31/03 2015

922 — Ensembles récursifs, récursivement énumérables. Exemples.

[CL]

But : définir des ensembles de fonctions qui soient "calculables", dans le sens où il existe un algorithme de calcul.

I Construction des ensembles récursifs et récursivement énumérables

A) Fonctions récursives primitives.

Soit $\mathcal{F}_p$ l'ensemble des applications de $\mathbb{N}^p$ dans $\mathbb{N}$ et $\mathcal{F} := \bigcup_p \mathcal{F}_p$.

Déf 1 : Soit $n \in \mathbb{N}$. La fonction composée de $g \in \mathcal{F}_n$ et $(f_1,\dots,f_n) \in \mathcal{F}_p$ est la fonction $h \in \mathcal{F}_p$:

$\forall (x_1,\dots,x_p) \in \mathbb{N}^p$, $h(x_1,\dots,x_p) = g(f_1(x_1,\dots,x_p),\dots,f_n(x_1,\dots,x_p))$.

Déf 2 : La fonction définie par récursion primitive à partir de $g \in \mathcal{F}_p$ et $h \in \mathcal{F}_{p+2}$ est la fonction $f \in \mathcal{F}_{p+1}$:

$\forall (x_1,\dots,x_p)$, $f(x_1,\dots,x_p,0) = g(x_1,\dots,x_p)$,

$\forall (x_1,\dots,x_p,y)$, $f(x_1,\dots,x_p,y+1) = h(x_1,\dots,x_p,y,f(x_1,\dots,x_p,y))$.

Déf 3 : L'ensemble des fonctions récursives primitives E est le plus petit sous-ensemble $E \subset \mathcal{F}$ contenant :

- les constantes : $\mathbb{N}^p \to \mathbb{N}$, $x \mapsto a$,
- le successeur : $\mathbb{N} \to \mathbb{N}$, $x \mapsto x+1$,
- la projection : $P_p^i : \mathbb{N}^p \to \mathbb{N}$, $(x_1,\dots,x_p) \mapsto x_i$, où $1 \leq i \leq p$,

et qui soit stable par composition et récursion primitive.

Les ensembles récursifs primitifs $A \subseteq \mathbb{N}^p$ sont ceux tels que la fonction caractéristique χ_A soit récursive primitive.

Exemples 4 : $+, \times, \dot{-} : (x,y) \mapsto \begin{cases} x-y & \text{si } x > y \\ 0 & \text{sinon} \end{cases}$ sont récursives primitives.

$A = \{x > y\}$ est récursif primitif.

Prop 5 : E est stable par schéma de définition par cas, sommes et produits finis et schéma μ borné (minimisation) :

si A est un sous-ensemble récursif primitif de $\mathbb{N}^{p+1}$, on définit : $\begin{cases} f(x_1,\dots,x_p,z) = 0 & \text{si } \nexists\, t \leq z \text{ t.q. } (x_1,\dots,x_p,t) \in A \\ f(x_1,\dots,x_p,z) = t, & \text{le plus petit des } t \leq z \text{ tel que } (x_1,\dots,x_p,t) \in A. \end{cases}$

Prop 6 : la fonction d'Ackermann définie par : DEV

$A(0,x) = 2^x$, $A(y,0) = 1$, $A(y+1,x+1) = A(y, A(y+1,x))$

n'est pas primitive récursive.

B) Fonctions partielles récursives.

Déf 7 : Une fonction partielle de $\mathbb{N}^p$ dans $\mathbb{N}$ est une application $f : A \to \mathbb{N}$, où $A \subseteq \mathbb{N}^p$. $\mathcal{F}_p^*$: ensemble des fonctions partielles de $\mathbb{N}^p$ dans $\mathbb{N}$, et $\mathcal{F}^* = \bigcup_p \mathcal{F}_p^*$. [CL]

Déf 8 : Minimisation non bornée (schéma μ) : [WOL]

Soit P un prédicat (i.e. la fonction caractéristique de $A \subseteq \mathbb{N}^p$), alors la minimisation non bornée de P, notée $\mu y\, P(x_1,\dots,x_p,y)$ est :

$\mu y\, P(x_1,\dots,x_p,y) = \begin{cases} \text{le plus petit } i \text{ tel que } P(x_1,\dots,x_p,i) = 1 \\ 0 \text{ si un tel } i \text{ n'existe pas.} \end{cases}$

Déf 9 : Un prédicat P est sûr si : $\forall (x_1,\dots,x_p), \exists y, P(x_1,\dots,x_p,y) = 1$.

Déf 10 : Les fonctions et prédicats μ-récursifs sont ceux obtenus à partir des fonctions constantes, successeur, projections et stable par composition, récursion primitive et minimisation non bornée de prédicats sûrs. Par abus, on dira récursif. $A \subseteq \mathbb{N}^p$ est dit récursif si sa fonction caractéristique est récursive.

Déf 11 : De la même manière, on définit les fonctions récursives partielles en remplaçant fonctions par fonctions partielles et en appliquant la minimisation bornée à des prédicats non sûrs. [WOL] [CL]

$A \subseteq \mathbb{N}^p$ est dit récursivement énumérable si sa fonction caractéristique est récursive partielle.

CL p42 Propo 12 : Tout ensemble récursif est récursivement énumérable.

L'ensemble des sous-ensembles récursifs de $\mathbb{N}^p$ est clos par les opérations booléennes.

II Lien avec les machines de Turing (MT)

A) Fonctions calculables.

Rappel : on a l'équivalence entre les machines de Turing à 1 ruban et celles à p rubans.

