914 . DÉCIDABILITÉ ET INDÉCIDABILITÉ. EXEMPLES

I/ Décidabilité : formalisme

1) Classes R et RE

déf 1 Une machine de Turing M est un 7-uplet $(Q, \Gamma, \Sigma, q_0, \delta, F, \#)$ où Q est un ensemble fini d'états, $q_0 \in Q$ est l'état initial, $F \subset Q$ sont des états finaux, $\Sigma \subset \Gamma$ est l'alphabet d'entrée, Γ l'alphabet de ruban et $\# \in \Gamma$ le symbole blanc.

déf 2. Un mot $w \in \Sigma^*$ est accepté par la machine M si l'exécution de M sur la configuration initiale $q_0 w$ mène à une configuration gqd où $q \in F$.

- Le langage accepté par une machine M, noté $\mathcal{L}(M)$ est l'ensemble des mots de Σ^* acceptés par M.

déf 3 Un langage $L \subset \Sigma^*$ est récursivement énumérable s'il existe une machine de Turing M telle que $L = \mathcal{L}(M)$

On note RE leur ensemble

déf 4 Un langage $L \subset \Sigma^*$ est décidable (= récursif) s'il existe une machine de Turing M sans calcul infini telle que $L = \mathcal{L}(M)$

On note R leur ensemble.

2) Quelques propriétés

déf 5 Un énumérateur est une machine de Turing dont la tête de lecture se déplace seulement vers la droite et qui écrit des mots de Σ^* séparés par # sur son ruban

prop 6 $L \in RE$ ssi L est l'ensemble des mots énumérés par un énumérateur.

prop 7 Il existe des langages sur Σ qui ne sont pas dans RE

prop 8 . $R \subset RE$

prop 9 Stabilités :

- Si L et $L' \in RE$ alors $L \cup L'$ et $L \cap L' \in RE$
- Si L et $L' \in R$ alors $L \cup L'$, $L \cap L'$ et $\Sigma^* \setminus L \in R$.
- Si L et $\Sigma^* \setminus L \in RE$ alors $L \in R$

3) Problèmes décidables ou indécidables

déf 10 Un problème de décision est un problème pour lequel la question posée ne peut avoir que "oui" ou "non" comme réponse.

déf 11 Soit P un problème dont on note I l'ensemble des instances (= entrées possibles pour P). Un codage est une fonction

$$<\,>: \begin{cases} I \longrightarrow \Sigma^* \\ i \mapsto <i> \end{cases}$$ calculable et injective

déf 12 Un problème P est dit décidable si le langage $L_P = \{<i> \mid i \text{ est une instance positive pour } P\}$ l'est (au sens de 1) et indécidable sinon

rq 13 La notion de décidabilité d'un problème ne dépend pas du codage

4) Réduction

déf 14 Une fonction $f : \Sigma^* \longrightarrow \Gamma^*$ est dite calculable s'il existe une machine de Turing M qui pour toute entrée w s'arrête avec $f(w)$ sur son ruban

déf 15 Soit A et B deux problèmes de langages L_A et L_B sur les alphabets Σ_A et Σ_B. Une réduction de A à B est une fonction $f : \Sigma_A^* \longrightarrow \Sigma_B^*$ calculable et telle que $w \in L_A \Leftrightarrow f(w) \in L_B$

prop 16 . Si A se réduit à B et B est décidable alors A aussi

- Si A se réduit à B et A est indécidable alors B aussi.

II/ Indécidabilité et machines de Turing

prop 17 Le problème de l'arrêt est

ARRET : $\left\{\begin{array}{l}\text{entrée : une machine de Turing } M\text{, un mot } w\\ \text{sortie = oui si l'exécution de } M \text{ sur } w \text{ s'arrête, non sinon}\end{array}\right.$

Il est indécidable. cf Annexe A

prop 18 Le problème du mot est (pour les machines de Turing)

MOT_MT : $\left\{\begin{array}{l}\text{entrée = une machine de Turing } M\text{, un mot } w\\ \text{sortie = oui si } w \in L(M)\text{, non sinon}\end{array}\right.$

Il est indécidable.

rq 19 Le langage $LU = \{\langle M, w\rangle \mid w \in L(M)\}$ est néanmoins dans RE. Une machine qui accepte LU est appelée machine universelle

Th20 Théorème de Rice (DEV1)

Soit $\mathcal{P}$ telle que $\emptyset \subsetneq \mathcal{P} \subsetneq RE$. Alors

$L_{\mathcal{P}} = \{\langle M\rangle \text{ machine de Turing} \mid L(M) \in \mathcal{P}\}$ est indécidable

ex21 Le problème du langage vide :

VIDE_MT : $\left\{\begin{array}{l}\text{entrée : une machine de Turing } M\\ \text{sortie : oui si } L(M) = \emptyset\text{, non sinon}\end{array}\right.$

est indécidable.

