Université Côte d'Azur
Polytech'Nice-Sophia
SI4
2021–2022

Examen Complexité & algorithmique avancée
du 6 décembre 2021

Nom : NIGET
Prénom : Tom
Groupe : SI4

Note : 15,5

Durée : 2 heures

1	2	3	4	3	4
4	4	6	6	6	6

L'épreuve est composée de questions indépendantes. Veuillez répondre sur la copie avec clarté et concision (le nombre de lignes laissé pour les réponses est souvent excessif). Les réponses *non raisonnables* pourront être pénalisées.
Dans la deuxième partie vous devez choisir deux questions parmi les 4 proposées.

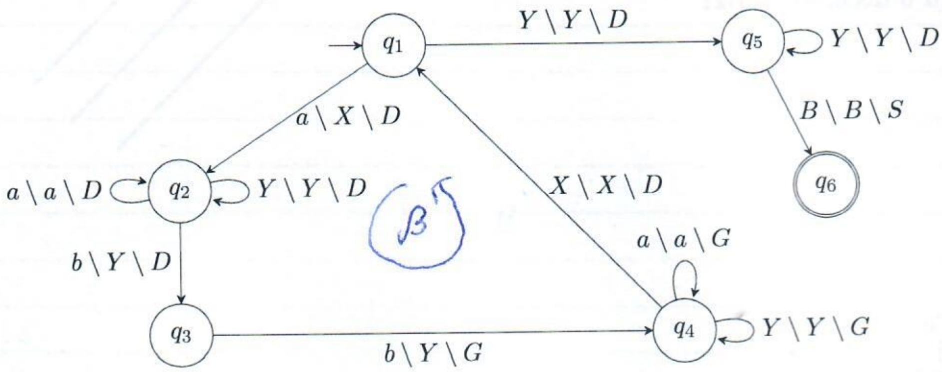
Si une question vous semble comporter des erreurs ou imprécisions, ne posez pas de question oralement, mais signalez-le ci-dessous en précisant :

- le numéro de la question concernée
- vos interrogations sur cette question
- éventuellement l'interprétation ou les choix faits pour votre (vos) réponse(s) à cette question.

Première partie - Machines de Turing

1 Machine de Turing [4 points]

Vous avez la machine de Turing suivante (alphabet des données $\{a, b\}$ et blanc B) :



a) Quel est le langage reconnu par cette machine?

Le langage des mots de la forme $a^n b^m$ avec $m = 2n$, et $n \geq 1$.

b) Quelle est sa complexité? Expliquez votre réponse.

Pour un mot $a^n b^m$: à chaque itération i de la "boucle" β, on parcourt $(n-i)$ "a", $(2i)$ "Y", 2 "b", puis on revient avec $(n-i+2i+1)=(n+i+1)$ étapes. Autrement dit, l'itération i prend $(2n+2i+3)$ étapes. On peut développer via Gauss mais dans tous les cas, on a n itérations de $O(n)$ étapes, d'où une complexité totale de $O(n^2)$

(nb total d'étapes : $\frac{5}{2}n^2 + \frac{9}{2}n$)

c) Décrivez (sans le dessiner, mais en expliquant en texte) une machine plus efficace, pour accepter le même langage et donnez sa complexité.

On lit un "a", on va à droite, on efface deux "b", si blanc on valide. Sinon on efface encore deux "b", on va à gauche et on recommence. Ici mouvement typique en aller-retours, toujours en $O(n^2)$, mais qui fait moins d'étapes que l'énoncé

2 Machine de Turing non-déterministe [4 points]

Décrivez de manière informelle mais claire les machines de Turing non déterministes - à plusieurs bandes si vous voulez - qui acceptent les langages suivants. Essayez de tirer parti du non-déterminisme pour gagner du temps au sens non déterministe. Donnez la complexité de chaque machine proposée.

a) Soit L_a le langage sur l'alphabet binaire $\{0, 1\}$ des mots tels qu'ils contiennent une répétition disjointe d'un facteur de longueur 100, c'est à dire des mots qui peuvent être factorisés en $wxyxz$ avec x de longueur 100 et w, y et z quelconques.

