

Programmation : TP 6

Objectifs : [Ocaml, C] Abstraction. Passage de fonction en paramètre.

On continue ... à programmer en style récursif (pas de `for` ni de `while`, ni de références).

[O'Caml,C] Factorisation des calculs. Abstraction

1. Définir en O'Caml la fonction $f(x, y) = (x+y)^3$ sans `let in` et sans définir d'autres fonctions préliminaires. Réécrire cette fonction en utilisant un `let in` permettant de mettre en facteur une expression. Réécrire f sans utiliser `let in` et en faisant en sorte que l'expression $x + y$ n'apparaisse qu'une fois dans la définition (utiliser la sémantique donnée dans le cours pour le `let in`). La récursivité terminale n'est pas obligatoire.
2. Faire la démarche analogue en C : écrire la fonction f sans définir de variable intermédiaire ni de fonctions préliminaires. Réécrire cette fonction en utilisant une variable intermédiaire permettant de mettre en facteur une expression. Réécrire f sans utiliser de variable intermédiaire et en faisant en sorte que l'expression $x + y$ n'apparaisse qu'une fois (définir une fonction intermédiaire).

[O'Caml] Composition de fonctions, passage de fonction en paramètre

En O'Caml, une fonction est une expression comme une autre, que l'on peut passer en paramètre d'une autre fonction.

1. Écrire la fonction `sigma` qui correspond au symbole Σ dans une sommation $\sum_{i=a}^b f(i)$:
 - (a) On suppose que a et b sont des entiers ; $\sum_{i=a}^b f(i)$ signifie alors $f(a) + f(a+1) + \dots + f(b)$. La fonction `sigma` a 3 arguments (a , b et f).
 - (b) Généraliser avec un incrément quelconque (i.e. pas nécessairement 1).
 - (c) Généraliser en supposant qu'il existe une fonction s qui calcule l'argument suivant : $f(a) + f(s(a)) + f(s(s(a))) + \dots + f(b)$
 - (d) Généraliser avec une condition quelconque pour les éléments : $f(a) + f(s(a)) + \dots + f(x)$ où x est le dernier argument tel que $cond(x)$ soit vrai ($cond$ est un prédicat).
 - (e) Généraliser en permettant le remplacement de $+$ par n'importe quelle fonction binaire (possédant un élément neutre).

[C] Passage de fonctions en paramètres

En C, il n'est pas possible de passer des fonctions en paramètres, mais seulement des **noms de fonctions**. Cette restriction ne doit pas en limiter l'usage. Il faut donc écrire la signature complète de la fonction passée en paramètre dans le prototype.

1. Reprendre les différentes variantes de la fonction `sigma` précédente en C.

[C, O'Caml] Quelques applications de sigma

1. Écrire en C la fonction factorielle (en utilisant la fonction `sigma`).
2. Écrire en O'Caml (toujours en utilisant la fonction `sigma`) :
 - (a) la fonction factorielle ;
 - (b) le calcul d'une approximation de π :

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

- (c) une fonction `integrale` qui calcule une approximation de $\int_a^b f(x)dx$ en prenant pour dx un «petit» réel.