Leçon 912 : Fonctions recursives primitives et non primitives, Exemples.

L'objectif des fonctions recursives est de formaliser le concept de calculabilité pour les fonctions, et de caracteriser les fonctions pour lesquelles il est possible de trouver l'image de tout argument par un procédé systématique.

Pour des raisons de simplicité, seules les fonctions à valeurs entières seront présentées. (ie : $\mathbb{N}^k \to \mathbb{N}$).

On notera le n-uplet $(x_1, \dots, x_n)$ $\overline{x}^n$.

I Fonctions récursives primitives.

1.1 Construction

Def 1 : Les fonctions suivantes sont dites fonctions de base :

$0 : x \mapsto 0$, $\quad S : x \mapsto x+1$, $\quad \pi_i^n : \overline{x}^n \mapsto x_i$

Def 2 : Schéma de composition : la composition d'une fonction $f : \mathbb{N}^p \to \mathbb{N}$ et des fonctions $(g_i)_{i \leq p}$ $\mathbb{N}^k \to \mathbb{N}$. est la fonction

$\overline{x}^k \mapsto f(g_1(\overline{x}^k), \dots, g_n(\overline{x}^k))$

Def 3 : Schéma de recursion primitive (ou récurrence primitive) associé à $f : \mathbb{N}^p \to \mathbb{N}$ et $g : \mathbb{N}^{p+2} \to \mathbb{N}$

$$R_{f,g} : \begin{cases} (\overline{x}^p, 0) \mapsto f(\overline{x}^p) \\ (\overline{x}^p, S(n)) \mapsto g(\overline{x}^p, n, R_{f,g}(\overline{x}^p, n)) \end{cases}$$

Def 4 : Les fonctions recursives primitives sont les fonctions construites à partir des fonction de base et closes par les schémas de composition et de recursion primitive.

Exemples 5 : Les fonctions suivantes sont recursives

$+ : n, m \mapsto n+m = R_{f,g}(n,m)$ pour $f = \pi_1^1$, $g = n,m,k \mapsto S(k)$.

$\times : n,m \mapsto n \times m = R_{f,g}(n,m)$ pour $f = \pi_1^1$, $g = n,m,k \mapsto +(n,k)$

pred : $n \mapsto n-1$ si $n>0$, 0 sinon $= R_{f,g}(n,n)$ pour $f = 0$, $g = n,m,k \mapsto m$.

sg : $n \mapsto 1$ si $n > 0$, 0 sinon $= R_{f,g}(n,n)$ pour $f = 0$, $g = n,m,k \mapsto n$.

$- : n,m \mapsto n-m$ si $n \geq m$, 0 sinon $= R_{f,g}(n,m)$ pour $f = \pi_1^1$, $g = n,m,k \mapsto$ pred k

Application 6 : Schéma de boucle for.

Si le corps d'une boucle for est calculable par la fonction recursive primitive c alors n tours de cette boucle est aussi une fonction recursive primitive

$n, x \mapsto R_{f,g}(x,n)$ où $f = \pi_1^1$ et $g = x,n,k \mapsto c(k,n)$

Rq : Une fonction $f : \mathbb{N}^k \to \mathbb{N}^r$ est recursive primitive ssi $\forall i \leq r$ $\pi_i^r(f)$ l'est

1.2 Predicats récursifs primitifs

Def 7 : Un prédicat P est dit récursif primitif si sa fonction caractéristique $\pi_P\ \overline{x}^p \mapsto \begin{cases} 1 & \text{si } P(\overline{x}^p) \\ 0 & \text{sinon} \end{cases}$

est recursive primitive.

Exemples 8 : Les prédicats suivants sont recursifs primitifs :

$- \{m,n \mid m > n\}$ $\quad \pi_> = m,n \mapsto sg(-(m,n))$

$- \{n \mid n = 0\}$ $\quad \pi_0 = -(S(0(n)), sg(n))$

Proposition 9 : Si P et Q sont des predicats recursifs primitifs alors $P^c = \{\overline{x} \mid$ on a pas $P(\overline{x})\}$, $P \vee Q = \{\overline{x} \mid P(\overline{x})$ ou $Q(\overline{x})\}$ et $P \wedge Q = \{\overline{x} \mid P(\overline{x})$ et $Q(\overline{x})\}$ sont recursifs primitifs.

Exemples 10 : Les prédicats suivants sont récursifs primitifs

$= \{m,n \mid m = n\}$, $\quad \pi_= = n \mapsto -(S(0(n)), \times(\pi_>, \pi_<))$

$\geq \{m,n \mid m \geq n\}$ $\quad \pi_\geq = m,n \mapsto sg(+(\pi_>(m,n), \pi_=(m,n)))$

Application 11 : Schéma if ... then ... else

Si la condition d'un test P_{if} est un prédicat récursif primitif et que ses issues f_{then} et f_{else} sont recursives primitives alors le schema est récursif primitif et calculé par

$f_{if} = \overline{x}^n \mapsto +(\times(\pi_{P_{if}}(\overline{x}^n), f_{then}(\overline{x}^n)), \times(sg(-(S(0(\pi_1^n(\overline{x}))), \pi_{P_{if}}(\overline{x}))), f_{else}(\overline{x})))$

1.3 Minimisation bornée

Proposition 12 : La fonction de minimisation bornée par rapport au prédicat récursif primitif P notée $\mu i \leq n\,(P(\overline{x},i))$
$\overline{x}, n \mapsto \min\{i \mid P(\overline{x},i) \text{ et } i \leq n\}$ est une fonction récursive primitive.

