Université de Strasbourg **UFR de Mathématiques et Informatique**
Master 1 ILC et ISI **semestre d'automne**

Complexité et Calculabilité

Fonction d'Ackermann-Péter : μ-récursive, mais pas récursive primitive

La fonction d'Ackermann-Péter est une fonction à deux variables naturelles, définie par double récurrence comme suit :

$$A(0,n) = n+1 \ ,$$
$$A(m+1,0) = A(m,1) \ ,$$
$$A(m+1,n+1) = A\big(m, A(m+1,n)\big) \ .$$

Cette double récurrence peut être implémentée par une algorithme utilisant une boucle de type **while** :

> $Liste = [m,n]$ `;`
> `Tant que` $longueur(Liste) > 1$ `faire`
>   `si` $Liste$ `se termine par` $0,y]$`,`
>   `alors remplacer` $0,y$ `par` $y+1$
>   `sinon si` $Liste$ `se termine par` $x,0]$ `avec` $x>0$`,`
>   `alors remplacer` $x,0$ `par` $x-1,1$
>   `sinon si` $Liste$ `se termine par` $x,y]$ `avec` $x,y>0$`,`
>   `alors remplacer` x,y `par` $x-1,x,y-1$ `;`
> `Extraire le résultat` z `de la liste finale` $[z]$`.`

Cette fonction a les propriétés suivantes (pour tous naturels m,n) :

$$A(m,n) > n \ ,$$
$$A(m,n+1) > A(m,n) \ ,$$
$$A(m+1,n) > A(m,n) \ ,$$
$$A(m+1,n) \geq A(m,n+1) \ .$$

Elle prend les valeurs :

$$A(0,n) = n+1 \ ,$$
$$A(1,n) = n+2 = 2+(n+3)-3 \ ,$$
$$A(2,n) = 2n+3 = 2(n+3)-3 \ ,$$
$$A(3,n) = 2^{(n+3)}-3 \ ,$$
$$A(4,n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}-3 \text{ (avec } n+3 \text{ occurrences de 2)} \ ,$$

et ainsi de suite.

Pour tout naturel m *fixé*, la fonction à une variable $n \mapsto A(m,n)$ est récursive primitive. Pour toute fonction récursive primitive f à une variable, il existe un naturel m tel que pour tout naturel n, $A(m,n) > f(n)$; par conséquent pour $m' \geq m$ on a $A(m',n) > f(n)$, donc pour n suffisamment grand, $A(n,n) > f(n)$. Il s'ensuit que la fonction à une variable $n \mapsto A(n,n)$ n'est pas récursive primitive, donc la fonction A (à deux variables) ne l'est pas.