Master DAPM - Session 1 - Complexité Algorithmique

Vendredi 21/02/2014. 9h00-12h00. Salle T' 302

La précision et la clarté de votre rédaction sont *fondamentales*. Chaque réponse doit être accompagnée d'une justification. Lisez attentivement l'énoncé avant toute tentative de raisonnement. Le barème est donné à titre indicatif. Documents interdits. Durée 3 heures.

Exercice 1. [2 pts] Démontrez que l'ensemble des entiers pairs est dénombrable.

Exercice 2. [4 pts] On rappelle la fonction $J : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ du cours utilisée pour démontrer que l'ensemble $\mathbf{N} \times \mathbf{N}$ est dénombrable :

$$J(m,n) := \frac{(m+n)(m+n+1)}{2} + n.$$

On se propose de démontrer que J est surjective. Pour tout entier k, on définit la somme $S_k := \sum_{i=0}^{k} i$ et l'intervalle semi-ouvert $I_k := [S_k, S_{k+1}[$ de $\mathbf{N}$.

(1) Calculez la somme S_k.

(2) Montrez que la famille (I_k), $k \in \mathbf{N}$ est une partition[1] de $\mathbf{N}$.

(3) En déduire que pour tout $y \in \mathbf{N}$, il existe un unique entier k tel que $y \in I_k$.

(4) Déterminez alors à partir de k la valeur du couple (m,n) tel que $J(m,n) = y$.

Exercice 3. [6pts] On se propose de montrez que la fonction $f : \mathbf{N} \to \mathbf{N} \times \mathbf{N}$ définie par $f(n) := (n,n)$ est Turing-calculable. On note $\boxtimes$ une case contenant un bâton, et $\square$ (ou $\blacksquare$ par commodité) une case vide de la bande et on utilise la notation exponentielle $\boxtimes^n$ pour désigner une séquence de n bâtons par exemple. On note donc $u = \boxtimes^{n+1}$ la séquence initiale sur la bande.

L'algorithme consiste dans un premier temps à initialiser une séquence $v := \boxtimes$ à droite de la séquence u en séparant u et v d'une case vide. Dans un deuxième temps, on efface successivement chacun des n derniers bâtons de la séquence u en partant de sa droite et en les rajoutant à droite de la séquence v. La dernière étape consiste à reconstituer les n bâtons de u précédemment effacés.

1. Si X et E sont deux ensembles, une famille (A_x), indexée par X, de sous-ensembles d'un ensemble E est une *partition* de E si et seulement si aucun des A_x n'est vide, les A_x sont deux-à-deux disjoints, et leur réunion est E tout entier.

(1) On se déplace à droite de la séquence initiale $u = \boxtimes^{n+1}$, on laisse une case vide et on écrit un bâton pour initialiser $v := \boxtimes$. à ce stade la bande contient :

$$\boxtimes^{n+1}\underline{\blacksquare}\boxtimes$$

(2) Tant que u contient au moins 2 bâtons, on efface le bâton le plus à droite et on écrit un bâton à droite de v. Par exemple, quand le premier bâton a été effacé dans u, la bande contient :

$$\boxtimes^n\square\blacksquare\boxtimes\boxtimes$$

(3) Quand il ne reste plus qu'un bâton à effacer dans la séquence u, c'est que les n bâtons auront été recopiés dans la séquence v. La bande contient donc :

$$\boxtimes\square^n\blacksquare\boxtimes^{n+1}$$

(4) Il reste alors à reconstituer u pour finir avec :

$$\boxtimes^{n+1}\blacksquare\boxtimes^{n+1}$$

Écrivez le programme correspondant en explicitant clairement le rôle de chaque bloc d'instructions. Quelle est la complexité de cet algorithme en fonction de n ?

Exercice 4. [3 pts] Rappelez la numérotation de Gödel d'une instruction de la machine de turing unaire. On considère la machine de Turing sur l'alphabet unaire $\Sigma := \{I\}$ définie par le programme suivant :

$$q_0, I, >, q_0$$
$$q_0, \square, I, q_1$$
$$q_1, I, <, q_1$$
$$q_1, \square, >, q_2$$

Quelle est la fonction calculée par cette machine ? Quel est son numéro de Gödel ?

