

**Définition 6.16** - - *prédécesseur, successeur immédiats*

Soit  $E$  un ensemble ordonné non vide et  $e \in E$ . On dit que :

- $p \in E$  est un *prédécesseur immédiat* de  $e$  si  $p < e$  et il n'existe pas d'élément  $a \in E$  tel que  $p < a < e$
- $s \in E$  est *successeur immédiat* de  $e$  si  $e < s$  et il n'existe pas d'élément  $a \in E$  tel que  $e < a < s$

**Exemple 6.17** - - *prédécesseur, successeur immédiats*

$\forall n \in \mathbb{N}$ ,  $n + 1$  est le successeur immédiat de  $n$  pour l'ordre usuel.

$\forall n \in \mathbb{N}^*$ ,  $n - 1$  est le prédécesseur immédiat de  $n$  pour l'ordre usuel.

0 n'a pas de prédécesseur (en particulier immédiat) pour l'ordre usuel.

**Exemple 6.18** - - *prédécesseur, successeur immédiats*

Dans  $\mathbb{R}$  muni de l'ordre usuel, aucun élément  $e$  n'a de prédécesseur (resp.successeur) immédiat puisque si  $a < e$  alors en particulier  $a < \frac{a+e}{2} < e$ .

**Définition 6.19** - - *éléments minimal, maximal*

Soit  $E$  un ensemble ordonné non vide et  $e \in E$ . On dit que

- $e$  est un *élément minimal* de  $E$  s'il n'admet pas de prédécesseur.
- $e$  est un *élément maximal* de  $E$  s'il n'admet pas de successeur.

**Exemple 6.20** - - *éléments minimal, maximal*

Soit  $E$  un ensemble. L'ensemble  $A = \mathcal{P}(E) \setminus \{\emptyset\}$  des parties non vides de  $E$  muni de l'inclusion et ordonné. Si  $E \neq \emptyset$ ,  $E$  est l'élément maximal de  $A$ , et  $\forall e \in E$ ,  $\{e\}$  est un élément maximal de  $A$ .

**Remarque 6.21** - - *sur le dernier exemple*

L'ensemble précédent montre en particulier qu'un ensemble peut tout à fait avoir plusieurs éléments minimaux ou maximaux.

**Définition 6.22** - - *plus grand, plus petit éléments*

Soit  $E$  un ensemble ordonné non vide et  $e \in E$ . On dit que :

- $e$  est le *plus grand élément* de  $E$  si  $\forall x \in E, x \leq e$ .
- $e$  est le *plus petit élément* de  $E$  si  $\forall x \in E, x \geq e$ .

**Démonstration**

On démontre l'unicité.

Supposons par l'absurde, qu'il n'y a pas unicité du plus petit élément.

Soit  $e$  et  $e'$  deux plus petits éléments distincts de  $E$ .

Alors, par définition, ( $e$  est un plus petit élément  $E$ ,  $e \leq e'$ ). de même,  $e' \leq e$ .

Par antisymétrie de  $\leq$ ,  $e = e'$ . Absurde.

On montre de même l'unicité du plus grand élément, s'il existe.

**Définition 6.23** - - *ordre bien fondé*

Soit  $(E, \leq)$  un ensemble ordonné. On dit que  $\leq$  est un *ordre bien fondé* si toute partie non vide de  $E$  admet au moins un élément minimal.

**Exemple 6.24** - - *ordres bien fondés*

l'ordre usuel sur l'ensemble  $\mathbb{N}$  des entiers naturels est bien fondé.

l'inclusion sur les parties d'un ensemble fini est bien fondée.

la relation de divisibilité sur l'ensemble  $\mathbb{N}^*$  est un ordre bien fondé.

**Exemple 6.24** - - *ordres non bien fondés*

l'ordre usuel sur  $\mathbb{Z}$  ou sur  $\mathbb{R}_+$

L'inclusion sur les parties d'un ensemble infini n'est pas bien fondée.

**Proposition 6.26** - - *"bonne fondation" de l'ordre lexicographique*

Soit  $(A, \leq_A)$  et  $(B, \leq_B)$  deux ensembles ordonnés. Si  $\leq_A$  et  $\leq_B$  sont bien fondées, l'ordre lexicographique défini sur  $A \times B$  est bien fondé.

