \preccurlyeq_A)$  et  $(B, \preccurlyeq_B)$  sont bien fondés, alors  $(A \times B, \preccurlyeq_\ell)$  l'est aussi.

*Preuve :*  
À rédiger.

□



**REMARQUE :**

On peut généraliser à un ensemble  $(A^n, \preceq_\ell)$ .

Par exemple, avec  $n = 3$ , on a

$$(a, b, c) \preceq_\ell (a', b', c') \stackrel{\text{def.}}{\iff} a \prec_A a' \text{ ou } (a = a' \text{ et } b \prec_B b') \text{ ou } (a = a' \text{ et } b = b' \text{ et } c \prec_C c').$$

Même question qu'avec l'ordre produit, l'ensemble  $(A^\mathbb{N}, \preceq_\ell)$  est-il bien fondé? De même, la réponse est non, la même suite de suite est un contre-exemple.

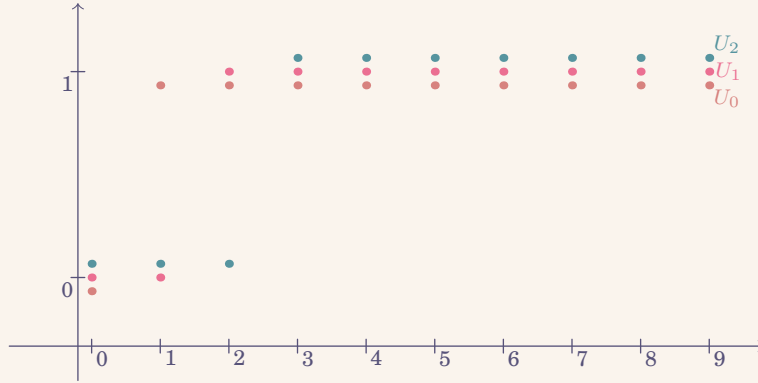


FIGURE 5 – Contre exemple :  $(A^\mathbb{N}, \preceq_\ell)$  est-il bien fondé?

L'ordre lexicographique est, comme son nom l'indique, l'ordre utilisé dans le dictionnaire. La seule différence est que l'on peut comparer des mots de longueurs différentes.

**RAPPEL :**

Si  $A$  est un ensemble, alors  $A^* = \bigcup_{n \in \mathbb{N}} A^n$ . L'ensemble  $A^*$  contient toutes les suites finies d'éléments de  $A$ .

Par exemple, avec  $A = \{0, 1\}$ , on a

$$A^* \supseteq \{(), (1), (0), (0, 1), (1, 0), (1, 1), (0, 0), (1, 1, 0), (0, 0, 0), \dots\}.$$

**Définition :** Si  $(A, \preceq_A)$  est un ensemble ordonné, on définit sur  $A^*$  :

$$(u_p)_{p \in \llbracket 1, n \rrbracket} \prec_\ell (v_p)_{p \in \llbracket 1, m \rrbracket} \stackrel{\text{def.}}{\iff} \begin{aligned} &\exists i \in \llbracket 1, \min(n, m) + 1 \rrbracket, \\ &(\forall j \in \llbracket 1, i - 1 \rrbracket, u_j = v_j) \\ &\text{et } (i = n + 1 \text{ ou } u_i \prec_A v_i). \end{aligned}$$

**Propriété :** C'est une relation d'ordre. Elle est totale si  $\preceq_A$  est totale.  $\square$

Même question avec cette nouvelle définition de l'ordre lexicographique, l'ensemble  $((A^*)^\mathbb{N}, \preceq_\ell)$  est-il bien fondé? De même, la réponse est non. Voici un contre-exemple : on considère la suite de suite  $U_0 = (1)$ ,  $U_1 = (0, 1)$ ,  $U_2 = (0, 0, 1)$ . On crée une suite infiniment strictement décroissante.

