UNIVERSITÉ LOUIS PASTEUR STRASBOURG

Département d'Informatique
LICENCE 2ème année
Mercredi 10 octobre 2007

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

I Suite arithmétique

1. *Suite de Fibonacci* : La suite de Fibonacci est définie ainsi :

$$
\begin{aligned}
&fib(0) = 0,\\
&fib(1) = 1,\\
&fib(n) = fib(n-1) + fib(n-2), \quad n{>}1
\end{aligned}
$$

   Traduire cette définition récursive en Ocaml.

2. *Triangle de Pascal* : Le triangle de Pascal est ce triangle infini d'entiers naturels réprésenté ci-dessous. Les nombres qui y figurent, appelés les coefficients binomiaux (car ils apparaissent comme coefficients dans une formule célèbre, dite du binôme de Newton), jouent un rôle important dans diverses branches des mathématiques. Ces nombres satisfont à une loi simple : chaque nombre est la somme des deux nombres voisins placés au dessus de lui. En déduire la définition récursive d'une fonction qui retourne le nombre situé à une position donnée dans le triangle.

$$
\begin{array}{ccccccccccccccccc}
 & & & & & & & & 1 & & & & & & & & \\
 & & & & & & & 1 & & 1 & & & & & & & \\
 & & & & & & 1 & & 2 & & 1 & & & & & & \\
 & & & & & 1 & & 3 & & 3 & & 1 & & & & & \\
 & & & & 1 & & 4 & & 6 & & 4 & & 1 & & & & \\
 & & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & & \\
 & & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & & \\
 & 1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1 & \\
1 & & \ldots & & & & & & & & & & & & \ldots & & 1
\end{array}
$$

II 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. Ecrire 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.

III Séries entières

1. *Elévation à la puissance* : On cherche à calculer x^n avec x, un réel et n, un entier naturel. Écrire une première définition récursive en utilisant l'équation $x^n = x^{n-1} \times x$ pour $n > 0$. Écrire une deuxième version en utilisant l'équation $x^n = x^{n/2} \times x^{n/2}$ si n pair.
2. *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 :

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

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 peut-on généraliser notre solveur à n quelconque ?