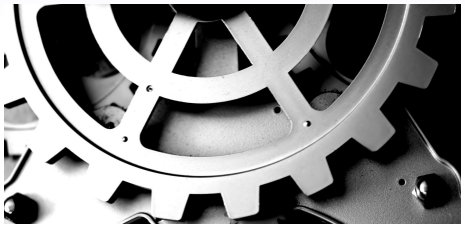
Algorithmique et structures de données
Induction



Gaël Mahé

slides : Elise Bonzon et Gaël Mahé

Université Paris Descartes

Licence 2

Induction

Induction

Quelques définitions liées aux relations d'ordre

- Elément x appartenant à un ensemble E : $x \in E$
- Ensemble n'ayant aucun élément, appelé l'**ensemble vide** : $\emptyset$
- A **sous-ensemble** de B, ou A **inclus** dans B : $A \subseteq B$
- $\mathcal{P}(E)$: ensemble des **parties** de l'ensemble E
  - Les éléments de $\mathcal{P}(E)$ sont des ensembles
  - $\mathcal{P}(E)$ contient toujours E et $\emptyset$
  - Exemple : $E = \{0, 1\}$, $\mathcal{P}(E) = \{\{0\}, \{1\}, E, \emptyset\}$.
- **Produit cartésien** de deux ensembles A et B, noté $A \times B$: ensemble des couples (a, b) tels que $a \in A$ et $b \in B$.
  Généralisation à une famille finie d'ensembles : $A_1 \times A_2 \times \ldots \times A_p$

Quelques définitions liées aux relations d'ordre

- Une **relation binaire** $\mathcal{R}$ sur E est
  une partie de $E \times E$
  ou une application de $E \times E \to \{vrai, faux\}$
- Propriétés des relations binaires :
  - **réflexive** si $\forall x \in E$ on a $x\mathcal{R}x$
  - **irréflexive** si $\forall x, y \in E$, $x\mathcal{R}y \Rightarrow x \neq y$
  - **symétrique** si $\forall x, y \in E$, $x\mathcal{R}y \Rightarrow y\mathcal{R}x$
  - **antisymétrique** si $\forall x, y \in E$, $x\mathcal{R}y$ et $y\mathcal{R}x \Rightarrow x = y$
  - **asymétrique** si $\forall x, y \in E$, $x\mathcal{R}y \Rightarrow \mathrm{not}(y\mathcal{R}x)$
  - **transitive** si $\forall x, y, z \in E$, $x\mathcal{R}y$ et $y\mathcal{R}z \Rightarrow x\mathcal{R}z$
- **Exemples :**
  - $E =$mots, $\mathcal{R} =$ "a le même nombre de lettres que"
  - $E = \mathbb{R}$, $\mathcal{R} = \leq$ ou A un ensemble, $E = \mathcal{P}(A)$, $\mathcal{R} = \subseteq$
  - $E = \mathbb{R}$, $\mathcal{R} = <$ ou A un ensemble, $E = \mathcal{P}(A)$, $\mathcal{R} = \subset$

Quelques définitions liées aux relations d'ordre

- Une **relation d'équivalence** est réflexive, symétrique et transitive
  Ex :
- Une **relation d'ordre large** est réflexive, antisymétrique et transitive
  Ex :
- Une relation **d'ordre strict** est irréflexive, asymétrique et transitive
  Ex :
- L'ordre est **total** lorsque deux éléments quelconques de l'ensemble sont comparables par la relation, sinon, on dit que l'ordre est **partiel**
  - L'ordre habituel sur les réels est total
  - L'ordre de divisibilité sur les entiers est partiel :
    $a\mathcal{R}_{div}b$ si et seulement si $\exists c$ tel que $b = ac$
- Plusieurs ordres peuvent être définis sur un même ensemble

