Master Sciences-Technologies-Santé / Mention Informatique
Info0705 : Informatique théorique / Pascal Mignot
Année universitaire 2016-2017
Première session anticipée de janvier 2017 / Examen terminal

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

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.

Exercice 1 : questions indépendantes

1. Donner un exemple d'un langage non régulier (différent de ceux donnés dans le polycopié), et démontrer son irrégularité.
2. Donner un exemple d'une grammaire non libre du contexte (et différente de celles données dans le polycopié), et lui appliquer le lemme de l'étoile pour démontrer son irrégularité.
3. Démontrer que $\texttt{COMPOSITE} \in \mathbf{NP}$ en utilisant un vérificateur. Puis, l'illustrer par un exemple concret.
4. Démontrer que $\texttt{COMPOSITE} \in \mathbf{NP}$ en utilisant une machine de Turing non déterministe. Puis, l'illustrer par un exemple concret.
5. On reprend la machine de Turing non déterministe précédente en inversant dans son code accepter par rejeter et vice-versa. Quel est alors le langage reconnu par cette machine ?
6. Quelle est l'intérêt de savoir qu'il existe des langages **NP**-complet pour la classe **NP** ?
7. Pourquoi est-il possible de trouver un exemple tel que $A \leq_P B$ et $A \in$ **NP**-complet sans qu'il soit possible de conclure que B est **NP**-complet ?
8. Montrer que pour tout langage régulier L, alors $L \in \mathbf{SPACE}(\log(n))$.
9. Quel est l'argument théorique qui permet d'affirmer qu'il existe des langages non récursivement énumérable ?
10. Soit la propriété P = au $5^{\text{ème}}$ pas, la tête de lecture de la machine de Turing est au début de la bande. Déterminer si la propriété P est décidable, soit en appliquant rigoureusement le théorème de Rice, soit en donnant le code de son décideur.

Exercice 2 : exécution d'une machine de Turing

Soit $\texttt{HALT}_n = \{\langle M, w\rangle \mid M(w)$ s'arrête en au plus n transitions$\}$.

1. Rappeler ce qu'est une machine de Turing universelle U.
2. Donner la machine de Turing, basée sur une machine de Turing universelle, afin qu'elle accepte le langage $\texttt{HALT}_n$?
3. La machine donnée à la question précédente est-elle décidable ? On justifiera.
4. Donner le nombre maximum de configurations d'une machine $M \in \texttt{HALT}_n$. On l'exprimera notamment en fonction du nombre g de symboles de bande, q le nombre d'états de la machine.
5. Que peut-on dire des machines dans $\texttt{HALT}_n$ si $|w| > n$?
6. Dans quelle plus petite classe de complexité temporelle **TIME** se trouve $\texttt{HALT}_n$? On justifiera.
7. Dans quelle plus petite classe de complexité spatiale **SPACE** se trouve $\texttt{HALT}_n$? On justifiera.

Exercice 3 : décidabilité

Soit $A = \{\langle M_1, M_2\rangle \mid \exists w \in \Sigma^*, M_1(w)$ et $M_2(w)$ acceptent tous les deux$\}$.

1. Montrer que A est récursivement énumérable.
2. Montrer que A n'est pas décidable.
3. Montrer que $\overline{A}$ est indécidable.

Exercice 4 : calculabilité

Soit $\Sigma = \{0, 1\}$. Soit $f(\langle M, n\rangle)$ la fonction qui retourne le nombre maximum d'étapes nécessaires pour la machine de Turing M pour s'arrêter sur un mot de taille n.

1. Donner le code de la fonction qui calcule $f(\langle M, n\rangle)$.
2. $f(\langle M, n\rangle)$ est-elle totalement calculable, partiellement calculable ou non calculable si $M \in \mathcal{RE}$?
3. Supposons que $f(\langle M, n\rangle)$ soit totalement calculable. Montrer qu'il est alors possible d'écrire un décideur $N(\langle M\rangle)$ qui décide le problème de l'arrêt pour M.
4. Que peut-on alors déduire de la question précédente ?
5. $f(\langle M, n\rangle)$ est-elle totalement calculable, partiellement calculable ou non calculable si $M \in \mathcal{R}$?
6. Quelle est la complexité temporelle de $f(\langle M, n\rangle)$ si $M \in \mathbf{P}$?
7. Quelle est la complexité spatiale de $f(\langle M, n\rangle)$ si $M \in \mathbf{L}$?

