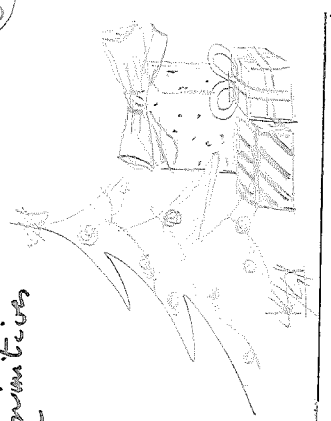
(J12)

1/3

Leçon 912 : Fonctions récursives primitives et non primitives

Olivier Baldellon

Ref : Cori-Lascar T2 (pour H)



Objectif : Depuis Cantor on sait qu'il y a plus de fonction de $\mathbb{N} \to \mathbb{N}$ que de fonction calculable par un algorithme fini. Quelles sont donc ces fonctions calculables ? Trois approches ont été proposées : les machines de Turing, le λ calcul et les fonctions μ-récursive.

I) Fonctions primitive récursive : une première approche

Soit $p \in \mathbb{N}$, on note $X_p = x_1, \dots, x_p$.

Def L'ensemble F_n des fonctions récursive est donné par induction. C'est le plus petit ensemble de fonction vérifiant

à nommer → 1) les fonction constantes de $\mathbb{N}^k \to \mathbb{N}$ sont dans F_n, $\forall k$ K_n^0

2) les projections P_i^p sont dans F_n : π_i^j

$$P_2^4(x_1, x_2, x_3, x_4) = x_2$$

3) la fonction succ est dans F_n : S^1

$\forall n \in \mathbb{N}$, $succ(n) = n+1$

4) F_n stable par composition

* soient $f_1 \dots f_p$ des fonctions de $\mathbb{N}^q \to \mathbb{N}$ de F_n
* soit g une fonction de $\mathbb{N}^p \to \mathbb{N}$. de F_n

alors on définit $g(f_1, \dots, f_p)$ par $\forall X_q \in \mathbb{N}^q$

$$g(f_1(X_q) \dots f_p(X_q)) = g(f_1 \dots f_p)(X_q)$$

$\sigma^n(g^m, f_1^n, \dots f_m^n)$.

5) F_n est stable par récurrence.

Soit $g \in F_n$ de $\mathbb{N}^p \to \mathbb{N}$ $\rho^1(g^n, n+2)$

et $h \in F_n$ de $\mathbb{N}^{p+2} \to \mathbb{N}$

alors la fonction suivante est bien définie :

$$f(X_p, 0) = g(X_p)$$

$$f(X_p, n+1) = h(X_p, n, f(X_p, n)) + \rho^2(\pi_1^1, \pi_3^3, \pi_3^4, \pi_3^3)$$

récurrence composée

proj

succ

Exemple de fonctions récursifs primitives

① somme : $+(x,0) = P_1^1(x)$

$+(x, y+1) = succ(P_3^3(x, y, +(x,y)))$

② produit : $\times(x, 0) = 0$

$\times(x, y+1) = +(x, \times(x,y))$

③ puissance $x^0 = 1$

$x^{y+1} = x \cdot x^y$

④ prédecesseur $pred(0) = 0$

$pred(n+1) = n$

⑤ différence $n - 0 = n$ ⚠ $3-5=0$!

$n - (m+1) = pred(n,m)$

⑥ signe $sg(x) = 1-(1-x)$ $\left(\begin{array}{l} = 0 \text{ si } x = 0 \\ 1 \text{ sinon} \end{array}\right)$

— démontrer mieux écrit ou intuition formellement les fonc° prim

mais plus informellement

Def Un ensemble A est dit récursif primitif ($A \in E_n$) si sa fonction caractéristique l'est.

$$A \in E_n \iff \mathbb{1}_A \in F_n$$

Prop E_n est stable par $\cap$, $\cup$ et complémentaire

Def par cas : $f = \sum f_i \mathbb{1}_{A_i}$

Prop : Fonction définies par cas. Soit f_1, f_2 et $f_3 \in F_n$ d'arité p alors f définie par :

$$f(x_p) = f_1(x_p) \text{ si } x_p \in A_1$$
$$f(x_p) = f_2(x_p) \text{ si } x_p \in A_2 \setminus A_1$$
$$f(x_p) = f_3 \text{ sinon.}$$

est récursive primitive si les ensemble A_1 et A_2 le sont.