### Démonstration

Soit  $X$  une partie non vide de  $A \times B$ . Montrons qu'elle admet un élément minimal.

On note  $A_X = \{a \in A, \exists b \in B, (a, b) \in X\}$ .  $X$  est non vide, donc  $A_X$  l'est également.

De plus, comme  $\leq_A$  est bien fondé,  $A_X$  admet un élément minimal.

Soit donc  $a_0 \in A$  un élément minimal de  $A_X$ . On considère alors l'ensemble  $B_0 = \{b \in B, (a_0, b) \in X\}$ .

Par définition de  $a_0$ ,  $B_0$  est non vide, alors,  $\leq_B$  étant aussi bien fondé,  $B_0$  admet un élément minimal  $b_0$ . l'élément  $x_0 = (a_0, b_0)$  est alors un élément minimal de  $X$ .

En effet, Soit  $(a, b) \in X$  tel que  $(a, b) \leq (a_0, b_0)$  i.e. tel que  $a < a_0$  ou  $(a = a_0 \text{ et } b \leq_B b_0)$ .

$a \in A_X$  donc, par minimalité de  $a_0$ ,  $a \not< a_0$ , on a donc  $a = a_0$  et  $b \leq_B b_0$ .

De même,  $b \in B_0$  puisque  $(a, b) = (a_0, b)$ . Par minimalité de  $b_0$  dans  $B_0$ , on a donc  $b = b_0$  d'où  $(a, b) = (a_0, b_0)$ .

### Proposition 6.27 - - "bonne fondation" de l'ordre produit

Soit  $((E_i, \leq_i))_{i \in \llbracket 1, n \rrbracket}$  une famille finie d'ensembles munis d'ordres bien fondés. ( $n \geq 2$ ). L'ordre produit défini sur  $\prod_{i=1}^n E_i = E_1 \times \dots \times E_n$  est bien fondé.

### Démonstration

Soit  $A$  une partie non vide de  $E_1 \times \dots \times E_n$ .

On pose  $A_1 = \{a_1 \in E_1, \exists (x_1, \dots, x_n) \in A^n, x_1 = a_1\}$ . Comme  $A$  est non vide,  $A_1$  est une partie non vide de  $E_1$  qui admet donc un élément minimal  $m_1$ .

On pose  $A_2 = \{a_2 \in E_2, \exists (x_1, \dots, x_n) \in A^n, x_2 = a_2 \text{ et } x_1 = m_1\}$ . Comme  $A$  est non vide,  $A_2$  est une partie non vide de  $E_2$  qui admet donc un élément minimal  $m_2$ .

On construit ainsi  $n$  ensembles non vides définis pour tout  $i \in \mathbb{N}$  par :

$$\begin{cases} A_{i+1} = \{a_{i+1} \in E_{i+1}, \exists (x_1, \dots, x_n) \in A^n, \forall j \in \llbracket 1, i \rrbracket, x_j = m_j \text{ et } x_{i+1} = a_{i+1}\} \\ m_i \text{ est un élément minimal de } A_i, \forall i \in \llbracket 1, n \rrbracket \end{cases}$$

L'élément  $m = (m_1, \dots, m_n)$  est alors, par construction, un élément minimal de  $A$ .

### Remarque 6.28

Si  $E$  est muni d'un ordre total et bien fondé, alors toute partie non vide de  $E$  admet un plus petit élément. On parle alors de bon ordre et d'ensemble bien ordonné.

Définition 6.29 Soit  $E$  un ensemble. On appelle prédicat sur  $E$  toute propriété  $P$  dépendant d'éléments de  $E$ . Lorsque  $P$  dépend de  $n$  paramètres, on dit que  $P$  est d'arité  $n$ . On note alors  $\forall (x_1, \dots, x_n) \in E^n$ .

- $P(x_1, \dots, x_n)$  lorsque la propriété est vraie.
- $\neg P(x_1, \dots, x_n)$  lorsque la propriété est fausse.

Remarque 6.30 Une relation binaire est en fait un prédicat d'arité 2.

