Info0705 : Informatique théorique
Année universitaire 2017-2018
Première session anticipée de janvier 2018
Examen terminal

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE
Master Sciences-Technologies-Santé
Mention Informatique

Notes :
- Seul l'aide-mémoire est autorisé. Tout autre document est interdit.
- Toute propriété de l'aide mémoire utilisée devra être citée (même brièvement). Si une propriété utilisée n'est pas dans l'aide-mémoire, alors cette propriété devra être démontrée.

QUESTIONS INDÉPENDANTES

1. Donner un exemple d'un langage non récursivement énumérable, et démontrer qu'il n'est pas récursivement énumérable.
2. Soit la fonction f calculable qui prend un mot w en entrée, et le convertit en $0^{|w|}$ si la longueur $|w|$ du mot est paire et en $1^{|w|}$ si la longueur $|w|$ du mot est impaire. Donner le graphe de la machine de Turing associée.
3. Démontrer que $\mathbf{P} = co\mathbf{P}$.
4. Pourquoi une démonstration similaire à celle faite à la question précédente ne peut pas être reprise afin de démontrer que $\mathbf{NP} = co\mathbf{NP}$.
5. Deux graphes non orientés G_1 et G_2 sont dits isomorphes s'ils représentent le même graphe (évidemment à l'ordre des sommets et arêtes près). On définit le langage $\mathtt{GRAPHISO} = \{\langle G_1, G_2\rangle \mid G_1$ et G_2 sont deux graphes isomorphes$\}$. Donner la classe de complexité temporelle d'un décideur déterministe de `GRAPHISO`.
6. Donner la classe de complexité temporelle d'un décideur non déterministe de `GRAPHISO`. Ce décideur devra utiliser de façon active le non déterminisme.
7. On rappelle que, dans un graphe, un triangle est un triplet de trois sommets qui sont reliés entre eux. On définit $\mathtt{TRIANGLEFREE} = \{\langle G\rangle \mid G$ est un graphe non orienté qui n'a pas de triangle$\}$. Montrer que $\mathtt{TRIANGLEFREE} \in L$.

EXERCICE 1: Régularité

Soit l'alphabet $\Sigma = \{1\}$. Dans cet exercice, on considère les langages sur Σ^*.

1. Soit $\mathtt{NOTDIV}_n = \{w \mid |w|$ n'est pas divisible par $n\}$. Montrer que $\mathtt{NOTDIV}_n$ est un langage régulier.
2. Démontrer que l'ensemble des langages réguliers est fermé par intersection.
3. Soit $\mathtt{PRIME} = \{w \mid |w|$ est un nombre premier$\}$. Montrer comment il est possible de définir `PRIME` à partir des langages $\mathtt{NOTDIV}_n$.
4. `PRIME` est-il un langage régulier ? La démonstration devra être rigoureuse.

EXERCICE 2: Différence symétrique

La différence symétrique de deux langages A et B (notée $DS(A, B)$) est l'ensemble des chaines qui sont uniquement dans l'un des deux ensembles. Par exemple, si $A = \{00, 101\}$ et $B = \{11, 00\}$, alors $DS(A, B) = \{11, 101\}$.

1. Supposons que $A = 0^*1^*$ et $B = 1^*0^*$. Quelles sont les chaines de longueur inférieure ou égale à 3 dans $DS(A, B)$?
2. Écrire l'expression régulière pour $DS(A, B)$.
3. Écrire l'expression de $DS(A, B)$ à partir d'opération sur les ensembles ($\cap, \cup, \ldots$).
4. Est-ce-que l'ensemble des langages réguliers est fermé par différence symétrique ?
5. Même question pour les langages libres de contexte.
6. Même question pour les langages décidables.
7. Même question pour les langages récursivement énumérables.

EXERCICE 3: $\mathtt{MINRUN}_n$ et $\mathtt{MAXRUN}_n$

On considère les deux langages suivants où M est une machine de Turing :
$\mathtt{MINRUN}_n = \{\langle M, w\rangle \mid M$ accepte w en n transitions au plus$\}$.
$\mathtt{MAXRUN}_n = \{\langle M, w\rangle \mid M$ accepte w en plus de n transitions$\}$.

1. $\mathtt{MINRUN}_n$ est-il récursivement énumérable ?
2. $\mathtt{MINRUN}_n$ est-il décidable ?
3. $\mathtt{MAXRUN}_n$ est-il récursivement énumérable ?

4. $\texttt{MAXRUN}_n$ est-il décidable ?
5. Soit $E = \texttt{MINRUN}_n \cup \texttt{MAXRUN}_n$. Que représente l'ensemble E ?
6. Nous avons démontré en TD que $\mathcal{RE}$ et $\mathcal{RE}$ étaient fermés par union. Peut-on en déduire alors quelque chose sur la récursive-énumérabilité et la décidabilité de E ?

