Informatique Théorique, TD 4 : Indécidabilité (2/2)

1 Le Problème de Correspondance de Post (PCP)

Le Problème de Correspondance de Post (PCP) prend en entrée un alphabet fini A, un entier N, une suite finie de paires de mots $(u_i, v_i)_{i\in[1..N]}$, et demande s'il existe une suite finie non vide d'indices $(i_p)_{p\in[1..P]}$ telle que $\Pi_{p=1}^{P} u_{i_p} = \Pi_{p=1}^{P} v_{i_p}$ i.e. telle que $u_{i_1}u_{i_2}u_{i_3}...u_{i_P} = v_{i_1}v_{i_2}v_{i_3}...v_{i_P}$

La variante avec début imposé impose que le premier indice soit $i_1 = 1$

Le problème de Post avec début imposé $(aa, aabaa), (baabbabaab, bbab), (bbb, bb)$ a-t-il une solution ?

Le problème de Post avec début imposé $(abab, ab)$, (a, aa), (b, aba), (a, b) a-t-il une solution ?

Le problème de Post avec début imposé $(Y, Y000000pX)$ (X, X) $(0, 0)$, $(1, 1)$ $(0p, 1q)$, $(1p, p0)$, $(q0, 0q)$, (qX, pX), $(Xp0, Xp)$, (XpX, X) a-t-il une solution ?

Montrez que les problèmes de Post avec début imposé, et sans début imposé, sont indécidables.

2 Grammaires

Une grammaire G est donnée par un alphabet "terminal" A (traditionnellement noté avec des minuscules), un alphabet "non-terminal" B disjoint du précédent (t. noté avec des majuscules), un symbole de départ dans B (t.: S) et des règles de dérivation $X \to r$ avec $X \in B$ et $r \in (A + B)^*$. Une dérivation est une suite de mots commençant par S, finissant par un mot m de A^*, chaue mot étant obtenu à partir du précédent en remplaçant un non-terminal X par le mot r où $X \to r$ est une dérivation. Le mot m est alors dit généré par la grammaire G. Le langage $L(G)$ est l'ensemble des mots qui peuvent être générés par G. Un langage L est algébrique ssi il existe une grammaire G telle que $L = L(G)$

Généralités préliminaires. Les agébriques sont-ils stables par union, intersection, complémentaire ? Montrez que si L est reconnaissable, alors L est algébrique. Savoir si un mot est généré par une grammaire est-il décidable ?

Intersection Soit A un alphabet, $(u_i, v_i)_{i\in[1..N]}$ des paires de mots de A^*, et $x \notin A$. On note $m(w)$ le miroir du mot w. Donnez une grammaire qui génère $\{m(u_{i_1}u_{i_2}u_{i_3}...u_{i_P})\, x\, v_{i_1}v_{i_2}v_{i_3}...v_{i_P} \mid (i_p)_{p\in[1..P]} \text{ suite non vide}\}$ et une autre qui génère $\{m(w)\, x\, w \mid w \in A^*\}$. Etant donné deux grammaires G_1 et G_2, est-il décidable de savoir si $G_1 \cap G_2 \neq \emptyset$?

Universalité (1) On considère les deux grammaires G_1 et G_2 suivantes sur l'alphabet $\{a, b, c\}$

		Y	$\to a\,Y\,c$	A	$\to a\,A$	G_2	$\to a\,G_2\,a$	V	$\to a\,V$
G_1	$\to Y$		$\to a\,A\,B$		$\to \epsilon$		$\to b\,G_2\,b$		$\to b\,V$
	$\to Z\,C$		$\to B\,C\,c$	B	$\to b\,B$		$\to a\,V\,c\,V\,b$		$\to \epsilon$
		Z	$\to a\,Z\,b$		$\to \epsilon$		$\to b\,V\,c\,V\,a$	W	$\to a\,V$
			$\to a\,A$	C	$\to c\,C$		$\to W\,c$		$\to b\,V$
			$\to B\,b$		$\to \epsilon$		$\to c\,W$		

Montrez que $aa\,bb\,cccc$ et $aa\,bbbb\,cc$ sont dans $L(G_1)$. Qu'en est-il de $aaa\,bbb\,ccc$? Montrez que $aaababb\,c\,abbbaaa$ et $aaababb\,c\,abaaa$ sont dans $L(G_2)$. Qu'en est-il de $aaababb\,c\,bbabaaa$? Décrire en français ce que sont $L(G_1)$ et $L(G_2)$.

(2) On se donne N paires de mots $(u_i, v_i)_{i\in[1..N]}$ sur $\{a, b\}$. $\binom{u_1}{v_1}=\binom{bbb}{abb}$, $\binom{u_2}{v_2}=\binom{ab}{a}$, $\binom{u_3}{v_3}=\binom{b}{bb}$
Par exemple (exemple pour toute la question), $N = 3$,

Soit $X = \{n_i \mid i \in [1..N]\}$ (c.-à-d. $\{n_1, n_2, n_3\}$ pour l'exemple), $A = \{a, b\}$ et $\Gamma = X \cup A \cup \{c\}$.