Théorème 6.31 Soit  $(E, \leq)$  un ensemble ordonné. Les propositions suivantes sont équivalentes :

1.  $\leq$  est un ordre bien fondé.
2. Il n'existe pas de suite infinie strictement décroissante d'éléments de  $E$ .
3. Pour tout prédicat  $P$  sur  $E$ , si :

$$\forall (x, y) \in E^2, x > y \implies P(x)$$

### Démonstration

- (1)  $\implies$  (2) : Supposons que  $\leq$  est un ordre bien fondé, et par l'absurde, que  $(x_n)_{n \in \mathbb{N}}$  est une suite infinie strictement décroissante d'éléments de  $E$ . Alors l'ensemble non vide  $\{x_n, n \in \mathbb{N}\} \subset E$  admet un élément minimal  $x_k$  ; Ainsi, par décroissance stricte de  $(x_n)$ ,  $x_{k+1} < x_k$ , ce qui contredit la minimalité de  $x_k$
- (2)  $\implies$  (3) : Soit  $P$  un prédicat sur  $E$ . On suppose que

$$(\forall (x, y) \in E^2, y < x \implies P(y)) \implies P(x)$$

. On note (A) cette propriété. Montrons que

$$\forall x \in E, P(x)$$

. Pour cela, on considère l'ensemble  $A = \{x \in E, \neg P(x)\} \subset E$ .

Supposons, par l'absurde que  $A$  est non vide : soit  $x_0$  tel que  $\neg P(x_0)$ .

Alors par contraposée de (A), il existe  $x_1 \in E$  tel que  $x_1 < x_0$  et  $\neg P(x_1)$ .

En itérant ce raisonnement, on construit une suite infinie, strictement décroissante d'éléments  $x_i \in E$  telle que  $\forall i, \neg P(x_i)$ , ce qui contredit la propriété (2).

- (3)  $\implies$  (1) : Soit  $A$  une partie non vide de  $E$ . on note  $P(x)$  le prédicat  $x \notin A$ . Puisque  $A \neq \emptyset$ , la proposition " $\forall x \in E, P(x)$ " est fausse.

Par contraposée de la proposition (3), on en déduit que :

$$\exists m \in E, y < m, P(y) \text{ et } \neg P(m)$$

Donc on choisit  $m \in A$  tel que

$$\forall y \in E, y < m \implies y \notin A$$

$m$  est alors minimal dans  $A$

donc  $\leq$  est bien fondée.

**Remarque 6.32** - À propos du théorème 6.31

La propriété (2) peut permettre de justifier la terminaison de d'un algorithme en utilisant un ordre produit, par exemple, bien fondé, sur l'ensemble des valeurs que peuvent prendre les variables de l'algorithme (on retrouve la notion de variant).

Remarque 6.33 : La proposition (3) définit un principe de récurrence sur n'importe quel ensemble d'un ordre bien fondé. dans le cas où  $E = \mathbb{N}$ , on retrouve le principe de récurrence forte.

**Définition 6.35** - *stabilité*

Soit  $E$  un ensemble et  $f : E^n \rightarrow E$  (pour  $n \in \mathbb{N}$ ) une application.  
On dit que  $X \subset E$  est *stable* par  $f$  si

$$\forall x_1, \dots, x_n \in X, f(x_1, \dots, x_n) \in X$$

**Définition 6.36** - *ensemble inductif*

Soit  $E$  un ensemble. On se donne au plus :

- une partie  $B \subset E$  correspondant aux éléments de base
- un ensemble de *constructeurs*  $\mathcal{K} \subset \bigcup_{n \in \mathbb{N}^*} \mathcal{F}(E^n, E)$  i.e. un ensemble d'applications  $f : E^n \rightarrow E$  où  $n \in \mathbb{N}^*$  est appelé *arité* de  $f$ .

L'*ensemble inductif* associé est alors le plus petit ensemble  $X \subset E$  contenant  $B$  est stable par les constructeurs de  $\mathcal{K}$ .

