Master DAPM - Session 1 - Complexité Algorithmique

Mardi 01/03/2016. 13h00-16h00. Salle Amphi 400

La précision et la clarté de votre rédaction sont *fondamentales*. Chaque réponse doit être accompagnée d'une justification. Le barème est indicatif. Documents interdits. Durée 3 heures.

Exercice 1. [2 pts] Soient E et F deux ensembles et $f : E \to F$ une application. Sans faire de preuves, que peut-on déduire :

- sur E si f est injective et F est fini de cardinal n ?
- sur E si f est injective et F est dénombrable ?
- sur F si f est injective et E est dénombrable ?
- sur E si f est surjective et F est dénombrable ?
- sur E si f est surjective et E est fini de cardinal n ?
- sur F si f est surjective et E est dénombrable ?

Exercice 2. [2 pts] Démontrez que pour tout entier naturel k, l'ensemble $\{kn \mid n \in \mathbf{N}\}$ des multiples de k est au plus dénombrable.

Exercice 3. [4 pts] Soit $k \in \mathbf{N}$ et $c_k : \mathbf{N} \to \mathbf{N}$ la *fonction constante* égale à k, définie par

$$\forall n \in \mathbf{N} \quad c_k(n) = k.$$

Dans la suite l'alphabet de la machine de Turing est l'alphabet unaire $\Sigma = \{\,!\,\}$. On rappelle qu'un entier n est codé avec $n+1$ *bâtons* et que la tête de lecture écriture est placée par convention sur le bâton le plus à gauche au début et à la fin de l'exécution d'un programme qui calcule une fonction de $\mathbf{N}$ dans $\mathbf{N}$.

1. [2 pts] Démontrez que la fonction c_k est Turing-calculable.

2. [1 pt] Quelle est la fonction de complexité de votre programme ?

3. [1 pt] Déduire de la première question que l'ensemble des machines de Turing est infini.

Exercice 4. [5 pts] Soit $f : \mathbf{N} \to \mathbf{N}$ une fonction Turing-calculable avec l'alphabet unaire $\Sigma := \{!\}$. On suppose que l'algorithme qui la calcule n'utilise aucune des cellules à gauche des $n+1$ bâtons qui représentent l'entier n. On note k le nombre de ses états et on suppose que l'algorithme s'arrête dans l'état q_{k-1}. Par convention, la tête de lecture/écriture est placée sur le bâton le plus à gauche au début et la fin de l'exécution.

1. [1 pt] Démontrez que la fonction de transition δ n'est pas définie en $(q_{k-1}, !)$.

2. [4 pts] Démontrez que la fonction $g : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ définie par $g(r,n) := f^r(n)$ est Turing-calculable où f^r est la composée de r fois $f : f \circ f \circ \cdots \circ f$. Expliquez votre algorithme en détail et explicitez les blocs d'instructions de votre algorithme.

Exercice 5. [10 pts] On considère le problème du k-coloriage suivant :

> **Problème k-COL** [k-COLORIAGE]
> *Instance :* Un graphe non-orienté $G = (X, U)$.
> *Question :* Peut-on colorier les sommets de X sans que deux sommets adjacents aient la même couleur avec au plus k couleurs ?

On rappelle que le problème 3-SAT est NP-complet :

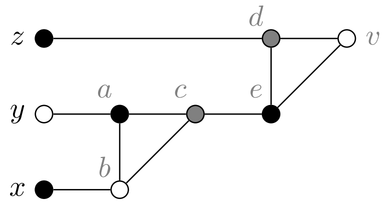
> **Problème 3-SAT** [SATISFAISABILITÉ À 3 LITTÉRAUX]
> *Instance :* Un ensemble de clauses $C = \{C_1, C_2, \ldots, C_m\}$ à trois littéraux avec n variables booléennes $\{u_1, u_2, \ldots, u_n\}$.
> *Question :* Existe-t-il une table de vérité pour les variables u_i qui permette de satisfaire l'ensemble des clauses C_i de C ?

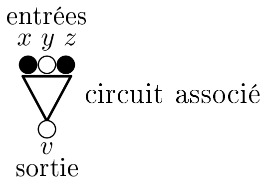
1. [1 pt] Montrez que le problème 3-COL est dans la classe NP.

2. [2 pts] Si l'on suppose qu'une instance de 3-SAT ne contient que des clauses distinctes et que chaque clause contient des variables distinctes, et par conséquent des littéraux distincts, calculez le nombre maximum de clauses possibles dans une instance du problème 3-SAT.

