Examen d'Algorithmique - Maitrise d'Informatique

26 Janvier 2001

Durée: 3 heures.
Documents: Seuls sont autorisés les notes manuscrites de cours et TD.
NB: Le barême est indicatif.

Exercice 1 (14 points)

On considère des relations n-aires entre les réels positifs $\mathbb{R}_{\geq 0}$, pour $n \geq 1$, décrites par des contraintes arithmétiques. Soit $X = \{x_1, \dots, x_n\}$ un ensemble de variables. On considère des ensembles de contraintes de la forme $x_i - x_j \leq c$, où c est un entier (positif ou négatif). Un tel ensemble correspond en fait à un ensemble d'inéquations où l'on compare des différences entre variables avec des constantes entières.

Un ensemble de constraintes ϕ représente l'ensemble des n-uplets $(x_1, \dots, x_n) \in \mathbb{R}^n_{\geq 0}$ qui satisfont *toutes* les contraintes dans ϕ. On note $[\![\phi]\!]$ cet ensemble de n-uplets (i.e., l'ensemble des solutions du système).

Considérons par exemple l'ensemble de contraintes $\phi_0 = \{y - x \leq 3, x - y \leq -1\}$ sur $X = \{x, y\}$. On a alors $[\![\phi_0]\!] = \{(x, y) \in \mathbb{R}^2_{\geq 0} \mid 1 \leq y - x \leq 3\}$. L'ensemble $\phi_1 = \{y - x \leq 1, x - y \leq -1\}$ représente quant à lui la droite réelle d'équation $y = x + 1$.

Question 1: Soient $\phi_1 = \{x - y \leq 1, x - z \leq 5, x - y \leq -1, y - z \leq 2\}$, et $\phi_2 = \{x - z \leq 3, x - y \leq -1, y - z \leq 2\}$. Montrer que $[\![\phi_1]\!] = [\![\phi_2]\!]$. Donner un autre ensemble de contraintes ϕ_3 tel que $[\![\phi_3]\!] = [\![\phi_1]\!]$.

On veut trouver une transformation T des ensembles de contraintes satisfaisant les deux propriétés suivantes:

1. $[\![T(\phi)]\!] = [\![\phi]\!]$,
2. $[\![\phi_1]\!] = [\![\phi_2]\!] \Rightarrow T(\phi_1) = T(\phi_2)$.

La propriété (1) signifie que la transformation T préserve l'ensemble des solutions des contraintes. La propriété (2) signifie que la transformation T donne une forme canonique pour les ensembles représentables par les contraintes, i.e., deux ensembles de contraintes qui représentent le même ensemble de n-uplets ont la même image par T. (De ces deux propriétés, il est clair que $[\![\phi_1]\!] = [\![\phi_2]\!]$ ssi $T(\phi_1) = T(\phi_2)$.)

Soit ϕ un ensemble de contraintes sur l'ensemble des variables X. On associe à ϕ le graphe orienté pondéré $\Gamma[\phi] = (S, A)$ où $S = X$, et $A \subseteq X \times \mathbb{Z} \times X$ est donnée par $A = \{(x, c, y) \mid x - y \leq c \in \phi\}$. Noter qu'il peut y avoir dans $\Phi[G]$ plusieurs arcs de poids différents allant d'un sommet x à un sommet y, mais il n'y a jamais d'arcs allant d'un sommet x vers lui même.

Inversement, étant donné un graphe pondéré $G = (S, A)$ où $S = X$ et $A \subseteq X \times \mathbb{Z} \times X$, on définit $\Phi[G] = \{x - y \leq c \mid (x, c, y) \in A\}$.

Question 2: Soit ϕ un ensemble de contraintes et soient deux arcs $x \xrightarrow{4} y$ et $y \xrightarrow{1} z$ dans $\Gamma(\phi)$. Quelle est la contrainte la plus forte que l'on peut en déduire sur x et z ?

De manière plus générale, soit θ un chemin dans le graphe $\Gamma(\phi)$ entre deux sommets x et y. Quelle est la contrainte la plus forte qu'il induit sur les variables x et y ?

Considérons un autre chemin θ' entre x et y. Quelle la plus forte contrainte $x - y \leq c$ que l'on peut déduire des deux chemins θ et θ' ? Que dire de l'ensemble obtenu en ajoutant $x - y \leq c$ à ϕ ? Si ϕ contient déjà une contrainte $x - y \leq c'$, peut-on omettre l'une des deux contraintes sans modifier $[\![\phi]\!]$?