Quelques définitions liées aux relations d'ordre

Soit E' une partie d'un ensemble ordonné $(E, \leq)$

- $x \in E$ est un **majorant** de E' si $\forall y \in E'$, $y \leq x$
- Un élément $y \in E'$ qui n'a aucun majorant dans E' est dit **élément maximal**
- $Maj(E')$ est l'ensemble des majorants de E'
- $Maj(E') \cap E'$ a au plus un élément
- Si $Maj(E') \cap E'$ n'est pas vide, son unique élément est appelé **maximum**
- On définit de manière similaire un **minorant**, un **élément minimal**, le **minimum** et **Min(E)**

Induction

Ensemble bien fondé

Définition

Un ensemble ordonné $(E, \leq)$ est **bien fondé**
(on dit aussi : la relation d'ordre $\leq$ sur E est **bien fondée**)
si l'une des deux conditions suivantes est vérifiée :

- Il n'y a pas de suite infinie strictement décroissante d'éléments de E
- Toute partie de E admet au moins un élément minimal.

Les deux conditions sont équivalentes.

- C'est le cadre d'application du raisonnement par récurrence
- $\rightarrow$ Induction : généralisation de la récurrence aux ordres bien fondés

Exemples d'ordres bien fondés ou non

- $<$ est un ordre bien fondé sur $\mathbb{N}$, pas sur $\mathbb{Z}$
- Sur un ensemble de mots, la relation *"être une sous-chaîne de"* est un ordre bien fondé
- Si $<_1$ et $<_2$ sont des ordres bien fondés sur E_1 et E_2 respectivement, alors l'ordre lexicographique $<_{1,2}$ est bien fondé sur $E_1 \times E_2$

  Ordre lexicographique :
  Soient $a, b \in E_1$ et $c, d \in E_2$
  $(a, c) <_{1,2} (b, d)$ si $a <_1 b$ ou $(a = b \,\text{et}\, c <_2 d)$.

Induction

Premier principe d'induction sur $\mathbb{N}$ = principe de récurrence

Théorème

Soit $P(n)$ un prédicat dépendant de l'entier n
Si les deux conditions suivantes sont satisfaites :

(B) $P(0)$ est vrai

(I) $\forall n \in \mathbb{N},\ P(n) \Rightarrow P(n+1)$

Alors $\forall n \in \mathbb{N},\ P(n)$ est vrai
(B) est la base de la récurrence, (I) l'hypothèse d'induction

Premier principe d'induction sur $\mathbb{N}$ = principe de récurrence

Théorème

Soit $P(n)$ un prédicat dépendant de l'entier n
Si les deux conditions suivantes sont satisfaites :

(B) $P(0)$ est vrai

(I) $\forall n \in \mathbb{N},\ P(n) \Rightarrow P(n+1)$

Alors $\forall n \in \mathbb{N},\ P(n)$ est vrai
(B) est la base de la récurrence, (I) l'hypothèse d'induction

Démonstration :

- Soit $X = \{k \in \mathbb{N} / P(k) \text{ faux}\}$.
- Si X non vide, comme il est bien fondé, il admet un plus petit élément n_0.
- D'après (B), $n_0 \neq 0$.
- Donc $n_0 - 1$ est un entier et $P(n_0 - 1)$ est vrai.
- Par (I), on déduit que $P(n_0)$ est vrai, ce qui est contradictoire.
- Donc X est vide.

Autre forme du premier principe d'induction sur $\mathbb{N}$

Proposition

Soit n_0 un entier positif ou nul
Si les deux conditions suivantes sont satisfaites :

(B_{n_0}) $P(n_0)$ est vrai

(I_{n_0}) $\forall n \geq n_0,\ P(n) \Rightarrow P(n+1)$

Alors $\forall n \geq n_0,\ P(n)$ est vrai

Deuxième principe d'induction sur $\mathbb{N}$

Théorème

Soit $P(n)$ un prédicat dépendant de l'entier n
Si la propriété (I') suivante est satisfaite :

$$\forall n \in \mathbb{N}, ((\forall k_{\in \mathbb{N}} < n, P(k)) \Rightarrow P(n))$$

Alors $\forall n \in \mathbb{N}$, $P(n)$ est vrai

Deuxième principe d'induction sur $\mathbb{N}$

Théorème

Soit $P(n)$ un prédicat dépendant de l'entier n
Si la propriété (I') suivante est satisfaite :

$$\forall n \in \mathbb{N}, ((\forall k_{\in \mathbb{N}} < n, P(k)) \Rightarrow P(n))$$

Alors $\forall n \in \mathbb{N}$, $P(n)$ est vrai

Et l'hypothèse de base (B) ?

