914 — Décidabilité, indécidabilité. Exemples.

AUT p13/92 WOL p3

I Définitions et outils pour la décidabilité

1) Problèmes et codages

Déf Un problème de décision P est la donnée de
→ E un ensemble
→ p un prédicat sur E
Les éléments de E sont appelés instances du problème, ceux vérifiant p instances positives, les autres instances négatives. On notera resp P^+ et P^- leur ensemble.

Rq On donnera plutôt les problèmes sous la forme entrée/sortie.

ex PAIR [entrée : n un entier naturel — ici $P = \mathbb{N}$
sortie : oui s'il est pair, non sinon — $P^+ = 2\mathbb{N}$, $P^- = 2\mathbb{N}+1$.

Déf Un codage pour un problème (de décision) $P = (E, p)$ est une fonction injective de E dans Σ^* où Σ est un alphabet fini, on la notera souvent $\langle - \rangle$.
Le langage associé au problème est alors $\mathcal{L}_P = \{\langle x \rangle \mid x \in P^+\}$ l'ensemble des codages des instances positives.

ex L'écriture en binaire est un codage possible des entiers naturels sur $\{0,1\}$.
$\mathcal{L}_{PAIR} = \omega \in \{0,1\}^* \mid \omega = \omega_n \dots \omega_1$ où $\omega_1 = 0\}$

2) Rappels sur les machines de Turing

Déf Une machine de Turing* (MT) $\mathcal{M}$ décide un langage $\mathcal{L} \subset \Sigma^*$ si pour tout mot $\omega \in \Sigma^*$ { $\mathcal{M}$ accepte ω si $\omega \in \mathcal{L}$; $\mathcal{M}$ termine et rejette ω si $\omega \notin \mathcal{L}$
Une MT $\mathcal{M}$ accepte $\mathcal{L} \subset \Sigma^*$ si pour tout mot $\omega \in \mathcal{L}$, $\mathcal{M}$ accepte ω.

Pté [$\mathcal{M}$ décide $\mathcal{L}$ ⇒ $\mathcal{M}$ accepte $\mathcal{L}$.

* déterministe.

WOL p108

Déf Un langage est dit récursif s'il existe une MT qui le décide. On note R la classe des langages récursifs.
Un langage est dit récursivement énumérable s'il existe une MT qui l'accepte. On note RE la classe des lang. réc. énum.

WOL p109

Pté [$R \subseteq RE$ **Pté** [R est stable par passage au complémentaire

Soit Σ un alphabet fini. Σ^* est dénombrable, on note $(\omega_i)_{i \in \mathbb{N}}$ ses mots. L'ensemble des MT sur Σ est dénombrable, on les note $(\mathcal{M}_i)_{i \in \mathbb{N}}$.
On introduit alors $L^\circ = \{\omega \in \Sigma^* \mid \omega = \omega_i$ et $\mathcal{M}_i$ n'accepte pas $\omega\}$

Pté [$L^\circ \notin RE$. En revanche $\overline{L^\circ} \in RE$ donc $\overline{L^\circ} \in RE \setminus R$

Cor [• l'inclusion $R \subsetneq RE$ est stricte
• RE n'est pas stable par passage au complémentaire

WOL p142 → 145

Rq Si $\mathcal{L}$ est accepté par une MT non dét, il l'est aussi par une MT dét. C'est pourquoi on ne précise pas le genre de MT pour définir RE

WOL p115

3) Décidabilité des problèmes

Déf Un problème P est décidable s'il existe un codage tel que $\mathcal{L}_P \in R$
——— indécidable sinon
——— semi-décidable s'il existe un codage tq $\mathcal{L}_P \in RE \setminus R$

cf fig 1

AUT p92

Pté [P décidable ⇒ $\overline{P}$ décidable où $\overline{P}$ désigne le problème

⚠ C'est faux pour semi-décidable. ci-dessous un contre exemple :

PL° [entrée $\omega \in \Sigma^*$
sortie oui si $\omega = \omega_i$ et $\omega_i \notin \mathcal{L}_{acc}(\mathcal{M}_i)$
est indécidable non semi-décidable

$\overline{PL^\circ}$ [entrée $\omega \in \Sigma^*$
sortie oui si $\omega = \omega_i$ et $\omega_i \in \mathcal{L}_{acc}(\mathcal{M}_i)$
est semi-décidable (et donc indécidable)

Déf Soient A et B deux problèmes de décision.
Une réduction de A à B est une procédure effective pour transformer une instance a de A en une instance b de B tq $a \in A^+$ ssi $b \in B^+$

cf fig 2.

AUT p92

Pté [A se réduit à B ; A est indécidable } ⇒ B est indécidable.

