

Preuves par induction

Récapitulatif

MP2I Lycée Pierre de Fermat

Nous disposons maintenant de trois outils de preuves :

- la récurrence (faible ou forte) ;
- l'induction bien fondée ;
- l'induction structurelle.

Il n'est pas nécessaire, dans la suite, de comprendre les fondements de l'induction structurelle ou de l'induction bien fondée, mais il faut savoir faire des preuves utilisant ces trois schémas. Lorsque l'on étudie une fonction récursive, on commence par voir quel type de preuve sera plus adapté

- La récurrence sera utilisable si un des paramètres est un entier positif strictement décroissant, ou si l'on peut fabriquer un variant entier positif strictement décroissant. Par exemple : exponentiation rapide, dichotomie, etc...
- L'induction bien fondée sera utilisable si les paramètres peuvent être munis d'un ordre (produit, lexicographique, ou autre) bien fondé. Ex : fonction d'Ackermann, calcul du pgcd de deux entiers par la méthode naïve, etc...
- L'induction structurelle pour les types sommes (sauf cas particuliers)

Preuve par induction structurelle

On commence par énoncer le principe d'induction pour le type que l'on considère. Lorsque vous serez à l'aise avec l'induction, vous n'aurez plus besoin de faire cette étape, mais au début ça pourra vous aider à comprendre ce que l'on fait.

Par exemple, pour le type des listes :

Théorème 1 (Principe d'induction structurelle sur les listes). Soit P une propriété sur les listes telle que :

- $P([])$ est vraie
- Pour toute liste l de la forme $x :: q$ où x est un élément et q une liste, $P(q) \Rightarrow P(l)$

Alors P est vraie pour toute liste.

Autre exemple, pour le type suivant :

```
1 type expr =  
2   | Const of int (* constante entière *)  
3   | Var of string (* variable *)  
4   | Add of expr * expr (* somme de deux expressions *)
```

Théorème 2 (Principe d'induction structurelle sur les expressions). Soit P une propriété sur les expressions telle que :

- Pour tout $n \in \mathbb{Z}$, $P(\mathbf{Const}(n))$ est vraie
- Pour toute variable x , $P(\mathbf{Var}(x))$ est vraie
- Pour toute expression e de la forme $Add(e_1, e_2)$ où e_1, e_2 sont des expressions, si $P(e_1)$ et $P(e_2)$ alors $P(e)$

Alors P est vraie pour toute expression.

Et ainsi de suite pour tous les types que l'on peut imaginer. Ensuite, lorsqu'il faut faire une preuve, on applique simplement ce principe, en suivant quelques règles :

- On annonce la preuve que l'on va faire
- On annonce le nombre de cas qu'il y aura à traiter
- On n'oublie pas de mentionner quand apparaissent les hypothèses d'induction (HI)

Exemple 1. On considère le type suivant pour représenter les couleurs :

```

1 type couleur =
2   | Rouge
3   | Jaune
4   | Bleu
5   | Melange of couleur*couleur (* 50% de chaque *)
6
7 (* f_r c renvoie la proportion de rouge totale dans c *)
8 let f_r (c: couleur) : float =
9   match c with
10  | Rouge -> 1.0
11  | Melange(c1, c2) -> (f_r c_1 +. f_r c_2) /. 2.
12  | _ -> 0.

```

Notons $\mathcal{C}$ l'ensemble des couleurs.

On suppose avoir défini de même des fonctions $\boxed{f_j, f_b}$ renvoyant les proportions de jaune et de bleu. Montrons par induction structurelle sur $\mathcal{C}$ que :

$$\forall c \in \mathcal{C}, f_r(c) + f_j(c) + f_b(c) = 1$$

On a 4 cas :

- Si $c = \mathbf{Rouge}$, alors $f_r(c) = 1, f_j(c) = 0$ et $f_b(c) = 0$, donc la somme fait bien 1.
- Si $c = \mathbf{Jaune}$: idem
- Si $c = \mathbf{Bleu}$: idem
- Si c est de la forme $\mathbf{Melange}(c_1, c_2)$ avec $c_1, c_2 \in \mathcal{C}$: on a $f_r(c) = \frac{f_r(c_1) + f_r(c_2)}{2}$ et pareil avec f_j, f_b . Donc :

$$f_r(c) + f_j(c) + f_b(c) = \frac{f_r(c_1) + f_j(c_1) + f_b(c_1)}{2} + \frac{f_r(c_2) + f_j(c_2) + f_b(c_2)}{2}$$

Or, par HI, $f_r(c_1) + f_j(c_1) + f_b(c_1) = 1$, et $f_r(c_2) + f_j(c_2) + f_b(c_2) = 1$. Donc, $f_r(c) + f_j(c) + f_b(c) = \frac{1}{2} + \frac{1}{2} = 1$

D'où la propriété annoncée.