- (I') pour $n = 0$: $((\forall k < 0, P(k)) \Rightarrow P(0))$
- Comme il n'y a pas d'entier naturel < 0,
  $(\forall k < 0, P(k))$ =Vrai
- Donc (I') pour $n = 0$: $P(0)$
- Conclusion : (I') inclut (B) et **il faut toujours vérifier** $P(0)$

Sur $\mathbb{N}$, les deux principes sont équivalents.

Deuxième principe d'induction sur $\mathbb{N}$: démonstration

Théorème

Soit $P(n)$ un prédicat dépendant de l'entier n
Si la propriété (I') suivante est satisfaite :

$$\forall n \in \mathbb{N}, ((\forall k_{\in \mathbb{N}} < n, P(k)) \Rightarrow P(n))$$

Alors $\forall n \in \mathbb{N}$, $P(n)$ est vrai

Deuxième principe d'induction sur $\mathbb{N}$: démonstration

Théorème

Soit $P(n)$ un prédicat dépendant de l'entier n
Si la propriété (I') suivante est satisfaite :

$$\forall n \in \mathbb{N}, ((\forall k_{\in \mathbb{N}} < n, P(k)) \Rightarrow P(n))$$

Alors $\forall n \in \mathbb{N}$, $P(n)$ est vrai

Démonstration Soit $X = \{k \in \mathbb{N} / P(k) \text{ faux}\}$.
Si X n'est pas vide, il admet un élément minimal k_{min} (ordre bien fondé).
Donc $\forall k < k_{min}$, $k \notin X$, donc $P(k)$ est vraie.
D'après (I'), $P(k_{min})$ est donc vraie : contradiction avec $k_{min} \in X$.
Donc X est vide. Donc, $\forall n \in \mathbb{N}$, $P(n)$.

Généralisation : principe d'induction sur les ordres bien fondés

Théorème

Soit $\leq$ un ordre large bien fondé sur E, $<$ l'ordre strict correspondant, et P une proposition dépendant d'un élément $x \in E$.
Si la propriété suivante (I') est vérifiée :

$$\forall x \in E, ((\forall y_{\in E} < x, P(y)) \Rightarrow P(x))$$

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

Généralisation : principe d'induction sur les ordres bien fondés

Théorème

Soit $\leq$ un ordre large bien fondé sur E, $<$ l'ordre strict correspondant, et P une proposition dépendant d'un élément $x \in E$.
Si la propriété suivante (I') est vérifiée :

$$\forall x \in E, ((\forall y_{\in E} < x, P(y)) \Rightarrow P(x))$$

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

Et l'hypothèse de base (B) ?

- (I') pour tout x_{min} élément minimal : $((\forall y < x_{min}, P(y)) \Rightarrow P(x_{min}))$
- Comme il n'y a pas d'élément de $E < x_{min}$,
  $(\forall y < x_{min}, P(y))$ =Vrai
- Donc (I') pour $x = x_{min}$: $P(x_{min})$
- Conclusion : (I') inclut (B) et **il faut toujours vérifier $P(x_{min})$ pour tous les éléments minimaux x_{min}**

Principe d'induction sur les ordres bien fondés : démonstration

Théorème

Soit $\leq$ un ordre large bien fondé sur E, $<$ l'ordre strict correspondant, et P une proposition dépendant d'un élément $x \in E$.
Si la propriété suivante (I') est vérifiée :

$$\forall x \in E, ((\forall y_{\in E} < x, P(y)) \Rightarrow P(x))$$

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

Principe d'induction sur les ordres bien fondés : démonstration

Théorème

Soit $\leq$ un ordre large bien fondé sur E, $<$ l'ordre strict correspondant, et P une proposition dépendant d'un élément $x \in E$.
Si la propriété suivante (I') est vérifiée :

$$\forall x \in E, ((\forall y_{\in E} < x, P(y)) \Rightarrow P(x))$$

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

Démonstration :
Soit $X = \{x \in E / P(x) \text{ faux}\}$.
Si X n'est pas vide, il admet un élément minimal x_{min} (ordre bien fondé).
Donc $\forall y < x_{min}$, $y \notin X$, donc $P(y)$ est vraie.
D'après (I'), $P(x_{min})$ est donc vraie : contradiction avec $x_{min} \in X$.
Donc X est vide. Donc, $\forall x \in E$, $P(x)$ est vraie

