

Programmation fonctionnelle en Caml

L3 Semestre 5

2015

TP 4 : Récursivité sur les nombres

1 Puissance naïve

1. Ecrire une fonction **puissance** qui calcule la puissance $n^{\text{ème}}$ (n entier) d'un réel x .
Méthode : il est facile d'exprimer x^n en fonction de x^{n-1} . Le cas de base et l'appel récursif seront distingués par filtrage.
2. Combien d'appels récursifs le calcul de **puissance (3.0 , 59)** exige-t-il ? Et plus généralement de puissance (x , n) ?

2 Écrire les fonctions récursives suivantes :

1. *repet(c,n)* qui construit le nombre obtenu en répétant n fois le chiffre c .

```
#repet(5,7);;  
- : int = 5555555
```
2. *unChiffre(n,c)*, fonction booléenne vrai ssi le nombre n n'est constitué que du chiffre c .

```
#unChiffre(555,5);;  
- : bool = true  
#unChiffre(55575,5);;  
- : bool = false
```
3. *pgd(n,d)*, qui retourne le plus grand nombre entier inférieur ou égal à d et divisant n .
(on rappelle que 1 divise tous les entiers...)

```
#pgd(18,11);;  
- : int = 9  
#pgd(11,9);;  
- : int = 1
```
4. *nbPairChif(n)*, fonction booléenne vrai ssi n contient un nombre pair de chiffres.

```
#npc(456789);;  
- : bool = true  
#npc(4567899);;  
- : bool = false
```

3 Multiplication égyptienne.

Les égyptiens du deuxième millénaire avant J.C. Utilisaient une numération additionnelle de base 10 (ce que vous connaissez de plus proche est la numération romaine...). Avec ce type de numération il est très simple d'ajouter deux nombres, donc de faire des produits par 2 et de faire des divisions entières par deux.. Pour les autres produits, c'est moins simple...

Les scribes égyptiens ont mis au point une technique de multiplication récursive n'utilisant que des sommes et des partages en 2. Elle peut se décrire avec le vocabulaire d'aujourd'hui de la manière suivante :

- Si n est pair, alors $n * p = (n/2) * (p+p)$
 - si n est impair, alors $n * p = (n-1) * p + p$
1. Ecrire une fonction ***mult_egypt*** (n , p) qui calcule $n * p$ en exploitant cette idée récursive.
 2. Décrivez à la main la suite des appels (récursifs) engendrés par ***mult_egypt*** (59,17).
 3. Vérifiez votre prédiction en traçant l'exécution de la fonction ***mult_egypt*** grâce à la fonction CAML **trace**.

4 Somme de chiffres :

On souhaite écrire en Caml des fonctions calculant la somme des chiffres d'un nombre entier représenté en base 10.

1. Définir une fonction récursive **s_chif** qui calcule la somme des chiffres d'un entier naturel. Exemple $s_chif(302091)=15$.
2. Ecrire une fonction **som_chif** qui poursuit le calcul de la somme des chiffres d'un nombre.jusqu'à se ramener à un seul chiffre.
 - Exemple $som_chif_n(18925) = 7$