Exercice 5 : Antilogie, satisfiabilité et NP-complétude

Une formule booléenne ϕ est une antilogie si elle est fausse quelque soit la valeur de vérité de ses littéraux. Soit le langage `ANTILOGY` = $\{\langle\phi\rangle$ tel que ϕ est une antilogie $\}$.

1. Donner un exemple d'une formule booléenne ϕ telle que $\phi \in$ `ANTILOGY`.
2. Donner la complexité temporelle de `ANTILOGY`.
3. Quel est l'ensemble complémentaire de `ANTILOGY` ? Par la suite, on appellera `A` cet ensemble.
4. Démontrer que `A` $\in$ **NP** en utilisant un vérificateur.
5. Démontrer que `A` $\in$ **NP** en utilisant une machine de Turing non déterministe.
6. Donner, dans les très grandes lignes, la démonstration que `A` est **NP**-complet.
7. Peut-on en déduire quelque chose sur `ANTILOGY` ?
8. Soit `FALSIFIABLE` = $\{\langle\phi\rangle$ tel qu'il existe une assignation des littéraux de ϕ qui rend ϕ faux $\}$. Montrer que `A` peut être réduit en temps polynomial à `FALSIFIABLE`.
9. Peut-on alors déduire immédiatement quelque chose sur `FALSIFIABLE` ? (**Attention**)

Exercice 6 : complexité temporelle

On définit `3COLOR` = $\{\langle G\rangle \mid$ les sommets de G peuvent être coloriés avec 3 couleurs de telle façon que toute arête relie des sommets de couleurs différentes $\}$.
On suppose que la propriété `3COLOR` **NP**-complet est admise comme vraie dans le reste de l'exercice.

1. Soit `BIPARTITE` = $\{\langle G\rangle \mid G$ est un graphe biparti non orienté$\}$. On rappelle qu'un graphe $G = (V, E)$ est biparti si l'ensemble des sommets V se partitionne en deux ensembles V_1 et V_2 (*i.e.* $V = V_1 \cup V_2$ et $V_1 \cap V_2 = \emptyset$) tels que pour toute arête $(u, v) \in E$, $(u, v) \in V_1 \times V_2$ ou $(u, v) \in V_2 \times V_1$. Montrer que `BIPARTITE` $\in \mathbf{P}$.
2. Soit `2COLOR` = $\{\langle G\rangle \mid$ les sommets de G peuvent être coloriés avec 2 couleurs de telle façon que toute arête relie des sommets de couleurs différentes $\}$.
   Montrer que `2COLOR` $\leq_{\mathbf{P}}$ `BIPARTITE`.
3. Montrer que `BIPARTITE` $\leq_{\mathbf{P}}$ `2COLOR`.
4. Quel sont les propriétés que l'on peut déduire des trois questions précédentes ? On justifiera.
5. Montrer que `2COLOR` $\leq_{\mathbf{P}}$ `3COLOR`.
6. Peut-on alors en déduire que `2COLOR` est **NP**-complet ? La réponse devra être démontrée.
7. Si la réponse à la question 6 est vraie, peut-on en déduire quelque chose ?
8. Montrer que `3COLOR` $\leq_{\mathbf{P}}$ `6COLOR`.
9. Peut-on alors en déduire que `6COLOR` est **NP**-complet ? La réponse devra être démontrée.
10. Donner une machine de Turing déterministe qui décide `6COLOR`.
11. Donner la complexité temporelle de cette machine de Turing.
12. Si la réponse à la question 9 est vraie, peut-on en déduire quelque chose ?