Exemple de raisonnement par induction

Soit $P(n)$ la propriété "n décomposable en produit de nombre premiers".
Démontrons par induction : $\forall n \geq 2, \quad P(n)$

1. **S'assurer que l'ensemble ordonné est bien fondé**
   - $\mathbb{N}$ est bien fondé, donc ses sous-ensembles aussi, dont $[2; \infty]$.
2. **Démontrer (B), c'est-à-dire (I') pour tous les éléments minimaux**
   - $[2; \infty]$ a un élément minimal, 2, et $P(2)$ est vraie.
3. **Démontrer (I') pour tout élément non minimal**
   - $(\text{I'}) = \forall n \in \mathbb{N} \geq 2, ((\forall k < n, P(k)) \Rightarrow P(n))$
   - Soit $n > 2$
   - Supposons que $\forall k \in \{2, \ldots, n-1\}$, $P(k)$ est vrai. (hypothèse d'induction)
   - Si n est premier, $P(n)$
   - Si n n'est pas premier,
     il existe a et b deux entiers entre 2 et $n-1$ tels que $n = a * b$.
     Comme $a, b \in \{2, \ldots, n-1\}$, $P(a)$ et $P(b)$ par l'hypothèse d'induction.
     n est donc décomposable en produit de nombres premiers
4. **Appliquer le principe d'induction**
   - On a montré que $\forall n \in \mathbb{N} \geq 2, ((\forall k < n, P(k)) \Rightarrow P(n))$
   - D'après le principe d'induction, $\forall n \geq 2, \quad P(n)$

Raisonnement erroné par induction

- Ensemble $\mathbb{R}$
- Ordre $\leq$ usuel
- Propriété $P(x) : x > 0$
- Appliquons le principe d'induction : la propriété

$$\forall x \in \mathbb{R}, ((\forall y < x, P(y)) \Rightarrow P(x))$$

est vérifiée.

Raisonnement erroné par induction

- Ensemble $\mathbb{R}$
- Ordre $\leq$ usuel
- Propriété $P(x)$: $x > 0$
- Appliquons le principe d'induction : la propriété

$$\forall\, x \in \mathbb{R}, ((\forall y < x, P(y)) \Rightarrow P(x))$$

est vérifiée.
Donc $\forall\, x \in \mathbb{R}, \quad x > 0$.

Application à la programmation

Le principe d'induction permet :

- de démontrer l'efficience et la terminaison d'algorithmes récursifs ;
- de démontrer les propriétés de structures définies par induction.

Exemple : fonction modulo récursive

- Soient $a, b \in \mathbb{N}$. On cherche r tel que :

$$\exists\, q \in \mathbb{N} \quad | \quad a = bq + r, \quad \text{avec } 0 \leq r < b$$

- Fonction : `mod(a,b) = if a<b then a else mod(a-b,b)`
- Preuve de la terminaison :
  - Ordre bien fondé : ordre usuel $\leq$ sur $\mathbb{N}$
  - Si $a \geq b$, $a - b \in \mathbb{N}$ et $a - b < a$ (car $b \neq 0$)

  $\rightarrow$ La suite des arguments des appels récursifs successifs est strictement décroissante dans un ensemble bien ordonné, donc est finie.
- Preuve de l'efficience :
  - Sur l'ensemble constitué de la suite des arguments, application du principe d'induction avec $P(a) =$ "mod(a,b) renvoie bien a modulo b"

Preuve de l'efficience de la fonction mod

`mod(a,b) = if a<b then a else mod(a-b,b)`

$P(a) =$ "mod(a,b) renvoie bien a modulo b"

