Récursivité II

Tuan Nguyen

27 Janvier 2006

1 Exemples de fonctions récursives

Afin de poursuivre notre discussion sur la notion de récursivité, qui a été entamé par Mathieu Bélanger, nous débuterons par un résultat sur les fonctions et les relations récursives. Par la suite nous donnons quelques exemples de fonctions récursives.

Exercice 3.18 [1]
Montrons qu'une fonction $f(x_1, ..., x_n)$ est récursive si et seulement si la relation *graphe* $f(x_1, ..., x_n) = y$ est récursive.

(1) M.q. Si $f(x_1, ..., x_n)$est récursive, alors $f(x_1, ..., x_n) = y$ est récursive.
Nous avons qu'une relation est récursive si et seulement si sa fonction caractéristique est récursive.
Soit $C_{graf}(x_1, ..., x_n, y)$ la fonction caractéristique de la relation *graphe* de $f(x_1, ..., x_n)$.
Nous avons que

$$\begin{aligned} C_{graf}(x_1, ..., x_n, y) &= \begin{cases} 0 & \text{si } f(x_1, ..., x_n) = y \text{ est vrai} \\ 1 & \text{si } f(x_1, ..., x_n) = y \text{ est faux} \end{cases} \\ &= \begin{cases} 0 & \text{si } f(x_1, ..., x_n) = y \\ 1 & \text{si } f(x_1, ..., x_n) \neq y \end{cases} \\ &= Sg|f(x_1, ..., x_n) - y| \end{aligned}$$

(2) M.q. si la relation $f(x_1, ..., x_n) = y$ est récursive, alors $f(x_1, ..., x_n)$ est récursive. Nous avons que $C_{graf}(x_1, ..., x_n, y)$ est récursive et que $C_{graf}(x_1, ..., x_n, y) = 0$ si $f(x_1, ..., x_n) = y$ est vrai. Donc, par l'opérateur μ nous avons que $f(x_1, ..., x_n) = \mu y(C_{graf}(x_1, ..., x_n, y) = 0)$.

Exercice 3.19
(1)Montrons que $[\sqrt{n}]$ dénotant le plus grand entier $\leqslant \sqrt{n}$ est récursive primitive. Nous savons que pour un entier y quelconque si $y \leqslant \sqrt{n}$, alors $y^2 \leqslant n$, dans ce cas $[\sqrt{n}]$ dénote le plus grand entier y tel que $y^2 \leqslant n$. Prenons $R(y, n) : y^2 \leqslant n$ et sa fonction caractéristique

$$\begin{aligned} C_R(y, n) &= \begin{cases} 0 & \text{si } y^2 \leqslant n \text{ est vrai} \\ 1 & \text{si } y^2 \leqslant n \text{ est faux} \end{cases} \\ &= \begin{cases} 0 & \text{si } ((y^2 < n) \vee (y^2 = n)) \text{ est vrai} \\ 1 & \text{si } ((y^2 < n) \vee (y^2 = n)) \text{ est faux} \end{cases} \end{aligned}$$

[1] Mendelson E., Introduction to Mathemathical Logic, D. Van Nostrand,3 ed, P.139.

$C_R(y,n)$ est clairement récursive primitive et $[\sqrt{n}] = \left(\sum_{y<n} \overline{sg}(C_R(y,n)\right) \dot{-} 1$

(2) Montrons que $\prod(n)$ dénotantt le nombre de nombres premiers $\leqslant n$ est récursive primitive.
Soit $Pr(y)$: y est premier, sa fonction caractéristique

$$\begin{aligned} C_{Pr}(y) &= \begin{cases} 0 & \text{si } (\tau(y) \dot{-} 2) \wedge \overline{sg}|x-0| \wedge \overline{sg}|x-1| \text{ est vrai} \\ 1 & \text{si } (\tau(y) \dot{-} 2) \wedge \overline{sg}|x-0| \wedge \overline{sg}|x-1| \text{ est faux} \end{cases} \\ C_{Pr}(y) &= sg((\tau(y) \dot{-} 2) + \overline{sg}|x-0| + \overline{sg}|x-1|) \end{aligned}$$

Enfin, nous avons que $\prod(n) = \sum_{y \leqslant n} \overline{sg}(C_{Pr}(y)$.

2 Définitions par cas

Une autre manière d'obtenir des fonctions récursives à partir de fonctions et de relations récursives est de les définir par cas. La fonction caractéristique d'une relation est donnée par une définition par cas.