Etant donné un graphe orienté pondéré $G = (X, A)$ où $A \subseteq X \times \mathbb{Z} \times X$, on définit un graphe orienté pondéré $\overline{G} = (X, \overline{A})$ où $\overline{A}$ est l'ensemble des arcs pondérés $(x, c, y) \in X \times \mathbb{Z} \times X$ tels que $x \neq y$ et $x - y \leq c$ est la plus forte contrainte $x - y$ imposée par $\Phi(G)$. Le graphe $\overline{G}$ est appelé *fermeture* de G. Noter que $\overline{G}$ n'a pas d'arc allant d'un sommet vers lui même.

Question 3: Donner $\overline{\Gamma(\phi_1)}$ et $\overline{\Gamma(\phi_2)}$ pour les deux ensembles de contraintes définies dans la question 1. Proposer un algorithme qui construit $\overline{G}$ à partir de G. Donner la complexité en temps et en espace de cet algorithme.

On pose, pour tout ensemble de contraintes ϕ, $T(\phi) = \Phi[\overline{\Gamma[\phi]}]$.

Question 4: Montrer que la transformation T satisfait bien les propriétés (1) et (2) données plus haut.

On dit qu'un arc (x, c, y) est redondant si $\overline{G - \{(x, c, y)\}} = \overline{G}$, c'est-à-dire que l'on peut enlever cet arc du graphe G sans modifier sa fermeture $\overline{G}$.

On considère maintenant (sans perte de généralité) uniquement des graphes pondérés ayant au plus un arc allant d'un sommet x à un autre sommet y. On dit qu'un graphe est sans 0-circuit s'il ne contient aucun circuit de poids 0.

Question 5: Donner un algorithme pour déterminer si un graphe $\overline{G}$ est sans 0-circuits. Quel est sa complexité ?

Question 5: Soient $G_1 = (X, A_1)$ et $G_2 = (X, A_2)$ deux graphes pondérés sans 0-circuits tels que $\overline{G_1} = \overline{G_2}$. Montrer que s'il existe un arc $(x, c, y) \in A_1$ tel que $(x, c, y) \notin A_2$, alors $\overline{G_1 - \{(x, c, y)\}} = \overline{G_1}$.
Indication: Répondre aux questions suivantes:

1. Que dire du poids minimum d'un chemin entre x et y dans chacun des graphes G_1 et G_2 ?
2. Pourquoi le chemin de poids minimum dans G_2 passe au moins par un autre sommet z ? Que dire des poids minimum de chemins de x à z, et de z à y, dans les graphes G_1 et G_2 ?
3. Considérer le chemin de poids minimum dans G_1 de x vers z. Supposer qu'il utilise l'arc (x, c, y) et en déduire une contradiction. Faire de même pour un chemin de poids minimum allant de z à y.

4. En déduire qu'il existe un chemin dans G_1 entre x et y qui ne passe pas par l'arc (x, c, y) et qui a un poids inférieur ou égal à c.

Question 6: Soit $\overline{G}$ un graphe sans 0-circuit et α un arc de $\overline{G}$. Y-a-il une taille t des sous-graphes de $\overline{G}$ contenant α qu'il suffit de considérer pour déterminer si α est redondant ? Proposer un algorithme qui calcule tous les arcs redondants dans un graphe $\overline{G}$ sans 0-circuit. Quelle est la complexité de cet algorithme ?

Exercice 2 (6 points)

Question 1: Donner un algorithme sur le modèle EREW calculant en $O(\log n)$ la somme des éléments d'une liste d'entiers de taille n, avec $n = 2^k$, utilisant n processeurs.

Question 2: Donner un algorithme sur le modèle CREW PRAM calculant en $O(\log n)$ le produit de deux matrices $n \times n$ d'entiers, avec $n = 2^k$, utilisant n^3 processeurs.

Question 3: Donner un algorithme efficace sur le modèle CRCW PRAM calculant le produit de deux matrices $n \times n$ de booléens, avec $n = 2^k$, utilisant n^3 processeurs. Préciser la complexité. (Ici, la multiplication est interprétée comme un “et”, et l'addition comme un “ou”).