$P(a) = (\exists\, q \mid \quad a = bq + mod(a,b) \quad \text{et} \quad 0 \leq mod(a,b) < b)$

Démontrons que pour tout élément a' de l'ensemble E des arguments des appels récursifs successifs, $P(a')$

1. L'ordre $\leq$ sur $\mathbb{N}$ est bien fondé.
2. (B) E a un élément minimal $a_0 < b$.
   Dans ce cas, $mod(a_0, b) = a_0$ et c'est bien le modulo.
3. Démontrons (I')
   - Soit a' élément non minimal de E, donc tel que $a' \geq b$.
   - $mod(a', b) = mod(a' - b, b)$
   - Supposons que $\forall x_{\in E} < a'$, $P(x)$.
   - Comme $a' - b < a'$ (rappel : $b \neq 0$),
     $\exists\, q \mid \quad a' - b = bq + mod(a' - b, b) \quad \text{et} \quad 0 \leq mod(a' - b, b) < b$
   - Donc $\exists\, q \mid \quad a' = b(q+1) + mod(a', b) \quad \text{et} \quad 0 \leq mod(a', b) < b$
   - Donc $P(a')$
4. On a montré que $\forall a' \in E, ((\forall x < a', P(x)) \Rightarrow P(a'))$
   D'après le principe d'induction, $\forall\, a' \in E, \quad P(a')$.

Preuve de l'efficience de la fonction mod

`mod(a,b) = if a<b then a else mod(a-b,b)`

$P(a) =$ "mod(a,b) renvoie bien a modulo b"

$P(a) = (\exists\, q \mid \quad a = bq + mod(a,b) \quad \text{et} \quad 0 \leq mod(a,b) < b)$

Démontrons que pour tout élément a' de l'ensemble E des arguments des appels récursifs successifs, $P(a')$

1. L'ordre $\leq$ sur $\mathbb{N}$ est bien fondé.
2. (B) E a un élément minimal $a_0 < b$.
   Dans ce cas, $mod(a_0, b) = a_0$ et c'est bien le modulo.
3. Démontrons (I')
   - Soit a' élément non minimal de E, donc tel que $a' \geq b$.
   - $mod(a', b) = mod(a' - b, b)$
   - Supposons que $\forall x_{\in E} < a'$, $P(x)$.
   - Comme $a' - b < a'$ (rappel : $b \neq 0$),
     $\exists\, q \mid \quad a' - b = bq + mod(a' - b, b) \quad \text{et} \quad 0 \leq mod(a' - b, b) < b$
   - Donc $\exists\, q \mid \quad a' = b(q+1) + mod(a', b) \quad \text{et} \quad 0 \leq mod(a', b) < b$
   - Donc $P(a')$
4. On a montré que $\forall a' \in E, ((\forall x < a', P(x)) \Rightarrow P(a'))$
   D'après le principe d'induction, $\forall\, a' \in E, \quad P(a')$.

Preuve de l'efficience de la fonction mod

`mod(a,b) = if a<b then a else mod(a-b,b)`

$P(a) =$ "mod(a,b) renvoie bien a modulo b"

$P(a) = (\exists\, q \mid \quad a = bq + mod(a,b) \quad \text{et} \quad 0 \leq mod(a,b) < b)$

Démontrons que pour tout élément a' de l'ensemble E des arguments des appels récursifs successifs, $P(a')$

1. L'ordre $\leq$ sur $\mathbb{N}$ est bien fondé.
2. (B) E a un élément minimal $a_0 < b$.
   Dans ce cas, $mod(a_0, b) = a_0$ et c'est bien le modulo.
3. Démontrons (I')
   - Soit a' élément non minimal de E, donc tel que $a' \geq b$.
   - $mod(a', b) = mod(a' - b, b)$
   - Supposons que $\forall x_{\in E} < a'$, $P(x)$.
   - Comme $a' - b < a'$ (rappel : $b \neq 0$),
     $\exists\, q \mid \quad a' - b = bq + mod(a' - b, b) \quad \text{et} \quad 0 \leq mod(a' - b, b) < b$
   - Donc $\exists\, q \mid \quad a' = b(q+1) + mod(a', b) \quad \text{et} \quad 0 \leq mod(a', b) < b$
   - Donc $P(a')$
4. On a montré que $\forall a' \in E, ((\forall x < a', P(x)) \Rightarrow P(a'))$
   D'après le principe d'induction, $\forall\, a' \in E, \quad P(a')$.

