Examen d'Algorithmique - Maitrise d'Informatique

04 Septembre 2001

Durée: 3 heures.
Documents: Seuls sont autorisés les notes manuscrites de cours et TD.
NB: Toutes les réponses doivent être clairement justifiées.

Exercice 1 (6 points)

On désigne par *forêt* un ensemble d'*arbres dirigés vers la racine*. On suppose que les sommets d'une telle forêt sont numérotés de 1 à n. La forêt est définie par une fonction tabulée P qui associe à chaque sommet son *père*. On suppose que pour tout sommet i qui est une *racine*, on a $P(i) = i$.

Question 1: Donner un algorithme parallèle (PRAM) qui, étant donnée une forêt P, calcule une table S telle que, pour tout sommet i, $S(i)$ est la *racine* de l'arbre contenant i. On demande un algorithme dont le temps d'execution est *logarithmique* en fonction de la taille de la forêt. Etudier de manière précise la complexité de votre algorithme. Dire quel est le modèle PRAM choisi (CRCW, EREW, ou autre).

Question 2: En supposant que la forêt est réduite à un seul arbre, écrire un algorithme CREW en $O(1)$ pour résoudre le problème précédent en utilisant n processeurs. Pourquoi tout algorithme EREW résolvant ce problème nécessite un temps en $\Omega(log(n))$?

Exercice 2 (14 points)

Soit $G = (S, A)$ un graphe fini, soit P un alphabet fini désignant un ensemble fini de propositions atomiques, et soit $\Pi : S \rightarrow 2^P$ une fonction associant à chaque sommet du graphe G un ensemble de propriétés atomiques. On suppose que tout sommet dans G a au moins un successeur.

On définit un langage de formules $\mathcal{F}$ décrivant des propritétés de sommets du graphe. On définit l'ensemble des formules $\mathcal{F}$ comme étant le plus petit ensemble tel que:

- si $p \in P$, alors $p \in \mathcal{F}$,
- si $\phi \in \mathcal{F}$, alors $\neg\phi \in \mathcal{F}$,
- si $\phi_1, \phi_2 \in \mathcal{F}$, alors $\phi_1 \vee \phi_2 \in \mathcal{F}$, et $\phi_1 \wedge \phi_2 \in \mathcal{F}$,
- si $\phi \in \mathcal{F}$, alors $\exists\bigcirc \in \mathcal{F}$, $\forall\bigcirc \in \mathcal{F}$, $\exists\Diamond\phi \in \mathcal{F}$, $\exists\Box\phi \in \mathcal{F}$, $\forall\Diamond\phi \in \mathcal{F}$, et $\forall\Box\phi \in \mathcal{F}$.

Chaque formule ϕ de $\mathcal{F}$ désigne un ensemble de sommets de G noté $[\![\phi]\!]$. Nous définissons cet ensemble inductivement selon la structure des formules:

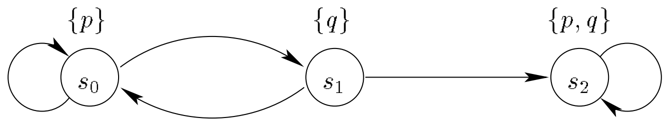

Figure 1: Exemple de Graphe

- si $p \in P$, alors $[\![p]\!] = \{s \in S \ : \ p \in \Pi(s)\}$,
- $[\![\neg\phi]\!] = S \setminus [\![\phi]\!]$,
- $[\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!]$,
- $[\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!]$,
- $[\![\exists\bigcirc\phi]\!] = \{s \in S \ : \ \exists s' \in S.\ (s, s') \in A \text{ et } s' \in [\![\phi]\!]\}$,
- $[\![\forall\bigcirc\phi]\!] = \{s \in S \ : \ \forall s' \in S.\ (s, s') \in A \text{ implique } s' \in [\![\phi]\!]\}$,
- $[\![\exists\Diamond\phi]\!] = \{s \in S \ : \ \exists s_0, s_1, \cdots$ un chemin infini dans G tel que $s_0 = s$ et $\exists i \geq 0.\ s_i \in [\![\phi]\!]\}$,
- $[\![\exists\Box\phi]\!] = \{s \in S \ : \ \exists s_0, s_1, \cdots$ un chemin infini dans G tel que $s_0 = s$ et $\forall i \geq 0.\ s_i \in [\![\phi]\!]\}$,
- $[\![\forall\Diamond\phi]\!] = \{s \in S \ : \ \forall s_0, s_1, \cdots$ chemin infini dans G tel que $s_0 = s.\ \exists i \geq 0.\ s_i \in [\![\phi]\!]\}$,
- $[\![\forall\Box\phi]\!] = \{s \in S \ : \ \forall s_0, s_1, \cdots$ chemin infini dans G tel que $s_0 = s.\ \forall i \geq 0.\ s_i \in [\![\phi]\!]\}$.

