Master CSI 2 2010-2011

Théorie de la complexité : MHT 913

Examen du 14 décembre 2010

Master Sciences et Technologies, mention Mathématiques ou Informatique, spécialité Cryptologie et Sécurité informatique

Durée : 3 heures. Sans document.

Responsable : Gilles Zémor

Les exercices sont indépendants.

– EXERCICE 1. On rappelle qu'un graphe non orienté à n sommets est dit hamiltonien s'il existe un cycle (un chemin se terminant en son origine) de longueur n passant par tous les sommets du graphe. Un tel cycle est dit «hamiltonien».

Montrer que si P=NP, alors il existe un algorithme polynomial qui prend en entrée un graphe (non orienté) G à n sommets et qui
- répond «non hamiltonien» si G est non hamiltonien,
- *détermine* (exhibe) un cycle hamiltonien de G si G est hamiltonien.

On pourra exhiber un algorithme qui utilise comme sous-programme un algorithme auxiliaire $\mathcal{A}_{\text{aux}}$ qui *décide* si G est hamiltonien.

– EXERCICE 2. Soit f une formule booléenne sous la forme $f = C_1 \wedge \ldots \wedge C_k$ où dans chaque clause C_i n'interviennent que l'opérateur $\vee$. On dira qu'une clause $C_i = y_1 \vee y_2 \vee y_3$ est *$\neq$-satisfaite* par un choix des variables y_i si C_i est satisfaite mais l'on n'a pas simultanément $y_1 = 1$, $y_2 = 1$, $y_3 = 1$.

a) Montrer que si la formule booléenne à n variables f est $\neq$-satisfaite par le vecteur v de $\{0,1\}^n$, alors la formule f est $\neq$-satisfaite également par le vecteur $\overline{v} = (1, 1 \ldots, 1) + v$.

b) Soit $\neq$SAT le problème :
   I : Une formule booléenne $f = C_1 \wedge \ldots \wedge C_k$
   Q : la formule f est-elle $\neq$-satisfaisable ?
   Montrer que $\neq$SAT est dans NP. Existe-t-il une réduction polynomiale de $\neq$-SAT vers SAT ?

c) Montrer que l'on obtient une réduction polynômiale de 3-SAT vers $\neq$-SAT en remplaçant chaque clause $C_i = y_1 \vee y_2 \vee y_3$ par la sous-formule

$$(y_1 \vee y_2 \vee z_i) \wedge (\overline{z}_i \vee y_3 \vee b)$$

où z_i est une variable auxiliaire associée à la clause C_i, et où b est une variable supplémentaire unique.

d) En déduire que le problème ≠SAT est NP-complet.

– EXERCICE 3. Soit $L \subset \Sigma^*$ un langage de la classe IP. Il existe donc un protocole interactif entre un prouveur P et un vérificateur V qui décide si une instance $x \in \Sigma^*$ appartient au langage L. Montrer que si le protocole est déterministe pour V, c'est-à-dire si V ne fait appel à aucun bit aléatoire pendant le déroulement du protocole, alors L est un langage de la classe NP.

– EXERCICE 4. Soient donnés p un grand nombre premier, q un diviseur de $p-1$, et g un élément d'ordre q dans $(\mathbb{Z}/p\mathbb{Z})^*$. Soit $P = g^s \bmod p$ une quantité publique. On considère le protocole suivant, destiné à démontrer la connaissance de s.
- Le prouveur P choisit un entier aléatoire r modulo q, puis calcule $x_1 = g^r \bmod p$ et $x_2 = g^{s-r} \bmod p$. Il communique x_1 et x_2 au vérificateur V.
- V choisit au hasard $i = 1$ ou $i = 2$ et demande à P un entier y_i tel que $g^{y_i} = x_i \bmod p$.
- P s'exécute.
- V vérifie que $x_1 x_2 = P$ et que $g^{y_i} = x_i \bmod p$.

Expliquer pour quoi ce protocole démontre la connaissance de s par P, et montrer qu'il est sans divulgation.

– EXERCICE 5. On souhaite réaliser un protocole sans divulgation de ce que l'entier y n'est *pas* un carré modulo l'entier n.

a) On considère un premier protocole :
   - Le vérificateur V choisit un entier modulo n aléatoire r ainsi qu'un bit $\varepsilon \in \{0,1\}$. Il calcule $x = r^2 y^\varepsilon$ et le communique au prouveur P.
   - Le prouveur révèle un bit $b \in \{0,1\}$. Si $b = \varepsilon$ le vérificateur accepte, sinon il rejette.

   Montrer que ce protocole est bien complet et valide, mais qu'il n'est pas sans divulgation.

b) Soit f une fonction à sens unique sur les entiers modulo n, par exemple $f : z \mapsto g^z \bmod n$ pour un certain g. On fait l'hypothèse cryptographique que la donnée de $f(z)$ ne révèle aucun bit d'information sur z à un vérificateur ne disposant que d'une capacité de calcul en temps polynomial (sauf peut-être sur une infime proportion d'entiers z). On considère maintenant le protocole suivant :
   - Le vérificateur V choisit un entier modulo n aléatoire r ainsi qu'un bit $\varepsilon \in \{0,1\}$. Il calcule $x = r^2 y^\varepsilon$ et le communique au prouveur P.
   - Le prouveur choisit un bit $b \in \{0,1\}$. Puis il choisit aléatoirement un entier $z \in \{0,1,\ldots,n-1\}$ où z est pair si $b = 0$ et impair si $b = 1$. Il calcule $e = f(z)$ et le donne à V.
   - Le vérificateur V révèle à P les quantités r et ε.

