

Variant

On rappelle que pour montrer la terminaison d'un programme, on utilise la notion de variant que nous avons défini comme une quantité qui dépend des variables du programme et :

- ne prend que des valeurs entières,

Variant

On rappelle que pour montrer la terminaison d'un programme, on utilise la notion de variant que nous avons défini comme une quantité qui dépend des variables du programme et :

- ne prend que des valeurs entières,
- positives

Variant

On rappelle que pour montrer la terminaison d'un programme, on utilise la notion de variant que nous avons défini comme une quantité qui dépend des variables du programme et :

- ne prend que des valeurs entières,
- positives
- et est *strictement* décroissante à chaque itération.

Variant

On rappelle que pour montrer la terminaison d'un programme, on utilise la notion de variant que nous avons défini comme une quantité qui dépend des variables du programme et :

- ne prend que des valeurs entières,
- positives
- et est *strictement* décroissante à chaque itération.

Comme il n'existe pas de suites d'entiers positifs strictement décroissante, l'existence d'un variant prouve la terminaison.

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :
 - 1 **réfléxive** : pour tout $x \in E, x \mathcal{R} x$.

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :
 - 1 **réfléxive** : pour tout $x \in E, x \mathcal{R} x$.
 - 2 **antisymétrique** : pour tout $(x, y) \in E^2$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :
 - 1 **réfléxive** : pour tout $x \in E, x \mathcal{R} x$.
 - 2 **antisymétrique** : pour tout $(x, y) \in E^2$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$
 - 3 **transitive** : pour tout $(x, y, z) \in E^3$, si $x \mathcal{R} y$ et $y \mathcal{R} z$ alors $x \mathcal{R} z$.

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :
 - 1 **réfléxive** : pour tout $x \in E, x \mathcal{R} x$.
 - 2 **antisymétrique** : pour tout $(x, y) \in E^2$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$
 - 3 **transitive** : pour tout $(x, y, z) \in E^3$, si $x \mathcal{R} y$ et $y \mathcal{R} z$ alors $x \mathcal{R} z$.

On dit alors que (E, \mathcal{R}) est un ensemble ordonné.

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :
 - 1 **réfléxive** : pour tout $x \in E, x \mathcal{R} x$.
 - 2 **antisymétrique** : pour tout $(x, y) \in E^2$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$
 - 3 **transitive** : pour tout $(x, y, z) \in E^3$, si $x \mathcal{R} y$ et $y \mathcal{R} z$ alors $x \mathcal{R} z$.

On dit alors que (E, \mathcal{R}) est un ensemble ordonné.

- Soit (E, \mathcal{R}) un ensemble ordonné, \mathcal{R} est **total** si pour tout $x, y \in E^2$, $x \mathcal{R} y$ ou $y \mathcal{R} x$, sinon \mathcal{R} est **partiel**.

Définition : ensemble ordonné

- Une **relation binaire** sur un ensemble E est un sous ensemble de $E \times E$.
- Une **relation d'ordre** \mathcal{R} sur un ensemble E est une relation binaire sur E ayant les 3 propriétés suivantes :
 - ① *réfléxive* : pour tout $x \in E, x \mathcal{R} x$.
 - ② *antisymétrique* : pour tout $(x, y) \in E^2$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$
 - ③ *transitive* : pour tout $(x, y, z) \in E^3$, si $x \mathcal{R} y$ et $y \mathcal{R} z$ alors $x \mathcal{R} z$.

On dit alors que (E, \mathcal{R}) est un ensemble ordonné.

- Soit (E, \mathcal{R}) un ensemble ordonné, \mathcal{R} est **total** si pour tout $x, y \in E^2$, $x \mathcal{R} y$ ou $y \mathcal{R} x$, sinon \mathcal{R} est **partiel**.

Remarques

- On notera \preccurlyeq une relation d'ordre dans un cadre général, afin de la différencier de l'ordre usuel \leq sur les ensembles de nombres.
- A toute relation d'ordre \preccurlyeq est associé l'*ordre strict* correspondant défini par $x \prec y$ si et seulement si $x \preccurlyeq y$ et $x \neq y$.

Exemples

- (\mathbb{N}, \leq) est un ensemble ordonné.

Exemples

- (\mathbb{N}, \leq) est un ensemble ordonné.
- En notant $\mathcal{P}(\mathcal{E})$ l'ensemble des parties d'un ensemble \mathcal{E} , $(\mathcal{P}(\mathcal{E}), \subset)$ est un ensemble ordonné.

Exemples

- (\mathbb{N}, \leq) est un ensemble ordonné.
- En notant $\mathcal{P}(\mathcal{E})$ l'ensemble des parties d'un ensemble \mathcal{E} , $(\mathcal{P}(\mathcal{E}), \subset)$ est un ensemble ordonné.
- Soit $\mathcal{E} = \{a, b, c\}$, la relation $\mathcal{R} = \{(a, a), (a, b), (b, c), (b, b)\}$ est-elle une relation d'ordre? Sinon, quels couples faut-il ajouter à \mathcal{R} pour qu'elle le devienne?

Exemples

- (\mathbb{N}, \leq) est un ensemble ordonné.
- En notant $\mathcal{P}(\mathcal{E})$ l'ensemble des parties d'un ensemble \mathcal{E} , $(\mathcal{P}(\mathcal{E}), \subset)$ est un ensemble ordonné.
- Soit $\mathcal{E} = \{a, b, c\}$, la relation $\mathcal{R} = \{(a, a), (a, b), (b, c), (b, b)\}$ est-elle une relation d'ordre? Sinon, quels couples faut-il ajouter à \mathcal{R} pour qu'elle le devienne?