Proposition 3.18

Soit

$$f(x_1,...,x_n) = \begin{cases} g_1(x_1,...,x_n) \text{ si } R_1(x_1,...,x_n) \text{ vrai} \\ g_2(x_1,...,x_n) \text{ si } R_2(x_1,...,x_n) \text{ vrai} \\ . \\ . \\ g_k(x_1,...,x_n) \text{ si } R_k(x_1,...,x_n) \text{ vrai} \end{cases}$$

Si les fonctions $g_1,...,g_k$ et les relations $R_1,...,R_k$ sont récursives primitives (ou récursives), et si, pour tout $x_1,...,x_n$ il y a exactement une relation $R_m(x_1,...,x_n)$ qui est vraie pour $1 \leqslant m \leqslant k$, alors $f(x_1,...,x_n)$ est récursive primitive (ou récursive).

Preuve :
Soit $C_{R_1}(x_1,...,x_n),...,C_{R_k}(x_1,...,x_n)$ les fonctions caractéristiques respectives de $R_1(x_1,...,x_n),...,R_k(x_1,...,x_n)$ qui sont récursives, nous avons que

$f(x_1,...,x_n) = \sum_{l \leqslant k}(g_l(x_1,...,x_n) \cdot \overline{sg} C_{R_l}(x_1,...,x_n))$

Exercice 3.25
Montrons que

$$f(x) = \begin{cases} x^2 \text{ si x est pair} \\ x+1 \text{ si x est impair} \end{cases}$$

est récursive primitive.

$$\begin{aligned} f(x) &= \begin{cases} x^2 \text{ si x est pair} \\ x+1 \text{ si x est impair} \end{cases} \\ &= \begin{cases} x^2 \text{ si } 2|x \text{ vrai} \\ x+1 \text{ si } 2|x \text{ faux} \end{cases} \\ &= (x^2 \cdot \overline{sg}C_{(2|x)}) + ((x+1) \cdot sgC_{(2|x)}) \end{aligned}$$

3 Le codage de couple d'entiers

Notre étude des fonctions récursives a porté jusqu'à présent sur des fonctions définis dans les entiers et dans un contexte plus général la récursivité est attribuables aux fonctions qui font correspondre une suite d'entiers à un entier. En ce sens, nous allons considérer dans ce qui suit une foncion nous permettant de coder des couples d'entiers. La fonction que nous allons considérer nous permet d'obtenir une bijection entre l'ensemble des pairs d'entiers ordonnés et l'ensemble des entiers.
Une méthode systématique d'énumérer l'ensemble des pairs ordonnés est de les grouper comme suit :

$$\underbrace{\underbrace{\underbrace{(0,0)}_{1-suite}}_{1-ordre}}_{2-ordre} , \underbrace{\underbrace{(0,1),(1,0),(1,1)}_{2-suite}}_{2-ordre}, \underbrace{\underbrace{(0,2),(2,0),(1,2),(2,1),(2,2)}_{3-suite}}_{3-ordre},...$$

Donc, nous commençons par constituer la suite de tous les couples dont les éléments sont $\leqslant k$, puis nous constituons la suite de tous les nouveaux couples dont les éléments sont$\leqslant k+1$ dans l'ordre suivant :
$(0,k+1),(k+1,0),(1,k+1),(k+1,1),...,(k,k+1),(k+1,k),(k+1,k+1)$, en adjoignant cette nouvelle suite à la précédente nous obtenons une suite ordonné de tous les couples dont tous les éléments sont $\leqslant k+1$.

Ainsi, nous avons pour chaque $n-ordre$ il y a n^2 couples résultant d'une combinaison exhaustive de n éléments. De plus, le nombre de couples dans un $n-$ordre est égale à la somme des couples contenus dans les suite $n-suite$ et celles qui la précède.
Si $m=n$, alors (m,n) se situe dans le $(n+1)-ordre$ et (m,n) est au $(n+1)^2$ rang dans l'ordonnance.
Si $m<n$, alors (m,n) apparaît au $2m+1$ rang dans la $(n+1)-suite$ et il apparaît au (n^2+2m+1) ième rang dans le $(n+1)-ordre$, l'ajout de n^2 correspond aux n^2 couples de la $n-ordre$. De plus, (m,n) précède (n,m), donc (n,m) est au (n^2+2m+2) ième rang dans le $(n+1)-ordre$.
Cependant, si $m=n$, $(n+1)^2=(n^2+2m+1)$

En convenant que le couple (0,0) correspond à 0, nous définissons la fonction recherchée comme suit :

$$\begin{aligned} \sigma^2(m,n) &= \begin{cases} m^2+2n+1 \text{ si } m>n \\ n^2+2m \qquad \text{si } m \leqslant n \end{cases} \\ &= ((sg(m \dot{-} n)) \cdot (m^2+2n+1)) + ((\overline{sg}(m \dot{-} n)) \cdot (n^2+2m)) \end{aligned}$$

Définissons les fonctions inverses σ_1^2, σ_2^2 tel que

$$\begin{aligned} \sigma_1^2(\sigma^2(m,n)) &= m \\ \sigma_2^2(\sigma^2(m,n)) &= n \\ \sigma^2(\sigma_1^2(z), \sigma_2^2(z)) &= z \end{aligned}$$

