

Examen de substitution – Calculabilité avancée (SINB40A)

Durée : 2 heures

(Barème indicatif)

Documents : non autorisés

Exercice 1.

Réductions (12 points)

Soit $L_{\forall\in} = \{\langle M \rangle \mid \forall w : w \in L(M)\}$.Soit $L_{\exists\in} = \{\langle M \rangle \mid \exists w : w \in L(M)\}$.Soit $L_{\forall\notin} = \{\langle M \rangle \mid \forall w : w \notin L(M)\}$.Soit $L_{\exists\notin} = \{\langle M \rangle \mid \exists w : w \notin L(M)\}$.

1. Indiquer lequel(s) est(sont) semi-décidable(s), en justifiant.
2. Indiquer lequel(s) est(sont) équivalent(s) au problème de l'arrêt des machines de Turing, c'est-à-dire qui se réduit à $L_{\downarrow} = \{\langle M \rangle, w \mid M(w) \downarrow\}$ et auquel $L_{\downarrow}$ se réduit, en justifiant.
3. Selon votre intuition, indiquer pour chacun des quatre langages $L_{\forall\in}$, $L_{\exists\in}$, $L_{\forall\notin}$, $L_{\exists\notin}$, s'il est décidable ou non, semi-décidable ou non.
4. Justifier précisément le plus possible de vos réponses à la question 3, à l'aide de réductions, du théorème de Rice, et du fait que le problème de l'arrêt est indécidable.

Exercice 2.

Automate cellulaire élémentaire (3 points)

On s'intéresse dans cet exercice aux automates cellulaires en dimension $d = 1$, avec deux états $Q = \{0, 1\}$ et voisinage $\{-1, 0, 1\}$ (la cellule elle-même et ses deux plus proches voisines).

1. Quel est le type de la règle locale d'un tel automate cellulaire ?
(c'est-à-dire, la règle locale est une fonction de quel ensemble vers quel ensemble?)
2. Donner la règle locale de l'automate cellulaire élémentaire 30 (sous la forme d'un tableau).

Un automata cellulaire en dimension $d = 1$ est *permutatif-gauche* lorsque pour toutes configurations $x, y \in Q^{\mathbb{Z}}$ égales partout sauf en position $i \in \mathbb{Z}$ (c'est-à-dire $x(i) \neq y(i)$, et $x(j) = y(j)$ pour tout $j \in \mathbb{Z} \setminus \{i\}$), on a $F(x)(i+1) \neq F(y)(i+1)$.

3. Montrer que l'automate cellulaire élémentaire 30 est permutatif-gauche.

Exercice 3.

Arrêt ? (5 points)

Soit F un système formel récursivement énumérable et permettant d'exprimer l'arrêt des machines de Turing.

1. Que signifie *récursivement énumérable* ici ?
2. Si un programme M termine sur une entrée w , alors peut-on toujours démontrer l'énoncé « $M(w) \downarrow$ » ? Justifier.

Soit le programme suivant :

- (a) obtenir son propre code,
- (b) énumérer toutes les preuves de F , jusqu'à :
 - rencontrer une preuve de sa propre terminaison, et alors *boucler*,
 - rencontrer une preuve de sa propre non-terminaison, et alors *s'arrêter*.
3. En analysant la terminaison ou la non-terminaison de ce programme, en déduire que F ne pas être à la fois cohérent (ne jamais démontrer à la fois un énoncé E et sa négation $\neg E$), et complet (pour tout énoncé E , démontrer soit E soit sa négation $\neg E$).