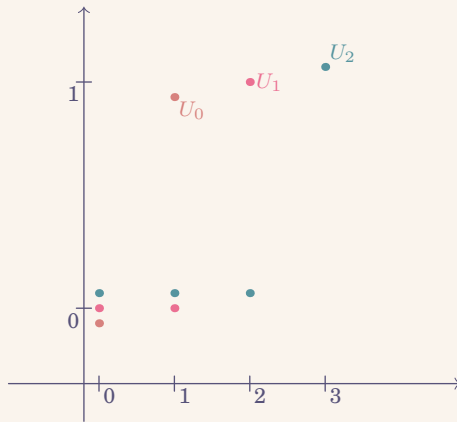


FIGURE 6 – Contre exemple :  $((A^*)^{\mathbb{N}}, \preceq_\ell)$  est-il bien fondé? (2)

```

1 let rec mystere3 m n =
2   if m = 0 then n + 1
3   else if n = 1 then mystere (n - 1) 1
4   else mystere (m - 1) (mystere m (n-1));;

```

CODE 6 – La fonction ACKERMANN

La fonction ACKERMANN utilise l'ordre lexicographique ; dans ce cas ci, l'ensemble ordonné est bien fondé. C'est comme cela que l'on a la terminaison de cette fonction.

Prenons un autre exemple :

```

1 let rec mystere n m p =
2   if m > 0 then 1 + mystere n (m - 1) p
3   else if m = 0 && n > 0 then 1 + mystere (n-1) p (p+1)
4   else 0

```

CODE 7 – Une fonction mystère (5)

À faire :

- s'assurer de la terminaison (comme celle du PGCD)
- démontrer (par induction) que

$$\forall (m, n, p) \in \mathbb{N}^3, (\text{mystere } n \ m \ p) = \frac{n(n+1)}{2} + pn + m.$$

Les preuves de correction et de terminaison sont basées sur la supposition qu'un entier en OCaml est identique à un entier mathématique.

## 2 Induction nommée

**Définition (Règle de Construction nommée) :** On appelle *règle de construction nommée* la donnée de

- un symbole  $S$ ,
- un entier  $r \in \mathbb{N}$ ,
- un ensemble non vide  $C$ .

On écrira alors cette règle

$$“S \Big|_C^r” \quad \text{ou encore} \quad “S(y, \underbrace{\square, \square, \dots, \square}_r) \text{ pour } y \in C.”$$

**REMARQUE :**

On a parfois besoin d'un ensemble  $C$  trivial (de taille 1 et contenant un objet inutile), on note alors la règle  $S \Big|_C^r$ .

**EXEMPLE :**

Les symboles sont écrits en **rouge** afin de les différencier.

$$\begin{array}{cc} \textcolor{red}{0} \Big|_{}^0 & \textcolor{red}{S} \Big|_{}^1 \\ \textcolor{red}{[]} \Big|_{}^0 & \textcolor{red}{::} \Big|_{\mathbb{N}}^1 \\ \textcolor{red}{V} \Big|_{}^0 & \textcolor{red}{N} \Big|_{\mathbb{N}}^2 \end{array}$$

**Définition :** — On appelle *règle de base* une règle de la forme  $S \Big|_C^0$ .  
— On appelle *règle d'induction* une règle de la forme  $S \Big|_C^n$ .

**Définition :** Étant donné un ensemble fini de règles  $R = \underbrace{B}_{\text{règle de base}} \cup \underbrace{I}_{\text{règle d'induction}}$  avec  $B \neq \emptyset$ , on

définit alors

$$X_0 = \left\{ (S, a) \mid S \Big|_C^r \text{ et } a \in C \right\}$$

puis, pour tout  $n \in \mathbb{N}$ ,

$$X_{n+1} = X_n \cup \left\{ (S, a, t_1, t_2, \dots, t_r) \mid S \Big|_C^r \in \underbrace{R}_C, a \in C, t_1 \in X_n, t_2 \in X_n, \dots, t_r \in X_n \right\}.$$

On appelle alors  $\bigcup_{n \in \mathbb{N}} X_n$  l'ensemble défini par induction nommée à partir des règles de  $R$ .

**REMARQUE (Notation) :**

On note un  $n$ -uplet ayant pour premier élément un symbole  $S$  puis  $n - 1$  éléments  $(a_1, \dots, a_{n-1})$ . Au lieu de  $(S, a_1, \dots, a_{n-1})$ , on note  $S(a_1, \dots, a_{n-1})$ .

**EXEMPLE :**

On pose

$$R = \left\{ A \Big|_{\mathbb{N}}^0, B \Big|_{}^1, C_{\{0,1\}}^2 \right\}.$$

À faire : À finir

**REMARQUE :**

Pour l'ensemble  $A$  défini par induction à partir de  $R = \{0|^0, S|^1\}$ , on dira plutôt

“Soit  $A$  l'ensemble défini par induction tel que  $0 \in A$  et  $\forall a \in A, S(a) \in A$ .”

