# TD 3 - Fonctions Récursives de Cro-Magnon

**Exercice 1.** *Echauffement*

Montrer que les fonctions suivantes sont récursives primitives :

**1.** $\mathrm{mult}(n, m) = n \times m$

**2.** $\mathrm{pred}(n) = \begin{cases} 0 & \text{si } n = 0 \\ n - 1 & \text{sinon} \end{cases}$

**3.** $\mathrm{moins}(n, m) = \max(0, n - m)$

**4.** $\mathrm{power}(n, m) = n^m$

**5.** $\mathrm{sgn}(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{sinon} \end{cases}$

**Exercice 2.** *Un peu d'exotisme*

**1.** Montrer que la fonction qui à $n$ associe 1 si le chiffre 7 apparaît $n$ fois consécutives dans l'écriture décimale de $\pi$ et 0 sinon est récursive primitive.

**2.** Montrer que la fonction qui à $n$ associe $n^2$ si $n$ est pair, $n^3$ si $n$ est impair est récursive primitive.

**3.** Montrer que l'ensemble des nombres premiers est récursif primitif

**Exercice 3.** *Prédicats et minimisation bornée*

**1.** Montrer que les prédicats $\mathrm{egal}(u, v)$ ($u = v$) et $\mathrm{pref}(u, v)$ ($\exists x, v = ux$) sont récursifs primitifs.

**2.** Soit $Q$ un prédicat récursif primitif. Montrer que les prédicats

$$\begin{array}{rl} & \mathrm{exists}(n, \overline{m}) = \exists i \leq n,\ Q(i, \overline{m}) \\ \text{et} & \mathrm{forall}(n, \overline{m}) = \forall i \leq n,\ Q(i, \overline{m}) \end{array}$$

sont récursifs primitifs. Définir les analogues pour des prédicats sur les mots

**3.** Montrer que la fonction $f(n, \overline{m})$ qui vaut 0 si il n'existe pas d'entier $i \leq n$ tel que $Q(i, \overline{m})$, et qui vaut le $z$ tel que $Q(z, \overline{m})$ sinon est encore primitive récursive.

**4.** Montrer que la fonction qui à $n$ fait correspondre le $(n + 1)$-ième nombre premier est récursive primitive

**Exercice 4.** *Réduction de l'ensemble des fonctions de base*

**1.** Exprimer $p_1^2$ à l'aide de $p_1^3$, $s$ et de la fonction nulle. En déduire que l'ensemble des fonctions de base n'est pas minimal.

**2.** Se convaincre que de toute façon, on ne peut pas avoir un ensemble de fonctions de base *minimal* qui définisse le même ensemble de fonctions récursives.

**Exercice 5.** *Hacker Man*

Histoire d'être bien d'accord, on choisit comme définition de la fonction d'Ackermann :

$$\begin{cases} A(0,x) = x+1 \\ A(p,0) = p+1 \\ A(p+1,x+1) = A(p, A(p+1,x)) \end{cases}$$

**1.** Donner les expressions explicites de $A(1,x)$, $A(2,x)$, $A(3,x)$.

**2.** Que vaut $A(4,x)$ ?

**3.** Montrer que
- $\forall (p,x) \in \mathbb{N}^2, A(p,x) > x$ ;
- $A$ est strictement croissante en chacune de ses variables ;
- $A(p+1,x) > x + A(p,x)$ pour $p \geq 1$ ;
- $A(p,x+1) \leq A(p+1,x)$ pour $x \geq 1$.

**4.** On se propose de montrer que pour toute fonction récursive primitive $f$, il existe $p$ tel que,

$$\forall x_1, \ldots, x_k, \qquad f(x_1, \ldots, x_k) < A(p, \sup(x_1, \ldots, x_k))$$

On dit alors que $f$ est dominée par la fonction d'Ackermann.

1. Montrer que l'ensemble des fonctions dominées par la fonction d'Ackermann contient les fonctions de base.
2. Montrer que cet ensemble est clos par composition.
3. Montrer que cet ensemble est clos par récursion primitive, en établissant les points suivants :
   - montrer qu'il existe ue constante $c_1^*$ telle que

     $$\sup(g(\vec{n}), n_1, \ldots, n_k) < A(c_1^*, \sup(n_1, \ldots, n_k))$$

   - montrer qu'il existe une constante $c_2^*$ telle que

     $$\sup(h(r, n_0, \vec{n}), r, n_0, \ldots, n_k) < A(c_2^*, \sup(r, n_0, \ldots, n_k))$$

4. En déduire le résultat.

**5.** En déduire que la fonction $A$ n'est pas récursive primitive.

**Exercice 6.** *Il reste un bout de place sur la feuille*

**1.** Montrer que les fonctions $\mathrm{div}(n,m)$ et $\mathrm{rem}(m,n)$ (le quotient et le reste de la division euclidienne de $m$ par $n$) sont récursives primitives.