Examen – Calculabilité avancée (SINB40A)

Durée : 2 heures *(Barème indicatif)*
Documents : non autorisés

Exercice 1. *Réductions (11 points)*

Soit $L_{\forall\downarrow} = \{\langle M \rangle \mid \forall w : M(w)\downarrow\}$ l'ensemble des codes de machines de Turing qui s'arrêtent sur toutes les entrées,
et soit $L_* = \{\langle M \rangle \mid L(M) = \Sigma^*\}$ l'ensemble des codes de machines de Turing qui acceptent tous les mots.

1. Appliquer le théorème de Rice pour montrer que L_* n'est pas un langage décidable.
2. Est-ce que $L_{\forall\downarrow} \subseteq L_*$? Est-ce que $L_* \subseteq L_{\forall\downarrow}$? Justifier.
3. Proposer une transformation calculable pour la réduction $L_{\forall\downarrow} \leq_m^T L_*$ (sans justification).
4. Proposer une transformation calculable pour la réduction $L_* \leq_m^T L_{\forall\downarrow}$ (sans justification).

Soit $L_\downarrow = \{(\langle M \rangle, w) \mid M(w)\downarrow\}$ l'ensemble des couples composés d'un code de machine de Turing et d'une entrée tels que la machine s'arrête sur cette entrée,
et soit $L_\uparrow = \{(\langle M \rangle, w) \mid M(w)\uparrow\}$ l'ensemble des couples composés d'un code de machine de Turing et d'une entrée tels que la machine ne s'arrête pas sur cette entrée.
Rappel : $L_\downarrow$ est semi-décidable, mais pas décidable.

5. En supposant que $L_\uparrow$ et $L_\downarrow$ sont complémentaires (ce qui est presque exact), que peut-on en déduire (à partir du rappel) sur la décidabilité et la semi-décidabilité de $L_\uparrow$?
6. Montrer que $L_\downarrow \leq_m^T L_{\forall\downarrow}$.
7. Montrer que $L_\uparrow \leq_m^T L_{\forall\downarrow}$.
   Indication : utiliser la taille de l'entrée pour tester de plus en plus d'étapes.

Suite au verso ⥁

Exercice 2. *Automate cellulaire élémentaire (3 points)*

On s'intéresse dans cet exercice aux automate cellulaire 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. *λ-calcul (3 points)*

Rappel : un λ-terme $t = (\lambda x.t')\ t''$ se β-réduit en $t'[x := t'']$, et peut être α-converti en $t[x := y]$.

$$\begin{aligned} \mathtt{n} &= \lambda f.\lambda x.(f\ \dots (f\ x)\dots) \\ \star &= \lambda m.\lambda n.\lambda f.(m\ (n\ f)) \end{aligned}$$

1. β-réduire le λ-terme suivant : $\star\ \mathtt{2}\ \mathtt{2}$.

2. Proposer un λ-terme dont la β-réduction ne termine pas.

Exercice 4. *Incomplétude (3 points)*

On dit qu'un système formel est *décidable* s'il existe un algorithme qui, étant donné un énoncé, décide s'il admet une démonstration ou s'il n'en admet pas.

1. Montrer que si un système formel est *complet* (pour tout énoncé E, soit E soit $\neg E$ est démontrable), alors il est décidable.

2. En déduire que si les énoncés d'un système formel permettent exprimer la notion de calcul et d'arrêt des machines de Turing, alors ce système formel ne peut pas être à la fois complet et *correct* (tout énoncé démontré est vrai).
Indication : utiliser vos connaissances sur le problème de l'arrêt des machines de Turing.