**Définition :** Soit  $R$  un ensemble fini de règles et  $A$  l'ensemble défini par induction à partir de ces règles. Sur  $A$ , on définit la relation binaire  $\diamond$  par

$$x \diamond y \stackrel{\text{def.}}{\iff} y = S(\dots, x, \dots) \text{ avec } S|_C^r \in R.$$

On définit alors

$$x \preceq y \stackrel{\text{def.}}{\iff} \exists p \in \mathbb{N}, \exists (a_1, \dots, a_p) \in A^p, x \diamond a_1 \text{ et } a_1 \diamond a_2 \text{ et } \dots \text{ et } a_p \diamond y \text{ ou } x = y.$$

**Définition (hauteur) :** Soit  $R$  un ensemble fini de règles d'induction nommée définissant un ensemble  $A = \bigcup_{n \in \mathbb{N}} X_n$ . On définit alors

$$\begin{aligned} h : A &\longrightarrow \mathbb{N} \\ x &\longmapsto \min\{n \in \mathbb{N} \mid x \in X_n\}. \end{aligned}$$

**REMARQUE :**

Si  $x \in a$ , il existe alors  $n_0 \in \mathbb{N}$  tel que  $x \in X_{n_0}$  donc  $\{n \in \mathbb{N} \mid x \in X_n\} \neq \emptyset$  donc le minimum existe.

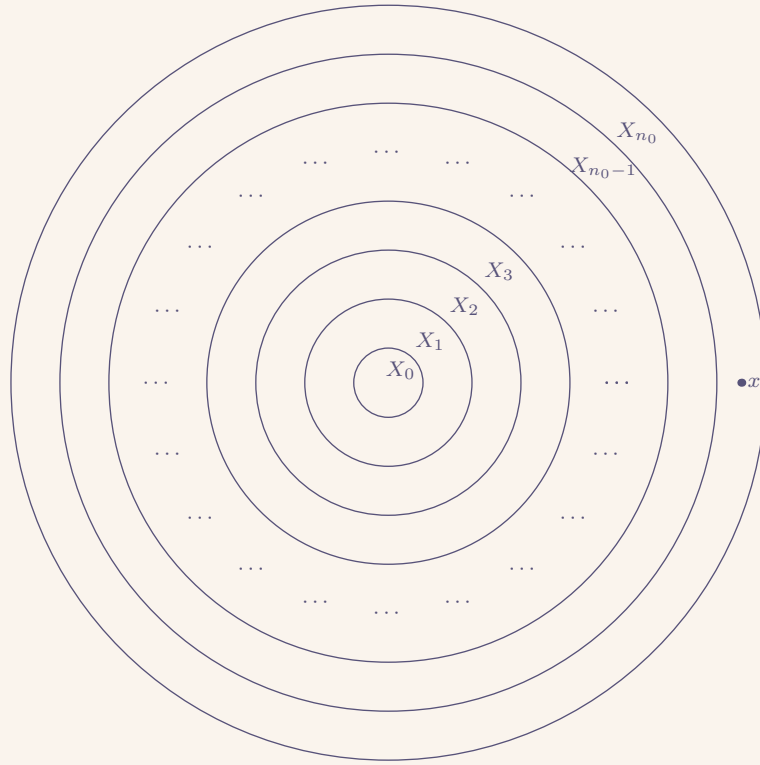


FIGURE 7 – Structure des ensembles  $X_1, \dots, X_n$

**Propriété :** Si  $x \diamond y$ , alors  $h(x) < h(y)$ .

*Preuve :*

Soit  $x \diamond y$ . Alors,  $y \in A$ . Soit  $n_0 = h(y) \neq 0$ , on a donc  $y \in X_{n_0}$ .

- Si  $y \in X_{n_0-1}$  ce qui est absurde par définition de  $h$ .
- Si  $y \in \left\{ S(a, t_1, \dots, t_r) \mid S|_C^r \in R \text{ et } a \in C \text{ et } t_1 \in X_{n_0-1} \text{ et } \dots \text{ et } t_r \in X_{n_0-1} \right\}$ . Or, soit  $i_0$  tel que  $x = t_{i_0}$  donc  $x \in X_{n_0-1}$  et donc  $h(x) \leq n_0 - 1$ .

□

**Corollaire :** Si  $n \leq y$ , alors  $h(x) = h(y) \iff x = y$  et  $h(x) < h(y) \iff x \prec y$ .

**Corollaire :** La relation  $\preccurlyeq$  est antisymétrique.

**REMARQUE :**

La relation  $\preccurlyeq$  est trivialement transitive et réflexive. Elle est donc d'ordre.