Exercice 4: Automate Linéaire Borné

On appelle un automate borné linéaire (ou ABL) est une machine de Turing non déterministe telle que :

- la bande de travail dont on dispose a la taille n du mot d'entrée w (*i.e.* $n = |w|$),
- pour simplifier, on suppose que deux marqueurs de fin de bande \$ sont présents au début et à la fin de la bande,
- la tête de lecture reste bloquée si l'on tente de dépasser l'une des extrémités de la bande, à savoir bloquée à droite (resp. à gauche) si l'on est déjà à l'extrémité droite (resp. gauche) de la bande.

1. Entre une machine de Turing et un ABL, lequel se rapproche le plus d'un modèle ordinateur physique ?
2. Soit $L = \left\{1^{n!} \mid n \geq 1\right\}$. Donner le code de l'ABL qui reconnait ce langage (= idem code d'une machine de Turing mais qui n'utilise aucune place supplémentaire).

   Soit $A_{\texttt{ABL}} = \{< A, w > \mid A \text{ est un ABL qui accepte } \langle w\rangle \}$ le problème d'acceptation pour les ABLs.
3. Montrer que $A_{\texttt{ABL}}$ est récursivement énumérable.
4. Après avoir rappelé ce qu'est une configuration pour une machine de Turing, indiquez ce que l'on peut dire d'une machine de Turing qui, lors de son exécution, passe par deux configurations identiques ?
5. Le nombre de configurations possibles pour un ABL est-il fini, dénombrable ou infini ? S'il est fini, on donnera alors le nombre de configurations possibles ; s'il est dénombrable, on le mettra en bijection avec $\mathbb{N}$; s'il est infini, on utilisera la méthode de diagonalisation.
6. Est-on dans les conditions de l'application du théorème de Rice. On justifiera.
7. $A_{\texttt{ABL}}$ est-il décidable ?
8. Peut-on en conclure sur la faisabilité de détecter les boucle infinies sur un ordinateur physique ?
9. Déterminer à la classe **NSPACE** à laquelle $A_{\texttt{ABL}}$ appartient.
10. Déterminer à la classe **SPACE** à laquelle $A_{\texttt{ABL}}$ appartient.

Exercice 5: Problème du sac à dos

Le problème du sac à dos peut être décrit de la manière suivante. Je dispose d'un ensemble de n objets tels que chaque objet a un poids w_i et une valeur v_i. Est-il possible de remplir un sac à dos en choisissant un sous-ensemble S de ces objets, tel que mon sac à dos ne pèse pas plus lourd que w et contiennent une valeur d'au moins v ?

Ainsi, $\texttt{KNAPSACK} = \{\langle\{w_1, \ldots, w_n\}, w, \{v_1, \ldots, v_n\}, v\rangle \mid \exists U \subset \{1, \ldots, n\} \text{ tel que } \sum_{i\in U} w_i \leq w \text{ et } \sum_{i\in U} v_i \geq v\}$.

1. Montrer que $\texttt{KNAPSACK} \in \mathbf{NP}$ en utilisant un vérificateur.
2. Montrer que $\texttt{KNAPSACK} \in \mathbf{NP}$ en utilisant une machine de Turing non déterministe.
3. En quoi le problème SUBSET_SUM est-il très similaire au problème KNAPSACK ?
4. Démontrer $\texttt{SUBSET\_SUM} \leq_P \texttt{KNAPSACK}$ (on se souviendra que si $a \leq b$ et $a \geq b$, alors $a = b$).
5. Démontrer alors que KNAPSACK est **NP**-complet.

Exercice 6: Chemin Hamiltonien

On rappelle qu'un chemin simple dans un graphe non orienté est un chemin sans répétition de sommet. On définit les langages suivants :

- $\texttt{SPATH} = \{\langle G, a, b, k\rangle \mid G$ contient un chemin simple de a à b de longueur **au plus** k $\}$.
- $\texttt{HAMPATH} = \{\langle G, a, b\rangle \mid$ il existe un chemin Hamiltonien dans le graphe **orienté** G entre a et b $\}$
- $\texttt{UHAMPATH} = \{\langle G, a, b\rangle \mid$ il existe un chemin Hamiltonien dans le graphe **non orienté** G entre a et b $\}$

1. Montrer que $\texttt{SPATH} \in \mathbf{P}$.

   On veut montrer que UHAMPATH est **NP**-complet, en utilisant le fait que HAMPATH est **NP**-complet.
2. Montrer que $\texttt{UHAMPATH} \in \mathbf{NP}$.

   On veut proposer une fonction de transformation f qui transforme un graphe orienté $< G, a, b > \in \texttt{HAMPATH}$ en un graphe non orienté $< G', a, b > \in \texttt{UHAMPATH}$ (indice : ajouter des sommets supplémentaires de façon à forcer le parcours des sommets à avoir lieu dans un sens particulier), et vérifiant en plus les questions suivantes.
3. Montrer qu'avec cette transformation f, $w \in \texttt{HAMPATH} \Rightarrow f(w) \in \texttt{UHAMPATH}$.
4. Montrer qu'avec cette transformation f, $f(w) \in \texttt{UHAMPATH} \Rightarrow w \in \texttt{HAMPATH}$.
5. Montrer que f s'exécute en temps polynomial.
6. Que peut-on en conclure ? On justifiera précisément la conclusion.