Donc, $\sigma_1^2(z), \sigma_2^2(z)$ corespond respectivement au premier et au second élément du z-ième couple ordonné de notre énumération. Puisque, $\sigma^2(0,0) = 0$, nous avons que $\sigma_1^2(0) = 0$ et $\sigma_2^2(0) = 0$. Remarquons que notre énumération est faite de manière tel que pour un couple (m,n) :
si $m < n$ alors le couple qui suit (m,n) correspond au couple résultant d'une permutation entre m et n,
si $m > n$ alors le couple qui suit (m,n) est $(n+1, m)$,
si $m = n$ alors le couple qui suit (m,n) est $(0, n+1)$
Ainsi, nous avons que

$$\begin{aligned} \sigma_1^2(n+1) &= \begin{cases} \sigma_2^2(n) & \text{si } \sigma_1^2(n) < \sigma_2^2(n) \\ \sigma_2^2(n)+1 & \text{si } \sigma_1^2(n) > \sigma_2^2(n) \\ 0 & \text{si } \sigma_1^2(n) = \sigma_2^2(n) \end{cases} \\ &= (\sigma_2^2(n) \cdot (sg(\sigma_2^2(n) \dot{-} \sigma_1^2(n)))) + ((\sigma_2^2(n)+1) \cdot (sg(\sigma_1^2(n) \dot{-} \sigma_2^2(n)))) \\ &= \varphi(\sigma_1^2(n), \sigma_2^2(n)) \end{aligned}$$

Dans le cas de σ_2^2, pour un couple ordonné (m,n) :
si $m < n$ alors le couple qui suit (m,n) correspond au couple résultant d'une permutation entre m et n,
si $m > n$ alors le couple qui suit (m,n) est $(n+1, m)$,
si $m = n$ alors le couple qui suit (m,n) est $(0, m+1)$.
Par conséquent, si $m \neq n$, le couple qui suit (m,n) est $(-, m)$ et si $m = n$ le couple qui suit (m,n) est $(0, m+1)$.

$$\begin{aligned} \sigma_2^2(n+1) &= \begin{cases} \sigma_1^2(n) & \text{si } \sigma_1^2(n) \neq \sigma_2^2(n) \\ \sigma_1^2(n)+1 & \text{si } \sigma_1^2(n) = \sigma_2^2(n) \end{cases} \\ &= (\sigma_1^2(n) \cdot (sg|\sigma_1^2(n) - \sigma_2^2(n)|)) + ((\sigma_1^2(n)+1) \cdot (\overline{sg}|\sigma_1^2(n) - \sigma_2^2(n)|))) \\ &= \psi(\sigma_1^2(n), \sigma_2^2(n)) \end{aligned}$$

Il est clair que les fonctions $\sigma^2, \sigma_1^2, \sigma_2^2$ sont primitives récursives.

Nous pouvons généraliser par induction le résultat précédant pour obtenir des bijections récursives primitives entre des n-tuples ordonnés d'entiers et les entiers.
Soit $\sigma^k(x_1, ...x_k), \sigma_1^k(x), ..., \sigma_k^k(x)$ des fonctions primitives récursives tel que

$$\begin{aligned} \sigma_i^k(\sigma^k(x_1, ...x_k)) &= x_i \quad \text{pour} \quad 1 \leqslant i \leqslant k \\ \sigma^k(\sigma_1^k(x), ..., \sigma_k^k(x)) &= x \end{aligned}$$

pour $n = k = 2$ le fait est établi. Par hypothèse inductive, assumons que pour tout k et $n = k$ le fait est établi. À présent pour $n = k + 1$ en définissant

$$\begin{aligned}\sigma^{k+1}(x_1, ...x_k, x_{k+1}) &= \sigma^2(\sigma^k(x_1, ...x_k), x_{k+1}).\text{pour} 1 \leqslant i \leqslant k \\ \sigma_i^{k+1}(x) &= \sigma_i^k(\sigma_1^2(x)) \\ \sigma_{k+1}^{k+1}(x) &= \sigma_2^2(x))\end{aligned}$$

Nous avons que $\sigma^k(x_1, ...x_k), \sigma_1^k(x), ..., \sigma_k^k(x)$ sont primitives récursives.

4 Courses of values recursion

Comme nous avons vu, nous pouvons définir des fonctions par récursion en faisant dépendre les valeurs de $f(x_{1,..,x_n,y+1})$ de $f(x_{1,...,x_n,y})$, mais il est possible de définir des fonctions lesquelles dépendent de plusieurs ou de toutes les valeurs de $f(x_1, ..., x_n, u)$ pour $u \leqslant y$.

