Examen de Calculabilité

Documents et calculatrices non autorisés. Durée de l'épreuve : 2h.
Les questions les plus difficiles sont marquées d'un ♠.

Les exercices qui suivent sont indépendants et peuvent être traités dans le désordre. Toutes les réponses doivent être justifiées sauf mention contraire.

Exercice 1. Lever de rideau [6 points] Soient A et B des langages sur l'alphabet Σ.

(1) Si A est récursif, peut-on décider le langage $B = \{n \in \mathbb{N} : \exists u \in A, |u| = n\}$?

(2) Montrer que, si A et B sont r.e., et si $A \cup B$ et $A \cap B$ sont récursifs, alors A et B sont récursifs.

(3) Montrer que si A est r.e. et si le nombre de mots de A de chaque taille $n \in \mathbb{N}$ est calculable, alors on peut décider A.

Exercice 2. *Machines de Turing multi-coeurs* [6 points] Soient 2 machines de Turing M et M' sur le même alphabet d'entrée Σ. Soit N la machine de Turing qui sur l'entrée $w\#w'$, où $w, w' \in \Sigma^*$ et $\# \notin \Sigma$, simule une étape de calcul de M sur w puis une étape de M' sur w' et recommence jusqu'au terme du calcul. On remplace les états de rejet de M (resp. M') par des états sur lesquels M (resp. M') boucle infiniment sans se déplacer ou changer la lettre sur le ruban. Lorsqu'une des 2 simulations arrive dans un état acceptant de M ou M', N accepte. Si la première simulation nécessite plus d'espace à droite, N commence par décaler toute la partie du ruban correspondant à la seconde simulation, de même pour étendre la seconde simulation à gauche.

(1) Discuter, selon le comportement (mot accepté, mot rejeté, calcul qui ne s'arrête pas) de M sur w et de M' sur w', du comportement de N sur $w\#w'$.

(2) Démontrer en utilisant cette construction que si 2 langages L et L' sont récursivement énumérables alors $L \cup L'$ l'est aussi.

(3) Conclure pour toute union finie de langages récursivement énumérables.

(4) Est-ce vrai pour une union dénombrable de langages récursivement énumérables ? Justifier.

(5) En réutilisant cette idée de simulation, montrer que, pour $w, w' \in \Sigma^*$ fixés, le langage des machines M qui s'arrête sur w ou w' (possiblement les 2) est récursivement énumérable.

(6) ♠ Généraliser cette technique pour montrer que le langage des machines de Turing qui s'arrêtent sur une entrée au moins (dans Σ^*) est récursivement énumérable.
Indice : La suite $0, 0, 1, 0, 1, 2, 0, 1, 2, 3, 0, 1, 2, 3, 4 \ldots$ passe infiniment souvent par chaque entier.

Exercice 3. *R.e. espionnage* [8 points] On se donne une fonction totale calculable $f : \mathbb{N} \to \mathbb{N}$. On note $Im(f)$ l'ensemble image ($Im(f) = \{f(n) : n \in \mathbb{N}\}$).

(1) Montrer que $Im(f)$ est récursivement énumérable.

(2) ♠ Montrer que tout ensemble récursivement énumérable infini peut s'écrire $Im(f)$ pour une fonction f totale injective ($\forall n, n' \in \mathbb{N}, f(n) \neq f(n')$). On pourra utiliser la technique de la question **(6)** de l'exercice 2.

Soit M une machine de Turing calculant la fonction f totale injective. On note $A = \{x \in \mathbb{N} : \exists y > x, f(y) < f(x)\}$.

(3) Montrer que A est récursivement énumérable.

(4) Montrer que le complémentaire de A est infini.
Indice : Par l'absurde, on prend n tel que $n - 1$ soit le plus grand entier qui n'est pas dans A, puis on construit une suite infinie strictement décroissante d'entiers partant de n.