Clôture

Soit \mathcal{R} une relation binaire sur un ensemble E ,

Exemples

- (\mathbb{N}, \leq) est un ensemble ordonné.
- En notant $\mathcal{P}(\mathcal{E})$ l'ensemble des parties d'un ensemble \mathcal{E} , $(\mathcal{P}(\mathcal{E}), \subset)$ est un ensemble ordonné.
- Soit $\mathcal{E} = \{a, b, c\}$, la relation $\mathcal{R} = \{(a, a), (a, b), (b, c), (b, b)\}$ est-elle une relation d'ordre? Sinon, quels couples faut-il ajouter à \mathcal{R} pour qu'elle le devienne?

Clôture

Soit \mathcal{R} une relation binaire sur un ensemble E ,

- la **clôture réflexive** de \mathcal{R} est la plus petite relation réflexive contenant \mathcal{R} .

Exemples

- (\mathbb{N}, \leq) est un ensemble ordonné.
- En notant $\mathcal{P}(\mathcal{E})$ l'ensemble des parties d'un ensemble \mathcal{E} , $(\mathcal{P}(\mathcal{E}), \subset)$ est un ensemble ordonné.
- Soit $\mathcal{E} = \{a, b, c\}$, la relation $\mathcal{R} = \{(a, a), (a, b), (b, c), (b, b)\}$ est-elle une relation d'ordre? Sinon, quels couples faut-il ajouter à \mathcal{R} pour qu'elle le devienne?

Clôture

Soit \mathcal{R} une relation binaire sur un ensemble E ,

- la **clôture réflexive** de \mathcal{R} est la plus petite relation réflexive contenant \mathcal{R} .
- la **clôture transitive** de \mathcal{R} est la plus petite relation transitive contenant \mathcal{R} .

⚠ La clôture réflexive-transitive d'une relation \mathcal{R} , notée \mathcal{R}^* , n'est pas forcément une relation d'ordre (car l'antisymétrie n'est pas garantie).

Exemples

- La clôture réflexive d'un ordre strict est l'ordre associé.

Exemples

- La clôture réflexive d'un ordre strict est l'ordre associé.
- Soit \mathcal{R} la relation sur \mathbb{N} définie par $\mathcal{R} = \{(n, n + 1), n \in \mathbb{N}\}$ alors \mathcal{R}^* est un ordre (c'est l'ordre usuel sur \mathbb{N}).

Exemples

- La clôture réflexive d'un ordre strict est l'ordre associé.
- Soit \mathcal{R} la relation sur \mathbb{N} définie par $\mathcal{R} = \{(n, n + 1), n \in \mathbb{N}\}$ alors \mathcal{R}^* est un ordre (c'est l'ordre usuel sur \mathbb{N}).
- Soit \mathcal{R} la relation sur $\{0, 1, 2\}$ définie par $\mathcal{R} = \{(0, 1), (1, 2), (2, 0)\}$, \mathcal{R}^* est-elle antisymétrique ?

Définitions : prédécesseur, successeur

Soit (E, \preccurlyeq) un ensemble ordonné et x, y deux éléments de E

Définitions : prédécesseur, successeur

Soit (E, \preccurlyeq) un ensemble ordonné et x, y deux éléments de E

- Si $x \prec y$, x est un **prédécesseur** de y (et y est un **successeur** de x).

Définitions : prédécesseur, successeur

Soit (E, \preccurlyeq) un ensemble ordonné et x, y deux éléments de E

- Si $x \prec y$, x est un **prédécesseur** de y (et y est un **successeur** de x).
- Si $x \prec y$ et s'il n'existe pas d'éléments $z \in E$ tel que $x \prec z \prec y$, on dit que y est un **successeur immédiat** de x (ou que x est un **prédécesseur immédiat** de y).

Définitions : prédécesseur, successeur

Soit (E, \preccurlyeq) un ensemble ordonné et x, y deux éléments de E

- Si $x \prec y$, x est un **prédécesseur** de y (et y est un **successeur** de x).
- Si $x \prec y$ et s'il n'existe pas d'éléments $z \in E$ tel que $x \prec z \prec y$, on dit que y est un **successeur immédiat** de x (ou que x est un **prédécesseur immédiat** de y).

Exemple

Déterminer, lorsqu'ils existent les successeurs et prédécesseur immédiat de x dans les cas suivants

Définitions : prédécesseur, successeur

Soit (E, \preccurlyeq) un ensemble ordonné et x, y deux éléments de E

- Si $x \prec y$, x est un **prédécesseur** de y (et y est un **successeur** de x).
- Si $x \prec y$ et s'il n'existe pas d'éléments $z \in E$ tel que $x \prec z \prec y$, on dit que y est un **successeur immédiat** de x (ou que x est un **prédécesseur immédiat** de y).

Exemple

Déterminer, lorsqu'ils existent les successeurs et prédécesseur immédiat de x dans les cas suivants

- (\mathbb{N}, \leq) et $x \in \mathbb{N}$

Définitions : prédécesseur, successeur

Soit (E, \preceq) un ensemble ordonné et x, y deux éléments de E

- Si $x \prec y$, x est un **prédécesseur** de y (et y est un **successeur** de x).
- Si $x \prec y$ et s'il n'existe pas d'éléments $z \in E$ tel que $x \prec z \prec y$, on dit que y est un **successeur immédiat** de x (ou que x est un **prédécesseur immédiat** de y).