?

b) Soit L_b le langage sur l'alphabet $\{0, 1, *\}$ des mots de la forme $w_1 * w_2 \ldots * w_n$ tels que chaque w_i est un mot binaire et il existe un j tel que w_j est j en binaire.

non dét)

On place bande 2 un compteur binaire qu'on incrémente à chaque étoile, et dont la tête est toujours au début. À chaque début de nombre, on fait 2 choses :
- on "compare" le nombre avec le compteur (acceptation si égal)
- on avance au nombre suivant

L'incrémentation est en $\log_2(n)$, le reste en n.
Complexité totale : $O(n)$.

Deuxième partie - Preuves [deux exercices à choisir]

Exercices choisis (seulement les deux marqués ici seront corrigés)

3	4	5	6
✗		✗	

3 Cliques [6 points]

Prouvez que les deux problèmes suivants sont polynomialement équivalents :

Nom : **CMI (clique maximum impaire)**
Instance : Un graphe fini $G(V,E)$ sous forme de liste de voisins
Question : Est-ce que la clique maximum de G est de taille impaire?

Nom : **CMP (clique maximum paire)**
Instance : Un graphe fini $G(V,E)$ sous forme de liste de voisins
Question : Est-ce que la clique maximum de G est de taille paire?

Soit $G(V,E)$ un graphe tel que $CMI(G)$ est vrai.
On rajoute à G un sommet relié à tous les autres, soit :
$E' = E \cup \{x\}$
$V' = V \cup \{\forall k \in V ; (k,x)\}$
$G' = (V', E')$
La taille de la clique maximum augmente de 1 ; sa parité s'inverse.
$\Rightarrow CMP(G')$ est vrai. On peut faire le chemin inverse pour obtenir $CMP(G') \Rightarrow CMI(G)$.
Même raisonnement pour aller de CMP à CMI.
(d'où $CMI(G) \Leftrightarrow CMP(G')$)
Le traitement de G, qu'on pourrait noter $G' = f(G)$, prend un temps polynomial, ergo : $CMI \leq_p CMP \wedge CMP \leq_p CMI$
$\Leftrightarrow CMI =_p CMP$. ∎

l'idée OK
transformation OK !
preuve.....

5 Pas tous égaux [6 points]

Montrer la NP-difficulté du problème suivant :

Nom : **k Satisfiabilité "pas tous égaux" (k-SAT-PTE)**
Instance : Φ une formule de Xk-SAT, $k \geq 4$.
Question : Existe-t'il une assignation de vérité aux variables qui satisfait Φ, telle que chaque clause contient au moins un littéral vrais et un littéral faux.

Indication : introduire une nouvelle variable z et l'insérer dans toutes les clauses.

Soit F une formule de Xk-SAT.
F est de la forme $C_1 \wedge C_2 \wedge \cdots$ où C_n est un OU de k termes.

Xk-SAT(F) = vrai est équivalent à dire que chaque clause contient au moins 1 littéral vrai.

Soit $\phi = \bigwedge_{i=1}^{k} \underbrace{(C_n \vee z)}_{C_n'}$. Si z = FAUX, alors $\phi \Leftrightarrow F$.

On a C_n qui contient ≥ 1 littéral vrai. Avec z = FAUX, C_n' contient ≥ 1 littéral faux, et donc (k_{+1})-SAT-PTE(ϕ) = VRAI.

D'où Xk-SAT(F) $\Leftrightarrow$ (k_{+1})-SAT-PTE(ϕ)

Et donc Xk-SAT $\leq_p$ k-SAT-PTE,

Or Xk-SAT est NP-complet, donc k-SAT-PTE est NP-difficile. ■