AUT p93

Résumé En pratique pour MQ P est décidable on donne une procédure effective, tandis que pour MQ P est indécidable on cherche à réduire un pb indéc connu à P.

II Problèmes indécidables théoriques

1) Problème de l'arrêt et d'acceptation

AUT p93 WOL p146

ARRET [entrée : $\mathcal{M}$ une machine de Turing sur Σ, ω un mot de Σ^* ; sortie : oui si $\mathcal{M}$ s'arrête lancée sur ω, non sinon

Pté $\boxed{\text{ARRET est semi décidable}}$

NB : on montre que $\mathcal{L}_{ARRET}$ est dans RE grâce à une machine universelle
on montre que ARRET est indécidable par l'absurde.

ACCEPT [entrée : $\mathcal{M}$ une machine de Turing sur Σ, ω un mot de Σ^* ; sortie : oui si $\mathcal{M}$ accepte ω, non sinon.

Pté $\boxed{\text{ACCEPT est semi-décidable}}$

NB : on montre que $\mathcal{L}_{ACCEPT}$ est dans RE grâce à une autre machine-univ.
on montre que ACCEPT est indécidable en réduisant ARRET à ACCEPT.

2) Théorème de Rice

AUT p94 WOL p150

Déf Soit p un prédicat sur RE (ie p est une partie de RE)
On dit que p est trivial si $p = \emptyset$ ou $p = RE$

P_p [entrée : $\mathcal{M}$ une MT ; sortie : oui si $\mathcal{L}acc(\mathcal{M}) \in p$, non sinon

Pté [(Théorème de Rice)
Si p est un prédicat non trivial sur RE
alors le problème P_p est indécidable

Csq Le théorème de Rice signifie qu'on ne peut pas construire de procédure permettant de vérifier qu'un programme est valide, si on considère tous les programmes possibles.

3) Le problème de correspondance de Post.

AUT p96

PCP [entrée : $(u_i, v_i)_{i \in [1..m]}$ un nombre fini de couples de mots ; sortie : oui s'il existe $(\alpha_i)_{i \in [1..k]} \in [1..m]^k$ tq $\begin{cases} u_{\alpha_1} \dots u_{\alpha_k} = v_{\alpha_1} \dots v_{\alpha_k} \\ k \geq 1 \end{cases}$.

4 fig 3

Pté $\boxed{\text{PCP est semi-décidable.}}$

NB on montre que PCP est indécidable par réduction d'ACCEPT à PCP.

III Problèmes issus de la logique

1) Théories et problèmes associés

DNR p79

Déf Une théorie est un ensemble d'énoncés (ie formules closes) appelés axiomes.

⚠ Ici une théorie n'est pas nécessairement close par conséquence.
Ne pas confondre avec les théorèmes d'une théorie ou d'un modèle.

Pour la suite on fixe $\vdash$ un système de déduction correct et complet.

DNR p80

Déf Une théorie T est consistante ssi $T \nvdash \bot$.
En particulier pour toute formule F on n'a pas simultanément $T \vdash F$ et $T \vdash \neg F$

Déf Une théorie T est complète ssi $\begin{cases} T \text{ est consistante et} \\ \text{pour toute formule close } \varphi \text{ on a } T \vdash \varphi \text{ ou } T \vdash \neg\varphi \end{cases}$

DNR p125

Déf Une théorie T est récursive ssi l'appartenance à T est décidable. APP(T)
——— décidable ssi être conséquence de T est décidable. CSQ(T)

⚠ Connaître les axiomes est différent de connaître les conséquences :
Il ne faut donc pas confondre les deux problèmes suivants

APP(T) [entrée : φ une formule close ; sortie : oui si $\varphi \in T$, non sinon $\neq$ CSQ(T) [entrée : φ formule close ; sortie : oui si $T \vdash \varphi$, non sinon

DNR p126

Pté [Soit T une $\mathcal{L}$-théorie. $\left.\begin{array}{c} T \text{ complète et récursive} \\ \mathcal{L} \text{ dénombrable} \end{array}\right\} \Rightarrow$ T est décidable

2) Arithmétique de Péano

DNR p111

Déf Le langage de l'arithmétique de Péano est $\mathcal{L}_{Péano} = \{0, s, +, \times, =\}$

cst, f unaire, f binaire, relation binaire

PÉANO [entrée : φ une $\mathcal{L}_{Péano}$ formule close ; sortie : oui si $\mathbb{N} \models \varphi$, non sinon.

