Examen terminal - Théorie de la complexité

THX

11 décembre 2018

> La précision et la clarté de la rédaction est prise en compte dans l'évaluation.
> Écrivez assez grand. Rédigez soigneusement. Les exercices sont indépendants.

Exercice 1 *Dire pour chaque problème de la liste s'il est dans L, NL-complet, P-complet, NP-complet, coNP-complet, Σ_2-complet, Π_2-complet, PSPACE-complet, EXPTIME-complet, NEXPTIME-complet, ou indécidable :*

	Vos réponses
3SAT	
TQBF	
2SAT	
Géographie	
HORNSAT	
Echecs généralisées	
Décider si le langage d'un automate fini déterministe est vide	
Décider si le langage d'un automate fini non-déterministe est vide	
Décider si le langage d'un automate fini déterministe est Σ^*	
Décider si le langage d'un automate fini non-déterministe est Σ^*	
Décider si le langage d'une grammaire algébrique est vide	
Décider si le langage d'une grammaire algébrique est Σ^*	

Exercice 2 *Montrer que $P = NP$ implique $EXPTIME = NEXPTIME$.*

Indice : Pour un langage L, on pourra considérer le langage $\mathsf{pad}(L) = \{w\#^{f(|w|)} \mid w \in L\}$ pour une fonction f bien choisie.

Exercice 3 *On considère des expressions, similaires aux expressions rationnelles, mais qui définissent des ensembles d'entiers naturels, plutôt que des ensembles de mots, engendrées par la grammaire :*

$$e \quad ::= \quad \overline{n}^{2} \quad | \quad (e + e) \quad | \quad (e \cup e)$$

où $\overline{n}^{2}$ est l'écriture en binaire d'un entier naturel n. La sémantique est définie par induction sur e :

- $S(\overline{n}^{2}) = \{n\}$ *;*
- $S(e_1 + e_2) = \{n_1 + n_2 \mid n_1 \in S(e_1)$ *et* $n_2 \in S(e_2)\}$ *;*
- $S(e_1 \cup e_2) = S(e_1) \cup S(e_2)$.

On s'intéresse aux deux problèmes de décision suivants :

N-DEDANS :
- *Entrée : une expression e, l'écriture en binaire $\overline{n}^{2}$ d'un entier naturel n ;*
- *Sortie : oui si $n \in S(e)$; non, sinon.*

N-INEQ :
- *Entrée : deux expressions e_1, e_2 ;*
- *Sortie : oui si $S(e_1) \neq S(e_2)$; non, sinon.*

1. *Définir par induction sur e, la taille de $|e|$.*
2. *Montrer que si $n \in S(e)$ alors $|\overline{n}^{2}| \leq |e|$.*
3. *Montrer que N-DEDANS est dans NP.*
4. *Montrer que N-INEQ est dans Σ_2^P.*
5. *Expliquer une démarche pour montrer que N-INEQ est Σ_2^P-dur (démonstration complète non exigée).*

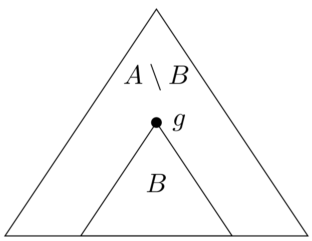
Exercice 4 *On rappelle qu'une formule est un circuit booléen qui est un arbre (binaire). L'objectif de l'exercice est de démontrer qu'un langage est décidé par une famille de formules de taille polynomiale ssi il est dans NC^1 (non-uniforme).*

1. *Donner un exemple de circuit à 3 entrées qui est une formule, et un autre qui n'est pas une formule.*
2. *Expliquer ce que l'adjectif 'non-uniforme' signifie.*
3. *Montrer que tout langage de NC^1 est décidé par une famille de formules de taille polynomiale.*

   Indice : on pourra montrer par induction que toute porte de profondeur d est équivalente à une formule de taille au plus $2^{d+1} - 1$.
4. *Donner une famille de formules $(C_n)_{n \in \mathbb{N}}$ de taille $O(n)$ et de profondeur $\Omega(n)$.*
5. *Nous allons montrer que toute formule A de taille t est équivalente à une formule φ_A de taille $t^{O(1)}$ et de profondeur $O(\log t)$.*

   (a) *Montrer que, toute formule A, vue comme un arbre binaire de taille t, admet un sommet g tel que le sous-arbre B de racine g contient entre $\frac{t}{3}$ et $\frac{2t}{3}$ sommets.*

   Indice : on pourra exhiber un algorithme qui cherche g.



   (b) *Raisonner par récurrence sur les formules C_0, C_1 et B, où C_0 (resp. C_1) sont obtenues à partir de $A \setminus B$ en remplaçant g par $\perp$ (resp. $\top$).*

6. *En conclure que tout langage décidé par une famille de formules de taille polynomiale est dans NC^1 (non-uniforme).*