## Feuille 10: Plus sur les listes

**Exercice 10.1** Les fonctions sum, prod et op\_prod vues précédemment suivent le même schéma de récursion. On peut écrire deux fonctions pour décrire de tels schémas :

- Si 1 est la liste [b1; ...; bn], left\_fold op a 1 vaut op (...(op (op a b1) b2)...) bn.
- Si 1 est la liste [a1; ...; an], right\_fold op 1 b vaut op a1 (op a2 (...(op an b)...)).

Cet exercice demande de comprendre et programmer ces fonctions. Elles seront ensuite utilisées pour reprogrammer des fonctions déjà vues.

- 1. Pour comprendre ces schémas, calculer à la main
  - left\_fold op a 1 et
  - right\_fold op 1 b

pour op la fonction fun  $x y \rightarrow x + y$ , 1 la liste [1;2;3], et a et b valant 0.

- 2. Écrire la fonction right\_fold.
- 3. Écrire la fonction left fold.

Note : Ces fonctions existent dans la bibliothèque standard sous les noms List.fold\_right et List.fold\_left.

Exercice 10.2 En utilisant les fonctions List.fold\_left et/ou List.fold\_right, écrire ou réécrire les fonctions suivantes.

- 1. Une fonction length 1 qui calcule la longueur de la liste 1.
- 2. Une fonction reverse 1 qui calcule la liste renversée de 1.
- 3. Une fonction maximum 1 qui calcule le maximum de la liste d'entiers 1.
- 4. Une fonction filter p 1 , où p est un prédicat s'appliquant aux éléments de 1 , qui calcule la liste des éléments de 1 qui satisfont le prédicat p .
- 5. Une fonction remove\_duplicates 1 qui ne garde qu'une occurrence de chaque élément de 1.
- 6. La fonction append qui concatène deux listes.
- 7. La fonction map f 1 écrite dans la feuille 1.