Tournez la page SVP ↘

À gauche ci-dessous, on montre un 3-coloriage d'un graphe particulier à 9 sommets avec les trois couleurs BLANC, GRIS et NOIR. Ce graphe sera synthétisé sous la forme d'un "circuit" $(x, y, z) \mapsto v$ où x, y et z sont les *entrées* et où v est la *sortie* (cf. schéma à droite) :





3. [2 pts] Supposons que l'on ait un 3-coloriage du circuit. Montrez que si les trois sommets/entrées x, y et z sont NOIRS, alors la sortie v est nécessairement NOIRE. Montrez que pour les 7 autres combinaisons possibles des entrées x, y et z avec les couleurs NOIR ou BLANC, il existe un 3-coloriage avec une sortie v BLANCHE.

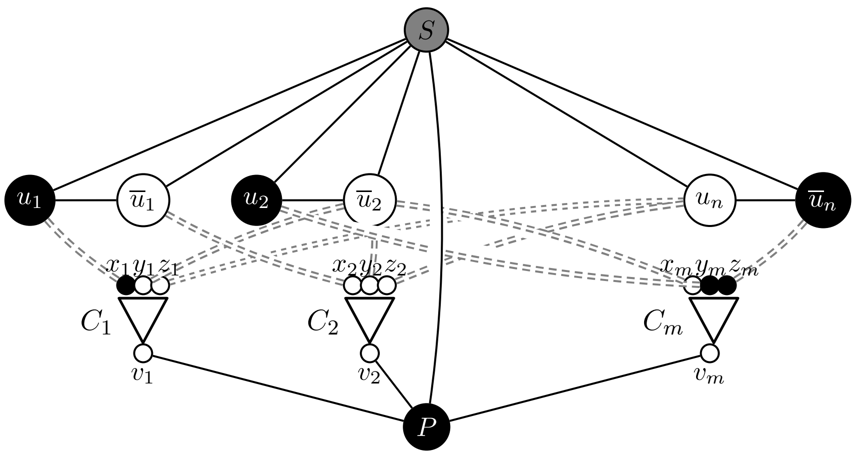
À partir d'une instance C du problème 3-SAT, on construit un graphe de la manière suivante (de haut en bas) :

(1) On définit un sommet *source* S que l'on connecte à chacun des $2n$ sommets/littéraux $u_1, \overline{u}_1, \ldots, u_n, \overline{u}_n$.

(2) On connecte chaque littéral u avec son opposé $\bar{u}$.

(3) On connecte les trois littéraux x_i, y_i, z_i de chaque clause $C_i = \{x_i, y_i, z_i\}$ aux 6 sommets $a_i, b_i, c_i, d_i, e_i, v_i$ d'un circuit $(x_i, y_i, z_i) \mapsto v_i$.

(4) On termine avec un arc entre chaque sortie de circuit v_i et un sommet *puit* P et un dernier arc reliant le puit P à la source S.



Attention ! Dans le schéma, pour faciliter la compréhension de la construction du graphe, les "entrées" (x_i, y_i, z_i) des circuits sont distinguées des littéraux $u_i, \overline{u}_i$ alors qu'ils sont *confondus*. Ces égalités sont matérialisées en traitillés, les vrais arcs du graphe sont en traits pleins. Dans l'exemple,

$$C_1 = \{\underset{x_1}{u_1}, \underset{y_1}{\overline{u}_2}, \underset{z_1}{u_n}\} \qquad C_2 = \{\underset{x_2}{\overline{u}_1}, \underset{y_2}{\overline{u}_2}, \underset{z_2}{u_n}\} \qquad C_m = \{\underset{x_3}{\overline{u}_2}, \underset{y_3}{u_2}, \underset{z_3}{\overline{u}_n}\}$$

4. [2 pts] Démontrez que si l'instance de 3-SAT est satisfaisable, alors le graphe est 3-coloriable. Indication : interprétez les couleurs BLANC et NOIR respectivement comme VRAI et FAUX.

5. [2 pts] Réciproquement, démontrez que si le graphe est 3-coloriable alors l'instance de 3-SAT est satisfaisable. Indication : vous supposerez que P est noir et S est gris (n'importe quelle permutation des trois couleurs d'un 3-coloriage définit un autre 3-coloriage).

6. [1 pt] Dénombrez le nombre de sommets s et le nombre d'arcs a de ce graphe en fonction de n et m et prouvez que cette transformation est polynomiale en n et m. Concluez