Exemple

Déterminer, lorsqu'ils existent les successeurs et prédécesseur immédiat de x dans les cas suivants

- (\mathbb{N}, \leq) et $x \in \mathbb{N}$
- (\mathbb{Q}, \leq) et $x \in \mathbb{Q}$

Définitions : prédécesseur, successeur

Soit (E, \preccurlyeq) un ensemble ordonné et x, y deux éléments de E

- Si $x \prec y$, x est un **prédécesseur** de y (et y est un **successeur** de x).
- Si $x \prec y$ et s'il n'existe pas d'éléments $z \in E$ tel que $x \prec z \prec y$, on dit que y est un **successeur immédiat** de x (ou que x est un **prédécesseur immédiat** de y).

Exemple

Déterminer, lorsqu'ils existent les successeurs et prédécesseur immédiat de x dans les cas suivants

- (\mathbb{N}, \leq) et $x \in \mathbb{N}$
- (\mathbb{Q}, \leq) et $x \in \mathbb{Q}$
- $(\mathcal{P}(\mathcal{E}), \subset)$ avec $\mathcal{E} = \{a, b, c, d\}$ et $x = \{a\}$.

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).
- on dit que $m \in F$, est **le plus petit élément** de F si pour tout x dans F , $m \preceq x$ (m est plus petit que tous les autres éléments de F).

Définition : élément minimal, plus petit élément

Soit (E, \preccurlyeq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).
- on dit que $m \in F$, est **le plus petit élément** de F si pour tout x dans F , $m \preccurlyeq x$ (m est plus petit que tous les autres éléments de F).

On définit de même, les élément maximaux et *le plus grand élément*.

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).
- on dit que $m \in F$, est **le plus petit élément** de F si pour tout x dans F , $m \preceq x$ (m est plus petit que tous les autres éléments de F).

On définit de même, les élément maximaux et *le plus grand élément*.

Exercices

Etudier l'existence d'éléments minimaux et du plus petit élément dans les cas suivants :

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).
- on dit que $m \in F$, est **le plus petit élément** de F si pour tout x dans F , $m \preceq x$ (m est plus petit que tous les autres éléments de F).

On définit de même, les élément maximaux et *le plus grand élément*.

Exercices

Etudier l'existence d'éléments minimaux et du plus petit élément dans les cas suivants :

- $E = (\mathbb{N}, \leq)$, et $F = \mathbb{N}$.

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).
- on dit que $m \in F$, est **le plus petit élément** de F si pour tout x dans F , $m \preceq x$ (m est plus petit que tous les autres éléments de F).

On définit de même, les élément maximaux et *le plus grand élément*.

Exercices

Etudier l'existence d'éléments minimaux et du plus petit élément dans les cas suivants :

- $E = (\mathbb{N}, \leq)$, et $F = \mathbb{N}$.
- $E = (\mathbb{R}, \leq)$ et $F =]0; 1]$.

Définition : élément minimal, plus petit élément

Soit (E, \preceq) un ensemble ordonné et F une partie de E ,

- on dit que $e \in F$, est **minimal** dans F s'il n'existe pas d'éléments x dans F tel que $x \prec e$ (aucun autre élément n'est plus petit que e).
- on dit que $m \in F$, est **le plus petit élément** de F si pour tout x dans F , $m \preceq x$ (m est plus petit que tous les autres éléments de F).

On définit de même, les élément maximaux et *le plus grand élément*.

Exercices

Etudier l'existence d'éléments minimaux et du plus petit élément dans les cas suivants :

- $E = (\mathbb{N}, \leq)$, et $F = \mathbb{N}$.
- $E = (\mathbb{R}, \leq)$ et $F =]0; 1]$.
- $E = (\mathcal{P}(\mathcal{E}), \subset)$ et $F = \mathcal{P}(\mathcal{E}) \setminus \{\emptyset\}$ où $\mathcal{E} = \{a, b, c, d\}$.

Ordre sur un produit cartésien

Etant deux ensembles ordonnés (E, \preceq_E) et (F, \preceq_F) , on définit sur $E \times F$:

Ordre sur un produit cartésien

Etant deux ensembles ordonnés (E, \preceq_E) et (F, \preceq_F) , on définit sur $E \times F$:

- La relation \preceq_p par $(e, f) \preceq_p (e', f')$ si et seulement si $e \preceq_E e'$ et $f \preceq_F f'$, \preceq_p est une relation d'ordre sur $E \times F$ appelé **ordre produit**.

Ordre sur un produit cartésien

Etant deux ensembles ordonnés (E, \preceq_E) et (F, \preceq_F) , on définit sur $E \times F$:

- La relation \preceq_p par $(e, f) \preceq_p (e', f')$ si et seulement si $e \preceq_E e'$ et $f \preceq_F f'$, \preceq_p est une relation d'ordre sur $E \times F$ appelé **ordre produit**.
- La relation \preceq_l par $(e, f) \preceq_l (e', f')$ si et seulement si $e \preceq_E e'$ ou $(e = e'$ et $f \preceq_f f')$, \preceq_l est une relation d'ordre sur $E \times F$ appelé **ordre lexicographique**.

Ordre sur un produit cartésien

Etant deux ensembles ordonnés (E, \preceq_E) et (F, \preceq_F) , on définit sur $E \times F$:

- La relation \preceq_p par $(e, f) \preceq_p (e', f')$ si et seulement si $e \preceq_E e'$ et $f \preceq_F f'$, \preceq_p est une relation d'ordre sur $E \times F$ appelé **ordre produit**.
- La relation \preceq_l par $(e, f) \preceq_l (e', f')$ si et seulement si $e \preceq_E e'$ ou $(e = e'$ et $f \preceq_f f')$, \preceq_l est une relation d'ordre sur $E \times F$ appelé **ordre lexicographique**.