La fonction m est la fonction miroir : $m(abb) = bba$, $m(n_1n_3n_2n_2) = n_2n_2n_3n_1$

La fonction t_u de X^* dans A^* remplace chaque n_i par u_i. Exemple $t_u(n_1n_1n_2n_2n_3) = bbb\,bbb\,ab\,ab\,b$. De même, la fonction t_v remplace chaque n_i par v_i. Exemple $t_v(n_1n_1n_2n_2n_3) = abbabbaabb$.

On considère les langages sur Γ :

$L_0 = X^*c\,A^*c\,A^*c\,X^*$, et L_1 = complémentaire de L_0. [Ex: $n_3n_2n_1\,c\,abba\,c\,bab\,c\,n_1n_1 \in L_0, \in L_2, \in L_3, \in L_{4g}, \in L_{4d}$]

L_2 = les mots $x_g\,c\,w_g\,c\,w_d\,c\,x_d$ de L_0 tels que $w_g \neq m(w_d)$ [Ex.: $n_2n_3\,c\,bab\,c\,abb\,c\,n_3n_2 \in L_2$ car $bab \neq m(abb)$]

L_3 = les mots $x_gcw_gcw_dcx_d$ de L_0 t.q. $x_g \neq m(x_d)$ [Ex.: $n_1n_2\,c\,abbbb\,c\,bbbba\,c\,n_1n_3 \in L_3$ car $n_1n_2 \neq m(n_1n_3)$]

L_{4g} = les mots $x_gcw_gcw_dcx_d$ de L_0 t.q. $w_g \neq t_u(m(x_g))$ [Ex.: $n_2n_3\,c\,bba\,c\,abb\,c\,n_3n_2 \in L_{4g}$ car $bba \neq t_u(m(n_2n_3)) = bab$]

L_{4d} = les mots $x_gcw_gcw_dcx_d$ de L_0 t.q. $w_d \neq m(t_v(x_d))$ [Ex.: $n_2n_3\,c\,bab\,c\,bab\,c\,n_3n_2 \in L_{4d}$ car $bab \neq m(t_v(n_3n_2)) = abb$]

$L_5 = L_1 \cup L_2 \cup L_3 \cup L_{4g} \cup L_{4d} \cup \{ccc\}$

Donnez un automate pour L_0, déduire que L_1 est algébrique. Décrire des grammaires pour L_2, L_3, L_{4g}, L_{4d} et L_5.

(3) Montrez que pour les $(u_i, v_i)_{i\in[1..N]}$ de l'exemple du (2), on a $L_5 \neq \Gamma^*$? En règle générale, à quelle condition sur $(u_i, v_i)_{i\in[1..N]}$ a-t-on $L_5 = \Gamma^*$?

Une grammaire $\mathcal{G}$ sur un alphabet Γ est dite universelle ssi $L(\mathcal{G}) = \Gamma^*$. Savoir si une grammaire est universelle est-il décidable ?, semi-décidable ?, co-semi-décidable ?

3 Rice

En génie logiciel, on cherche à vérifier que les programmes vérifient certaines propriétés, dites "spécifications".

Une spécification fonctionnelle repose uniquement sur la fonction produite par la machine, et non sur son fonctionnement interne. Par exemple, "la machine rend un entier non nul sur les entrées paires et ne rend pas de résultat sur les entrées impaires" est une spécification fonctionnelle. Par contre, "la machine est de complexité polynomiale" ou bien, "il existe une entrée qui engendre une exécution qui passe par tous les états de la machine" ne sont pas des spécifications fonctionnelles.

Une telle spécification est totalement décrite par le sous-ensemble X des fonctions partielles de $\mathbb{N}$ dans $\mathbb{N}$ qui vérifient la propriété souhaitée. Savoir si une machine M vérifie la propriété souhaitée se ramène alors au problème $\mathcal{P}_X$ qui prend en entrée une machine de Turing M et demande si $f_M \in X$, où f_M désignera la fonction réalisée par la machine M.

Exemples: Si Y est l'ensemble des fonctions totales, le problème $\mathcal{P}_Y$ demande si la machine en entrée s'arrête sur toute entrée. Si *prime* est la fonction indicatrice des nombres premiers, alors le problème $\mathcal{P}_{\{prime\}}$ demande si la machine en entrée est une machine qui décide si son entrée est un nombre premier.

(1) Soit $X_1 = \emptyset$, X_2 l'ensemble de toutes les fonctions partielles de $\mathbb{N}$ dans $\mathbb{N}$, et X_3 l'ensemble de toutes les fonctions partielles calculables. Est-ce que $\mathcal{P}_{X_1}$, $\mathcal{P}_{X_2}$ et $\mathcal{P}_{X_3}$ sont décidables ? Justifiez.

