Informatique théorique : feuille numéro 7
Récurrence et récursivité

Introduction

Dans la fiche précédente, on a vu le *principe de récurrence transfinie*, appelé aussi *principe de récurrence bien fondée.*

Une version "algorithmique" de ce principe permet d'écrire des *définitions récursives* de fonctions :
- Soit $(E, <)$ un ensemble bien fondé,
- Une définition *récursive* d'une fonction f de E dans F consiste à donner un moyen de calculer pour tout x de E la valeur de $f(x)$ en fonction de valeurs $f(y)$ *tels que* $y < x$. La fonction ainsi définie est totale (partout définie).

1 Exemples

1.1

L'exemple le plus banal est la fonction factorielle, définie sur $\mathbb{N}$ par

$$\begin{array}{rcl} 0! & = & 1 \\ n! & = & n \times (n-1) \textit{pour tout } n > 0 \end{array}$$

Cette définition est correcte car $(\mathbb{N}, <)$ est bien fondé et que $n - 1 < n$ pour tout $n > 0$.

1.2 La fonction de Fibonacci

C'est encore une fonction définie sur $\mathbb{N}$

$$\begin{array}{rcl} F(0) & = & 1 \\ F(1) & = & 1 \\ F(n) & = & F(n-1) + F(n-2) \textit{ pour tout } n \geq 2 \end{array}$$

1.3 La fonction d'Ackerman

C'est une fonction de domaine $\mathbb{N}^2$ et à valeurs dans $\mathbb{N}$

$$\begin{array}{rcl} A(0, n) & = & n + 1 \textit{ pour tout } n \in \mathbb{N} \\ A(m, 0) & = & A(m-1, 1) \textit{ pour tout } m \geq 1 \\ A(m, n) & = & A(m-1, A(m, n-1)) \textit{ pour tous } m, n \geq 1 \end{array}$$

1.3.1

Essayer de calculer $A(5,5)$ pour voir.

1.4 Exponentiation binaire

Soit $x \in \mathbb{R}$. Considérons la fonction définie sur $\mathbb{N}$ qui calcule x^n pour tout entier naturel n.

$$\begin{array}{rcl} Exp(0) & = & 1 \\ Exp(n) & = & x * Exp(n-1) \textit{ pour tout } n \geq 1 \end{array}$$

1.4.1

Ecrire une autre définition de l'exponentiation , qui utilise une fonction auxiliaire définie sur $\mathbb{N}^2$. Cette fonction auxiliaire calcule $a * x^n$ pour tout triplet (a, n, x) de $\mathbb{N}^2 \times \mathbb{R}$
L'entier a sert d'accumulateur (on y accumule les calculs intermédiaires). De plus votre fonction utilisera les égalités bien connues :

$$\begin{array}{rcl} x^n & = & (x^2)^{n/2} \text{ si } n \text{ est pair} \\ x^n & = & x \times x^{n-1} \text{ si } n > 0 \end{array}$$

2 Fonctions non arithmétiques

2.1 Chaînes de caractères

La relation "être plus courte" sur les chaînes de caractères est bien fondée. On peut donc définir une fonction sur les chaînes de caractères en donnant l'image de la chaîne vide et en exprimant l'image d'une chaîne non vide en fonction de l'image de chaînes plus courtes.

Si s est une chaîne non vide, on peut l'écrire soit cs', où c et son premier caractère et s' le reste de la chaîne. De même, on peut l'écrire $s'c'$ si c' est son dernier caractère.

La definition ci-dessous permet de calculer la valeur numérique d'une chaîne composée exclusivement de chiffres. On considèrera pour simplifier que la chaîne vide représente le nombre 0.

$$\begin{array}{rcl} \texttt{atoi ("")} & = & 0 \\ \texttt{atoi}(sc) & = & (\texttt{atoi}(s) \times 10 + c) \textit{ pour tout chiffre } c \textit{ et toute chaîne de chiffres } s \end{array}$$

Par exemple, on a le calcul suivant :

$$
\begin{aligned}
\texttt{atoi("342")} &= \texttt{atoi("34")} \times 10 + 2 \\
&= (\texttt{atoi("3")} \times 10 + 4) \times 10 + 2 \\
&= ((\texttt{atoi("")} \times 10 + 3) \times 10 + 4) \times 10 + 2 \\
&= 342
\end{aligned}
$$

3 Exercice

On considère des chaînes sur l'alphabet $\{\texttt{a,b,...,z}\} \cup \{0, 1, \ldots, 9\}$

3.1

Définir récursivement une fonction qui efface tous les chiffres d'une chaîne.

3.2

Définir récursivement une fonction qui teste si une chaine ne contient que des lettres.

4 Autres types de donnée

On pourra montrer des exemples sur des arbres binaires, entre autres.