CL p28 Déf 13 : Soit f une fonction partielle à p variables ; M une MT à au moins $(p+1)$ bandes. On dit que M calcule f si, pour tout $(x_1,\dots,x_p)\in\mathbb{N}^p$, en faisant fonctionner M avec en entrée x_1 sur la bande 1, ..., x_p sur la bande p :

- si $f(x_1,\dots,x_p)$ non définie, la machine ne s'arrête jamais.
- sinon, la machine s'arrêtera et aura $f(x_1,\dots,x_p)$ écrit sur la bande $p+1$.

f est calculable s'il existe une machine calculant f.

Exemple 14 : successeurs, projecteurs et constantes sont calculables.

CL p32 Propo 15 (Th) : Toutes les fonctions partielles récursives sont calculables.

Exemple 16 : $\pi : \mathbb{N}\to\mathbb{N}$ qui à un entier i associe le $i^{\text{ème}}$ nombre premier, est récursive, donc calculable.

WOL p135 Th 17 : Toutes les fonctions calculables par une machine de Turing sont partielles récursives.

IVT p140 Th 18 : Une fonction est récursive si et seulement si elle est calculable par une MT. Puisque la fonction est définie partout, la MT s'arrête quelque soit l'entrée ; on dit que la fonction est décidable.

Th 19 : Un ensemble est récursif (resp. récursivement énumérable) si et seulement si sa fonction caractéristique est décidable (resp. calculable).

Déf 20 : La classe de décidabilité R (Récursifs) est l'ensemble des langages décidables par une machine de Turing. [WOL p141]

La classe de décidabilité RE est l'ensemble des langages acceptés par une MT.

Rem 21 : Les machines de Turing de par leurs définitions, sont dénombrables ; il en est de même pour les mots finis. Soit donc $(M_i)_{i\in\mathbb{N}}$ et $(w_i)_{i\in\mathbb{N}}$ deux énumérations.

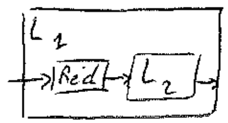
Propo 22 : Le langage $L_0 = \{w \mid w = w_i$ et M_i n'accepte pas $w_i\}$ n'est pas dans RE.

Th 23 : Le langage $\overline{L_0} = \{w \mid w = w_i \text{ et } M_i \text{ accepte } w_i\}$ est indécidable. De plus, $\overline{L_0}\in RE$. (cf annexe).

B) Quelques propriétés. [WOL p245]

Étude de l'indécidabilité des problèmes $\Rightarrow$ peut-on les comparer ?

Technique 24 de la réduction : Soient L_1 et L_2 deux langages, tel que $L_1 \notin R$. Si on sait que :
s'il existe un algorithme qui décide L_2,
alors il existe un algorithme qui décide L_1,
Alors L_2 n'est pas décidable.



Propo 25 : Un langage L est dans RE si et seulement si il est engendré par une grammaire. [NVT p145]

Propo 26 : La famille des langages RE est close par union, produit, étoile et homomorphisme.

Propo 27 : La famille des langages RE est close par intersection.

Propo 28 : La famille des langages RE n'est pas fermée par complémentaire ; en notant $co(RE) = \{L \dots ; \overline{L}\in RE\}$, on a : $R = RE \cap co(RE)$.

III Applications à différents langages.

[WOL] Problème du langage universel : $LU = \{\langle M, w\rangle \mid M \text{ accepte } w\}$.

Th 29 : le langage LU est indécidable.

Problème de l'arrêt : $H = \{\langle M, w\rangle \mid M \text{ s'arrête sur } w\}$.

Th 30 : le problème de l'arrêt est indécidable.

Problème du langage accepté vide : savoir si le langage accepté par une MT de Turing est vide.

Th 31 : le problème du langage accepté vide est indécidable.

[CAR] Th 32 (de Rice) : Pour toute propriété non triviale P sur les langages récursivement énumérables, le problème de savoir si le langage L reconnu par une MT vérifie P est indécidable. (réduction de LU).

Problème de correspondance de Post (PCP) :

entrée : une instance est une suite $(u_1, v_1), \dots, (u_n, v_n)$ de paires de mots sur un alphabet Σ.

sortie : oui s'il y a au moins une solution.

[C-L] Th 33 : PCP est indécidable.

L'arithmétique de Peano S_0 est la théorie constituée des 7 axiomes :

(1) $\forall v \; \neg Sv = 0$

(2) $\forall v \exists w \; (\neg v = 0 \Rightarrow S v_1 = v_0)$

(3) $\forall v \forall w \; (Sv = Sw \Rightarrow v = w)$

(4) $\forall v \quad v + 0 = v$

(5) $\forall v \forall w \quad v + Sw = S(v + w)$

(6) $\forall v \quad v \times 0 = 0$

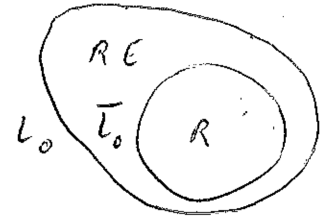
(7) $\forall v \forall w \quad v \times Sw = (v \times w) + v$

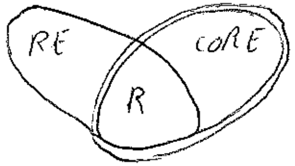
Th 34 : Soit T une théorie cohérente cohérente S_0 ; alors T est indécidable.

Th 35 : Cependant, l'arithmétique de Presburger (dans le langage $\{0, S, +, =\}$) est décidable. [DNR]

Th 36 : Soit R un système de réécriture ; le problème de savoir si R est terminant est indécidable. [BAA] DVT

Annexe 1 : Inclusions de classes de décidabilité :





[CL] : Cori-Lascar II, Logique mathématique II.
[CAR] : Carton, Langages formels.
[WOL] : Introduction à la calculabilité, Wolper.
[AUT] : Autebert, Théorie des langages et des automates.
[DNR] : David-Nour-Raffali,
[BAA] : Baader