Ces définitions se généralisent à un produit cartésien de n ensembles.

Ordre sur un produit cartésien

Etant deux ensembles ordonnés (E, \preceq_E) et (F, \preceq_F) , on définit sur $E \times F$:

- La relation \preceq_p par $(e, f) \preceq_p (e', f')$ si et seulement si $e \preceq_E e'$ et $f \preceq_F f'$, \preceq_p est une relation d'ordre sur $E \times F$ appelé **ordre produit**.
- La relation \preceq_l par $(e, f) \preceq_l (e', f')$ si et seulement si $e \preceq_E e'$ ou $(e = e'$ et $f \preceq_f f')$, \preceq_l est une relation d'ordre sur $E \times F$ appelé **ordre lexicographique**.

Ces définitions se généralisent à un produit cartésien de n ensembles.

Exemple

Comparer (lorsque cela est possible) les couples suivants pour l'ordre produit et l'ordre lexicographique sur $(\mathbb{N}, \leq) \times (\mathbb{N}, \leq)$:

Ordre sur un produit cartésien

Etant deux ensembles ordonnés (E, \preceq_E) et (F, \preceq_F) , on définit sur $E \times F$:

- La relation \preceq_p par $(e, f) \preceq_p (e', f')$ si et seulement si $e \preceq_E e'$ et $f \preceq_F f'$, \preceq_p est une relation d'ordre sur $E \times F$ appelé **ordre produit**.
- La relation \preceq_l par $(e, f) \preceq_l (e', f')$ si et seulement si $e \preceq_E e'$ ou $(e = e'$ et $f \preceq_f f')$, \preceq_l est une relation d'ordre sur $E \times F$ appelé **ordre lexicographique**.

Ces définitions se généralisent à un produit cartésien de n ensembles.

Exemple

Comparer (lorsque cela est possible) les couples suivants pour l'ordre produit et l'ordre lexicographique sur $(\mathbb{N}, \leq) \times (\mathbb{N}, \leq)$:

- 1 $(3, 5)$ et $(7, 6)$
- 2 $(3, 5)$ et $(7, 4)$
- 3 $(2, 1)$ et $(2, 4)$

Définition

Soit (E, \preccurlyeq) un ensemble ordonné, on dit que \preccurlyeq est un ordre **bien fondé** lorsqu'il n'existe pas de suites d'éléments strictement décroissantes d'éléments de E . C'est à dire qu'il n'existe pas de suite $(x_n)_{n \in \mathbb{N}}$ telle que $x_k \prec x_{k-1}$ pour tout $k \in \mathbb{N}$.
Un ensemble **bien ordonné** est un ensemble muni d'un ordre bien fondé.

Définition

Soit (E, \preccurlyeq) un ensemble ordonné, on dit que \preccurlyeq est un ordre **bien fondé** lorsqu'il n'existe pas de suites d'éléments strictement décroissantes d'éléments de E . C'est à dire qu'il n'existe pas de suite $(x_n)_{n \in \mathbb{N}}$ telle que $x_k \prec x_{k-1}$ pour tout $k \in \mathbb{N}$.
Un ensemble **bien ordonné** est un ensemble muni d'un ordre bien fondé.

Exemple

- L'ordre usuel sur \mathbb{N} est bien fondé.

Définition

Soit (E, \preccurlyeq) un ensemble ordonné, on dit que \preccurlyeq est un ordre **bien fondé** lorsqu'il n'existe pas de suites d'éléments strictement décroissantes d'éléments de E . C'est à dire qu'il n'existe pas de suite $(x_n)_{n \in \mathbb{N}}$ telle que $x_k \prec x_{k-1}$ pour tout $k \in \mathbb{N}$. Un ensemble **bien ordonné** est un ensemble muni d'un ordre bien fondé.

Exemple

- L'ordre usuel sur \mathbb{N} est bien fondé.
- L'ordre usuel sur \mathbb{Z} n'est pas bien fondé.

C14 Ordre – Induction structurelle

3. Ordres bien fondés

Définition

Soit (E, \preccurlyeq) un ensemble ordonné, on dit que \preccurlyeq est un ordre **bien fondé** lorsqu'il n'existe pas de suites d'éléments strictement décroissantes d'éléments de E . C'est à dire qu'il n'existe pas de suite $(x_n)_{n \in \mathbb{N}}$ telle que $x_k \prec x_{k-1}$ pour tout $k \in \mathbb{N}$. Un ensemble **bien ordonné** est un ensemble muni d'un ordre bien fondé.

Exemple

- L'ordre usuel sur \mathbb{N} est bien fondé.
- L'ordre usuel sur \mathbb{Z} n'est pas bien fondé.

Caractérisation d'un ordre bien fondé

Soit (E, \preccurlyeq) un ensemble ordonné, l'ordre \preccurlyeq est bien fondé si et seulement si toute partie non vide de E admet un élément minimal.

Preuve de terminaison

La notion d'ordre bien fondé permet d'étendre la définition d'un variant. En effet, pour prouver la terminaison d'une boucle, on peut exhiber une quantité à valeur dans ensemble E muni d'un ordre bien fondé \preccurlyeq et strictement décroissante à chaque passage dans la boucle.

Preuve de terminaison

La notion d'ordre bien fondé permet d'étendre la définition d'un variant. En effet, pour prouver la terminaison d'une boucle, on peut exhiber une quantité à valeur dans ensemble E muni d'un ordre bien fondé \preccurlyeq et strictement décroissante à chaque passage dans la boucle.