Exercice 5. [8pts] On rappelle que $\propto$ est une relation binaire définie sur l'ensemble des problèmes de décision par $\Pi \propto \Pi'$ si et seulement s'il existe une machine de Turing polynomiale qui transforme les instances I de Π en instances I' de Π' et telle que $I \in Y_\Pi \Leftrightarrow I' \in Y_{\Pi'}$. On rappelle également qu'un problème de décision Π est dit NP-complet si et seulement s'il appartient à la classe NP et que $\forall \Pi' \in$ NP, $\Pi' \propto \Pi$.

TSVP

On considère les deux problèmes de décision suivants :

> **Problème de satisfaisabilité [sat]**
> *Instance :* Un ensemble de n variables booléennes $X = \{x_1, x_2, \ldots, x_n\}$ et un semble de m clauses $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$.
> *Question :* Existe-t-il une table de vérité $T : X \to \{\text{VRAI}, \text{FAUX}\}$ telle que $\mathcal{C}$ est satisfaisable ?

> **Problème de 3-satisfaisabilité [3-sat]**
> *Instance :* Un ensemble de n variables booléennes $X = \{x_1, x_2, \ldots, x_n\}$ et un semble de m clauses $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ à trois littéraux exactement.
> *Question :* Existe-t-il une table de vérité $T : X \to \{\text{VRAI}, \text{FAUX}\}$ telle que $\mathcal{C}$ est satisfaisable ?

Exemple 1 ▷ $X = \{x_1, x_2, x_3, x_4\}$ et $\mathcal{C} = \{\{x_1, \overline{x_2}\}, \{x_1, x_2, \overline{x_4}\}, \{\overline{x_1}, x_2, x_3, \overline{x_4}\}\}$ est une instance de SAT mais pas de 3-SAT car la dernière clause contient 4 littéraux. ◁

1. [1pt] Quelle est la réponse au problème SAT pour l'instance de l'exemple ?

On se propose de démontrer que 3-SAT est un problème NP-complet. Il faut transformer les instances I de SAT en instances J de 3-SAT de manière à ce que I soit satisfaisable si et seulement is J est satisfaisable. Pour cela, on remplace chaque clause C de SAT qui ne contient pas exactement 3 littéraux en un ensemble $f(C)$ de clauses à trois littéraux à l'aide de nouvelles variables booléennes.

2. [1pt] Montrez que 3-SAT appartient à NP.

3. [1pt] Considérons une clause à un seul littéral $C = \{u\}$. Donnons nous deux nouvelles variables v et w. Considérons l'ensemble $f(C)$ des 4 clauses $\{u, v, w\}$, $\{u, v, \overline{w}\}$, $\{u, \overline{v}, w\}$ et $\{u, \overline{v}, \overline{w}\}$. Montrez que C est satisfaisable si et seulement si l'ensemble des clauses de $f(C)$ est satisfaisable en vérifiant que l'expression logique

$$(u \vee v \vee w) \wedge (u \vee v \vee \overline{w}) \wedge (u \vee \overline{v} \vee w) \wedge (u \vee \overline{v} \vee \overline{w})$$

est vraie si et seulement si u est vrai. Indication : utilisez l'algèbre de Boole pour réduire l'expression.

4. [2pts] En vous inspirant de la question précédente trouvez comment transformer une clause à deux littéraux $C = \{u, v\}$ en un ensemble de clauses à trois littéraux $f(C)$ tel que C est satisfaisable si et seulement si l'ensemble des clauses de $f(C)$ est satisfaisable.

5. [2pts] Considérons une clause $C = \{u_1, u_2, \ldots, u_k\}$ à k littéraux, $k > 3$. On se donne $k-3$ variables $v_1, v_2, \ldots, v_{k-3}$ et l'ensemble $f(C)$ de $k-2$ clauses à trois littéraux suivant :

$$\{u_1, u_2, \boldsymbol{v}_1\}, \{\overline{\boldsymbol{v}}_1, u_3, \boldsymbol{v}_2\}, \{\overline{\boldsymbol{v}}_2, u_4, \boldsymbol{v}_3\}, \ldots, \{\overline{\boldsymbol{v}}_{k-4}, u_{k-2}, \boldsymbol{v}_{k-3}\}, \{\overline{\boldsymbol{v}}_{k-3}, u_{k-1}, u_k\}.$$

Montrez que C est satisfaisable si et seulement si les clauses de $f(C)$ sont satisfaisables.

6. [1pt] Montrez que la transformation f est polynomiale et concluez.