III/ Décidabilité et langages rationnels

prop 22 Le problème du mot pour les automates finis est

MOT : $\left\{\begin{array}{l}\text{entrée : un automate fini } A\text{, un mot } w\\ \text{sortie : oui si } w \in L(A) \text{ non sinon}\end{array}\right.$

Il est décidable.

prop 23 Le problème du langage du langage vide pour les automates finis est

VIDE : $\left\{\begin{array}{l}\text{entrée : un automate fini } A\\ \text{sortie : oui si } L(A) = \emptyset\text{, non sinon.}\end{array}\right.$

Il est décidable (accessibilité dans un graphe)

prop 24 Le problème du langage universel pour les automates finis est

UNIV : $\left\{\begin{array}{l}\text{entrée : un automate fini } A\\ \text{sortie : oui si } L(A) = \Sigma^*\text{, non sinon}\end{array}\right.$

Il est décidable (par réduction à VIDE)

IV/ Décidabilité et indécidabilité sur les langages algébriques

1) Problèmes décidables

prop 25 Le problème du mot pour les grammaires algébriques (on remplace l'automate A de la prop 22 par une grammaire algébrique) est décidable via l'algorithme CYK

prop 26 Le problème du langage vide pour les grammaires algébriques est décidable

2) Problèmes indécidables

def 27 Le problème de correspondance de Post est

PCP : $\left\{\begin{array}{l}\text{entrée = un ensemble } (u_1, v_1), \dots (u_m, v_m) \text{ de paire de mots sur } \Sigma \text{ (de cardinal } \geq 2)\\ \text{sortie : oui s'il existe une suite d'indice } i_1, \dots i_n \in [\![1, m]\!]\\ \text{tels que } u_{i_1} \dots u_{i_n} = v_{i_1} \dots v_{i_n}\end{array}\right.$

def 28. Le problème de correspondance de Post modifié est

PCPM : $\left\{\begin{array}{l}\text{entrée : un ensemble } (u_1, v_1) \dots (u_m, v_m) \text{ de paires de mots sur } \Sigma\\ \text{sortie oui s'il existe une suite d'indices } i_1 \dots i_n \in [\![1, m]\!]\\ \text{tels que } i_1 = 1 \text{ et } u_{i_1} \dots u_{i_n} = v_{i_1} \dots v_{i_n}\end{array}\right.$

prop 29 PCP et PCPM sont indécidables

prop 30 Le problème

INTER : $\begin{cases} \text{entrée} = G \text{ et } G' \text{ grammaires algébriques} \\ \text{sortie : oui si } L(G) \cap L(G') = \emptyset \end{cases}$

est indécidable par réduction de POST à INTER

prop 31 Le problème

EGAL : $\begin{cases} \text{entrée} = G \text{ et } G' \text{ grammaires algébriques} \\ \text{sortie : oui si } L(G) = L(G') \end{cases}$

est indécidable par réduction de INTER à EGAL.

rq 32 Ceci implique que le problème du langage universel pour les grammaires algébriques est aussi indécidable.

V/ Décidabilité et logique

prop 33 Le problème VALIDE $\begin{cases} \text{entrée : } \varphi \text{ formule du calcul propositionnel} \\ \text{sortie : oui si } \varphi \text{ est valide, non sinon} \end{cases}$

est décidable.

1) Théories décidables

Dans la suite, on se donne L un langage du premier ordre et $\vdash$ un système de déduction correct et complet

def 34 Une théorie est un ensemble de formules closes (sur L)

def 35 Une théorie T est complète si $T \nvdash \bot$ et pour toute formule close F, $T \vdash F$ ou $T \vdash \neg F$

def 36 Une théorie T est récursive si le problème

REC : $\begin{cases} \text{entrée : une formule } F \\ \text{sortie : oui si } F \in T\text{, non sinon} \end{cases}$ est décidable

def 37 Une théorie T est décidable si le problème

DEC : $\begin{cases} \text{entrée : une formule close } F \\ \text{sortie : oui si } T \vdash F\text{, non sinon} \end{cases}$ est décidable.

th 38 Une théorie complète et récursive sur un langage dénombrable est décidable

2) Presburger ...

def 39 Le langage de l'arithmétique de Presburger est $\mathcal{L} = \{0, s, +, =\}$

Th 40 Le problème

PRES : $\begin{cases} \text{entrée : une } \mathcal{L}\text{-formule close, } \varphi \\ \text{sortie : oui si } \mathbb{N} \models \varphi\text{, non sinon} \end{cases}$

est décidable

DEV 2

3) Et au delà

def 41 si on ajoute à L_{Pr} symbole de fonction $\times$ et aux axiomes de l'arithmétique de Presburger ceux relatifs à la multiplication, on obtient la théorie de l'arithmétique de Peano

prop 42 [Admis]

La théorie de Peano n'est pas décidable

En effet, toute théorie T non contradictoire (= admettant un modèle et contenant T_0 (les axiomes de Peano sauf la récurrence) est indécidable. cf annexe B

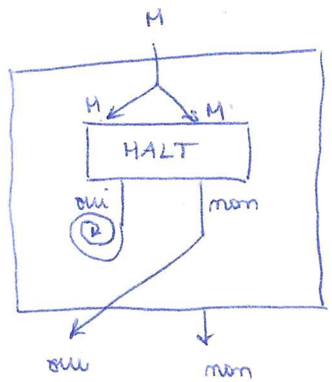
rq 43

PEANO : $\begin{cases} \text{entrée : } \varphi \text{ formule close sur } L' = \{+, \times, =, s, 0\} \\ \text{sortie : oui si } \mathbb{N} \models \varphi\text{, non sinon} \end{cases}$

est indécidable.

ANNEXE A

Si une machine HALT décide ARRET alors donner M' en entrée à M' donne une contradiction



$= M'$

ANNEXE B

Axiomes de P_0 :

$\forall x, \neg(s(x) = 0)$

$\forall x . \exists y \ (\neg x = 0 \rightarrow s(y) = x)$

$\forall x \forall y \ (s(x) = s(y) \Rightarrow x = y)$

$\forall x, (x + 0 = x)$

$\forall x \forall y \quad x + s(y) = s(x + y)$

$\forall x, \ x \times 0 = 0$

$\forall x \forall y \quad x \times s(y) = x \times y + x$