

Programmation fonctionnelle

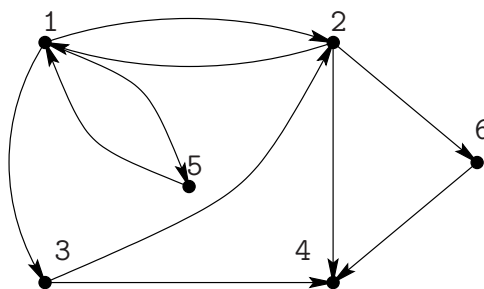
Répétition 7

26 avril 2007

Exercice sur les graphes

1.♣ Ecrire une fonction `inv` qui, à partir d'un graphe orienté, construit le graphe retourné. Nous convenons de représenter un graphe comportant n nœuds par une liste de n listes; la liste correspondant au nœud x a pour premier élément x et pour éléments suivants les successeurs (immédiats) de x .

Par exemple le graphe



sera représenté par la liste

((1 2 3 5) (2 1 4 6) (3 2 4) (4) (5 1) (6 4))

Fonctions comme argument et retour

2.♣ Définir les fonctions `symmetrize` et `anti-symmetrize`, qui prennent comme argument une fonction $f : R \rightarrow R$ et qui renvoient respectivement les fonctions

$$f' : R \rightarrow R \quad x \mapsto \frac{f(x) + f(-x)}{2}$$

et

$$f' : R \rightarrow R \quad x \mapsto \frac{f(x) - f(-x)}{2}$$

Définir ensuite une fonction `func-op` à trois arguments, un opérateur `op` de $R \times R \rightarrow R$, et deux fonctions unaires `f` et `g`. `func-op` renvoie la fonction unaire `h` telle que

$$\forall x \in R : h(x) = \text{op}(f(x), g(x))$$

Redéfinir ensuite `symmetrize` et `anti-symmetrize` à partir de `func-op`.

Exercices d'examen

3.* Ecrire une fonction `longest-inc` retournant la plus longue sous-liste croissante de la liste d'entiers donnée en argument.

4.* Une *tricoupure* d'une liste ℓ est une liste de trois listes non vides dont la concaténation (dans l'ordre) vaut ℓ . Ecrire une fonction `tricoup` qui à toute liste ℓ associe la liste des tricoupures de ℓ .

Par exemple, si ℓ est `(a b)` la liste des tricoupures de ℓ est la liste vide ; si ℓ est `(a b c d)` la liste des tricoupures de ℓ comporte, dans un ordre quelconque, les trois listes `((a b) (c) (d))`, `((a) (b c) (d))` et `((a) (b) (c d))`.

Variante

Une *tricoupure* d'une liste ℓ est une liste de trois listes dont la concaténation (dans l'ordre) vaut ℓ . Ecrire une fonction `tricoupvide` qui à toute liste ℓ associe la liste des tricoupures de ℓ .

Exercices proposés

5.* Ecrire une fonction qui renvoie la liste des permutations d'une liste donnée.

6.* Ecrire une fonction `nbsum` qui prend comme argument un nombre n et qui renvoie le nombre de façons d'écrire une somme égale à n (on comptera une seule fois les commutations).

7.* Ecrire une fonction qui prend en argument une liste et renvoie cette liste dont on a supprimé 3 éléments sur 5. On commence par garder les deux premiers éléments, puis on supprime les trois suivants, puis on garde les deux suivants, ...