

<p style="text-align: center;"><b>info401 : Programmation fonctionnelle</b> <b>TD 6 : références</b></p>
--

Pierre Hyvernat

Laboratoire de mathématiques de l'université de Savoie

bâtiment Chablais, bureau 22, poste : 94 22

email : [Pierre.Hyvernat@univ-savoie.fr](mailto:Pierre.Hyvernat@univ-savoie.fr)

www : <http://www.lama.univ-savoie.fr/~hyvernat/>

wiki : <http://www.lama.univ-savoie.fr/wiki>

*Question 1.* Imaginez que vous avez un gros programme comportant de nombreuses fonctions. Une de ces fonctions, la fonction  $f$ , vous intéresse particulièrement, et vous aimeriez savoir combien de fois vous l'appellez lors d'une exécution de votre programme.

Expliquez comment vous procédez. (Sans utiliser de références.)

*Question 2.* Même question, mais en utilisant une référence. Est-ce que la transparence référentielle est conservée ?

Quelle solution préférez-vous ?

*Question 3.* Toujours à propos de la fameuse fonction  $f$ , vous aimeriez optimiser le calcul des valeurs. Lors du calcul de  $f\ x$ , vous regardez si vous avez déjà calculé la valeur correspondante.

- Si oui, vous renvoyez la valeur déjà calculée
- si non, vous calculez  $f\ x$  normalement, et conservez la valeur quelque part.

(Cette technique s'appelle "mémoization". Le deuxième calcul de  $f\ x$  est en général beaucoup plus rapide que le premier...)

Expliquez comment vous procéderiez, sans utiliser de référence.

Que pensez-vous de la complexité de `memoization f` par rapport à celle de `f` ?

*Question 4.* Même question, mais en utilisant des références. Que pensez-vous de la complexité de `memoization f` par rapport à celle de `f` ?

Est-ce que la transparence référentielle est conservée ?

Quelle solution préférez-vous ?

*Question 5.* En utilisant les remarques des questions précédentes, écrivez une fonction `memoization : ('a->'b) -> ('a->'b)` qui transforme une fonction (presque) quelconque en une version mémoisée.

Est-ce que la transparence référentielle est conservée ?

*Question 6.* Expliquez comment vous pourriez améliorer encore un peu la complexité de votre fonction...