Partiel de Complexité
DEA Algorithmique 2001/2002

Auto-réductions

Le problème du commerçant voyageur TSP prend en entrée un graphe complet orienté $G = (V, V \times V)$, une fonction $w : V \times V \to N$ qui associe à chaque arête un entier positif (poids) et un entier W. La question est s'il existe un cycle hamiltonien dont le poids est infèrieur à W. (Un cycle hamiltonien est un cycle dans lequel chaque nœud de V apparait exactement une fois. Son poids est la somme des poids de ses arêtes.)

On suppose qu'on a un algorithme polynomial A pour décider TSP.

1. Décrire un algorithme polynomial qui construit pour (G, w, W) un cycle hamiltonien de poids au plus W (si la réponse de A est "oui").
2. Décrire un algorithme polynomial qui calcule pour (G, w) le plus petit W tel qu'il existe un cycle hamiltonien de poids W dans G.

Complétude

1. Montrer que le problème suivant est NL-complet :
   Entrée: Graphe orienté $G = (V, E)$, sommets $s, t \in V$.
   Question: Existe-t-il 2 chemins différents de s à t?
2. Montrer que le problème suivant est NP-complet :
   Entrée: Ensemble de clauses $C_1, \ldots, C_m$ tel que les C_i contiennent que des variables positives, et entier k.
   Question: Existe-t-il une valuation des variables qui associe au plus k variables à Vrai et qui satisfait la formule $C_1 \wedge C_2 \cdots \wedge C_m$?

Translation

On définit la fonction pad : $A^* \times N \to (A \cup \{\#\})^*$ par $\text{pad}(w, i) = w\#^j$, où $j = \max(0, i - |w|)$ et $\# \notin A$.

Pour un langage $L \subseteq A^*$ et une fonction $f : N \to N$ soit $\text{pad}(L, f) = \{\text{pad}(w, f(|w|)) \mid w \in L\}$.

1. Montrer que $L \in \text{DSPACE}(n^k)$ implique $\text{pad}(L, n^k) \in \text{DSPACE}(n)$.
2. Déduire (en utilisant le théorème de hiérarchie) que $P \neq \text{DSPACE}(n)$.

NP et densité polynomiale

Un langage $L \subseteq A^*$ est de densité polynomiale, s'il existe un polynôme p tel que pour tout n, L contient au plus $p(n)$ mots de longueur au plus n.

L'ordre du dictionnaire $\prec$ sur l'ensemble des mots de $\{0,1\}^n$ est défini par $\lambda \prec 0 \prec 00 \prec \cdots \prec 0\underbrace{1\cdots 1}_{n-1} \prec 1 \prec \cdots \prec \underbrace{1\cdots 1}_{n}$.

1. Soit LeftSAT le problème suivant :
   Entrée: Formule booléenne ϕ à n variables, mot $w \in \{0,1\}^j$, $j \leq n$.
   Question: Existe-t-il une valuation $v \in \{0,1\}^n$ qui satisfait ϕ et telle que $w \preceq v$?
   Décrire une réduction polynomiale de de SAT à LeftSAT et une autre réduction polynomiale de LeftSAT à SAT.
2. On suppose que S est un langage NP-difficile de densité bornée par le polynome $p(n)$. On déduira un algorithme polynomial pour décider si une formule booléenne ϕ est satisfaisable.
   Soit g une réduction polynomiale de LeftSAT à S et q un polynome tel que $|g(\phi, w)| \leq q(|\phi|)$.

   (a) Soit ϕ une formule satisfaisable à n variables et $v = v_1 \cdots v_n$, $v_i \in \{0,1\}$, la valuation maximale de ϕ par rapport à $\prec$.
   On considère un ensemble $W \subseteq \{0,1\}^j$, $0 \leq j \leq n$ tel que $v_1 \cdots v_j \in W$. Montrer qu'en appliquant les deux opérations suivantes l'ensemble obtenu contient toujours $v_1 \cdots v_j$:

   - Si $w \prec w' \in W$ sont tels que $g(\phi, w) = g(\phi, w')$, alors on supprime w de W.
   - Si $W = \{w_1 \prec \cdots \prec w_m\}$ est tel que $g(\phi, w_i) \neq g(\phi, w_j)$ pour tout $i \neq j$ et $m > k := p(q(|\phi|))$ alors on preserve que $w_1, \cdots, w_k$ dans W.

   (b) Décrire un algorithme polynomial pour décider si une formule est satisfaisable.
   Indication: On commence par $W = \{\lambda\}$ et on cherche la valuation maximale pour ϕ (si elle existe) en incrémentant j.