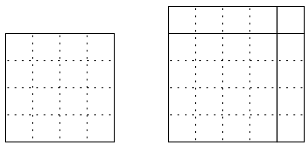
Département d'Informatique
LICENCE 2ème année
Mardi 7 octobre 2008

Travaux Dirigés de Programmation Fonctionnelle
«Programmer c'est prouver»

I Calcul figuré

Un nombre carré est l'aire d'une figure carrée (voir les exemples ci-dessous). En déduire une définition récursive du carré d'un nombre en utilisant uniquement l'addition et la multiplication par 2.



$4^2 = 16$ $\quad$ $5^2 = 25$

En procédant de même, donner une définition récursive du cube d'un nombre.

II Extraction de chaîne

Définir une fonction qui, étant données deux chaînes de caractères $s1$ et $s2$, teste si $s1$ est une *sous-chaîne* de $s2$ au sens où $s1$ est contenue à une certaine position dans $s2$. Exemple : `"BA"` est une sous-chaîne de `"ABACAB"`. On utilisera la fonction `String.sub` (cf. question II du TD précédent).

Définir une fonction qui, étant données deux chaînes de caractères $s1$ et $s2$, teste si $s1$ est une *chaîne extraite* de $s2$ au sens où $s1$ est obtenue à partir de $s2$ en otant des caractères. Exemple : `"BA"` est une chaîne extraite de `"ABCAB"`.

En déduire une fonction qui, étant données deux chaînes de caractères $s1$ et $s2$, compte le nombre de manières différentes d'extraire $s1$ de $s2$. Exemple : il y a 5 manières différentes d'extraire la chaîne `"BA"` de la chaîne `"ABBCABA"`.

III Divisibilité

1. *Plus grand commun diviseur (algorithme d'Euclide)* : D'après Euclide, si a et b sont des entiers vérifiant $a > b > 0$ et r est le reste de la division euclidienne de a par b, alors $pgcd(a, b) = pgcd(b, r)$. Utiliser cette équation pour inventer une définition récursive du pgcd de 2 nombres entiers relatifs.

2. *Nombres premiers* : Un nombre entier naturel p est premier si et seulement si l'ensemble de ses diviseurs est $\{1, p\}$ et p est non nul et différent de 1. Écrire une fonction récursive qui, étant donnés 2 entiers naturel n et d, détermine s'il existe un diviseur de n inférieur ou égal à d et différent de 1.

   En déduire une fonction qui teste si un entier naturel est premier.

IV Solutions entières d'une équation polynomiale

On s'intéresse aux *équations polynomiales* de la forme

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$$

où les $a_i, i \in \{0, ..., n\}$ sont des nombres entiers et $a_0 \neq 0$.

On cherche à résoudre une telle équation dans les entiers naturels, c'est-à-dire trouver l'ensemble des valeurs entières de l'inconnue x, appelées solutions entières de l'équation, pour lesquelles l'égalité est vraie. On procédera de la manière suivante :

1. Définir une fonction qui, étant donné 2 entiers naturels a et b (avec b non nul), retourne le plus grand entier inférieur ou égal à b et qui divise a.
2. Pour les questions suivantes, nous prendrons le cas particulier $n = 4$. Définir une fonction qui, étant donnés $(a_4, a_3, a_2, a_1, a_0)$ et un entier naturel x non nul, détermine si x est une solution entière de l'équation.
3. Nous pouvons remarquer que l'équation peut s'écrire sous la forme

   $$x(a_4 x^3 + a_3 x^2 + a_2 x + a_1) = -a_0$$

   et donc que les solutions entières de l'équation divisent a_0. Définir une fonction qui, étant donnés $(a_4, a_3, a_2, a_1, a_0)$ et un entier naturel x non nul, retourne la plus grande solution entière inférieure ou égale à x (ou lève une exception si il n'existe pas de solution inférieure ou égale à x).
4. Comment généraliser notre solveur à n quelconque ?

V Opérateur Σ

Définir une fonctionnelle qui, étant donnés p, un entier naturel, et u, une suite entière, calcule la somme des p+1 premiers termes de u :

$$\sum_{k=0}^{p} u_k$$