Exemple 13 : Les fonctions suivantes sont récursives primitives :
$\mathrm{div} : m,n \mapsto \left\lfloor \frac{m}{n} \right\rfloor$ (division euclidienne) $= m,n \mapsto \mu i \leq m\,(n(i+1) > m)$
$\min : m,n \mapsto \min(m,n) \quad = m,n \mapsto \mu i \leq m\,(i = m \vee i = n)$
Les prédicats suivants sont récursifs primitifs :
$| = \{m,n \mid m \mid n\} : \Pi_| = m,n \mapsto \mu i \leq n\,((m \times i = n) \wedge (i \geq 1) \vee n = 0)$
premier $= \{p \mid p \text{ est premier}\}$ $\Pi_{\text{premier}} = p \mapsto 1 - sg(\mu i \leq p\,(i \mid p \wedge i > 1 \wedge i < p))$
la fonction mod : $m,n \mapsto n \bmod m$ est recursive primitive
$\mathrm{mod}(n,m) \mapsto \mu i \leq m\,(m \mid n-i)$.

1.4 Limite des fonctions recursives primitives

Proposition 14 : L'ensemble des fonctions recursives primitives est dénombrable.

Proposition 15 : L'ensemble des fonction $\mathbb{N} \to \mathbb{N}$ n'est pas dénombrable.

Corollaire 16 : Il existe des fonctions non recursives primitives.

Exemple 17 La fonction d'Ackermann, défini par :
$$\forall x,y \in \mathbb{N}, \begin{cases} a(0,x) = 2^x \\ a(y,0) = 1 \\ a(y+1,x+1) = a(y, a(y+1,x)) \end{cases}$$
n'est pas récursive primitive. DEV 1

2 / Fonctions récursives

2.1 Construction.

Définition 18 Un prédicat $P : \mathbb{N}^k \times \mathbb{N} \longrightarrow \{0,1\}$ est dit sûr si
$\forall \overline{x} \in \mathbb{N}^k, \exists i \in \mathbb{N}, \quad P(i) = 1.$

Définition 19 (minimisation non bornée)
La minimisation non bornée d'un prédicat P est la fonction notée $\mu i\, P(\overline{x},i)$ et définie par
$$\mu i P(\overline{x},i) = \begin{cases} \text{le plus petit } i \text{ tel que } P(\overline{x},i) = 1 \\ 0 \text{ si un tel } i \text{ n'existe pas.} \end{cases}$$

Définition 20 Fonctions récursives totales
L'ensemble des fonctions récursives totales (ou "μ-récursives totales") est le plus petit ensemble contenant les fonctions de base et clos par composition, récursion primitive et minimisation non bornée de prédicats récursifs (totaux) sûrs

Rq 21 Savoir si un prédicat est sûr ou non est un problème indécidable

Exemple 22 La fonction suivante associe à n le plus petit nombre premier strictement supérieur à n :
$$f : n \mapsto \mu i\,(\text{premier}(i) \wedge (i > n))$$
Elle est récursive totale car premier et $>$ sont des prédicats sûrs.

Application 23 Schéma d'une boucle while
Si la condition d'arrêt C est un prédicat récursif total sûr et que f est récursive totale alors le schéma "while C do f" est récursif total et est construit ainsi :
$f_{\text{while}} : x \mapsto It(x, \mu i\, \neg C(It(x,i)))$ où

$It \cdot (x,i) \longmapsto R_{\Pi_1^1, g}(x,i)$ où $g : (x,m,k) \longmapsto h(k)$

Définition 24 Fonctions récursives partielles

L'ensemble des fonctions récursives partielles (ou μ-récursives partielles) est le plus petit ensemble contenant les fonctions de base et clos par composition, récursion primitive et minimisation non bornée de prédicats récursifs (quelconques)

Exemple 25 La fonction f suivante associe à n $\frac{n}{2}$ si n est pair et n'est pas définie sinon. $f : n \mapsto \mu i \; (2i = n)$

Elle est récursive partielle.

2.2. Fonction avec les machines de Turing

Définition 26 Une fonction f est Turing calculable au sens strict s'il existe une machine de Turing M telle que pour tout argument n de f placé sur le ruban d'entrée de M, le calcul de M sur l'entrée n s'arrête et à ce moment, $f(n)$ est écrit sur le ruban de M.

Proposition 27 • Toute fonction récursive totale est MT-calculable au sens strict

• Toute fonction MT-calculable au sens strict est récursive totale DEV 2

Définition 28 Une fonction f est Turing calculable au sens large s'il existe une machine de Turing M telle que pour tout argument n de f placé en entrée sur le ruban de M, le calcul de M sur n s'arrête si $f(n)$ est défini et à ce moment $f(n)$ est écrit sur le ruban de M

Remarque 29 : Si $f(n)$ n'est pas défini, M peut ne pas s'arrêter.

Proposition 30 Les fonctions MT-calculables au sens large sont les fonctions récursives partielles

Remarque 31 Il existe des fonctions qui ne sont pas récursives partielles c'est le cas de la fonction C (castor affairé) définie ainsi :

$$C(N) = \sup \{ \varphi_N(M) \mid M = \text{machine de Turing à } N \text{ états et d'alphabet de ruban } \{1, \#\} \}$$

où $\varphi_N(M) = \sup \{$ nombre de 1 écrits sur le ruban de M dans la configuration finale à partir d'une entrée de taille $\leq N \}$

Refs Dehornoy, Mathématiques de l'informatique (pour castor affairé)

Wolper, Introduction à la calculabilité (pour les defs et exo) ←

Cori - Lascar, Logique mathématique 2 (pour Ackermann)