Comme dans le cas de (N, \leq) , déjà rencontré, cela permet de prouver la terminaison de la boucle car comme (E, \preccurlyeq) est bien fondé, il n'existe pas de suite strictement décroissante d'éléments de E .

C14 Ordre – Induction structurelle

4. Applications aux preuves de terminaison

Preuve de terminaison

La notion d'ordre bien fondé permet d'étendre la définition d'un variant. En effet, pour prouver la terminaison d'une boucle, on peut exhiber une quantité à valeur dans ensemble E muni d'un ordre bien fondé \preceq et strictement décroissante à chaque passage dans la boucle.

Comme dans le cas de (N, \leq) , déjà rencontré, cela permet de prouver la terminaison de la boucle car comme (E, \preceq) est bien fondé, il n'existe pas de suite strictement décroissante d'éléments de E .

Ordre bien fondé sur un produit cartésien

Si (E, \preceq_E) et (F, \preceq_F) sont deux ensembles bien fondés alors,

C14 Ordre – Induction structurelle

4. Applications aux preuves de terminaison

Preuve de terminaison

La notion d'ordre bien fondé permet d'étendre la définition d'un variant. En effet, pour prouver la terminaison d'une boucle, on peut exhiber une quantité à valeur dans ensemble E muni d'un ordre bien fondé \preceq et strictement décroissante à chaque passage dans la boucle.

Comme dans le cas de (N, \leq) , déjà rencontré, cela permet de prouver la terminaison de la boucle car comme (E, \preceq) est bien fondé, il n'existe pas de suite strictement décroissante d'éléments de E .

Ordre bien fondé sur un produit cartésien

Si (E, \preceq_E) et (F, \preceq_F) sont deux ensembles bien fondés alors,

- L'ordre produit sur $E \times F$ est bien fondé,

C14 Ordre – Induction structurelle

4. Applications aux preuves de terminaison

Preuve de terminaison

La notion d'ordre bien fondé permet d'étendre la définition d'un variant. En effet, pour prouver la terminaison d'une boucle, on peut exhiber une quantité à valeur dans ensemble E muni d'un ordre bien fondé \preceq et strictement décroissante à chaque passage dans la boucle.

Comme dans le cas de (N, \leq) , déjà rencontré, cela permet de prouver la terminaison de la boucle car comme (E, \preceq) est bien fondé, il n'existe pas de suite strictement décroissante d'éléments de E .

Ordre bien fondé sur un produit cartésien

Si (E, \preceq_E) et (F, \preceq_F) sont deux ensembles bien fondés alors,

- L'ordre produit sur $E \times F$ est bien fondé,
- L'ordre lexicographique sur $E \times F$ est bien fondé.

Terminaison de la fusion de deux listes

- Ecrire en OCaml une fonction `fusion`, `int list -> int list -> int list` qui prend en arguments deux listes triées d'entiers et renvoie leur fusion triée.

Terminaison de la fusion de deux listes

- Ecrire en OCaml une fonction `fusion, int list -> int list -> int list` qui prend en arguments deux listes triées d'entiers et renvoie leur fusion triée.
- En utilisant un variant à valeurs dans $\mathbb{N} \times \mathbb{N}$ muni de l'ordre produit justifier la terminaison de cette fonction.

Terminaison de la fusion de deux listes

- Ecrire en OCaml une fonction `fusion`, `int list -> int list -> int list` qui prend en arguments deux listes triées d'entiers et renvoie leur fusion triée.

```
1  let rec fusion l1 l2 =  
2    match l1,l2 with  
3    | [], l2 -> l2  
4    | l1, [] -> l1  
5    | h1::t1, h2::t2 -> if h1<h2 then h1::fusion t1 l2 else  
    ↪ h2::fusion l1 t2;;
```

- En utilisant un variant à valeurs dans $\mathbb{N} \times \mathbb{N}$ muni de l'ordre produit justifier la terminaison de cette fonction.

Terminaison de la fonction d'Ackerman

On considère la fonction d'Ackerman $a : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ définie par :

$$\begin{cases} a(0, m) &= m + 1 \\ a(n, 0) &= a(n - 1, 1) \text{ si } n > 0 \\ a(n, m) &= a(n - 1, a(n, m - 1)) \text{ si } n > 0 \text{ et } m > 0 \end{cases}$$

Terminaison de la fonction d'Ackerman

On considère la fonction d'Ackerman $a : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ définie par par :

$$\begin{cases} a(0, m) &= m + 1 \\ a(n, 0) &= a(n - 1, 1) \text{ si } n > 0 \\ a(n, m) &= a(n - 1, a(n, m - 1)) \text{ si } n > 0 \text{ et } m > 0 \end{cases}$$

- 1 Compléter le tableau de valeurs suivant de la fonction a :

	0	1	2	3	4	5	6
0
1	
2				

Terminaison de la fonction d'Ackerman

On considère la fonction d'Ackerman $a : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ définie par par :

$$\begin{cases} a(0, m) &= m + 1 \\ a(n, 0) &= a(n - 1, 1) \text{ si } n > 0 \\ a(n, m) &= a(n - 1, a(n, m - 1)) \text{ si } n > 0 \text{ et } m > 0 \end{cases}$$

- ① Compléter le tableau de valeurs suivant de la fonction a :

	0	1	2	3	4	5	6
0
1	
2				

- ② Ecrire la fonction `ack int -> int -> int` qui permet de calculer $a(n, m)$

Terminaison de la fonction d'Ackerman

On considère la fonction d'Ackerman $a : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ définie par par :

$$\begin{cases} a(0, m) &= m + 1 \\ a(n, 0) &= a(n - 1, 1) \text{ si } n > 0 \\ a(n, m) &= a(n - 1, a(n, m - 1)) \text{ si } n > 0 \text{ et } m > 0 \end{cases}$$

- Compléter le tableau de valeurs suivant de la fonction a :

	0	1	2	3	4	5	6
0
1	
2				

- Ecrire la fonction `ack int -> int -> int` qui permet de calculer $a(n, m)$
- Prouver sa terminaison à l'aide d'un variant à valeurs dans \mathbb{N}^2 muni de l'ordre lexicographique.

