30/01 2015

914 Décidabilité et indécidabilité. Exemples.

pb de décision

Réf : C-L, WOL, Carton.

Nous nous intéressons aussi Problèmes binaires :
▷ question générique dont la réponse est oui ou non

Encodage :
→ On encode les instances par des mots sur un alphabet Σ fini.
→ On appelle instances positives les mots qui encodent les instances pour lesquelles la réponse est oui

Un problème est représenté par le langage formé de ces instances positives.

I Classes de Décidabilité

définition 01 : Mot accepté ; langage décidé par une MT
Soit $\omega \in \Sigma^*$ et soit M, une machine de Turing d'alphabet Σ. Soit L, un langage sur Σ.
On dit que M :
- accepte ω si le calcul de M à partir de ω mène à une configuration acceptante
- refuse ω si le calcul de M à partir de ω mène à une configuration refusante
- décide L si M accepte tout les mots de L et refuse tout les autres

définition 02 : Langages récursifs : classe R
La classe de décidabilité R est l'ensemble des langages décidables par une Machine de Turing.

La classe R est la classe des langages : récursifs ; décidables ; calculables décidés par une machine de Turing ; solubles algorithmiquement.

définition 03 : Langages récursivement enumérables : RE
La classe de décidabilité RE est l'ensemble des langages acceptés par une Machine de Turing.

La classe RE est la classe des langages : acceptés par une machine de Turing ; partiellement récursifs, décidables, calculables, solubles algorithmiquement.

Lemme 01 : ~~$RE \subseteq R$~~ $R \subseteq RE$.

II Un premier langage indécidable

WOL?

Lemme 02 : $\exists L \notin R$

L'ensemble des machines de Turing sur Σ est dénombrable. On note ces Machines $(M_i)_{i \in \mathbb{N}}$.
De même, on note $(\omega_i)_{i \in \mathbb{N}}$ l'ensemble des mots sur Σ.

définition 04 : Le langage diagonal
On appelle langage diagonal le langage
$D = \{ \omega_i, i \in \mathbb{N}, M_i \text{ n'accepte pas } \omega_i \}$

Théorème 01 : $D \notin RE$

preuve :
Si $D \in RE$, $\exists k \in \mathbb{N}$, M_k accepte D et donc
$\omega_k \in D \iff M_k$ n'accepte pas ω_k [définition de D] $\iff \omega_k \notin D$ par définition de M_k.

III Un deuxième langage indécidable

définition 05: La classe co-RE

La classe de décidabilité co-RE est l'ensemble des langages L tels que $\bar{L} \in RE$.

lemme03: clôture de R par complémentarité

$\forall L \in R, \bar{L} \in R$

Proposition 01:

$co\text{-}RE \cap RE = R$

Théorème 02:

$D \in co\text{-}RE$; c'est-à-dire :

$\bar{D} = \{ \omega_i, i \in \mathbb{N}, M_i \text{ accepte } \omega_i \} \in RE$

à mettre en valeur

remarque: Ce langage n'appartient pas à R.

IV La méthode de Réduction

définition 06: Fonction calculable

$f : \Sigma_1^* \to \Sigma_2^*$

Soient Σ_1 et Σ_2, deux alphabets, soient L_1 et L_2, deux langages sur Σ_1 et Σ_2. Soit $f : L_1 \to L_2$

On dit que f est calculable si il existe une machine de Turing qui accepte tout mot ω de L_1 et termine avec $f(\omega)$ en sortie

définition 07: Réduction

Soient L_1 et L_2, deux problèmes de décision d'alphabets respectifs Σ_1 et Σ_2.

On appelle réduction de L_1 vers L_2 toute fonction calculable $f : \Sigma_1^* \to \Sigma_2^*$ vérifiant : $\forall \omega \in \Sigma_1^*, \omega \in L_1 \Leftrightarrow f(\omega) \in L_2$

on note alors $L_1 \leq_m L_2$ et on dit que L_1 se réduit à L_2.

Théorème03: Théorème de réduction

Soit L, un problème de décision.

Si $\exists L'$, un problème décidable tel que

$L \leq_m L'$,

alors L est décidable

→ et celui aux problèmes indécidables

voir la figure 01.

Dans la suite de cette partie des exemples de problèmes indécidables sont présentés.

1. Le langage universel

Le langage $LU := \{(M, \omega), M \text{ accepte } \omega\}$ est indécidable.

2. Le problème de l'arrêt Dup

Le problème $Arret := \{(M, \omega), M \text{ s'arrête sur } \omega\}$ est indécidable.