**EXEMPLE :**

On prend  $R = \{A|, B|\}$  et  $X_0 = \{A, B\}$ ,  $X_1 = X_0, \dots$  L'ensemble  $S$  défini par

induction sur  $R$  est  $\{A, B\}$ . On en déduit que  $\preccurlyeq$  n'est pas totale.

**EXEMPLE :**

On prend  $R = \{0|{}^0, S|{}^1\}$ ,  $X_0 = \{0\}$ ,  $X_1 = \{0, S(0)\}$  (on a  $0 \prec S(0)$ ),  $X_2 = \{0, S(0), S(S(0))\}$  (on a  $0 \prec S(0) \prec S(S(0))$ ). L'ensemble obtenu est

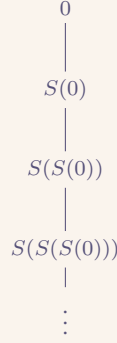


FIGURE 8 – Ensemble obtenu avec les règles  $S$  et  $0$

**EXEMPLE :**

On pose  $R = \{::|{}^1_{\mathbb{N}}, []|{}^0\}$  et  $X_0 = \{[]\}$ ,  $X_1 = \{::(0, []), [], ::(1, []), ::(2, [])\}$ , et  $X_2 = \{[], ::(0, []), ::(1, ::(0, []))\}$ . L'ensemble obtenu est

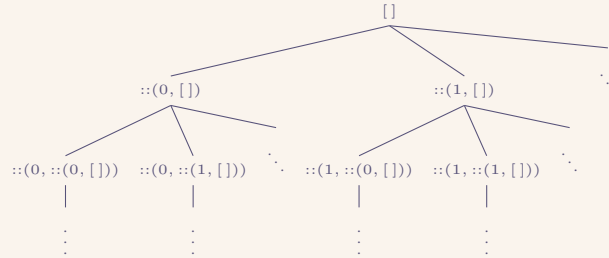


FIGURE 9 – Ensemble obtenu avec les règles  $::$  et  $[]$

**Propriété :** La relation  $\preccurlyeq$  est bien fondée.

*Preuve :*

Soit  $(x_n)_{n \in \mathbb{N}}$  une suite telle que  $x_0 \succ x_1 \succ x_2 \succ \dots \succ x_n \succ \dots$  alors

$$\underbrace{h(x_0) > h(x_1) > \dots > h(x_n) > \dots}_{\in (\mathbb{N}, \leq)}$$

Or,  $(\mathbb{N}, \leq)$  est bien fondé, c'est donc absurde. □

**REMARQUE :**

On peut faire des preuves par induction bien fondée.

**EXEMPLE :** — Soit l'ensemble trivial défini par  $\{A|^0, B|^0\}$ , le théorème est trivial.

— Soit  $\mathcal{N}$  défini par  $\mathcal{N} = \{0|^0, S|^1\}$ . Le théorème donne :

$$\text{si } P(0) \text{ vrai et } \forall n \in \mathbb{N}, (\forall p \in \mathbb{N}, p < n-1, P(\underbrace{S(S(\dots(S(0)\dots)))}_p)) \implies P(\underbrace{S(S(\dots(S(0)\dots)))}_n)$$

alors

$$\forall n \in \mathbb{N}, P(\underbrace{S(S(\dots(S(0)\dots)))}_n).$$

— Soit  $\mathcal{L}$  défini par  $\mathcal{L} = \{[]^0, ::|^1_{\mathbb{N}}\}$ . Si  $P([])$ , et  $\forall \ell \in \mathcal{L}, \forall n \in \mathbb{N}, P(\ell) \implies P(::(n, \ell))$  alors  $\forall \ell \in \mathcal{L}, P(\ell)$ .

**Propriété :** Étant donné,

- un ensemble  $A$  défini par induction à partir d'un ensemble de règles nommé  $R$ ,
- un ensemble  $\mathbb{I}$ ,
- une fonction  $f_T : C \times \mathbb{I}^r \rightarrow \mathbb{I}$  pour chaque règle  $T = S|_C^r \in R$ ,

on définit de manière unique une fonction  $f : A \rightarrow \mathbb{I}$  telle que, pour tout  $x = S(a, t_1, \dots, t_r) \in A$ , soit  $T = S|_C^r \in R$  alors  $f(x) = f_T(a, f(t_1), \dots, f(t_r))$ .