Preuve de l'efficience de la fonction mod

`mod(a,b) = if a<b then a else mod(a-b,b)`

$P(a) =$ "mod(a,b) renvoie bien a modulo b"

$P(a) = (\exists\, q \mid \quad a = bq + mod(a,b) \quad \text{et} \quad 0 \leq mod(a,b) < b)$

Démontrons que pour tout élément a' de l'ensemble E des arguments des appels récursifs successifs, $P(a')$

1. L'ordre $\leq$ sur $\mathbb{N}$ est bien fondé.
2. (B) E a un élément minimal $a_0 < b$.
   Dans ce cas, $mod(a_0, b) = a_0$ et c'est bien le modulo.
3. Démontrons (I')
   - Soit a' élément non minimal de E, donc tel que $a' \geq b$.
   - $mod(a', b) = mod(a' - b, b)$
   - Supposons que $\forall x_{\in E} < a'$, $P(x)$.
   - Comme $a' - b < a'$ (rappel : $b \neq 0$),
     $\exists\, q \mid \quad a' - b = bq + mod(a' - b, b) \quad \text{et} \quad 0 \leq mod(a' - b, b) < b$
   - Donc $\exists\, q \mid \quad a' = b(q+1) + mod(a', b) \quad \text{et} \quad 0 \leq mod(a', b) < b$
   - Donc $P(a')$
4. On a montré que $\forall a' \in E, ((\forall x < a', P(x)) \Rightarrow P(a'))$
   D'après le principe d'induction, $\forall a' \in E, \quad P(a')$.

Preuve de l'efficience de la fonction mod

`mod(a,b) = if a<b then a else mod(a-b,b)`

$P(a) =$ "mod(a,b) renvoie bien a modulo b"

$P(a) = (\exists\, q \mid \quad a = bq + mod(a,b) \quad \text{et} \quad 0 \leq mod(a,b) < b)$

Démontrons que pour tout élément a' de l'ensemble E des arguments des appels récursifs successifs, $P(a')$

1. L'ordre $\leq$ sur $\mathbb{N}$ est bien fondé.
2. (B) E a un élément minimal $a_0 < b$.
   Dans ce cas, $mod(a_0, b) = a_0$ et c'est bien le modulo.
3. Démontrons (I')
   - Soit a' élément non minimal de E, donc tel que $a' \geq b$.
   - $mod(a', b) = mod(a' - b, b)$
   - Supposons que $\forall x_{\in E} < a'$, $P(x)$.
   - Comme $a' - b < a'$ (rappel : $b \neq 0$),
     $\exists\, q \mid \quad a' - b = bq + mod(a' - b, b) \quad \text{et} \quad 0 \leq mod(a' - b, b) < b$
   - Donc $\exists\, q \mid \quad a' = b(q+1) + mod(a', b) \quad \text{et} \quad 0 \leq mod(a', b) < b$
   - Donc $P(a')$
4. On a montré que $\forall a' \in E, ((\forall x < a', P(x)) \Rightarrow P(a'))$
   D'après le principe d'induction, $\forall\, a' \in E, \quad P(a')$.