3. Un cas applicatif

Le problème de déterminer si un programme écrit dans un langage de programmation usuel s'arrête pour des valeurs fixées de ses données est indécidable.

4. Le mot vide & le langage vide

Les langages suivants sont indécidables:

$H_\emptyset := \{M, M \text{ s'arrête sur le mot vide}\}$

$L_\emptyset := \{M, L(M) = \emptyset\}$

III Le théorème de Rice Dvp

Théorème 04: Le théorème de Rice

Soit P, une propriété sur les langages. Si P est non triviale, alors le problème suivant est indécidable.

$R_P = \{M, L(M) \text{ vérifie } P\}$

Corollaire: Le drame de la vérification

Il n'existe pas d'Algorithme qui prend en entrée un programme et sa spécification et qui vérifie si il est correct

propriété ←

Ex: programme les entiers:

un

P: soustraction sur deux entiers

entrée: n, n, $n \geq m$.

sortie: $p(n, m)$

Spécification: $\forall n, m,\ n \geqslant m \Rightarrow p(n, m) + m = n$.

↳ logique du premier ordre; =, > : symboles de prédicat.

↳ p: symbole de fonction.

voir la logique de Hore ⇓ triplets de Hore.

V L'arithmétique de Peano

L'arithmétique de Peano vise à axiomatiser l'arithmétique, c'est-à-dire, les propriétés des entiers.

On s'attend donc à ce que ces axiomes possède au moins un modèle: le modèle standard des entiers.

On s'intéresse donc au problème:

$Th(\mathbb{N}) = \{$formules closes sur la signature $\{0, s, +, \cdot\}$ vraie sur les entiers$\}$

Théorème 05: Gödel

$Th(\mathbb{N})$ est indécidable

démonstration:

On montre $Th(\mathbb{N}) \geqslant_m \overline{Arret}$:

On construit γ, formule à partir d'une instance (M, ω) telle que

$\gamma \in Th(\mathbb{N}) \Leftrightarrow (M, \omega) \in \overline{Arret}$ en exprimant arithmétiquement:

→ des faits sur les mots

→ le diagramme Espace-Temps de M.

On construit ainsi une formule F telle que $F(n)$ une suite de configuration de M sur ω et est vérifié si la dernière est acceptante.

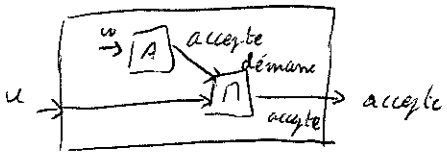
$\gamma = \exists n\, F(n)$ convient

* Commentaires :

- ajouter des exemples ! - ajouter du 1er ordre.

* sur le Th de Rice (8 min)

- rallonger : avec des exemples,



- exemple plus proche des MT :

$\Sigma = \{a, b\}$, (propriété contenir un a)

$R_P = \{ M ;\ L(M) \text{ contient } ab \}$

P = contenir ab.

- exemple de propriétés où l'on ne s'arrêterait pas : propriété sur la machine et pas sur les langages (ex : avoir un seul état final).

- ici, définir triviale : $P = \emptyset$ ou $P = \{M\}$, tlles les machines / $P \subset RE$. $P \neq 0$ et $P \neq RE$.

* sur le Th de l'arrêt (6 min).

(cf Sipser)

Autres dots : - décidabilité de Presburger.

- pb de Post.

- sur les langages (complémentaire d'un langage alg est alg)

- pb de terminaison d'un syst de réécriture ; indécidable.

Quest° : arithm Peano : indécidable. → et avec $\mathbb{Q}$ à la place de $\mathbb{N}$? f : {formule Th($\mathbb{N}$)} → {formule Th($\mathbb{Q}$)} calculable.

arithm Presburger : décidable ⇒ Théorie complet ?

F satisfaite sur $\mathbb{N}$ ⇔ $f(F)$ satisfaite sur $\mathbb{Q}$.

Soit $F = \exists x, x < 0$

$f(F) = \exists x, (x \geq 0) \wedge (x < 0)$.

$f(\forall x\, \varphi) = \forall x, x \geq 0 \Rightarrow f(\varphi)$.

$f(\exists x\, \varphi) = \exists x, x \geq 0 \wedge f(\varphi)$

↳ si on remplace $\mathbb{Q}$ par les réels

$\exists x, 1 < x < 2$.

$\neg \exists x, 1 < x < 2$.

Pb de savoir si une formule est finiment satisfiable RE \ R.

" " " est valide est dans RE. ⇒ on peut énumérer les preuves.

Th : φ valide ⇔ φ est un Th.

Figure 01 : Le principe de réduction