Logique & Calculabilité

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

Exercice 1
Dire si les problèmes suivants sont décidables ou non, en justifiant :

1. **Donnée:** le code $\langle M \rangle$ d'une machine de Turing ;
   Question: M s'arrête-t-elle sur le mot vide ?
2. **Donnée:** le code $\langle M \rangle$ d'une machine de Turing ;
   Question: M s'arrête-t-elle sur au moins une entrée ?
3. **Donnée:** les codes $\langle M \rangle$ et $\langle M' \rangle$ de deux machines de Turing ;
   Question: $L(M) = L(M')$?
4. **Donnée:** le code $\langle M \rangle$ d'une machine de Turing et un mot w ;
   Question: M boucle-t-elle sur w ?
   On dit qu'une machine boucle sur w si à partir de la configuration initiale $\gamma_0(w)$ on atteint une configuration γ, et qu'à partir de γ on atteint de nouveau γ en au moins une transition.
5. **Donnée:** le code d'une machine de Turing M, un mot w et un entier n codé en base 2 ;
   Question: M accepte-t-elle w après au plus n transitions ?
6. **Donnée:** le code d'une machine de Turing M, un mot w et un entier n codé en base 2 ;
   Question: M accepte-t-elle w après au moins n transitions ?
7. **Donnée:** le code $\langle M \rangle$ d'une machine de Turing ;
   Question: est-ce que le complémentaire de $L(M)$ est récursivement énumérable ?
8. **Donnée:** les codes de machines de Turing M_1 et M_2 tel que M_2 s'arrête sur toute entrée w après au plus $2 \times |w|$ transitions ;
   Question: $M_1(w) = M_2(w)$ pour tout w ?
9. **Donnée:** le code d'une machine de Turing M ;
   Question: existe-t-il deux mots distincts w_1 et w_2 de même longueur tels que $w_1, w_2 \in L(M)$?

10. **Donnée:** le code d'une machine de Turing M;
    Question: M calcule-t-elle en temps polynômial ?
11. **Donnée:** les codes de deux machines de Turing M_1 et M_2 qui calculent en temps polynômial ;
    Question: $L(M_1) = L(M_2)$?
12. 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 2
Donner une fonction calculable dont l'image ne soit pas décidable.

Exercice 3
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 $\to_R \subseteq \Sigma^* \times \Sigma^*$ définie par

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

La relation de réduction $\to_R^*$ est la clôture réflexive transitive de la relation de réécriture. Par exemple, avec

$$R = \{(c, abaca), (c, bcbabab), (aca, d), (bcb, d), (ada, d), (bdb, d)\}$$

on a $c \to_R^* d$:

$$\begin{aligned} c &\to_R abaca \to_R ababcbababa \to_R abababacabababa \to_R abababdbababa \\ &\to_R ababadababa \to_R ababdbaba \to_R abadaba \to_R abdba \to_R ada \to_R d \end{aligned}$$

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 $u \to_R^* v$?

Exercice 4
1. Soit f une fonction récursive des codes des machines de Turing dans les codes des machines de Turing. Montrer qu'il existe une machine M telle que, sur toute donnée x, $f(\langle M \rangle)(x) = M(x)$.
2. Montrer qu'il existe une machine qui, quand elle s'exécute sur le mot vide, produit son propre code. (On peut construire de tels programmes, appelés *quines*, dans tout langage de programmation Turing complet.)