~14'30

THÉORÈME DE RICE [Wolper p 150]

(Th) Soit $\emptyset \subsetneq \mathcal{P} \subsetneq RE$. Alors le langage $L_{\mathcal{P}} = \{$machines de Turing $M \mid L(M) \in \mathcal{P}\}$

est indécidable. Autrement dit → version du plan

Rq : on confond langage et pb décision et on peut !

(Notations) On se place sur 1 alphabet Σ fini

notations ≠ car w_i désigne plutôt la lettre i de w et pas le mot n° i

- Σ^* est dénombrable donc $\exists$ $N : \begin{cases} \Sigma^* \longrightarrow \mathbb{N} \\ w \longmapsto N(w) \end{cases}$ injective (m̂ bijective)

  qui numérote les éléments de Σ^*
- {machines de Turing} est aussi dénombrable donc {machines de Turing} $= \{M_i \mid i \in \mathbb{N}\}$
- $L_0 = \{w \in \Sigma^* \mid w \notin L(M_{N(w)})\}$
- $LU = \{(M, w) \in$ machines de Turing $\times \Sigma^* \mid w \in L(M)\}$

expliquer le plan de la démo

(Lemme 1) L_0 est indécidable (on a même $L_0 \notin RE$)

Sinon il serait accepté par une certaine machine M_k. / bij de $\mathbb{N}$, $\exists w \in \Sigma^*, N(w)=$

Si $\overset{w}{w_k} \in L_0 = L(M_k)$ alors $w_k \notin L_0$

si $\overset{w}{w_k} \notin L_0 = L(M_k)$ alors $w_k \in L_0$ donc contradiction

(Cqc) $\overline{L_0}$ est aussi indécidable

donner la déf d'1 réduction

(Lemme 2) LU est indécidable. On réduit $\overline{L_0}$ à LU via

$f(w) = \begin{cases} \text{déterminer } N(w) \\ \text{déterminer } M_{N(w)} \\ \text{renvoyer } (M_{N(w)}, w) \end{cases}$

instance pour $\overline{L_0}$

(calculable en énumérant les mots)

(calculable en énumérant les MT car on peut faire 1 automate qui, donnant 1 mot, dit si ce mot est le codage d'1 MT)

instance pour LU

On a bien $w \in \overline{L_0} \iff w \in L(M_{N(w)}) \iff (M_{N(w)}, w) \in LU$

(démo) du th. On réduit LU à $L_{\mathcal{P}}$

On peut supposer que le langage vide n'est pas dans $\mathcal{P}$

Sinon on travaille avec $RE \setminus \mathcal{P}$ (qui vérifie les mêmes conditions que $\mathcal{P}$, à savoir $\neq \emptyset$ et $\neq RE$, ou c'est le cas pour $\mathcal{P}$)

Comme $\mathcal{P} \neq \emptyset$ ($RE \setminus \mathcal{P} \neq \emptyset$ aussi puisque $\mathcal{P} \neq RE$), il existe $L \in \mathcal{P}$ accepté par une machine M_L

On construit alors $f(M, w) =$ la machine M' telle que

M' se comporte ainsi sur l'entrée x :

simuler M sur w

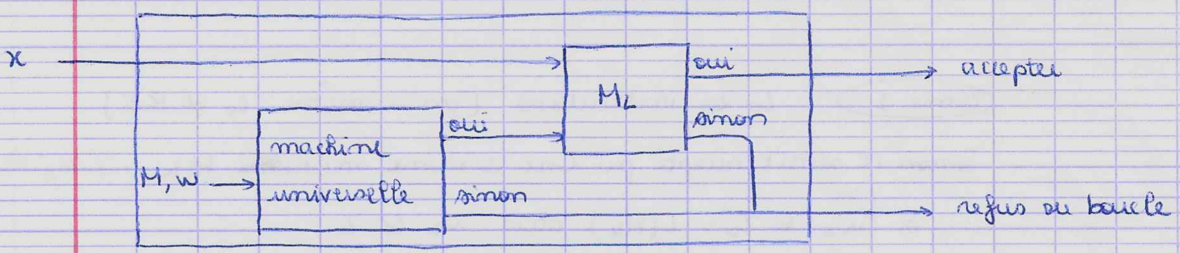
si M accepte w

simuler M_L sur x

si M_L accepte x, accepter

refuser

Cette fonction est calculable : voici comment construire la machine M'



$= M'$

Les "sinon" dans cette machine correspondent aux refus et aux boucles.

On a $x \in L(M') \Leftrightarrow x \in L(M_L)$ et $w \in L(M)$

donc $L(M') = \begin{cases} L(M_L) = L \in \mathcal{P} & \text{si } w \in L(M) \\ \emptyset \notin \mathcal{P} & \text{sinon.} \end{cases}$

Donc $M' = f(M, w) \in L_{\mathcal{P}} \Leftrightarrow L(M') \in \mathcal{P}$

$\Leftrightarrow L(M') = L$

$\Leftrightarrow w \in L(M)$

$\Leftrightarrow (M, w) \in LU$ ce qui conclut la réduction.

Ex Pour $\mathcal{P} = \{\emptyset\}$, on obtient que le problème suivant :

entrée : une machine de Turing M

sortie : oui si $L(M) = \emptyset$, non sinon, est indécidable.

Alors que dès les automates $\leq$, on peut décider si le langage reconnu est $\emptyset$