**Remarque 6.37** - *Vocabulaire*

L'ensemble  $B$  peut être défini par un ensemble d'*assertions* et l'ensemble  $\mathcal{K}$  par des *règles d'inférence*

**Exemple 6.38** - *arithmétique de Péano*

L'ensemble  $\mathbb{N}$  des entiers naturels est défini de manière inductive :

- $\{0\} \subset \mathbb{N}$  (assertion)
- si  $n \in \mathbb{N}$ , alors son successeur  $S(n)$  appartient à  $\mathbb{N}$  (règle d'inférence)

### Remarque 6.39

L'ensemble  $\mathcal{P}(E)$  des parties d'un ensemble  $E$  n'est pas nécessairement bien ordonné, l'existence d'un ensemble inductif n'est alors pas immédiate et découle de la propriété suivante.

#### Proposition 6.40 - Caractérisation d'un ensemble inductif

Soit  $E$  un ensemble. Soit  $B \subset E$  un ensemble d'éléments de base et  $\mathcal{K}$  un ensemble de constructeurs sur  $E$ . On note  $A$  l'ensemble des parties de  $E$  contenant  $B$  et stables par les constructeurs de  $\mathcal{K}$ . L'ensemble inductif défini par  $B$  et  $\mathcal{K}$  est alors :

$$X = \bigcap_{Y \in A} Y$$

#### Démonstration

— On a :

$$\forall Y \in A, B \text{ Donc } B \subset \bigcap_{Y \in A} Y$$

— Montrons que  $\bigcap_{Y \in A} Y$  est stable par les constructeurs de  $\mathcal{K}$ .

Soit  $n \in \mathbb{N}^*$ ,  $f : E^{\mathbb{N}} \rightarrow E$  un constructeur de  $\mathcal{K}$  et  $x_1, \dots, x_n \in \bigcap_{Y \in A} Y$

Soit  $Y \in A$ . Alors  $x_1, \dots, x_n \in Y$ . Comme, par définition de  $A$ ,  $Y$  est stable par  $f$ ,

$$f(x_1, \dots, x_n) \in Y$$

et ce, indépendamment du choix de  $Y$ . On en déduit que  $f(x_1, \dots, x_n) \in \bigcap_{Y \in A} Y$ , i.e. que  $\bigcap_{Y \in A} Y$  est stable par  $f$ .

— Les deux points précédents justifient que  $\bigcap_{Y \in A} Y \in A$ .

De plus, pour tout  $X \in A$ ,  $\bigcap_{Y \in A} Y \subset X$ .

$\bigcap_{Y \in A} Y$  est donc le plus petit élément de  $A$ , il s'agit ainsi de l'ensemble inductif recherché.

#### Exemple 6.41

On considère l'ensemble  $E = \mathcal{P}(\mathbb{R})$ . On pose  $B = \{[0, 2], [1, 3]\}$  et  $\mathcal{K} = \{\cap, \cup\}$ . L'ensemble inductif associé est alors  $\{[0, 2], [1, 3], [0, 3], [1, 2]\}$ .

**Proposition 6.42** - *Caractérisation récursive d'un ensemble inductif*

Soit  $E$  un ensemble,  $B \subset E$  un ensemble d'éléments de base et  $\mathcal{K}$  un ensemble de constructeurs sur  $E$ .

On définit la suite  $(Y_n)$  de parties de  $E$  par : 
$$\begin{cases} Y_0 = B \\ \forall n \in \mathbb{N}, Y_{n+1} = Y_n \cup \mathcal{K}(Y_n) \end{cases}$$

L'ensemble inductif défini par  $B$  et  $\mathcal{K}$  et alors :

$$X = \bigcup_{n \in \mathbb{N}} Y_n$$

**Démonstration**

Soit  $X$  l'ensemble inductif défini par  $B$  et  $\mathcal{K}$ .

$B = Y_0 \subset \bigcup_{n \in \mathbb{N}} Y_n$ . Montrons que  $\bigcup_{n \in \mathbb{N}} Y_n$  est stable par les constructeurs de  $\mathcal{K}$ . Soit  $f : E^p \rightarrow E$  ( $p \in \mathbb{N}^*$ ) un constructeur de  $\mathcal{K}$  et  $x_1, \dots, x_p \in \bigcup_{n \in \mathbb{N}} Y_n$ .

