INFO-F408: Computability & complexity

Rémy Detobel

9 Octobre, 2017

1 Cantor's Diagonal

Corollary 4.18 dans le livre :
Certain langages ne sont pas reconnaissable par une machine de Turing (Turing-recognizable).
Idée :
L'ensemble des machines des turing est comptable.

Exemple :
Tous les mots possibles dans l'alphabet : $\{0,1\}$:

	ε	0	1	00	01	10	11	000	001
M_1	**0**	1	1	0	1	1	1	0	0
M_2	1	**1**	1	1	0	1	0	1	1
M_3	0	1	**0**	1	1	0	1	1	1
M_4									
M_5									

On construit donc ici un table qui définit un langage.

2 Halting Problem (problème de l'arrêt)

$$A_{TM} = \{< M, w > | M \text{ est une machine de turing qui accepte } w\}$$

M = Une machine de turing
w = un mot en entrée (input)
Voir le livre, théorème 4.11 : le problème A_{TM} est indécidable.

2.1 Preuves

Par contradiction : supposons que A_{TM} est décidable. Cela signifie qu'il existe une machine $H(< M, w >) = \{$Accepte si M accepte w et rejette dans les autres cas$\}$
On définit ensuite une machine D tel que pour l'entrée M(une machine de turing) :

1. Exécuter H sur l'entrée $< M, < M >>$
2. On renvoie l'inverse de H : accepte si M n'accepte pas $< M >$ et rejette si M accepte $< M >$.

Enfin, on exécute D sur lui-même : $D(< D >)$.
Cela signifie que D accepte uniquement s'il n'accepte pas. $\rightarrow$ contradiction
Langage A définit $\bar{A}$ comme étant son complément : $\bar{A} = \{w | w \notin A\}$
Supposons que A et $\bar{A}$ sont ("recognizable") reconnaissable. Donc A et $\bar{A}$ sont aussi décidable.
Preuve : Posons M et M' sont respectivement reconnaissable pour A et $\bar{A}$. Construisons un "décideur" D pour A en exécutant M et M' en "parallèle" (en alternant étape par étape sur M et sur M'.

Posons maintenant A_{TM} comme étant indécidable. Est-il pour autant reconnaissable ("recognizable") ?
A_{TM} est reconnaissable (preuve : simuler M sur w).
On peut également écrire :

$$\begin{aligned} &A \text{ est décidable} \\ \Leftrightarrow\ &\text{l'ensemble } A \text{ et } \bar{A} \text{ est reconnaissable} \end{aligned}$$

$\Rightarrow \bar{A_{TM}}$ n'est pas reconnaissable.

$$\bar{A_{TM}} = \{< M, w > | w \text{ n'est pas accepté par } M\}$$

3 Reductibility (Réduction)

Pour une machine de turing M, noté $L(M)$ décrit le langage reconnu par M.

$$L(M) = \{w | M \text{ accepte w}\} \in \Sigma^*$$

$$REGULAR_{TM} = \{< M > | M \text{ est une machine de turing et } L(M) \text{ est régulier}\}$$

Rappel d'un langage régulier : qui peut être reconnu par un automate fini. Ce langage est indécidable.
Voir **livre**, théorème 5.3
Preuve :
Définissons une Machine de Turing M_2 comme une fonction de M (une machine de turing) et w ($\in \Sigma^*$)
$M_2 =$ pour une certaine entrée x :

1. si x a la forme $0^n 1^n$, on accepte
2. sinon, on exécute M sur w et on accepte si M accepte w.

M_2 n'est pas spécialement un "décideur". Cela va déprendre de M. Notons également que M_2 est une "fonction" de M et w.

Quel est le langage de $L(M_2)$?

1. Si M accepte w : alors $L(M_2) = \Sigma^*$ accepte tout.
   $\Rightarrow$ régulier
2. Si M n'accepte pas w, $L(M_2) = \{0^n 1^n | n \geq 0\}$
   $\Rightarrow$ pas régulier

On a donc réussi à réduire le problème de l'arrêt à ce problème.
Par contradiction, supposons que $REGULAR_{TM}$ est décidable et qu'il existe R, un "décideur" pour $REGULAR_{TM}$.
$S =$ en entrée $< M, w >=$:

1. Construire M_2 pour M et w
2. Exécuter R sur M_2 et on accepte si et seulement si R accepte.

En faisant cela, on montre que S décide $A_{TM} \rightarrow$ contradiction

4 Rice's Theorem

Chapitre 5 exercice 28
Posons P comme étant n'importe quel propriété de langage NON-TRIVIAL (NON-TRIVIAL).
Théorème : Déterminer si le langage d'une machine de turing a comme propriété P, est indécidable.

$$\{< M > | M \text{ est une machine de turing et } L(M) \text{ a une propriété } P\}$$

Une propriété trivial : est une propriété sans importance, par exemple une propriété est trivial si tous les langages ou aucun langage ne l'a.

4.1 Démonstration

Par contradiction :
Posons R_p un "décideur" pour L_p.
- Posons $T_\varnothing$ comme une machine de turing qui rejette toutes les possibilités ($L(T_\varnothing) = \varnothing$), supposons que $\varnothing \notin P$ (sans perte de généralité).
- Posons T comme une machine de turing tel que $L(T) \in P$.

Prenons, M, w, construit tel que M_w : pour l'entrée x :
1. Simuler M sur w. Si c'est accepté, aller en étape 2. Si c'est rejeté, on rejette.
2. Simuler T sur x, si c'est accepté on accepte, sinon on rejette.

M accepte w est équivalent à dire que $L(M_w) \in P$.
Maintenant nous pouvons donc créer un "décideur" pour A_{TM} tel que :
$S =$ pour une entrée M, w :
1. Construire M_w
2. Exécuter R_p sur M_w et donner la même réponse.

On a donc un "décideur" pour le problème de l'arrêt. Ce qui n'est pas possible. On a donc une contradiction.