Terminaison de la fonction d'Ackerman

Terminaison de la fonction d'Ackerman

① Tableau de valeurs :

	0	1	2	3	4	5	6
0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	
2	2	3	5				

Terminaison de la fonction d'Ackerman

① Tableau de valeurs :

	0	1	2	3	4	5	6
0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	
2	2	3	5				

② Fonction ack :

```
1 let rec ack n m =  
2   match n, m with  
3   | 0, m -> m+1  
4   | n, 0 -> ack (n-1) 1  
5   | n, m -> ack (n-1) (ack n (m-1));;
```

Terminaison de la fonction d'Ackerman

① Tableau de valeurs :

	0	1	2	3	4	5	6
0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	
2	2	3	5				

② Fonction ack :

```
1 let rec ack n m =  
2   match n, m with  
3   | 0, m -> m+1  
4   | n, 0 -> ack (n-1) 1  
5   | n, m -> ack (n-1) (ack n (m-1));;
```

③ On note (n, m) (resp. (n', m')) les valeurs des paramètres avant (resp. après un appel récursif) alors, soit $n' < n$ et donc $(n', m') \prec (n, m)$ soit $n' = n$ et $m' \leq m$ et on a encore $(n', m') \prec (n, m)$.

Rappel : arbre binaire

On rappelle qu'on a défini un arbre binaire (sur un ensemble d'étiquettes E) comme étant :

- soit l'arbre vide noté \emptyset
- soit la donnée d'un triplet (g, e, d) où $e \in E$ et g et d sont des arbres binaires.

Rappel : arbre binaire

On rappelle qu'on a défini un arbre binaire (sur un ensemble d'étiquettes E) comme étant :

- soit l'arbre vide noté \emptyset
- soit la donnée d'un triplet (g, e, d) où $e \in E$ et g et d sont des arbres binaires.

On a donc donné un objet de base (l'arbre vide) et une règle permettant de construire un arbre « plus grand » en donnant deux sous arbres.

C'est ce mécanisme (la donnée d'un ou plusieurs objets de base et d'une ou plusieurs règles permettant d'en construire d'autres) qui permet de définir des ensembles dits **inductifs**.

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

- Un ensemble $X_0 \subset E$, appelé **axiomes**

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

- Un ensemble $X_0 \subset E$, appelé **axiomes**
- Un ensemble fini de **règles d'inférences**.

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

- Un ensemble $X_0 \subset E$, appelé **axiomes**
- Un ensemble fini de **règles d'inférences**.

Une règle d'inférence R est une fonction de E^n dans E , n est appelé **arité** de R .

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

- Un ensemble $X_0 \subset E$, appelé **axiomes**
- Un ensemble fini de **règles d'inférences**.

Une règle d'inférence R est une fonction de E^n dans E , n est appelé **arité** de R .

X est alors la *plus petite partie* (au sens de l'inclusion) de E qui :

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

- Un ensemble $X_0 \subset E$, appelé **axiomes**
- Un ensemble fini de **règles d'inférences**.

Une règle d'inférence R est une fonction de E^n dans E , n est appelé **arité** de R .

X est alors la *plus petite partie* (au sens de l'inclusion) de E qui :

- contient X_0

Définition

Soit E un ensemble, on peut définir par **induction** une partie X de E en se donnant :

- Un ensemble $X_0 \subset E$, appelé **axiomes**
- Un ensemble fini de **règles d'inférences**.

Une règle d'inférence R est une fonction de E^n dans E , n est appelé **arité** de R .

X est alors la *plus petite partie* (au sens de l'inclusion) de E qui :

- contient X_0
- est stable par l'application des règles d'inférence c'est à dire que $\forall f \in R$ d'arité $n, \forall (x_1, \dots, x_n) \in X^n$ appartenant au domaine de définition de $f, f(x_1, \dots, x_n) \in X$.

Exemples

- L'ensemble \mathbb{N} peut-être défini de façon inductive par $X_0 = \{0\}$ et la règle d'inférence d'arité 1 : $n \mapsto n + 1$.

Exemples

- L'ensemble \mathbb{N} peut-être défini de façon inductive par $X_0 = \{0\}$ et la règle d'inférence d'arité 1 : $n \mapsto n + 1$.
- Les listes d'entiers de OCaml peuvent être définies de façon inductive par $X_0 = []$ et la règle d'inférence d'arité 1 : $l \mapsto n :: l$ où $n \in \mathbb{N}$.

Exemples

- L'ensemble \mathbb{N} peut-être défini de façon inductive par $X_0 = \{0\}$ et la règle d'inférence d'arité 1 : $n \mapsto n + 1$.
- Les listes d'entiers de OCaml peuvent être définies de façon inductive par $X_0 = []$ et la règle d'inférence d'arité 1 : $l \mapsto n :: l$ où $n \in \mathbb{N}$.
Le **filtrage par motif** permet alors de raisonner sur les listes de façon globale en envisageant à chaque fois le cas de base (la liste vide) et le cas d'une liste produite par la règle d'inférence.

