TD6 : Dérécursivation

TOP : Techniques and tOols for Programming – 1A

★ Exercice 1: Dérécursivation de fonctions sur les chaînes de caractères.

L'objectif de cet exercice est de revenir sur les fonctions sur les chaînes vues lors du TD2 afin de les dérécursiver. On rappelle les opérateurs de base du type chaîne :

$$\begin{cases} \texttt{Nil} \mapsto \text{La liste vide} & \\ \mathit{list}.\texttt{head} \mapsto \text{Premier caractère de la liste } \mathit{list} & \text{(défini ssi } \mathit{list} \text{ n'est pas vide)} \\ \mathit{list}.\texttt{tail} \mapsto \mathit{list} \text{ privée du premier élément} & \text{(défini ssi } \mathit{list} \text{ n'est pas vide)} \\ \mathit{entier} :: \mathit{list} \mapsto \text{Concaténation de l'entier } \mathit{entier} \text{ et de la liste } \mathit{list} & \end{cases}$$

▷ **Question 1:** $\mathit{est\_membre} : \begin{cases} \mathit{List} \times \mathit{Int} \mapsto \mathit{Bool} \\ \text{retourne VRAI ssi l'entier fait partie de la liste} \end{cases}$

▷ **Question 2:** $\mathit{occurence} : \begin{cases} \mathit{List} \times \mathit{Int} \mapsto \mathbb{N} \\ \text{retourne le nombre d'occurences de la valeur dans la liste} \end{cases}$

▷ **Question 3:** $\mathit{retourne} : \begin{cases} \mathit{List} \mapsto \mathit{List} \\ \text{retourne la liste lue en sens inverse} \end{cases}$

▷ **Question 4:** $\mathit{concat} : \begin{cases} \mathit{List} \times \mathit{List} \mapsto \mathit{List} \\ \text{le résultat est la concaténation des deux listes} \end{cases}$

▷ **Question 5:** $difference : \begin{cases} \mathit{List} \times \mathit{List} \mapsto \mathit{List} \\ \text{Le résultat est la liste de tous les éléments de list1 ne faisant pas partie de list2} \end{cases}$

▷ **Question 6:** $\mathit{nnaturels} : \begin{cases} \mathbb{N} \mapsto \mathit{List} \\ \text{résultat : une liste formée des n premiers entiers naturels} \end{cases}$

★ Exercice 2: Dérécursivation de l'exponentiation rapide.

On souhaite calculer (rapidement) x^n (x et n étant entiers).

- Si n est pair alors $x^n = \left(x^2\right)^{\frac{n}{2}}$. Il suffit alors de calculer $y^{n/2}$ avec $y = x^2$.
- Si n est impair et $n > 1$, alors $x^n = x \times \left(x^2\right)^{\frac{n-1}{2}}$. Il suffit de calculer $y^{\frac{n-1}{2}}$ avec $y = x^2$ et de multiplier le résultat par x.

Cela nous amène à l'algorithme récursif suivant qui calcule x^n pour un entier strictement positif n :

$$\mathit{puissance}(x,n) = \begin{cases} x, & \text{si } n = 1 \\ \mathit{puissance}(x^2, \frac{n}{2}), & \text{si } n \text{ pair} \\ x \times \mathit{puissance}(x^2, \frac{n-1}{2}), & \text{si } n \text{ impair } (n \neq 1) \end{cases}$$

▷ **Question 1:** Écrivez une fonction récursive calculant l'exponentiel d'un entier avec cet algorithme.

▷ **Question 2:** Quelle est la complexité de cet algorithme ?

▷ **Question 3:** Transformez cette fonction en une fonction récursive terminale.

▷ **Question 4:** Transformez la fonction obtenue en fonction itérative.

★ Exercice 3: Dérécursivation des tours de Hanoï.

```
HANOI(n,a,b) :
  si n = 1 alors deplacer(a,b)
           sinon hanoi(n-1, a, c)
                 deplacer(a, b)
                 hanoi(n-1, c, b)
  finsi
```

▷ **Question 1:** Dérécursivez cet algorithme. Comme cet algorithme n'est pas récursif terminal, il faut utiliser une pile. On y conservera l'état courant du programme, constitué des paramètres de la fonction récursive auxquels on ajoute un marqueur entier indiquant le numéro de l'appel récursif simulé (puisqu'il y en a 2).

▷ **Question 2:** Dessinez les états successifs de la pile lors de Hanoi(3,a,b)