

Démonstration d'équivalence (filter)

Exercice

Faire de même pour filter :

```
let rec filter (p : 'a -> bool) (l : 'a list) : 'a list = match l with
| [] -> []
| head :: tail when p head -> head :: (filter p tail)
| head :: tail -> filter p tail
```

```
let rec fold_right (f : 'b -> 'a -> 'a) (l : 'b list) (z : 'a) : 'a =
  match l with
  | [] -> z
  | head :: tail -> f head (fold_right f tail z)
```

Théorème :

```
filter p l = fold_right (fun x xs -> if p x then x :: xs else xs) l []
```

Preuve par récurrence structurelle sur la liste l

Cas de base: $l = []$

• à gauche: $\text{filter } p [] = []$

• à droite: $\text{fold_right } (\underbrace{\text{fun } x \text{ } xs \rightarrow \text{if } p \ x \text{ then } x :: xs \text{ else } xs}_{f}) \underbrace{[]}_l \underbrace{[]}_z$
 $= z = []$

les 2 côtés sont égaux, alors le cas de base est vérifié

Réurrence:

- Hypothèse de récurrence (HR): (supposons que c'est vrai pour une certaine liste $l = ys$)

```
filter p ys = fold_right (fun x xs -> if p x then x :: xs else xs) ys []
```

- Vérifions pour $l = y :: ys$

À gauche:
$$\text{filter } p (y :: ys) = \begin{cases} y :: (\text{filter } p \text{ } ys) & \text{si } (p \ y) = \text{true} \\ \text{filter } p \text{ } ys & \text{si } (p \ y) = \text{false} \end{cases}$$

$$\text{À droite: } \text{fold_right } \underbrace{(\text{fun } x \text{ } xs \rightarrow \text{if } p \ x \text{ then } x :: xs \text{ else } xs)}_f \ (y :: ys) \ []$$

$$= f \ y \ (\text{fold_right } f \ ys \ []) \quad (\text{par définition de fold_right})$$

$$= f \ y \ (\text{filter } p \ ys) \quad (\text{par HR})$$

Il reste à appliquer la fonction f :

$$= (\text{fun } x \text{ } xs \rightarrow \text{if } p \ x \text{ then } x :: xs \text{ else } xs) \ y \ (\text{filter } p \ ys)$$

$$= \begin{cases} y :: (\text{filter } p \ ys) & \text{si } (p \ y) = \text{true} \\ \text{filter } p \ ys & \text{si } (p \ y) = \text{false} \end{cases} \quad (\text{par définition de la fonction } f)$$

Comme l'expression à gauche est égale à l'expression à droite,
la récurrence est vérifiée.

