Caractérisation des fonctions récursivement énumérables

Julie Parreaux

2018-2019

Référence du développement : Cori–Lascar [1, p.41] (cette référence est difficile : elle ne contient pas l'ensemble du développement au même endroit)

Leçons où on présente le développement : 912 (Fonctions récursives) ; 914 (Décidabilité et indécidabilité).

1 Introduction

La classe de décidabilité RE est généralement définie à l'aide des machines de Turing. Cependant, par la thèse de Church et plus précisément l'équivalence entre les machines de Turing et les fonctions μ-récursives, nous pouvons la définir à l'aide de ces dernières. Nous allons énoncer et montrer un résultat permettant de caractériser les fonctions dans la classe de décidabilité RE (et plus généralement la classe en elle-même) à l'aide des fonctions primitives récursives et des fonctions μ-récursives.

Nous énonçons ce résultat avec la définition via les machines de Turing d'un langage dans RE. Pour prouver certaines implications, nous utiliserons librement l'équivalence entre les machines de Turing et les fonctions μ-récursives (que nous admettons). Pour plus de détail, voir le développement correspondant. Il faut donc faire attention de mentionner cette équivalence dans le plan avant le théorème afin de pouvoir utiliser librement sa preuve.

Remarques sur le développement

Ce développement présente une difficulté autours de la référence mais la technicité de celui-ci reste abordable. Cependant, il faut faire attention car la première implication contient un développement (équivalence entre les fonctions μ-récursives et les machines de Turing) qu'il nous faut donc admettre dans le cadre de celui-ci.

2 Caractérisation des fonctions récursivement énumérables

Théorème. *Soit $A \subseteq \mathbb{N}$. Les assertions suivantes sont équivalentes.*

1. $A \in RE$;
2. $A = \pi_2^2(B)$ *où* $B \subseteq \mathbb{N}^2$ *est primitif récursif ;*
3. $A = \emptyset$ *ou* A *est l'image d'une fonction primitive récursive ;*
4. A *est l'image d'une fonction* μ*-récursive.*

Démonstration. On montre la suite d'implication suivante : $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$.

Montrons $1 \Rightarrow 2$ Soit $\mathcal{M}$ une machine de Turing qui accepte A ($A \in RE$ donc par définition de RE, il existe bien une machine de Turing qui accepte A). Sans perte de généralité, on suppose que les états finaux de la machine de Turing boucle à l'infini sur eux-même (par la thèse de Church on obtient le résultat de l'équivalence de l'expressivité des machines de Turing dont la frontière est la frontière de la classe RE).

On pose le prédicat $P(t, m) =$ "$\mathcal{M}$ s'arrête en t étapes sur n". Ce prédicat est primitif récursif car il s'écrit :

$$\mathbb{1}_F(\text{CONFIG}(\text{INIT}(n), t))$$

(par l'équivalence entre les machines de Turing et les fonctions μ-récursives, on sait que ces fonctions sont bien primitive récursives). (Si on ne fait pas l'hypothèse supplémentaire sur la machine de Turing, le prédicat reste récursif car il peut s'écrire comme

$$\mathbb{1}_F(\text{CONFIG}(\text{INIT}(n), \mu i \leq t, \mathbb{1}_F(\text{CONFIG}(\text{INIT}(n), t))))$$

où on a juste ajouté une minimisation non-bornée.)

Notons $B = \{(t,n) \in \mathbb{N}^2 | P(t,n)\}$ (le prédicat P peut être vu comme l'indicatrice de B), donc B primitif récursif (car son indicatrice l'est). De plus,

$$\begin{array}{llll} n \in A & ssi & \exists t \in \mathbb{N}, P(t,n) & A \in RE \text{ et } \mathcal{M} \text{ accepte } A \\ & ssi & n \in \pi_2^2(B) & \Leftrightarrow \exists t \in \mathbb{N}, (t,n) \in B \end{array}$$

Donc $A \in \pi_2^2(B)$ où B est primitif récursif.

Montrons $2 \Rightarrow 3$ Supposons $A \neq \emptyset$ (sinon, on a terminé). Il existe donc $m \in A$. Soit $\begin{array}{lccc} \varphi : & \mathbb{N} & \to & \mathbb{N} \\ & k & \mapsto & (t_k, n_k) \end{array}$ une énumération primitive récursive de $\mathbb{N}^2$. Une telle énumération existe : on peut prendre, par exemple, $\varphi : k \mapsto (\text{DIAG}(k) - \text{ECART}(k), \text{ECART}(k))$ où

$$\begin{array}{ll} \text{DIAG} : & k \mapsto \left(\mu i \leq (k+1). \left(\frac{i(i+1)}{2} > k\right)\right) - 1 \\ \text{ECART} : & k \mapsto k - \frac{\text{DIAG}(k)(\text{DIAG}(k)+1)}{2} \end{array}$$

sont primitives récursives. Donc φ est bien primitive récursive comme composition de fonctions primitives récursives. (De plus, φ est une bijection de réciproque $(t,n) \mapsto \frac{(t+n)(t+n+1)}{2} + n$.)

Alors en notant P un prédicat récursif tel que $B = \{(t,n) | P(t,n)\}$ (existe par hypothèse). On cherche f tel que

$$f(n_k) = \begin{cases} n_k & \text{si } (t_k, n_k) \in B \\ m & \text{sinon} \end{cases}$$

On pose alors $f = (P \circ \varphi) \times (\pi_2^2 \circ \varphi) + (1 - (P \circ \varphi)) \times m$ qui est bien une fonction primitive récursive et $A = Im\ f$ (car si $n_k \in A$ alors il existe $(t_k, n_k) \in B$ et sinon pour tout t_k, $(t_k, n_k) \notin B$).

Montrons $3 \Rightarrow 4$ Soit $A \subseteq \mathbb{N}$, on distingue deux cas.

- Si $A = \emptyset$, alors $A = im\ (\mu i\ (\mathbb{O}(i)))$ où $\mathbb{O}(.)$ est le prédicat nul d'arité 1. Donc A est l'image d'une fonction μ-récursive.
- Sinon, A est l'image d'une fonction primitive récursive, donc en particulier, elle est l'image d'une fonction μ-récursive.

Montrons $4 \Rightarrow 1$ Si A est l'image d'une fonction primitive récursive, alors elle est calculable par une machine de Turing (encore l'équivalence entre les machines de Turing et les fonctions μ-récursives) définie par l'algorithme 1 qui prend en paramètre un entier n et renvoie OUI si $n \in A$. Donc $A \in RE$. □

Algorithm 1 Définition de la machine de Turing déterministe acceptant A

```
1: k ← 0
2: while f(k) ≠ n do
3:     k ← k + 1
4: end while
5: Accepte
```

Références

[1] D. Lascard R. Cori. *Logique mathématique, tome 2*. Masson, 1994.