



(Séquence 3.1

Récursion sur entiers naturels



Définition récursive : factorielle

Spécification **informelle** de la factorielle :

$$n! = 1 * 2 * 3 * \dots * n$$



Définition récursive : factorielle

Spécification **informelle** de la factorielle :

$$n! = 1 * 2 * 3 * \dots * n$$

Définition ambiguë : a-t-on $3! = 1 * 2 * 3 * 3$?



Définition récursive : factorielle

Spécification **informelle** de la factorielle :

$$n! = 1 * 2 * 3 * \dots * n$$

Définition ambiguë : a-t-on $3! = 1 * 2 * 3 * 3$?

Définition **récursive** de factorielle n :

$$n! = 1 \quad \text{pour} \quad n = 0$$

$$n! = n * (n - 1)! \quad \text{pour} \quad n \geq 1$$



Définition récursive : puissance

Spécification **informelle** de la puissance :

$$X^n = X * X * X * \dots * X$$



Définition récursive : puissance

Spécification **informelle** de la puissance :

$$x^n = x * x * x * \dots * x$$

Définition **récursive** de x puissance n :

$$\begin{aligned} x^0 &= 1 \\ x^n &= x * x^{n-1} \quad \text{pour } n \geq 1 \end{aligned}$$



Principes

- ▶ **Décomposition** : $f(n) = \dots f(p) \dots$
Exprimer $f(n)$ en fonction de $f(p)$ avec $p < n$
- ▶ **Cas de base** : $f(0) = \dots$
Donner la(les) valeur(s) de f pour la(les) valeur(s) de base



Principes

- ▶ **Décomposition** : $f(n) = \dots f(p) \dots$
Exprimer $f(n)$ en fonction de $f(p)$ avec $p < n$
- ▶ **Cas de base** : $f(0) = \dots$
Donner la(les) valeur(s) de f pour la(les) valeur(s) de base

Fonctionne car les entiers naturels forment un « domaine bien fondé » où il ne peut y avoir de suite infiniment décroissante.



Définition en 3 cas

Une autre définition de la puissance en trois cas :

$$\begin{aligned}x^0 &= 1 \\x^{2n} &= (x^n)^2 \quad \text{pour } n \geq 1 \\x^{2n+1} &= x * (x^n)^2 \quad \text{pour } n \geq 1\end{aligned}$$



Double récursion

Les nombres de Fibonacci sont définis par une double récursion :

$$fib(0) = 1$$

$$fib(1) = 1$$

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



Récursion enveloppée



La fonction 91 de McCarthy :

$$f(n) = n - 10 \quad \text{pour} \quad n > 100$$

$$f(n) = f(f(n + 11)) \quad \text{pour} \quad n \leq 100$$



En Scheme

- ▶ Définition (courante) d'une fonction récursive sur entier naturel (c'est-à-dire les nombres 0, 1, 2, etc.) :

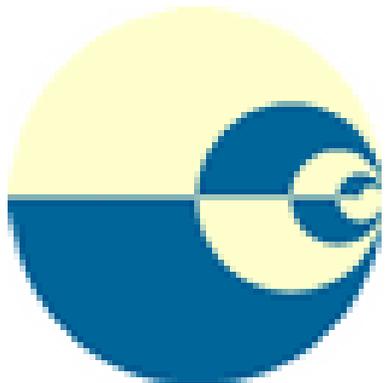
```
(define (f n)
  (if (> n 0)
    expression fonction de f(n - 1)
    cas de base
  ))
```

- ▶ Évaluation de la factorielle

```
(f 3) ≡ (* 3 (f 2)) ≡ (* 3 (* 2 (f 1)))
≡ (* 3 (* 2 (* 1 (f 0))))
≡ (* 3 (* 2 (* 1 1)))
≡ (* 3 (* 2 1))
≡ (* 3 2)
≡ 6
```

```
(f 0) cas de base: arrêt des appels récursifs
```





Fin séquence)