(5) On suppose qu'il existe B r.e. infini et tel que $B \cap A = \emptyset$. On veut montrer que $Im(f)$ est récursif :

(a) Pour $x \in \mathbb{N}$, montrer qu'il existe $y_x \in B$ tel que $f(y_x) > x$. Justifier que la fonction qui sur l'entrée x renvoie y_x est calculable.

(b) Montrer que, pour $z \in \mathbb{N}$, si $f(z) = x$ alors $z < y_x$.

(c) ♠ Conclure que $Im(f)$ est récursif.

(6) Montrer qu'il existe un ensemble récursivement énumérable de complémentaire infini qui intersecte tous les ensembles récursivement énumérables infinis.

Examen de Complexité

Documents et calculatrices non autorisés. Durée de l'épreuve : 2h.
Les questions les plus difficiles sont marquées d'un ♠.

Les exercices qui suivent sont indépendants et peuvent être traités dans le désordre. Toutes les réponses doivent être justifiées sauf mention contraire.

Exercice 1. Lever de rideau [10 points]

(1) Quelles sont les inclusions connues entre les classes P, NP, EXP et $NEXP$?

(2) Quelle réduction utilise-t-on pour comparer des langages de ces classes ? Pourquoi définir une classe de langages décidés en temps linéaire n'aurait pas d'intérêt vis à vis de cette réduction ?

(3) Donner la définition de NP-complétude et citer 4 problèmes NP-complets.

(4) Soit $L \in P$, a-t-on $\Sigma^* \setminus L \in P$? Et si $L \in NP$, a-t-on $\Sigma^* \setminus L \in NP$?

(5) On suppose l'existence d'une machine de Turing capable de résoudre une infinité d'instances distinctes de SAT en temps polynomial, peut-on en déduire $P = NP$?

Exercice 2. **3-COLOR** [10 points]

On définit le problème **3-COLOR** comme suit :
Entrée : Un graphe (S, A).
Question : Peut-on attribuer à chaque sommet de S une couleur parmi 3 tel que 2 sommets adjacents aient des couleurs différentes ?

On va montrer par réduction polynomiale de 3-SAT à 3-COLOR que 3-COLOR est NP-complet. On rappelle la définition de 3-SAT :
Entrée : Un ensemble fini de variables $\{x_1, x_2, \ldots, x_k\}$ et un ensemble fini de clauses $C_i = c_{i,1} \vee c_{i,2} \vee c_{i,3}$ où chaque $c_{i,j}$ est soit un x_l soit un $\neg x_l$, $1 \leq l \leq k$.
Question : Existe-t-il une affectation des variables qui satisfasse toutes les clauses ?

(1) Rappeler la définition de réduction polynomiale.

(2) Le graphe *Base* de la figure 1, contient un sommet V (Vrai), un sommet F (Faux) et un sommet I (Indéterminé). Montrer comment ajouter 2 sommets x et $\neg x$ à ce graphe de sorte que x et $\neg x$ aient les mêmes couleurs que V et F ou F et V respectivement.

(3) Étant donnés 2 sommets a et b chacun soit de la couleur de V, soit de celle de F, et un sommet o de la couleur de V, pour quelles couleurs de a et b peut-on colorier le graphe de la figure 2 ?

(4) ♠ Donner un graphe C contenant notamment les sommets a, b, c et o tel que :

- o ait la couleur de V ;
- a, b et c aient chacun soit la couleur de V, soit la couleur de F ;
- C soit 3-coloriable si et seulement si au moins 1 parmi a, b et c ait la couleur de V.

(5) ♠ Étant donnée une instance ϕ de 3-SAT, expliquer comment construire un graphe G tel que

$$\phi \text{ est satisfiable} \Leftrightarrow G \text{ est 3-coloriable}$$

Démontrer cette équivalence.

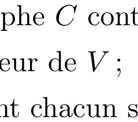

FIGURE 1 – Base

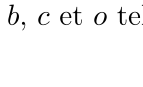

FIGURE 2 – Gadget