On suppose à partir de maintenant que X est un ensemble de fonction telle qu'il existe deux machines M_{oui} et M_{non} avec $f_{M_{oui}} \in X$ et $f_{M_{non}} \notin X$ (la machine M_{oui} vérifie la spécif. tandis que M_{non} ne la vérifie pas).

On notera $f_\emptyset$ la fonction définie nulle part (Attention, ne confondez pas $\mathcal{P}_\emptyset$ et $\mathcal{P}_{\{f_\emptyset\}}$)

(2) On suppose pour cette question que $f_\emptyset \notin X$. Pour toute machine M et toute entrée u, on note $\Pi_{M,u}$ la machine qui prend en entrée v, simule M sur u mais jette l'éventuel résultat à la poubelle PUIS simule M_{oui} sur v. Que vaut $f_{\Pi_{M,u}}$? Est-ce que $\mathcal{P}_X(\Pi_{M,u})$? Discutez en fonction de M et de u.

Montrez que $\mathcal{P}_X$ est indécidable.

(3) Montrez que $\mathcal{P}_X$ est également indécidable si $f_\emptyset \in X$.

4 Indécidabilité de l'arrêt par la fonction *Kolmogorov*

On considère un langage de programmation X (qui peut être C, Pascal, Caml, ..., peu importe)

Soit $A = \{0, 1\}$. On note A^* l'ensemble des mots sur A, i.e. des suites finies à valeur dans A. On note $|m|$ la longueur d'un mot m (Exemple, $|0010110| = 7$)

On Définit la fonction K qui à chaque mot $m \in A^*$, associe la longueur en nombre de bits du plus court programme X qui ne fait aucune lecture, qui termine, et ce après avoir eu comme effet d'écrire le mot m.

(1) Montrez $\exists D, H \in \mathbb{N}$, $\forall m \in A^*$, $K(m) \leq H * |m| + D$

(2) Montrez qu'il existe une suite de mots $(m_i)_{i \in \mathbb{N}}$ tq $|m_i|$ tende vers l'infini et $K(m_i) = o(ln(ln(ln(|m_i|)))))$

(3) Montrez que $K(m) \rightarrow_{|m| \rightarrow \infty} \infty$.

(4) Soit X_n un mot aléatoire de longueur n. Montrez que la probabilité que $K(X_n) < n - 10$ est inférieure à 0.1%

(5) Montrez que si l'arrêt etait décidable, alors K serait calculable. Qu'en déduisez vous pour la calculabilite de K ?

On définit la fonction f_K qui à n associe le plus petit mot m tq $K(m) > n$ où, par définition, le mot m_1 est plus petit que le mot m_2 ssi (il est plus court) ou (il est de même longueur, mais apparaît avant dans l'ordre alphabétique)

(6) Montrez que si K est calculable, alors il existe des constantes G et L telle que pour tout $n \in \mathbb{N}$, $K(f_K(n)) \leq G + L * ln\, n$ (On rapelle que le nombre de bits pour écrire n en base 2 est environ $ln_2 n$)

(7) Montrez que K n'est pas calculable. En déduire une nouvelle démonstration de l'indécidabilité du problème de l'arrêt.

5 Machines avec oracle Arrêt

Une machine déterministe avec oracle Arrêt (une MTDOA) peut à tout moment demander si une MTD M s'arrête sur un entrée u. Pour cela, elle écrit M et u sur une bande particulière, la bande-oracle, puis elle passe dans l'état $q_?$. Le prochain pas de la MTDOA la fait passer dans l'état q_{oui} si M sur u s'arrête et dans l'état q_{non} sinon.

(1) Une MTDOA peut-elle décider de l'arrêt des MTD ? (2) Une MTD peut-elle décider de l'arrêt des MTDOA ?

(3) Montrez que les MTDOA ne peuvent pas décider de l'arrêt des MTDOA.

Le problème *NegArretQuelqueSoit* prend en entrée une MTD M et demande s'il est faux qu'elle s'arrête sur toute entrée, ie il demande s'il existe au moins une entrée sur laquelle la MTD ne s'arrête pas.

Un problème est Π_2^{oracle} s'il peut être semi-décidé par une MTDOA. Un problème est $\Pi_2^{logique}$ s'il existe une propriété décidable D telle que le problème sur l'entrée m rend vrai ssi $\exists x, \forall y, D(m, x, y)$

(4) Montrez que *NegArretQuelqueSoit* est Π_2^{oracle} (5) Montrez que *NegArretQuelqueSoit* est $\Pi_2^{logique}$

(6) Montrez que tout problème $\Pi_2^{logique}$ est Π_2^{oracle} (7) [délicat] Montrez que tout problème Π_2^{oracle} est $\Pi_2^{logique}$

(8) Montrez que *NegArretQuelqueSoit* est Π_2 complet, ie qu'il y a une réduction de tout problème Π_2 dans *NegArretQuelqueSoit*.

(9) Montrez que *NegArretQuelqueSoit* n'est pas décidable par MTDOA.