Calculabilité avancée (SINBU06) – Examen – Session 1

Durée : 2 heures *(Barème indicatif)*
Documents : non autorisés
Ce sujet comporte **2 pages** et 4 exercices

Exercice 1. *Réduction et Rice (12 points)*
Soient l'ensemble des codes de machines de Turing dont le calcul sur le mot vide s'arrête :

$$L_{halt\epsilon} = \{\langle M\rangle \mid M(\epsilon)\downarrow\}$$

et l'ensemble des codes de machines de Turing qui reconnaissent le langage vide :

$$L_{\emptyset} = \{\langle M\rangle \mid L(M) \neq \emptyset\}.$$

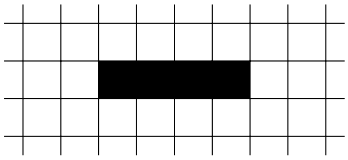
1. Montrer que $L_{halt\epsilon} \leq_m^T L_{\emptyset}$.
2. Appliquer le théorème de Rice pour montrer que $L_{\emptyset}$ est indécidable.

Exercice 2. *Jeu de la vie (4 points)*
Une *configuration* $c \in \{0,1\}^{\mathbb{Z}^2}$ du jeu de la vie associe à chaque case de l'espace ($\mathbb{Z}^2$) un état *mort* (0) ou *vivant* (1). Une configuration c est *finie* lorsque $\sum_{p\in\mathbb{Z}^2} c(p)$ est fini. On note

$$f : \{0,1\}^{\mathbb{Z}^2} \to \{0,1\}^{\mathbb{Z}^2}$$

la règle d'évolution *globale* du jeu de la vie. Le **problème de la mort** consiste à décider, étant donnée une configuration finie c, si $\exists t \in \mathbb{N} : \forall p \in \mathbb{Z}^2 : (f^t(c))(p) = 0$?

1. L'instance ci-dessous est-elle positive (réponse « oui ») ou négative (réponse « non ») ?



case vide = morte
case pleine = vivante

2. Donner en quelques phrases une idée de démonstration de l'indécidabilité du problème de la mort, en soulignant quelques difficultés techniques.
Indication : on attend une idée de réduction, avec un « $\iff$ ».

Exercice 3. *Auto-référence (4 points)*

ZFC désigne le *système formel de Zermelo–Fraenkel avec l'axiome du Choix*, qui est communément admis comme une formalisation de l'ensemble des mathématiques (si vous n'aimez pas l'axiome du Choix, vous pouvez considérer ZF). ZFC définit des axiomes et règles de déduction, permettant de construire des *démonstrations*. L'ensemble des démonstrations est infini mais récursivement énumérable (un programme sans fin peut les lister). Un système formel est *cohérent* s'il ne permet pas de démontrer un énoncé et sa négation. Un système formel est *complet* si pour tout énoncé A, il permet de démontrer A ou $\neg A$. Un système formel est *correct* s'il ne permet de démontrer que des énoncés vrais (nous ne définirons pas ici mathématiquement la *vérité*, votre intuition sur cette notion suffira : si un énoncé est vrai, ce qu'il dit est vrai).

1. Écrire une petite dissertation sur l'arrêt ou non du programme suivant, en supposant que l'on peut bien programmer ce qui est indiqué (c'est le cas). Ce programme n'a pas d'entrée.

```
1. Obtenir mon propre code source P
2. Énumerer les démonstrations de ZFC, jusqu'à rencontrer :
   — une preuve de l'arrêt du programme P, alors boucler
   — une preuve du non-arrêt du programme P, alors s'arrêter
```

Inspiration : le premier théorème d'incomplétude de Gödel affirme que tout système formel suffisamment expressif ne peut pas être à la fois complet et cohérent.

Exercice 4. ~~*Turing-Universalité*~~ *Degrés Turing (4 points)*

Les *réductions Turing* (avec oracle) notées $\leq^T$ sont plus générales que les *réductions many-one Turing* notées $\leq^T_m$. On a $A \leq^T B$ lorsque, à partir d'un programme qui décide B (l'oracle), on peut construire un programme qui décide A. Le programme pour B est hypothétique (il peut correspondre à un problème indécidable), mais la construction du programme pour A doit être effective en utilisant le modèle des machines de Turing (ou du pseudo-code, comme on a l'habitude). La relation $\leq^T$ est un ordre partiel (réflexive, transitive, antisymétrique). Le **degré Turing** d'un langage X est l'ensemble des A équivalents à X pour les réductions Turing, noté $[X] = \{A \mid X \leq^T A \text{ et } A \leq^T X\}$. Ainsi $[\emptyset]$ correspond aux langages décidables, et $[L_{halte}]$ aux langages semi-décidables. Ils sont traditionnellement notés **0** et **0'**, respectivement.

1. Sur une feuille de brouillon que vous joindrez à votre copie, dessiner le plus fidèlement possible la forme générale du graphe des degrés Turing (dont les sommets sont les degrés $[X]$, et les arcs représentent l'ordre partiel $\leq^T$ avec $[X] \leq^T [Y]$ ssi $X \leq^T Y$), sachant que <u>tous</u> les énoncés suivants sont vrais :
— il existe $2^{\aleph_0}$ degrés Turing (le cardinal de $\mathbb{R}$) ;
— pour tout degré il existe un nombre dénombrable de degrés qui lui sont inférieurs ;
— pour tout degré il existe $2^{\aleph_0}$ degrés qui lui sont supérieurs ;
— il existe des degrés strictement entre **0** et **0'** ;
— il existe des degrés $[X] \neq \mathbf{0}$ tels qu'il n'existe pas de degré strictement entre **0** et $[X]$;
— pour tout degré $[X] \neq \mathbf{0}$, il existe des degrés qui lui sont incomparables ;
— il existe des paires de degrés qui n'ont pas de plus grande borne inférieure ;
— aucune séquence infinie et strictement croissante de degrés n'a de borne supérieure.
Indiquer clairement les sommets **0** et **0'** sur votre graphe.