Université de Lille
FST- Département Informatique

ACT

2018–2019
Master d'Informatique-S1

Fiche TD : La classe NP

Exercice 1 : Emploi du temps

Soit le problème de décision:

Donnée: k un nombre d'examens, n un nombre d'étudiants, pour chaque étudiant i, la liste L_i (non vide) des examens qu'il doit passer, $j \leq k$ un nombre de slots.

On supposera que chaque examen est passé par au moins un étudiant.

Sortie: Oui, si on peut planifier tous les examens en j slots de façon à ce que bien sûr chaque étudiant passe au plus un examen par slot.

Montrer que la propriété est NP.

Exercice 2 : Solitaire

Un jeu de solitaire se joue sur un plateau $m \times m$. Au départ, à chaque position, il y a un pion rouge ou bleu ou aucun pion, et chaque ligne contient au moins un jeton. L'objectif est d'enlever des pions pour que chaque colonne ne contienne des jetons que d'une seule couleur et que chaque ligne contienne au moins un jeton.

On supposera qu'une configuration est représentée par un tableau $m \times m$ à valeurs dans $\{Rouge, Bleu, Vide\}$.

Q 1. Pour $m = 3$, exhiber plusieurs configurations gagnantes et plusieurs configurations perdantes.

Q 2. Montrer que déterminer si une configuration est gagnante est NP.

Exercice 3 : Le Sudoku

Soit le Sudoku généralisé: une grille est un carré de côté k^2 pour un certain k, qu'on peut diviser en k^2 carrés de k^2 cases, le but étant de remplir avec les entiers de 1 à k^2 avec les règles suivantes: un entier doit apparaître une et une seule fois dans chaque ligne, dans chaque colonne, dans chaque carré.

Q 1. Soit le problème de décision:

Donnée: un entier k et une grille $k^2 * k^2$ partiellement remplie par des entiers de 1 à k^2.

Sortie: Oui, Ssi la grille peut être complétée en respectant les règles.

Montrer qu'il est NP.[1]

Q 2. Que pensez-vous de la complexité de ce problème de décision:

Donnée: un entier k et une grille $k^2 * k^2$ partiellement remplie par des entiers de 1 à k^2.

Sortie: Oui, Ssi il existe **au plus** une complétion correcte de la grille.

Exercice 4 : Processus

Soit un système temps réel avec n processus asynchrones et m ressources. Quand un processus est actif, il bloque un certain nombre de ressources et une ressource ne peut être utilisée que par un seul processus. On cherche à activer simultanément k processus. Le problème de décision DecProc est donc:

Donnée: n, le nombre de processus
m, le nombre de ressources
pour chaque processus i, la liste P_i des ressources qu'il bloque.
k le nombre de processus que l'on souhaite activer

Sortie: Oui, si on peut activer k processus simultanément, non sinon.

Par exemple si $n = 4$, $m = 5$, et $P_1 = \{1, 2\}, P_2 = \{1, 3\}, P_3 = \{2, 4, 5\}, P_4 = \{1, 2, 4\}$ on peut activer simultanément les processus 2 et 3 et donc la réponse est *Oui* pour $k = 2$ mais la réponse est *Non* pour $k = 3$.

Q 1. Montrer que le problème DecProc est NP.

[1]Le problème est $NP-$dur, mais dans le cas $k = 3$, il existe bien sûr des algorithmes dont l'exécution est très rapide.

Exercice 5 : Temps versus espace

Q 1. Justifier que P ($PTime$) est inclus dans $PSpace$, la classe des propriétés pour lesquels il existe un algorithme de décision polynomial en espace.
Q 2. $Exptime$ est la classe des propriétés pour lesquels il existe un algorithme de décision exponentiel (en temps). Justifier que $PSpace$ est inclus dans $ExpTime$.
Q 3. On a donc $P \subset PSpace \subset EXPTime$. Comment se positionne NP?

Exercice 6 : Autour de SAT

Quelle est l'erreur dans le raisonnement suivant: *Toute formule booléenne peut être mise sous forme disjonctive; or tester la satisfiabilité d'une formule booléenne sous forme disjonctive est polynomial. Donc tester la satisfiabilité d'une formule booléenne est polynomial.*

Exercice 7 : Noir et Blanc

On a vu en cours que le 3-coloriage de graphes est NP. Que pensez-vous du 2-coloriage de graphes?

Exercice 8 : Test de Primalité

Soit l'algorithme:

```
//n est un entier naturel >=2
Booléen Composé(Entier n)
Pour i de 2 à racine-carree(n)
     si i divise n alors retourne Vrai;
retourne Faux
```

Q 1. Cet algorithme est-il correct, i.e. retourne-t-il Vrai Ssi n est composé? Est-il polynomial?
Q 2. Montrer que la propriété " être composé" est NP.
Q 3. Le test de primalité d'un entier a été montré P en 2002 après de longues années de recherche. Qu'en déduire pour la propriété "être composé"?

Pour les curieux: Jusque 2002, certains tests utilisés étaient supposés être polynomiaux mais cela n'avait pas été prouvé. Le test proposé en 2002 est le test de primalité d'Agarwal-Kayal-Saxena -"AKS primality test"-, la complexité étant en $O((\log n)^{12+\epsilon})$ (n est l'entier dont on teste la primalité). Des améliorations ont été proposées pour diminuer l'exposant. Avant 2002, on savait déjà que "être premier" était NP, donc était dans $co - NP \cap NP$. La notion de certificat,un peu plus difficile à trouver que pour "être composé", est basée sur le fait que n est premier Ssi il existe a tel $a^{n-1} \equiv 1 (mod\ n)$ et $a^m \not\equiv 1 (mod\ n)$ pour tout m, $1 \leq m < n$.

Exercice 9 : Génération d'objets combinatoires

Pour appliquer la méthode du British Museum, on peut être amené à générer tous les certificats pour une instance de problème. Si on revient au sens "strict" de la définition, cela revient à générer tous les mots d'une longueur donnée (ou inférieure à une longueur donnée).
Q 1. Combien existe-t-il de mots de longueur n sur un alphabet Σ de cardinal k? de longueur au plus n?
Q 2. Proposer une méthode pour générer tous les mots sur $\{a, b\}$ de longueur égale à une valeur donnée.
Q 3. Proposer une méthode pour générer tous les mots sur $\{a, b\}$ de longueur inférieure ou égale à une valeur donnée.