
Fonctions récursives sur les listes

Exercice 1.

Fonctions récursives

Écrivez les fonctions suivantes de manière récursive en utilisant les primitives `liste()`, `tete(l)`, `queue(l)`, `vide(l)` et `cons(a, l)` (on suppose que toutes les listes contiennent des entiers) :

1. `ajoute(x, l)` qui ajoute x à chacun des éléments de l ;
2. `paire(l)` qui teste si la liste l ne contient que des entiers pairs ;
3. `avant_dernier(l)` qui renvoie l'avant-dernier élément de la liste l ;
4. `multiple(l)` qui calcule le plus petit commun multiple des éléments de l . On pourra supposer que l'on a déjà une fonction `ppcm(a, b)` qui calcule le plus petit commun multiple de deux entiers ;
5. `incremente(l)` qui incrémente chaque élément de la liste de son rang (l'élément en position 0 n'est pas modifié, celui en position 1 est incrémenté de 1, celui en position 2 de 2, etc.). Par exemple, sur la liste 2, 9, 5, 22 la fonction doit renvoyer 2, 10, 7, 25.
6. `alterne(l)` qui teste si une liste est *alternée*, c'est-à-dire si son premier élément est plus petit que le second, que le second est plus grand que le troisième, le troisième est plus petit que le quatrième, le quatrième plus grand que le cinquième et ainsi de suite.
7. `repete(n, i, l)` qui répète n fois l'élément de l en position i . Par exemple si $n = 3$, $i = 2$ et $l = 1, 2, 3, 4$, la fonction doit renvoyer la liste 1, 2, 3, 3, 3, 4 (la numérotation des positions commence à 0) ;
8. `applique(f, l1, l2)` qui applique un certain nombre de fois la fonction f aux éléments de $l1$. On suppose que la liste $l2$ est de même longueur que $l1$ et c'est elle qui donne pour chacun des éléments de $l1$ le nombre de fois qu'il faut appliquer f . Par exemple, si $l1 = 1, 2, 3$ et $l2 = 2, 0, 1$, la fonction doit renvoyer la liste $f(f(1)), 2, f(3)$;