Preuve de l'efficience de la fonction mod

`mod(a,b) = if a<b then a else mod(a-b,b)`

$P(a) =$ "mod(a,b) renvoie bien a modulo b"

$P(a) = (\exists\, q \mid \quad a = bq + mod(a,b) \quad \text{et} \quad 0 \leq mod(a,b) < b)$

Démontrons que pour tout élément a' de l'ensemble E des arguments des appels récursifs successifs, $P(a')$

1. L'ordre $\leq$ sur $\mathbb{N}$ est bien fondé.
2. (B) E a un élément minimal $a_0 < b$.
   Dans ce cas, $mod(a_0, b) = a_0$ et c'est bien le modulo.
3. Démontrons (I')
   - Soit a' élément non minimal de E, donc tel que $a' \geq b$.
   - $mod(a', b) = mod(a' - b, b)$
   - Supposons que $\forall x_{\in E} < a'$, $P(x)$.
   - Comme $a' - b < a'$ (rappel : $b \neq 0$),
     $\exists\, q \mid \quad a' - b = bq + mod(a' - b, b) \quad \text{et} \quad 0 \leq mod(a' - b, b) < b$
   - Donc $\exists\, q \mid \quad a' = b(q+1) + mod(a', b) \quad \text{et} \quad 0 \leq mod(a', b) < b$
   - Donc $P(a')$
4. On a montré que $\forall a' \in E, ((\forall x < a', P(x)) \Rightarrow P(a'))$
   D'après le principe d'induction, $\forall\, a' \in E, \quad P(a')$.

Et en particulier, $P(a)$

Application à la programmation

Le principe d'induction permet :

- de démontrer l'efficience et la terminaison d'algorithmes récursifs ;
  - Définir un ordre bien fondé sur le domaine de définition de la fonction
  - Montrer que la suite des arguments des appels récursifs successifs est strictement décroissante
    $\rightarrow$ elle est finie, donc terminaison de l'algorithme
  - Application du principe d'induction sur la suite des arguments, avec P = "l'algorithme renvoie le résultat attendu"
    $\rightarrow$ efficience de l'algorithme
- de démontrer les propriétés de structures définies par induction.

Application à la programmation

Le principe d'induction permet :

- de démontrer l'efficience et la terminaison d'algorithmes récursifs ;
  - Définir un ordre bien fondé sur le domaine de définition de la fonction
  - Montrer que la suite des arguments des appels récursifs successifs est strictement décroissante
    $\rightarrow$ elle est finie, donc terminaison de l'algorithme
  - Application du principe d'induction sur la suite des arguments, avec P = "l'algorithme renvoie le résultat attendu"
    $\rightarrow$ efficience de l'algorithme
- de démontrer les propriétés de structures définies par induction.

Définitions inductives

- La définition d'un ensemble X peut être de la forme
  - (B) certains éléments de X sont donnés explicitement
  - (I) les autres éléments sont définis à partir d'éléments appartenant déjà à l'ensemble X
- Exemple : la partie X de $\mathbb{N}$ définie par
  - (B) l'élément $0 \in X$
  - (I) $n \in X \Rightarrow n + 1 \in X$

  n'est autre que $\mathbb{N}$ tout entier

Définitions inductives

- La définition d'un ensemble X peut être de la forme
  - (B) certains éléments de X sont donnés explicitement
  - (I) les autres éléments sont définis à partir d'éléments appartenant déjà à l'ensemble X
- Exemple : la partie X de $\mathbb{N}$ définie par
  - (B) l'élément $0 \in X$
  - (I) $n \in X \Rightarrow n + 1 \in X$

  n'est autre que $\mathbb{N}$ tout entier

Preuve de propriétés : exemple des mots de Dyck

- Δ = ensemble des mots sur l'alphabet $\{a, \overline{a}\}$ définis par :
  - Mot vide $\epsilon \in \Delta$
  - Si $x, y \in \Delta$, alors $ax\overline{a}y$ aussi
- Démontrons la propriété $P = (\forall\, x \in \Delta,\ x$ contient autant de a que de $\overline{a})$.
  - Ordre $\leq$: *est une sous-chaîne de*
  - Base de l'induction : $P(\epsilon)$ est vraie
  - Soit $x \in \Delta, > \epsilon$
  - Supposons $\forall\, t \in \Delta \,\text{et}\, < x,\ P(t)$
  - $\exists\, y, z \in \Delta \,|\quad x = ay\overline{a}z$
  - $P(y)$ et $P(z)$, et on ajoute un a et un $\overline{a}$
  - Donc $P(x)$
  - Par application du principe d'induction, $P(x)\,\forall\, x$