Exemples

- L'ensemble \mathbb{N} peut-être défini de façon inductive par $X_0 = \{0\}$ et la règle d'inférence d'arité 1 : $n \mapsto n + 1$.
- Les listes d'entiers de OCaml peuvent être définies de façon inductive par $X_0 = []$ et la règle d'inférence d'arité 1 : $l \mapsto n :: l$ où $n \in \mathbb{N}$.
Le **filtrage par motif** permet alors de raisonner sur les listes de façon globale en envisageant à chaque fois le cas de base (la liste vide) et le cas d'une liste produite par la règle d'inférence.
- Etant donné un ensemble A appelé **alphabet** dont les éléments sont appelés **lettres**, on peut définir de façon inductive l'ensemble A^* des mots sur A par la donnée de $X_0 = \{\epsilon\}$ où ϵ est le mot vide et la règle d'inférence d'arité 1 : $u \mapsto ua$ pour tout a dans A

Exercices

- 1 Donner l'ensemble X_0 , et la règle d'inférence pour la définition des arbres binaires rappelées en début de chapitre.

Exercices

- 1 Donner l'ensemble X_0 , et la règle d'inférence pour la définition des arbres binaires rappelées en début de chapitre.
- 2 Donner une définition inductive des entiers pairs.

Exercices

- 1 Donner l'ensemble X_0 , et la règle d'inférence pour la définition des arbres binaires rappelées en début de chapitre.
- 2 Donner une définition inductive des entiers pairs.
- 3 Sur $\mathbb{N} \cup \{+, -, \times, /\}$ on définit par induction l'ensemble E par $X_0 = \mathbb{N}$ et les règles d'inférences d'arité 2 : $(u, v) \mapsto u + v$, $(u, v) \mapsto u - v$, $(u, v) \mapsto u \times v$ et $(u, v) \mapsto u/v$
Montrer que $8 \times 4 + 10 \in E$ et donner deux suites différentes d'application des règles d'inférence permettant de l'obtenir.

Exercices

- 1 Donner l'ensemble X_0 , et la règle d'inférence pour la définition des arbres binaires appelées en début de chapitre.
- 2 Donner une définition inductive des entiers pairs.
- 3 Sur $\mathbb{N} \cup \{+, -, \times, /\}$ on définit par induction l'ensemble E par $X_0 = \mathbb{N}$ et les règles d'inférences d'arité 2 : $(u, v) \mapsto u + v$, $(u, v) \mapsto u - v$, $(u, v) \mapsto u \times v$ et $(u, v) \mapsto u/v$
Montrer que $8 \times 4 + 10 \in E$ et donner deux suites différentes d'application des règles d'inférence permettant de l'obtenir.

Ambigüité d'une définition inductive

Une définition inductive d'un ensemble X est dite **non ambiguë** si pour tout $x \in X$, il n'existe qu'une seule façon de construire x à partir des axiomes et des règles d'inférence. Cela permet notamment de construire des fonctions sur les objets inductifs.

Principe d'induction structurelle

Soit X un ensemble défini inductivement, et P une propriété sur X .

Principe d'induction structurelle

Soit X un ensemble défini inductivement, et P une propriété sur X .

- (*initialisation*) Si $P(x)$ est vraie pour tous les éléments de X_0 .

Principe d'induction structurelle

Soit X un ensemble défini inductivement, et P une propriété sur X .

- (*initialisation*) Si $P(x)$ est vraie pour tous les éléments de X_0 .
- (*hérédité*) Et si pour toute règle R d'arité n , et tout $(x_1, \dots, x_n) \in X$, $P(x_1), \dots, P(x_n)$ vraies $\implies P(x_1, \dots, x_n)$ vraie.

Principe d'induction structurelle

Soit X un ensemble défini inductivement, et P une propriété sur X .

- (*initialisation*) Si $P(x)$ est vraie pour tous les éléments de X_0 .
- (*hérédité*) Et si pour toute règle R d'arité n , et tout $(x_1, \dots, x_n) \in X$,
 $P(x_1), \dots, P(x_n)$ vraies $\implies P(x_1, \dots, x_n)$ vraie.

alors pour tout $x \in X$, $P(x)$ est vraie.

Principe d'induction structurelle

Soit X un ensemble défini inductivement, et P une propriété sur X .

- (*initialisation*) Si $P(x)$ est vraie pour tous les éléments de X_0 .
- (*hérédité*) Et si pour toute règle R d'arité n , et tout $(x_1, \dots, x_n) \in X$,
 $P(x_1), \dots, P(x_n)$ vraies $\implies P(x_1, \dots, x_n)$ vraie.

alors pour tout $x \in X$, $P(x)$ est vraie.

Remarque

Dans le cas où X est l'ensemble des entiers naturels défini de façon inductive, on retrouve le principe du raisonnement par récurrence.

Exemple

On définit inductivement (et de façon non ambiguë) l'ensemble D des mots de Dyck (ou parenthésages bien formés) par : $X_0 = \{\epsilon\}$ et la règle d'inférence d'arité 2 $(x, y) \mapsto \langle x \rangle y$.

Pour $d \in D$, on note $n_o(d)$ (resp. $n_f(d)$) le nombre de symboles \langle (resp. \rangle) apparaissant dans d . Montrer par induction structurelle que $n_o(d) = n_f(d)$ pour tout $d \in D$.