Autre exemple

$g(x) = \sum_{k=1}^{n} f_k(x)$ et $g(x, n) = \sum_{k=1}^{n} f(x, k)$ sont dans F_n

Def Minimisation bornée.
Soit $A \in E_n$ et m fixé. La fonction qui à x_p renvoit le plus petit i tel que $(x_p, i) \in A$ s'il existe et 0 sinon est dans F_n. On la note $\mu(i \leq m)\ (x_p, i) \in A$

Def Prédiquat primitif récursif : une fonction de F_n à image dans $\{0,1\}$ et de domaine $\mathbb{N}^k$, $k \geq 1$

Ex $>(x, y) = sg(x - y)$ $\quad$ ($x > y$ ssi $sg(x-y) = 1$)

Ex Les quantifications bornées sont elles aussi dans P_n.

$\forall i \leq m \quad p(x, i)$
$\exists i \leq m \quad p(x, i)$

II) Limites des fonctions primitives récursives

Th : Il existe des fonctions calculable que ne sont pas dans F_n.

Remarque Soit $F_n = \{f_0, f_1, f_2 \cdots\}$. $g : n \to f_n(n) + 1 \notin F_n$ □

Rq On ne peut pas dénombrer l'ensemble des fonctions calculable (qui n'existe) par des fonctions calculable.

Def Fonction d'Ackerman ← rec ms pas prim rec.

$A(0, n) = n+1$
$A(m+1, 0) = A(m, 1)$
$A(m+1, n+1) = A(m, A(m+1, n))$

) bien calculable

Prop On note $A_m(n) = A(m, n)$ alors
* A_m est $\nearrow$ $\quad$ * $A_m(n) > n$
* $A_{m+1} > A_m$ $\quad$ * $A_{m+1}(n) > A_m(n+1)$

Th $A \notin F_n$

III) Fonctions μ-récursives et partielles

Def Minimisation non bornée totale.
Soit A un ensemble de $\mathbb{N}^{p+1}$ tq $\forall x_p, \exists i, (x_p, i) \in A$.
On définit $\mu i\ (x_p, i) \in A$ la fonction qui renvoit le plus petit i tq $(x_p, i) \in A$

Def Les fonctions et les prédicats μ-récursif sont obtenues à partir : ① des fonctions de base ② de la composition et ③ par récurrence et ④ schéma de minimisation non bornée totale

Def On appelle fonction partielle un couple (A, f) avec $f : A \subseteq \mathbb{N}^k \to \mathbb{N}$. A est le domaine de f et f n'est pas définie autre part.

Par abus on associe (A, f) à f

⚠ $g + f - f \neq g$ $\quad$ si g et f ont des domaines distincts

(9/12) $s_j^i(j)$ Appli : tte fonc° μ-rec peut s'écrire ar 1 seul μ. récursive → μ-récursive

Def minimisation non bornée

$\mu i\ (X_p, i) \in A = \begin{cases} \text{le plus petit } i \text{ vérifiant } (X_p, i) \in A \\ \text{non définie sinon.} \end{cases}$

Def Fonction μ-récursive partielle. Plus petit ensemble contenant les fonctions de base, stable par récurrence, composition et minimisation non bornée. On le note F_μ^*

Rq Soit $(A_1, f_1) \dots (A_n, f_n)$

$g(f_1 \dots f_n)(X_p)$ est définie si $\forall i$ $f_i(X_p)$ est définie et si $\forall j$ $g(f_1(X_p) \dots f_n(X_p))$ est déf.