Soit  $n_1, \dots, n_p \in \mathbb{N}$  tels que

$$\forall i \in \llbracket 1, p \rrbracket, x_i \in Y_{n_i}$$

On note  $n_0 = \max\{n_i, i \in \llbracket 1, p \rrbracket\}$ . Alors,

$$\forall i \in \llbracket 1, p \rrbracket$$

,  $x_i \in Y_{n_0}$  ( $\supset Y_{n_i}$ ).

Par construction, on a donc  $f(x_1, \dots, x_p) \in \mathcal{K}(Y_{n_0}) \subset Y_{n_0+1} \subset \bigcup_{n \in \mathbb{N}} Y_n$ .

Par ailleurs, par récurrence immédiate,

$$\forall n \in \mathbb{N}, Y_n \subset X$$

$\bigcup_{n \in \mathbb{N}} Y_n$  est donc une partie de  $X$ , contenant  $B$ , et stable par les constructeurs de  $\mathcal{K}$ . Or  $X$  est le plus petit ensemble vérifiant cette propriété, on en déduit que  $X = \bigcup_{n \in \mathbb{N}} Y_n$ .

**Remarque 6.43**

Cette dernière propriété montre que l'ensemble  $E$  n'a finalement pas d'importance. En pratique, on définit le plus souvent des objets de manière inductive en indiquant uniquement un ensemble de base et des constructeurs.

**Exemple 6.44** - *Définition inductive des listes OCaml*

`[]` est la liste vide

si  $e$  est un élément  $q$  une liste, alors  $e :: q$  est une liste d'élément en tête  $e$ . C'est cette construction qui justifie le filtrage de motif sur les listes OCaml.

**Définition 6.45** - *non ambiguïté d'une définition*

Soit  $X \subset E$  un ensemble défini de manière inductive par un ensemble de base  $B$  et un ensemble de constructeurs  $\mathcal{K}$ .

On dit que la définition de  $X$  est *non ambiguë* si es deux propriétés suivantes sont vérifiées :

- $\forall f \in \mathcal{K}(\text{arité } p), \forall x_1, \dots, x_p \in X, f(x_1, \dots, x_p) \notin B$
- $\forall f, g \in \mathcal{K}(\text{arité } p), \forall x_1, \dots, x_p, y_1, \dots, y_p \in X, (f(x_1, \dots, x_p) = g(y_1, \dots, y_p)) \implies (f = g \text{ et } \forall i \in \llbracket 1, p \rrbracket, x_i = y_i)$

La définition non ambiguë d'un ensemble inductif permet de définir des fonctions de manière inductive sur cet ensemble.

**Exemple 6.46** - *Définition inductive de la taille d'une liste OCaml*

$$\begin{cases} |\square| = 0 \\ |e :: q| = 1 + |q| \end{cases}$$

**Définition 6.47** - *ordre induit*

Dans un ensemble inductif  $X \subset E$ , l'*ordre induit*  $<$  est défini à partir des constructeur (ou règles d'inférence) : pour  $n \in \mathbb{N}^*$ , si  $f : E^n \rightarrow E$  est un constructeur et étant donné les éléments  $x_1, \dots, x_n \in E$ , alors :

$$\forall i \in \llbracket 1, p \rrbracket, x_i < f(x_1, \dots, x_p)$$

**Théorème 6.48** - *preuve par induction structurelle*

Soit  $X \subset E$  un ensemble défini de manière inductive par un ensemble de base  $B$  et un ensemble de constructeurs  $\mathcal{K}$ . Soit  $\mathcal{P}$  un prédicat sur  $E$ .

si on a :

- $\forall b \in B, \mathcal{P}(b)$
- $\forall f : E^p \rightarrow E \in \mathcal{K}, \forall x_1, \dots, x_p \in X, (\forall i \in \llbracket 1, p \rrbracket, \mathcal{P}(x_i)) \implies (\mathcal{P}(f(x_1, \dots, x_p)))$

Alors,

$$\forall x \in X, \mathcal{P}(x)$$



### Démonstration

On définit la suite  $(Y_n)$  par 
$$\begin{cases} Y_0 = B \\ \forall n \in \mathbb{N}, Y_{n+1} = Y_n \cup \mathcal{K}(Y_n) \end{cases}$$

On a vu (6.42) qu'alors  $X = \bigcup_{n \in \mathbb{N}} Y_n$ .

