Logique & Calculabilité

Hubert Comon ∧ David Baelde
{comon,baelde}@lsv.ens-cachan.fr

Exercice -1

1. **Donnée:** les codes de deux machines de Turing M_1 et M_2 qui calculent en temps polynômial ;
   Question: $L(M_1) = L(M_2)$?
2. Un automate linéairement borné est une machine de Turing qui, lorsqu'elle lit un blanc, écrit un blanc et se déplace vers la gauche.
   Donnée: le code d'un automate linéairement borné M ;
   Question: M s'arrête-t-il sur toute donnée ?

Exercice 0

Un système de réécriture de mots sur l'alphabet Σ est donné par un ensemble fini de paires $(u, u') \in \Sigma^* \times \Sigma^*$. La relation de réécriture associée est $\rightarrow_R \subseteq \Sigma^* \times \Sigma^*$ définie par

$$w \rightarrow_R w' \text{ ssi } \exists (u, u') \in R.\ \exists w_1, w_2 \in \Sigma^*.\ w = w_1 u w_2 \wedge v = w_1 u' w_2$$

Montrer que le problème d'*accessibilité* est indécidable :

Donnée: un système de réécriture R et deux mots $u, v \in \Sigma^*$;
Question: est-ce que $u \rightarrow_R^* v$?

Exercice 1

Montrer que le problème suivant est indécidable :

Donnée: un ensemble fini de formules du premier ordre $\mathcal{S}$ et une formule ϕ ;
Question: est-ce que $\mathcal{S} \models \phi$ ou $\mathcal{S} \models \neg\phi$?

Exercice 2

1. Le problème suivant est-il décidable ?
   Donnée: Un ensemble fini de tuiles T, deux relations de compatibilité $H, V \subseteq T \times T$ et une tuile de bordure $t_0 \in T$;
   Question: Existe-t-il un rectangle pavable avec T ?

Un rectangle $(n,m) \in \mathbb{N}^2$ est pavable (par T, V, H, t_0) s'il existe une application $f : [0, ..., n+1] \times [0, m+1] \to T$ telle que
– pour tout i, $f(0,i) = f(i,m+1) = f(i,0) = f(n+1,i) = t_0$
– pour tout $i \leq n$, pour tout j, $(f(i,j), f(i+1,j)) \in H$
– pour tout $j \leq m$, pour tout i, $(f(i,j), f(i,j+1)) \in V$
– pour tout $1 \leq i \leq n$ et tout $1 \leq j \leq m$, $f(i,j) \neq t_0$

2. Le problème suivant est-il décidable ?
Donnée: un ensemble fini de formules du premier ordre S;
Question: S a un modèle fini ?

Exercice 3
L'idée de la *complexité descriptive* (ou complexité de Kolmogorov) est de quantifier la complexité d'une donnée par sa plus courte description dans un langage donné. On pose ainsi

$$K_f(x) = \min\{\ |d| \mid f(d) = x\ \}$$

On considère en particulier $K := K_T$ où T est la fonction d'exécution des machines de Turing : $T(\langle M \rangle) = M(\epsilon)$ et $T(d) = \perp$ quand d n'est pas un code d'une machine de Turing. On se restreint ici à des machines de Turing à un ruban et on fixe l'alphabet à $\{0,1\}$.

1. Montrer qu'il existe une constante c telle que $K(xx) \leq K(x) + c$ pour tout x. Montrer qu'il existe une constante c telle que pour tout x et n on a $K(x^n) \leq K(x) + \log_2(n) + c$.
2. Quelle serait la notion de complexité obtenue si l'on avait considéré des machines à plusieurs rubans, ou votre langage de programmation préféré ? Établir des majorations pour montrer que notre définition est relativement indépendante du formalisme tant que celui-ci est Turing complet.
3. Montrer que pour tout n il existe un mot w de taille au plus n tel que $K(w) \geq n$. (Selon le point de vue, on qualifie ces objets d'*aléatoires* ou *incompressibles*.)
4. Montrer que la fonction K n'est pas calculable.

Bonus : Soit une machine M telle que pour tout $i \in \mathbb{N}$, $M(i)$ renvoie une paire $\langle x, n \rangle$ telle que $K(x) \geq n$. Montrer que les valeurs de n sont nécessairement bornées. (Vous aurez montré le cœur du *théorème d'incomplétude de Chaitin*, qui dit qu'aucune logique raisonnable et correcte ne peut exhiber des objets de complexité arbitrairement grande.)