Soit $f\sharp(x_1, ..., x_n, y) = \prod_{u<y} P_u^{f(x_1,...,x_n,u)}$ et $f(x_1, ..., x_n, y) = (f\sharp(x_1, ..., x_n, y + 1))_y$

Rappel : soit la décomposition $x = p_0^{a_0} p_1^{a_1}, ..., p_k^{a_k}$ de x en produit de nombres premiers, la fonction $(x)_j$ est une fonction primitive récursive dénotant l'exposant a_j.

Proposition 3.19

Si $h(x_1, ..., x_n, y, z)$ est récursive primitive (ou récursive) et $f(x_1, ..., x_n) = h(x_1, ..., x_n, y, f\sharp(x_1, ..., x_n, y))$, alors $f(x_1, ..., x_n)$ est récursive primitive (ou récursive).

Preuve :

$$\begin{aligned}f\sharp(x_1, ..., x_n, 0) &= 1 \\ f\sharp(x_1, ..., x_n, y + 1) &= f\sharp(x_1, ..., x_n, y) \cdot P_y^{f(x_1,...,x_n,y)} \\ &= f\sharp(x_1, ..., x_n, y) \cdot P_y^{h(x_1,...,x_n,y,f\sharp(x_1,...,x_n,y))}\end{aligned}$$

Exemple

Prenons la suite de Fibonacci définit comme suit : $f(0) = f(1) = 1$ et $f(k + 2) = f(k) + f(k + 1)$ pour $k \geqslant 0$, montrons que cette fonction est récursive primitive. Nous avons que

$$\begin{aligned}f(0) = f(1) = 1 \\ f(2) = f(0) + f(1) \\ f(3) = f(1) + f(2) \\ ...\end{aligned}$$

Donc définissons $f(k)$ comme suit :

$$\begin{aligned}f(k) &= \begin{cases} 1 \text{ si } k = 0 \\ 1 \text{ si } k = 1 \\ ((f\sharp(k))_{\dot{k-1}} + (f\sharp(k))_{\dot{k-2}}) \text{ si } k > 1 \end{cases} \\ &= \overline{sg}(k) + \overline{sg}|k - 1| + ((f\sharp(k))_{\dot{k-1}} + (f\sharp(k))_{\dot{k-2}}) \cdot sg(\dot{k-1})\end{aligned}$$

puisque $h(y,z) = \overline{sg}(y) + \overline{sg}|y-1| + ((z)_{k-1} + (z)_{k-2}) \cdot sg(k \dot{-} 1)$
est primitive récursive et que
$f(k) = h(k, f\sharp(k))$
par la prop. 3.19 $f(k)$ est récursive primitive.

Corollaire 3.20
Si $H(x_1, ..., x_n, y, z)$ est une relation récursive primitive (ou récursive) et que $R(x_1, ..., x_n, y)$ est vrai si et seulement si $H(x_1, ..., x_n, y, C_R\sharp(x_1, ..., x_n, y))$, où C_R est la fonction caractéristique de $R(x_1, ..., x_n, y)$, alors $R(x_1, ..., x_n, y)$ est récursive primitive (ou récursive).

Preuve: $R(x_1, ..., x_n, y)$ est récursive primitive (ou récursive) si et seulement si sa fonction caractéristique $C_R(x_1, ..., x_n, y)$ est récursive primitive (ou récursive).

$$C_R(x_1, ..., x_n, y) \quad = \quad \begin{cases} 0 \text{ si } R(x_1, ..., x_n, y) \text{ est vrai} \\ 1 \text{ si } R(x_1, ..., x_n, y) \text{ est faux} \end{cases}$$

et $R(x_1, ..., x_n, y)$ est vrai si et seulement si $H(x_1, ..., x_n, y, C_R\sharp(x_1, ..., x_n, y))$ l'est.
Donc,$C_R(x_1, ..., x_n, y) = C_H(x_1, ..., x_n, y, C_R\sharp(x_1, ..., x_n, y))$, puisque $H(x_1, ..., x_n, y, z)$ est récursive primitive (ou récursive)sa fonction caractéristique l'est aussi nécessairement et par la prop. 3.19 $C_R(x_1, ..., x_n, y)$ est récursive primitive (ou récursive) et $R(x_1, ..., x_n, y)$ est donc récursive primitive (ou récursive).

Exercice 3.29
Soit $f_0, f_1, ...$ l'énumération de tous les fonctions récursives d'une variable, montrons que $f_x(y)$ n'est pas récursive primitive.

Supposons que $f_x(y)$ est récursive primitive. Dès lors, $f_x(x) + 1$ est aussi récursive primitive et $f_x(x) + 1 = f_k(x)$ pour tout x et un certain k. Mais, si $x = k$ alors $f_k(k) + 1 = f_k(k)$ ce qui est absurde. Donc, $f_x(y)$ n'est pas récursive primitive.