Master Sciences-Technologies-Santé / Mention Informatique
Info0809 : Informatique théorique / Pascal Mignot
Année universitaire 2014-2015
Première session d'avril 2015 / 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.

Questions indépendantes

1. Donner deux ADFs différents qui reconnaissent le même langage régulier L.
2. Soit $\texttt{DIFF}_{\texttt{ADF}} = \{\langle A_1, A_2\rangle \mid A_1$ et A_2 sont des ADFs qui reconnaissent des langages différents $\}$. Donner un machine de Turing non décidable qui reconnaît $\texttt{DIFF}_{\texttt{ADF}}$.
3. Expliquer la raison pour laquelle $\texttt{DIFF}_{\texttt{ADF}}$ est décidable (on montrera juste comment il possible de construire une telle machine de Turing).
4. Quel est le lien particulier entre les L-classes d'un langage régulier L et les ADFs qui reconnaissent ce langage ?
5. Donner une méthode permettant de convertir un ADF en automate à pile.
6. Si un langage libre de contexte est sous forme normale de Chomsky, est-il nécessairement non ambigu ?

Exercice 1 : machine de Turing

1. Soit L_{w_i} le langage qui reconnait un seul mot w_i (i.e. $L_{w_i} = \{w_i\}$). Montrer que $L_{w_i} \in \mathcal{RE}$
2. Montrer que $\mathcal{RE}$ est fermé par union.
3. En remarquant qu'un langage L non fini peut s'écrire $L = \cup_i L_{w_i}$, est-il possible de créer un machine de Turing M qui reconnait L à partir des machines de Turing $\{L_{w_i}\}$ qui reconnaissent $\{L_{w_i}\}$? Si oui, on décrira comment cette machine est construite. Si non, on expliquera pourquoi cette construction n'est pas possible.
4. Donner une raison théorique qui prouve qu'une telle machine existe (ou n'existe pas).
5. Conclure : lorsque $\#L$ n'est pas fini, L est-il toujours récursivement énumérable ?

Exercice 2 : machine de Turing non déterministe

Soit $\texttt{SAT}_k = \{\langle\Phi\rangle \mid \Phi$ est une expression booléenne avec k assignations de littéraux vraies $\}$.

1. Montrer que $\texttt{SAT}_k \in \mathbf{NP}$ en utilisant un certificat.
2. Montrer que $\texttt{SAT}_k \in \mathbf{NP}$ en utilisant une machine de Turing non déterministe.
3. En reprenant la MTND proposée à la question 2, reconnait-on un autre langage en permutant la décision de la machine (*i.e.* accepter→ rejeter et rejeter → accepter) ?
4. Montrer que $\texttt{SAT}_{k-1} \leq_{\mathbf{P}} \texttt{SAT}_k$.
5. Peut-on alors en déduire que $\texttt{SAT}_k$ est $\mathbf{NP}$-complet ?
6. Si on montre que $\overline{\texttt{SAT}_k} \in \mathbf{NP}$, démontrer qu'alors $\mathbf{NP} = co\mathbf{NP}$.

Exercice 3 : LIPOGRAM

1. $\texttt{LIPOGRAM}_a = \{\langle M, w\rangle \mid M \in \mathcal{RE} \text{ et } a \notin w\}$ est-il récursivement énumérable ?
2. $\texttt{LIPOGRAM}_a$ est-il décidable ? S'il est indécidable, on effectuera la démonstration par réduction d'un problème connu comme étant indécidable à ce problème.
3. $\texttt{LIPOGRAM} = \{\langle M, w\rangle \mid \langle M, w\rangle \in \texttt{LIPOGRAM}_a \text{ et } M(w) \text{ accepte}\}$ est-il récursivement énumérable ?
4. LIPOGRAM est-il décidable ? S'il est indécidable, on effectuera la démonstration par réduction d'un problème connu comme étant indécidable à ce problème.

Exercice 4 : complexité temporelle

On définit $\texttt{3COLOR} = \{\langle G\rangle \mid$ 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 $\texttt{BIPARTITE} = \{\langle G\rangle \mid G \text{ est un graphe biparti non orienté}\}$. 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 $\texttt{BIPARTITE} \in \mathbf{P}$.
2. Soit $\texttt{2COLOR} = \{\langle G\rangle \mid$ les sommets de G peuvent être coloriés avec 2 couleurs de telle façon que toute arête relie des sommets de couleurs différentes $\}$.
   Montrer que $\texttt{2COLOR} \leq_{\mathbf{P}} \texttt{BIPARTITE}$.
3. Montrer que $\texttt{BIPARTITE} \leq_{\mathbf{P}} \texttt{2COLOR}$.
4. Quel sont les propriétés que l'on peut déduire des trois questions précédentes ? On justifiera.
5. Montrer que $\texttt{2COLOR} \leq_{\mathbf{P}} \texttt{3COLOR}$.
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 $\texttt{3COLOR} \leq_{\mathbf{P}} \texttt{6COLOR}$.
9. Peut-on alors en déduire que 6COLOR est **NP**-complet ? La réponse devra être démontrée.
10. Donner une machine de Turing <u>déterministe</u> 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 ?

Exercice 5 : complexité spatiale

On définit $\texttt{PATH} = \{\langle G, s, t\rangle \mid G$ est un graphe orienté tel qu'il existe un chemin (orienté) entre s et $t\}$. Les résultats sur PATH dans l'aide-mémoire ne devront pas être utilisés dans cet exercice.

1. Montrer que $\texttt{PATH} \in \texttt{SPACE}(n \log n)$.
2. Montrer que $\texttt{PATH} \in \texttt{NSPACE}(\log n)$.
3. Montrer que PATH est **NL**-difficile, à savoir que pour tout $A \in \mathbf{NL}$, il existe une fonction f calculable en espace logarithmique qui réduit A à PATH. On se contentera de donner les grandes lignes de fonctionnement de f.
4. Montrer que PATH est **NL**-complet.
5. Montrer que $\forall A, B \in \texttt{PATH}$, $A \cup B \in \mathbf{NL}$.
6. Peut-on déduire des questions précédente que **NL** est fermé par union ? On justifiera.