Exemple

On définit inductivement (et de façon non ambiguë) l'ensemble D des mots de Dyck (ou parenthésages bien formés) par : $X_0 = \{\epsilon\}$ et la règle d'inférence d'arité 2 $(x, y) \mapsto \langle x \rangle y$.

Pour $d \in D$, on note $n_o(d)$ (resp. $n_f(d)$) le nombre de symboles \langle (resp. \rangle) apparaissant dans d . Montrer par induction structurelle que $n_o(d) = n_f(d)$ pour tout $d \in D$.

- On note $P(d)$ la propriété $n_o(d) = n_f(d)$

Exemple

On définit inductivement (et de façon non ambiguë) l'ensemble D des mots de Dyck (ou parenthésages bien formés) par : $X_0 = \{\epsilon\}$ et la règle d'inférence d'arité 2 $(x, y) \mapsto \langle x \rangle y$.

Pour $d \in D$, on note $n_o(d)$ (resp. $n_f(d)$) le nombre de symboles \langle (resp. \rangle) apparaissant dans d . Montrer par induction structurelle que $n_o(d) = n_f(d)$ pour tout $d \in D$.

- On note $P(d)$ la propriété $n_o(d) = n_f(d)$
- Initialisation : on vérifie la propriété pour les éléments de X_0 , comme $X_0 = \epsilon$ et $n_o(\epsilon) = 0$ et $n_f(\epsilon) = 0$. P est vraie pour tous les éléments de X_0 .

Exemple

On définit inductivement (et de façon non ambiguë) l'ensemble D des mots de Dyck (ou parenthésages bien formés) par : $X_0 = \{\epsilon\}$ et la règle d'inférence d'arité 2 $(x, y) \mapsto \langle x \rangle y$.

Pour $d \in D$, on note $n_o(d)$ (resp. $n_f(d)$) le nombre de symboles \langle (resp. \rangle) apparaissant dans d . Montrer par induction structurelle que $n_o(d) = n_f(d)$ pour tout $d \in D$.

- On note $P(d)$ la propriété $n_o(d) = n_f(d)$
- Initialisation : on vérifie la propriété pour les éléments de X_0 , comme $X_0 = \epsilon$ et $n_o(\epsilon) = 0$ et $n_f(\epsilon) = 0$. P est vraie pour tous les éléments de X_0 .
- Hérédité : on vérifie la conservation de la propriété par application de chacune des règles d'inférence. Si x et y sont deux mots de Dyck vérifiant P alors $n_o(\langle x \rangle y) = 1 + n_o(x) + n_o(y)$ et $n_f(\langle x \rangle y) = 1 + n_f(x) + n_f(y)$ et puisque x et y vérifient P , $n_o(\langle x \rangle y) = n_f(\langle x \rangle y)$

Exemple

On définit inductivement (et de façon non ambiguë) l'ensemble D des mots de Dyck (ou parenthésages bien formés) par : $X_0 = \{\epsilon\}$ et la règle d'inférence d'arité 2 $(x, y) \mapsto \langle x \rangle y$.

Pour $d \in D$, on note $n_o(d)$ (resp. $n_f(d)$) le nombre de symboles \langle (resp. \rangle) apparaissant dans d . Montrer par induction structurelle que $n_o(d) = n_f(d)$ pour tout $d \in D$.

- On note $P(d)$ la propriété $n_o(d) = n_f(d)$
- Initialisation : on vérifie la propriété pour les éléments de X_0 , comme $X_0 = \epsilon$ et $n_o(\epsilon) = 0$ et $n_f(\epsilon) = 0$. P est vraie pour tous les éléments de X_0 .
- Hérédité : on vérifie la conservation de la propriété par application de chacune des règles d'inférence. Si x et y sont deux mots de Dyck vérifiant P alors $n_o(\langle x \rangle y) = 1 + n_o(x) + n_o(y)$ et $n_f(\langle x \rangle y) = 1 + n_f(x) + n_f(y)$ et puisque x et y vérifient P , $n_o(\langle x \rangle y) = n_f(\langle x \rangle y)$

Par induction structurelle on conclut que P est vraie pour tout $x \in D$.

Définition - Théorème

Soit X un ensemble inductif défini de manière non ambiguë on définit la relation \mathcal{R} par $x_i \mathcal{R} f(x_1, \dots, x_n)$ pour toute règle d'inférence f d'arité n et tout n -uplet (x_1, \dots, x_n) .

Définition - Théorème

Soit X un ensemble inductif défini de manière non ambiguë on définit la relation \mathcal{R} par $x_i \mathcal{R} f(x_1, \dots, x_n)$ pour toute règle d'inférence f d'arité n et tout n -uplet (x_1, \dots, x_n) .

La fermeture réflexive-transitive de \mathcal{R} , notée \preccurlyeq est une relation d'ordre bien fondé sur X appelé **ordre induit**.

Définition - Théorème

Soit X un ensemble inductif défini de manière non ambiguë on définit la relation \mathcal{R} par $x_i \mathcal{R} f(x_1, \dots, x_n)$ pour toute règle d'inférence f d'arité n et tout n -uplet (x_1, \dots, x_n) .

La fermeture réflexive-transitive de \mathcal{R} , notée \preccurlyeq est une relation d'ordre bien fondé sur X appelé **ordre induit**.

Principe d'induction bien fondé

Soient (X, \preccurlyeq) un ensemble bien ordonné et P une propriété sur X , si

- P est vraie pour tous les éléments minimaux de X .
- Et si pour $z \preccurlyeq x$ $P(z)$ vraies $\implies P(x)$

Alors P est vraie pour tout $x \in X$.