Théorème de Rice

Benjamin Hellouin

Hopcroft 8.4

Théorème 1. *Soit Σ un alphabet et $P \subseteq \mathcal{P}(\Sigma^*)$ qui contient une partie non vide mais pas l'ensemble des langages récursivement énumérables. On note $Pr(P)$ le problème :*

Entrée *Le code d'une machine de Turing M*

Sortie *OUI si $L(M) \in P$, NON sinon.*

Alors $Pr(P)$ est indécidable.

On peut supposer, sans perte de généralité, que $\emptyset \notin P$. Puisque P n'est pas triviale, il existe L possédant la propriété P et M_L une machine de Turing qui accepte L. Supposons que P soit décidable, et soit M_P une machine de Turing qui le décide. On construit un algorithme A qui, étant donné une machine M et une entrée w, construit une machine M' telle qu'on ait :

$$L(M') \in P \Leftrightarrow M \text{ accepte } w.$$

Sur une entrée x, cette machine M' a le comportement suivant :

- elle simule la machine M sur l'entrée w ;
- si M rejette w, M' rejette x. Sinon :
- elle simule M_L sur l'entrée x ;
- M' accepte si et seulement si M_L accepte.

Ainsi, M accepte $\emptyset (\neq \in P)$ ou $L (\in P)$ selon que M accepte w ou pas. On dispose ainsi d'un algorithme pour décider si une machine quelconque M accepte une entrée donnée w : on applique A à M et w et on obtient M', puis on applique M_P pour décider si M' est dans P ou non. Or ce problème est réputé indécidable. On a une contradiction, donc P est indécidable.

Corollaire 1. *Les propriétés suivantes d'un langage récursivement énumérable sont indécidables :*

- *$L = \emptyset$;*
- *L est fini ;*
- *L est rationnel ;*
- *L est algébrique.*