DNR p111

Déf La théorie de l'arithmétique élémentaire est

$$P_0 = \left\{ \begin{array}{lll} \forall x \ s(x) \neq 0 & (A_1) ; & \forall x, \ x + 0 = 0 \quad (A_4) \\ \forall x \ (x = 0) \vee (\exists y, s(y) = x) & (A_2) ; & \forall x \ \forall y \ x + s(y) = s(x+y) \quad (A_5) \\ \forall x, \forall y \ s(x) = s(y) \rightarrow x = y & (A_3) ; & \forall x \ x \times 0 = 0 \quad (A_6) \\ & & \forall x \ \forall y \ x \times s(y) = x \times y + x \quad (A_7) \end{array} \right\}$$

DNR p111

Déf La théorie de l'arithmétique de Péano est l'arithmétique élémentaire à laquelle on ajoute le schéma de récurrence

$PA = P^0 \cup \{ [F[x:=0] \wedge \forall y\, F[x:=y] \to F[x:=s(y)]] \to \forall x\, F \mid F\ \mathcal{L}_{Péano}\text{-formule} \}$

NB : P^0 était fini donc récursif. PA est infini mais aussi récursive.

DNR p126

Pté ADMIS [Soit T une $\mathcal{L}_{Péano}$-théorie. T non contradictoire, $P_0 \subset T$ } ⟹ T est indécidable.

Csq
- Puisque $\mathbb{N} \models PA$, PA est non contradictoire et elle contient P_0 donc P_0 est indéc
- En notant $Th(\mathbb{N}) = \{\varphi$ un $\mathcal{L}_{Péano}$-énoncé $\mid \mathbb{N} \models \varphi\}$ les théorèmes de $\mathbb{N}$ on a
  - → $Th(\mathbb{N})$ est non contradictoire
  - → $P_0 \subset Th(\mathbb{N})$ car $\mathbb{N} \models P_0$.

  } donc $Th(\mathbb{N})$ est indécidable.

Or pour φ un $\mathcal{L}$-Péano énoncé $\mathbb{N} \models \varphi$ ssi $\varphi \in Th(\mathbb{N})$ ssi $Th(\mathbb{N}) \vdash \varphi$
donc PÉANO EST INDÉCIDABLE.

DNR p127

Pté ADMIS (Théorème d'incomplétude de Gödel)
Soit T une $\mathcal{L}_{Péano}$-théorie contenant PA et non contradictoire
Si T est récursive alors T n'est pas complète.

3) Arithmétique de Presburger

DNR p136

Déf Le langage de l'arithmétique de Presburger est $\mathcal{L}_{Pres} = \{0, s, +, =\}$

PRES [entrée : φ une $\mathcal{L}_{Pres}$-formule close.
sortie : oui si $\mathbb{N} \models \varphi$, non sinon

Déf "local" On dira qu'une $\mathcal{L}_{Pres}$-formule atomique est simple si elle est de la forme $x_i = 0$ ou $x_i = x_j$ ou $x_i + x_j = x_k$ ou $s(x_i) = x_k$ (pr $i, j, k \neq$).

Lemme [Pour F une combinaison booléenne de $\mathcal{L}_{Pres}$ formules atomiques simple il existe un automate – qu'on peut effectivem^t construire – qui reconnait exactem^t les codages des k-uplets d'entiers naturels satisfaisant φ, pour un codage bien choisi

Pté PRES est décidable.] DVP 1.

inspiré de Carton p 178

IV Problèmes sur les langages algébriques

1) Problèmes décidables

VIDE [entrée $G = (\Sigma, \Gamma, R, S)$
sortie oui si $L_G(S) = \emptyset$, non sinon

MOT-VIDE [entrée $G = (\Sigma, \Gamma, R, S)$
sortie oui si $\varepsilon \in L_G(S)$

VARIABLE [entrée $G = (\Sigma, \Gamma, R, S)$, $T \in \Gamma$ une variable
sortie oui si T apparaît ds $\widehat{L_G(S)}$

MOT [entrée $G = (\Sigma, \Gamma, R, S)$, ω un mot de Σ^*
sortie oui si $\omega \in L_G(S)$, non sinon.

Aut 94 p 80, 81, 83

Pté [VIDE, MOT-VIDE et VARIABLES sont décidables par une méthode de saturation.

Ca On peut effectivement transformer une grammaire en une grammaire réduite équivalente (grâce à VIDE et VARIABLE), ou en une grammaire propre quasi-équivalente (grâce à MOT-VIDE) et donc sous forme normale de Chomsky (FNC)

[FB] p320

Pté [L'algorithme C.Y.K, basé sur la prog. dynamique, détermine si un mot appartient à une grammaire sous FNC.

└ **Ca** MOT est aussi décidable.

