
Induction sur les entiers

Utilisations

- Pour définir des fonctions sur les entiers
- Pour définir des fonctions sur d'autres objects

Induction sur les entiers pour la multiplication

$$\begin{aligned}\text{mul}(0, m) &= 0 \\ \text{mul}(n + 1, m) &= \text{mul}(n, m) + m\end{aligned}$$

Complexité de `mul`

Quel est le nombre d'appels à la fonction `mul` ?

$$\begin{aligned}\text{mul}(4, 5) &\rightarrow \\ \text{mul}(3, 5) + 5 &\rightarrow \\ \text{mul}(2, 5) + 5 + 5 &\rightarrow \\ \text{mul}(1, 5) + 5 + 5 + 5 &\rightarrow \\ \text{mul}(0, 5) + 5 + 5 + 5 + 5 &\rightarrow \\ 0 + 5 + 5 + 5 + 5 &= 20\end{aligned}$$

Induction sur les entiers pour l'exponentiation

On veut définir la fonction

$$\begin{aligned}\text{exp}(m, r) &= \underbrace{m \times \dots \times m}_r \\ \text{exp}(m, r) &= \underbrace{m \times \dots \times m}_{r-1} \times m\end{aligned}$$

Ceci donne l'équation

$$\text{exp}(m, r) = \text{exp}(m, r - 1) \times m$$

Quelles sont les **bonnes** valeurs de r pour que cette équation soit bien définie ? $r > 0$.

Ceci donne **deux** équations :

$$\begin{aligned}\text{exp}(m, 0) &= 1 \\ \text{exp}(m, n + 1) &= \text{exp}(m, n) * m\end{aligned}$$

Complexité de `exp (I)`

Quel est le nombre d'appels à la fonction `exp` ?

$$\begin{aligned}\text{exp}(3, 6) &\rightarrow \\ \text{exp}(3, 5) * 3 &\rightarrow \\ \text{exp}(3, 4) * 3 * 3 &\rightarrow \\ \text{exp}(3, 3) * 3 * 3 * 3 &\rightarrow \\ \text{exp}(3, 2) * 3 * 3 * 3 * 3 &\rightarrow \\ \text{exp}(3, 1) * 3 * 3 * 3 * 3 * 3 &\rightarrow \\ \text{exp}(3, 0) * 3 * 3 * 3 * 3 * 3 * 3 &\rightarrow \\ 1 * 3 * 3 * 3 * 3 * 3 * 3 &= 729\end{aligned}$$

Est-ce qu'on peut faire mieux ?

$$\begin{aligned}r \text{ est pair} : & m^{2*n} = (m^2)^n \\ r \text{ est impair} : & m^{2*n+1} = (m^2)^n * m\end{aligned}$$

Quelles sont les **bonnes** valeurs de n pour que ces équations soient bien définies ? $n > 0$ et $n \geq 0$. Ceci donne

$$\begin{aligned}\text{exp}(m, 0) &= 1 \\ \text{exp}(m, 2 * (n + 1)) &= \text{exp}(m * m, n + 1) \\ \text{exp}(m, (2 * n) + 1) &= \text{exp}(m * m, n) * m\end{aligned}$$

Complexité de `exp (II)`

Quel est le nombre d'appels à la fonction `exp(n, m)` ?

$$\begin{aligned}\text{exp}(3, 6) &\rightarrow \\ \text{exp}(9, 3) &\rightarrow \\ \text{exp}(81, 1) * 9 &\rightarrow \\ \text{exp}(6561, 0) * 81 * 9 &\rightarrow \\ 1 * 81 * 9 &= 729\end{aligned}$$

$$\begin{aligned}\text{exp}(3, 8) &\rightarrow \\ \text{exp}(9, 4) &\rightarrow \\ \text{exp}(81, 2) &\rightarrow \\ \text{exp}(6561, 1) &\rightarrow \\ \text{exp}(6561 * 6561, 0) * 6561 &= 1 * 6561 = 6561\end{aligned}$$

$$2 + \log_2 n = \log_2(2n) \quad \text{pour } n > 0$$

Induction sur les entiers pour compter (I)

Définition : Une **permutation** d'un ensemble fini A est une bijection de A dans A .

Exercice : Soit $n \geq 0$. Le nombre de permutations d'un ensemble de n éléments (noté P_n) est $n!$.

Induction sur les entiers pour compter (II)

Définition : Soit $n \geq 0$. Un **arrangement** d'ordre $k \leq n$ d'un ensemble A de n éléments est un sous-ensemble totalement ordonné de A ayant k éléments.

Exercice : Le nombre d'**arrangements** d'ordre k d'un ensemble A de n éléments (noté A_n^k) est $\frac{n!}{(n-k)!}$.

Induction sur les entiers pour compter (III)

Définition : Soit $n \geq 0$. Une **combinaison** d'ordre $k \leq n$ d'un ensemble A de n éléments est un sous-ensemble de A ayant k éléments.

Exercice : Le nombre de combinaisons d'ordre k d'un ensemble A de n éléments (noté C_n^k) est $\frac{n!}{(n-k)!k!}$.

Fonction entières : la suite de Fibonacci

On définit par récurrence les entiers de Fibonacci comme la suite ayant les propriétés suivantes :

$$fib(0) = 1$$

$$fib(1) = 1$$

$$fib(n) = fib(n-1) + fib(n-2) \text{ pour } n \geq 2$$

On peut montrer par induction que, pour $n > 0$, on a

$$fib(n) = \frac{\sqrt{5}+1}{2\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{\sqrt{5}-1}{2\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n$$

C'est donc une fonction exponentielle!