IV) où l'on parle des Machines de Turing

On veut utiliser les MdeT pour calculer des fonction à plusieurs variables

Def On appelle machine de Turing ap bandes un n-uplet

$M = \Sigma \times E \times T \times \{e_i\} \times \{e_f\}$ tel que

1) Σ alphabet 2) E un ensemble d'état avec $(e_i, e_f) \in E^2$

3) $T \subseteq \Sigma^p \times E \to \Sigma^p \times E \times \{-1, 1\}$

l'exécution commence dans l'état initial e_i et s'arrête dans l'état final e_f. De plus si l'état initial est de la forme

$\begin{matrix} \vdash x_1 \\ \vdash x_2 \\ \vdots \\ \vdash x_p \\ \vdash \perp \end{matrix}$ alors l'état final sera ici $\begin{matrix} \vdash x_1 \\ \vdash x_2 \\ \vdots \\ \vdash x_n \\ \vdash y \end{matrix}$

Def une fonction est T-calculable si il existe une MT qui termine toujours tel que $f(x_1 \dots x_p) = y$

si la machine de Turing ne s'arrête pas forcément, alors son ensemble de définition est l'ensemble des valeurs pour lequel elle s'arrête et cela permet de définir les fonctions partielles T-calculable.

Th Les fonctions μ-récursive total sont les fonctions T-calculable ← dev. 1

Th Les fonctions μ-récursive partielle sont les fonctions T calculable partielle.

V) fonctions partielles et ensemble récursif énumérable

Def Soit $A \subseteq \mathbb{N}^p$, A est récursif énumérable ssi A est le domaine de définition d'une fonction partielle récursive

Prop A est récursif si A et $\mathbb{N}^p \setminus A$ sont réc. énumérable

VI) Théorème d'énumération et applications

On note $I_p = \{x \mid x$ est l'indice de MdeT de ar mois p+1 bandes$\}$

Def $\varphi_i^p(X_p)$ n'est pas définie si $i \notin I_p$ et vaut le résultat de la MdT ayant pour argument X_p sinon

Th (d'énumération) (passé sous le tapis)

1) $i \to \varphi_i^p$ est partielle récursive

2) si ψ est partielle récursive, $\exists i$ tq $\psi = \varphi_i^p$

φ_i^p ← nbre d'arguments = nbre d'arg. ; i ← code de la MT

Th (SNM) $\forall j, \forall p$ il existe une fonction récursive primitive s_j^p tq $\forall i, \forall X_p$ $\varphi_j^{p+1}(i, X_p) = \varphi_{s_j^p(i)}^p(X_p)$

Th (Point fixe) Soit α total récursive, $\forall j, \exists k$ tq $\varphi_k^p = \varphi_{\alpha(k)}^p$ ← dev 2

App. Acckerman

Th (Rice) Soit $\emptyset \subsetneq X \subsetneq F_n^*$ alors $\{x \mid \varphi_x^p \in X\}$ non récursif. ← dev 3

↳ arrêté pas impolite.

[COMMENTAIRES]

1) Plan

- D'où ça vient les fonc° prim ?
  // ce qu'on programme eg. schéma de réc = for
- Coût ce qui concerne les codages
  eg. utiliser un seul argument en encodant av des ℕ-uplets
  Résultats de Robinson
- Minimisa° ... a + préservar
- Exemples * fonc° calculables en tps poly
  * - - - - - - - - - - exp
  * - - -
- Intro : △ algorithme = ce qu'on essaye de définir
  Pas vraiment préexistant

— très détaillé au début et pas assez à la fin → pas très cohérent.

— Mc fonc° rec peut s'écrire avec une seule applica° de μ (appli de rec ← MT) (devr 1)

- Sets équivalents
  Avec le schéma de réc en moins (mais d'autres bucs en +)
  Alg on peut encoder la réc. avec
  Permet de rapprocher + facilement des MT, où de l'arithmo...

2) Devs

- devr 2 : * interprétat° = "M compil (= α) a au moins un prog correct (= le)"
  * d'autre manière : simpler pr démontrer que Acc ∈ rec.
  * dem pt fixe : construc° gaffi (- formel et + lg ds le SUPER)
- devr 3 : réduc° au plus de l'ami
  * △ Pas d'eq entre MT et fonc° ~~prim~~ rec !
  i une MT + nbre d'arg fixé → 1 fonc° rec. φ_i
  (sans go vers !)
  * Un peu hors sujet ?
  devr possible : mq déterminer pas prim. rec.
- devr 1 : * idée générale = utiliser un ppe de min. non bornée pour partir de là pour expliquer (minimisa° → langage MT)
  * se placer ds le cadre 1 seule bande.
  * utiliser les déf. par cas
  * appli : on peut écrire les fonc° μ-rec av 1 seul μ.