Pour tout  $x \in X$ , on définit alors la *hauteur* de  $x$  :  $h(x) = \min\{n \in \mathbb{N}, x \in Y_n\}$ .

On suppose que :

1.  $\forall b \in B, \mathcal{P}$
2.  $\forall f \in \mathcal{K}(\text{arité } p), \forall x_1, \dots, x_p \in X, (\forall i \in \llbracket 1, p \rrbracket, \mathcal{P}(x_i)) \implies (\mathcal{P}(f(x_1, \dots, x_p)))$

Montrons par récurrence forte sur la hauteur  $n$  des éléments de  $X$  que  $\forall x \in X, \mathcal{P}(x)$ .

- Si  $n = 0$  :  
Soit  $x \in X$  tel que  $h(x) = 0$ , i.e. tel que  $x \in Y_0 = B$ . Alors d'après la propriété (1),  $\mathcal{P}(x)$
- Hérédité : Soit  $n \in \mathbb{N}$  fixé. On suppose que

$$\forall y \in X, h(y) \leq n \implies \mathcal{P}(y)$$

Soit  $x \in X$  tel que  $h(x) = n + 1$

Montrons que  $\mathcal{P}(x)$  est encore vraie.

Par définition de la hauteur,  $x \in Y_{n+1} = Y_n \cup \mathcal{K}(Y_n)$ . De plus,  $x \notin Y_n$  (sinon on aurait  $h(x) \leq n$ ).

Alors,  $x \in \mathcal{K}(Y_n)$ . Soit donc  $f \in \mathcal{K}(\text{arité } p)$  et  $y_1, \dots, y_p \in Y_n$  tels que  $x = f(y_1, \dots, y_p)$ .

Alors  $\forall i \in \llbracket 1, p \rrbracket, h(y_i) \leq n$  donc  $\mathcal{P}(y_i)$  est vrai par hypothèse de récurrence.

D'après la proposition (2), on a donc  $\mathcal{P}(x)$ . Ainsi, par récurrence forte sur  $n$ ,

$$\forall n \in \mathbb{N}, \forall x \in X \text{ de hauteur } n, \mathcal{P}(x)$$

### Exemple 6.49

Montrons la correction de la fonction Ocaml `insert` : `'a`  $\rightarrow$  `'a list`  $\rightarrow$  `'a list` ci-dessous dont la spécification est la suivante :

Soit  $E$  un ensemble ordonné.

Si  $x \in E$  et  $u$  une liste triée d'éléments de  $E$ , alors `insert`  $x$   $u$  est une permutation triée de  $x::u$ .

```
let rec insert x u = match u with | [] -> [x] | e::q -> if x > e then e::(insert x q) else x::u
```

On raisonne par induction structurelle sur la liste  $u$  passée en argument.

- Si  $u = []$ , alors pour tout  $x$ , `insert`  $x$   $u$  est la liste  $[x]$ , la spécification est donc bien respectée
- Si  $u = e::q$ , a fortiori,  $q$  est triée. Supposons que pour tout  $x$ , `insert`  $x$   $q$  est une permutation triée de  $x::q$ .

Soit  $x$  un élément comparable aux éléments de  $u$ .

- Si  $x > e$ , alors comme  $u$  est triée, tous les éléments de  $q$  sont supérieurs à  $e$ .  
`insert`  $x$   $q$  est donc, par hypothèse d'induction, une permutation triée de  $x::q$ , dont tous les éléments sont supérieurs à  $e$ .  
On en déduit que  $e::(\text{insert } x \text{ } q)$  est une permutation triée de  $e::x::q$ , et donc de  $x::u$ .  
La spécification est donc bien respectée
- Sinon,  $x \leq e$ , et comme  $u = e::q$  est triée,  $x::u$  est encore triée.

Ainsi, par induction structurelle sur  $u$ , pour toute liste triée  $u$  et tout élément  $x$  compatible, `insert`  $x$   $u$  est une permutation triée de  $x::u$ . D'où la correction de la fonction `insert`.