2) Problèmes indécidables

INTER-VIDE [entrée $G = (\Sigma, \Gamma, R, S)$, $G' = (\Sigma, \Gamma', R', S')$
sortie oui si $L_G(S) \cap L_{G'}(S') = \emptyset$

EGAL [entrée $G = (\Sigma, \Gamma, R, S)$, $G' = (\Sigma, \Gamma', R', S')$
sortie oui si $L_G(S) = L_{G'}(S')$

UNIV [entrée $G = (\Sigma, \Gamma, R, S)$
sortie oui si $L_G(S) = \Sigma^*$

AMBIGU [entrée $G = (\Sigma, \Gamma, R, S)$
sortie oui si G ambiguë.

Pté [Ces quatre problèmes sont indécidables.) DVP 2.

Carton p 166

Rq En revanche pour des langages rationnels ces problèmes sont décidables puisqu'on sait
→ faire l'automate produit
→ complémenter un automate
→ décider la vacuité d'un langage reconnu par un automate.

fig 1

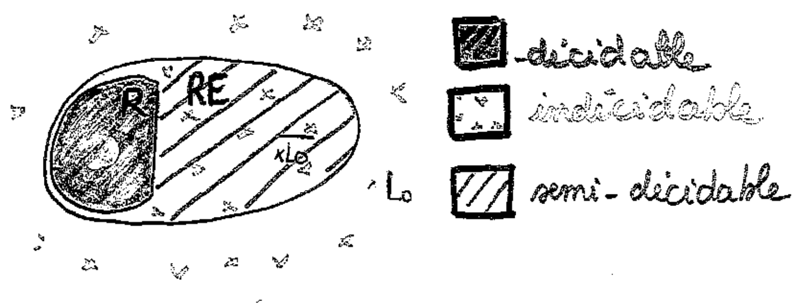
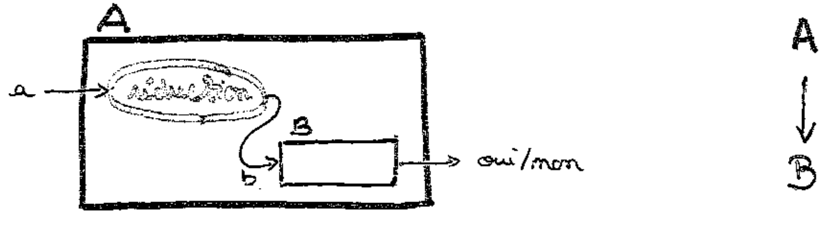


fig 2



réduction du problème A au problème B

fig 3

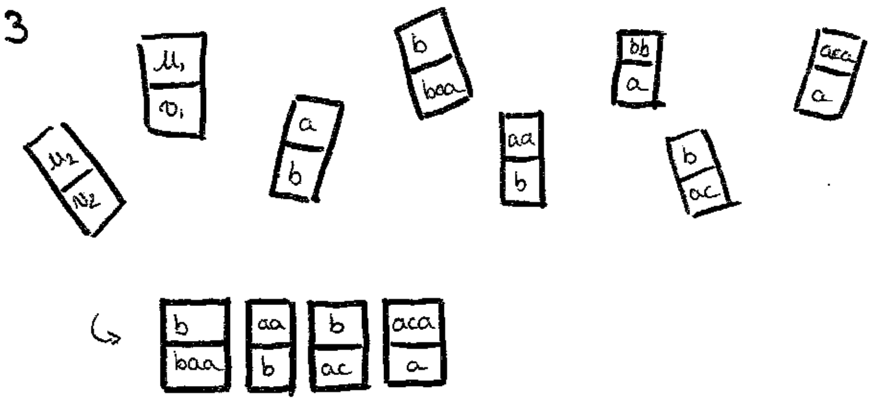
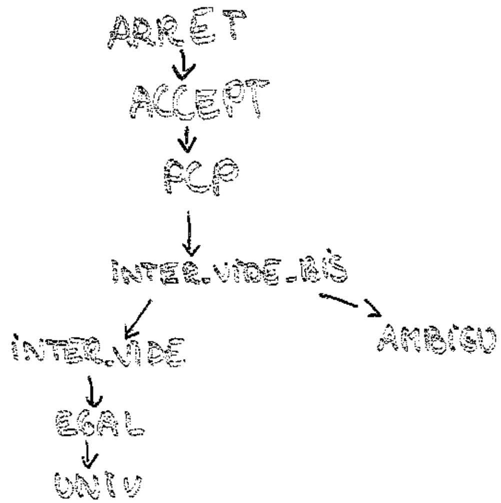


fig 4



Références.

[Aut94] J.M Autebert
[F-B] Floyd Biegel
[Cart] Olivier-Carton
[AUT] J.M Autebert. Calculabilité et décidabilité. éd Masson.
[WOL] P. Wolper Introduction à la calculabilité, éd. Dunod
[DNR] David, Nour, Raffali Introduction à la logique, éd Dunod