Fibonacci (II)

En Ocaml :

```
# let rec fib =
  function
    0 -> 1
  | 1 -> 1
  | n -> fib(n-1)+fib(n-2);;
val fib : int -> int = <fun>
```

Le nombre d'appels récursifs à la fonction fib suit la même loi que fib elle-même!

Fibonacci (III)

Peut-on faire mieux ?

Solution 1 : utiliser la forme close, mais cela dépend de la précision des flottants.

Solution 2 : travailler avec des couples

$$(fib(n), fib(n-1)) = (fib(n-1) + fib(n-2), fib(n-1)) \\ = (fun(x,y) -> (x+y,x))(fib(n-1), fib(n-2))$$

```
let fibetape (x,y) = x+y,x;;
val fibetape : int * int -> int * int = <fun>
```

```
let rec iter f a = function 0 -> a | n -> f(iter f a (n-1));;
val iter : ('a -> 'a) -> 'a -> int -> 'a = <fun>
```

```
let fibcuple n = iter fibetape (1,1) n;;
```

```
val fibcouple : int -> int * int = <fun>
```

```
let fastfib n = snd (fibcouple n);;  
val fib : int -> int = <fun>
```

Exercice : prouver par induction que $\text{fibcouple } n = (\text{fib } n+1, \text{fib } n)$.

En déduire que $\text{fastfib } n = \text{fib } n$.

Recherche d'équations récursives à un argument

Pour définir une fonction récursive f , on peut chercher à exprimer f sous la forme d'une équation de la forme :

$$f(x) = g(x, f(h(x)))$$

- h représente la façon dont on passe de la valeur x à une valeur "plus petite" $h(x)$ sur laquelle on procède à l'appel récursif $f(h(x))$. Cette fonction doit vérifier $h(x) < x$ pour un ordre < bien fondé.
- g traduit la façon dont on utilise le résultat de l'appel récursif pour décrire la valeur de $f(x)$.

Exemple de la fonction Sigma

$\text{Sigma} : \mathbb{N} \rightarrow \mathbb{N}$ est définie par :

$$\text{Sigma}(n) = n + (n-1) + (n-2) + \dots + 1 + 0$$

L'équation récursive est

$$\text{Sigma}(x) = x + \text{Sigma}(x-1)$$

On a donc $h(x) = x-1$ et $g(n, m) = n + m$.

Cas particuliers dans les équations récursives

Il faut identifier les cas pour lesquels les équations posent problème :

- soit parce qu'elles donnent des résultats faux,
- soit parce qu'elles ne correspondent pas à des expressions bien formées ($h(x)$ n'est pas dans le domaine de f).

On traite ces cas à part, ce sont les **cas de base**.

Exemple de la fonction Sigma

La valeur $h(0)$ n'est pas dans le domaine de la fonction Sigma . On pose donc

$$\begin{aligned} \text{Sigma}(0) &= 0 \\ \text{Sigma}(x) &= x + \text{Sigma}(x-1), \text{ pour } x > 0 \end{aligned}$$

Un autre exemple

$\text{Pi} : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ est définie par :

$$\text{Pi}(n) = \begin{cases} n^2 \times (n-2)^2 \times (n-4)^2 \times \dots \times 2, & \text{si } n \text{ pair} \\ n^2 \times (n-2)^2 \times (n-4)^2 \times \dots \times 1, & \text{si } n \text{ impair} \end{cases}$$

L'équation récursive est

$$\text{Pi}(x) = x^2 \times \text{Pi}(x-2)$$

On a donc $h(x) = x-2$ et $g(n, m) = n \times m$.

Les valeurs $h(2)$ et $h(1)$ ne sont pas dans le domaine de la fonction. On pose donc

$$\begin{aligned} \text{Pi}(1) &= 1 \\ \text{Pi}(2) &= 4 \\ \text{Pi}(x) &= x^2 \times \text{Pi}(x-2), \text{ pour } x > 2 \end{aligned}$$

Équations récursives à plusieurs arguments

On généralise le schéma récursif :

$$f(y, x) = g(y, x, f(i(y), h(x)))$$

- $h(x)$ produit une valeur "plus petite" que x . C'est l'argument sur lequel on fait la récurrence.
- $i(y)$ produit une valeur différente de y (pas forcément plus petite).
- g traduit la façon dont on utilise le résultat de l'appel récursif pour décrire la valeur de $f(y, x)$.

Exemple de la fonction exp (première version)

Mettre l'équation $\text{exp}(y, x) = y \times \text{exp}(y, x-1)$ sous la forme

$$\text{exp}(y, x) = g(y, x, \text{exp}(i(y), h(x)))$$

on obtient :

$$h(x) = x-1 \quad i(y) = y \quad g(y, x, z) = y \times z$$

ayant comme cas particulier :

$$\text{exp}(y, 0) = 1$$

Exemple de la fonction exp (deuxième version)

Mettre

$$\begin{aligned}\text{exp}(m, 0) &= 1 \\ \text{exp}(m, 2 * (n + 1)) &= \text{exp}(m * m, n + 1) \\ \text{exp}(m, (2 * n) + 1) &= \text{exp}(m * m, n) * m\end{aligned}$$

sous la forme

$$\text{exp}(y, x) = g(y, x, \text{exp}(i(y), h(x)))$$

On obtient

$$\begin{aligned}h(x) &= x \text{ div } 2 \\ i(y) &= y \times y \\ g(y, x, z) &= \begin{cases} z & \text{si } x \text{ pair} \\ y \times z & \text{si } x \text{ impair} \end{cases}\end{aligned}$$

ayant comme cas particulier

$$\text{exp}(y, 0) = 1$$

car $h(0) \neq 0$.