**EXEMPLE :**

Sur  $\mathcal{L}$  défini par  $\left\{ \underbrace{[]^0}_{R_1}, \underbrace{::|^1_{\mathbb{N}}}_{R_2} \right\}$ , on choisit  $\mathbb{I} = \mathbb{N}$  et

$$\begin{aligned} f_{R_1} : \underbrace{(-, -)}_{\in \text{Inutile} \times \mathbb{N}^0} &\mapsto 0 \\ f_{R_2} : \underbrace{(t, i)}_{\mathbb{N}} &\mapsto t + i. \end{aligned}$$

On définit alors la fonction

$$\begin{aligned} f : \mathcal{L} &\longrightarrow \mathbb{N} \\ ::(a_1, ::(a_2, :: \dots ::(a_n, []) \dots)) &\longmapsto \sum_{i=1}^n a_i. \end{aligned}$$

### 3 Synthèse du chapitre

#### — ORDRE —

##### Ordre bien fondé

Un ordre est *bien fondé* s'il n'existe pas de suite infiniment strictement décroissante, *i.e.* toute partie non vide de  $E$  admet un élément minimal.

Le théorème de l'induction bien fondée nous autorise à faire une récurrence forte sur un ensemble ayant un ordre bien fondé.

##### Ordre produit

On définit l'*ordre produit*  $\preceq_{\times}$  de  $(A, \preceq_A)$  et  $(B, \preceq_B)$  comme

$$(a, b) \preceq_{\times} (a', b') \stackrel{\text{def.}}{\iff} a \preceq_A a' \text{ et } b \preceq_B b'.$$

Cet ordre préserve le caractère bien fondé de  $\preceq_A$  et  $\preceq_B$ , mais pas le caractère total de la relation. On étend cet ordre à l'ensemble  $(A^n, \preceq_{\times})$ , en itérant l'ordre produit.

##### Ordre lexicographique

On définit l'*ordre lexicographique*  $\preceq_{\ell}$  de  $(A, \preceq_A)$  et  $(B, \preceq_B)$  comme

$$(a, b) \preceq_{\ell} (a', b') \stackrel{\text{def.}}{\iff} \begin{cases} (a \prec_A a') \\ \text{ou} \\ (a = a' \text{ et } b \preceq_B b'). \end{cases}$$

Cet ordre conserve le caractère bien fondé de  $\preceq_A$  et  $\preceq_B$ , ainsi que le caractère total. On peut également étendre cet ordre à un ensemble  $(A^n, \preceq_{\ell})$ . On étend également cet ordre à l'ensemble  $(A^*, \preceq_{\ell})$  comme

$$\bigcap_{A^n} u \prec_{\ell} v \stackrel{\text{def.}}{\iff} \begin{cases} \exists i \in \llbracket 1, n+1 \rrbracket, \\ (\forall j < i, u_j = v_j) \text{ et } \\ (i = n+1 \text{ ou } u_i \prec_A v_i). \end{cases}$$

#### — INDUCTION NOMMÉE —

Une *règle de construction nommée* est de la forme

$$S|_C^r \quad \text{ou} \quad S(y, \underbrace{\square, \dots, \square}_r)$$

où  $S$  est un symbole,  $r \in \mathbb{N}$  est l'arité de  $S$ , et  $C$  est un ensemble non vide. On omettra parfois l'ensemble  $C$  s'il est trivial (*i.e.* de taille 1). Une *règle de base* est de la forme  $S|_C^0$ ; une *règle d'induction* est de la forme  $S|_C^n$ .

On définit la *hauteur*  $h$  comme

$$h : \bigcup_{n \in \mathbb{N}} X_n \longrightarrow \mathbb{N} \\ x \longmapsto \min\{n \in \mathbb{N} \mid x \in X_n\}.$$

On définit la relation  $\preceq$  bien fondée telle que  $x \preceq y$  si  $x$  est défini à partir de  $y$ . Si  $x \preceq y$ , alors  $h(x) \leq h(y)$ .

Un ensemble  $\bigcup_{n \in \mathbb{N}} X_n$  est dit *défini par induction nommée* à partir des règles  $R$ , si  $X_0 = \{S(a) \mid S_C^0 \in R, a \in C\}$ , et

$$X_{n+1} = X_n \cup \{S(a, t_1, \dots, t_r) \mid S_C^r \in R, a \in C, (t_1, \dots, t_r) \in (X_n)^r\}.$$