- Le prouveur P vérifie que $x = r^2 y^\varepsilon$: si ce n'est pas le cas il arrête le protocole. Si c'est le cas il révèle z à V.
- Le vérificateur calcule $f(z)$ et vérifie que cette quantité est bien égale à e. Il vérifie également que z est pair si $\varepsilon = 0$ et impair si $\varepsilon = 1$. Si l'une de ces vérifications échoue il rejette la réponse, sinon il accepte.

Montrer que ce protocole est complet et valide, et qu'il est sans divulgation au sens calculatoire.

– EXERCICE 6. Alice souhaite *s'engager* sur un bit $b \in \{0,1\}$ auprès de Bob. Ceci veut dire qu'elle donne à Bob une quantité $E(b)$ qui ne donne pas d'information à Bob sur la valeur de b, et qu'il existe un algorithme de vérification pour Bob qui lui permet, lorsque Alice révèle la valeur de b, de vérifier que le bit révélé par Alice est bien le même que celui sur lequel elle s'est engagée.

On considère le protocole d'engagement suivant. Alice et Bob conviennent d'un nombre premier p et d'un entier g primitif modulo p.
- Bob donne à Alice un entier y de $(\mathbb{Z}/p\mathbb{Z})^*$.
- Alice choisit le bit b sur lequel elle souhaite s'engager, puis donne à Bob la quantité $E(b,r) = y^b g^r \bmod p$ où r est un entier aléatoire de $[0, 1, \ldots, p-2]$. L'engagement est donc pour Bob un entier z modulo p.
- Le moment venu, Alice révèle b et r.

a) Montrer que Alice est vraiment engagée, c'est-à-dire que après qu'elle a donné une quantité z à Bob, elle n'est capable de révéler qu'une valeur du bit, 0, ou 1, sinon cela veut dire qu'elle est capable de calculer un logarithme modulo p en base g de y. Montrer également qu'*avant* que Alice ne révèle la valeur de son bit, la quantité z ne donne strictement aucune information sur la valeur b du bit surlequel Alice s'est engagée.

b) On suppose maintenant que Alice s'est engagée sur deux bits, b et c, par l'intermédiaire des deux engagements $z = E(b,r)$ et $t = E(c,r')$. On suppose de plus qu'Alice ne soit pas encore prête à révéler b et c, mais qu'elle souhaite convaincre Bob que ses deux bits b et c sont différents. En d'autres termes, elle souhaite prouver à Bob que $(b,c) = (0,1)$ ou $(b,c) = (1,0)$, mais elle ne souhaite pas dévoiler à Bob d'information supplémentaire. Montrer qu'Alice peut le faire en exhibant un entier x tel que $zt = yg^x \bmod p$. Expliquer pourquoi Bob est convaincu et pourquoi il n'apprend rien d'autre.

c) On suppose maintenant que Alice souhaite convaincre Bob que ses deux bits b et c sont égaux, c'est-à-dire que $(b,c) = (0,0)$ ou $(b,c) = (1,1)$ mais sans rien révéler d'autre. Comment procède-t-elle dans ce cas ?

d) On suppose maintenant que Alice s'est engagée sur deux *séquences* ou n-uples de bits $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ et $c = (c_1, \ldots c_n) \in \{0,1\}^n$. Elle a donc soumis à Bob deux n-uples $z = (z_1, \ldots, z_n) \in (\mathbb{Z}/p\mathbb{Z})^n$ et $t = (t_1, \ldots, t_n) \in (\mathbb{Z}/p\mathbb{Z})^n$. Elle souhaite maintenant démontrer à Bob que

ses deux n-uples sont *différents* sans rien lui révéler d'autre. Elle pourrait appliquer le protocole précédent à un couple (b_i, c_i) et aux engagements associés z_i et t_i, mais cela révèlerait que non seulement b et c sont distincts, mais qu'ils diffèrent en la coordonnée i. Au lieu de cela on considère le protocole suivant :

- Alice choisit une permutation aléatoire π de l'ensemble $\{1, 2, \ldots, n\}$ des indices. Elle calcule ensuite un nouvel engagement $z' = (z'_1, \ldots, z'_n)$ du n-uple permuté $(b_{\pi(1)}, \ldots, b_{\pi(n)})$, ainsi qu'un nouvel engagement $t' = (t'_1, \ldots, t'_n)$ du n-uple permuté $(c_{\pi(1)}, \ldots, c_{\pi(n)})$, puis elle donne z' et t' à Bob.
- Bob choisit un bit aléatoire ε, et le donne à Alice.
- Si $\varepsilon = 0$, Alice révèle π et utilise le protocole de la question précédente pour prouver que les engagements $z_{\pi(i)}$ et z'_i encodent le même n-uple de $\{0,1\}^n$ et que $t_{\pi(i)}$ et t'_i encodent le même n-uple également. Si $\varepsilon = 1$, alors Alice ne révèle pas π mais choisit un indice i pour lequel z'_i et t'_i encodent des bits différents et le démontre.

Montrer que ce protocole démontre que les n-uples b et c sont différents et rien d'autre. Discuter sa complétude, validité, et caractère sans divulgation.