Etant donnés un sommet s et une formule ϕ, si $s \in [\![\phi]\!]$, on dit que s *satisfait* ϕ. Un sommet s satisfait une proposition atomique p si et seulement si p appartient à l'ensemble des propositions $\Pi(s)$. Les operateurs $\neg$, $\vee$, et $\wedge$ sont les connecteurs logiques usuels correspondant respectivement à la négation (le *non*), la disjonction (le *ou*), et la conjonction (le *et*). Ainsi, s satisfait $\neg\phi$ si et seulement s'il ne satisfait pas ϕ, et s satisfait $\phi_1 \vee \phi_2$ (resp. $\phi_1 \wedge \phi_2$) si et seulement s'il satisfait ou bien ϕ_1 ou alors ϕ_2 (resp. il satisfait à la fois ϕ_1 et ϕ_2). La formule $\exists\bigcirc\phi$ est satisfaite par un sommet s si et seulemet si celui ci possède un successeur immediat satisfaisant ϕ (autrement dit, s est un prédécesseur de ϕ). La formule $\forall\bigcirc\phi$ est satisfaite par s si *tous* les successeurs immediats de s satisfont ϕ. La formule $\exists\Diamond\phi$ (resp. $\exists\Box\phi$) est satisfaite par s si et seulement s'il *existe* un chemin infini à partir de s où la formule ϕ est satisfaite au moins une fois (resp. ϕ est toujours satisfaite). La formule $\forall\Diamond\phi$ (resp. $\forall\Box\phi$) est satisfaite par s si et seulement si sur *tous* les chemins infinis à partir de s, la formule ϕ est satisfaite au moins une fois (resp. ϕ est toujours satisfaite). Noter que si s satisfait ϕ alors il satisfait aussi $\exists\Diamond\phi$ et $\forall\Diamond\phi$.

Considérons par exemple le graphe décrit par la figure 1. Dans cette figure, l'étiquette de chaque sommet s_i du graphe correspond à l'ensemble $\Pi(s_i)$. D'après les définitions ci-dessus, on peut dire que s_0 satisfait p puisque $p \in \Pi(s_0)$, et il satisfait aussi $\neg q$ du fait que $q \notin \Pi(s_0)$. De plus, s_0 satisfait $\exists\bigcirc q$ puisqu'il a un successeur (s_1) qui satisfait q, mais s_0 ne satisfait pas $\forall\bigcirc q$ puisque ses successeurs ne satisfont pas tous q. En effet, s_0 est successeur de lui même, et s_0 ne satisfait pas q. On peut aussi voir que s_0 satisfait la formule $\exists\Diamond(p \wedge q)$ puisqu'il existe un chemin à partir de s_0 qui passe par un sommet (s_2) satisfaisant $p \wedge q$. Remarquer que s_0 ne satisfait pas $\exists\bigcirc(p \wedge q)$, mais il satisfait $\exists\bigcirc\exists\bigcirc(p \wedge q)$. Remarquer aussi que s_0 ne satisfait pas la formule $\forall\Diamond(p \wedge q)$ puisqu'il existe des chemins infinis à partir de s_0 où $p \wedge q$ n'est jamais satisfaite (par exemple le chemin obtenu en répétant le circuit $s_0 s_1 s_0$).

Question 1 En considérant le graphe de la figure 1, donner les ensembles $[\![\phi]\!]$ pour les formules ϕ suivantes: $\exists\bigcirc p$, $\exists\bigcirc\neg p$, $\forall\bigcirc p$, $\forall\bigcirc q$, $\exists\Diamond p$, $\forall\Box p$, $\exists\Diamond q$, $\forall\Box q$, $\exists\Box p$, $\exists\Box q$, $\forall\Diamond p$, $\forall\Diamond q$, $\forall\Diamond\forall\Box p$, $\forall\Box\exists\Diamond q$, $\forall\Box\forall\Diamond q$.

Dans la suite du problème, on veut définir un algorithme permettant de construire automatiquement l'ensemble $[\![\phi]\!]$ pour toute formule ϕ. On représentera l'ensemble $[\![\phi]\!]$ par un tableau de booléens T_ϕ indéxé par les sommets du graphe (si $s \in [\![\phi]\!]$ alors $T_\phi(s) = 1$ sinon $T_\phi(s) = 0$). On suppose que le graphe G est représenté par une matrice d'adjacence Adj (si $(s, s') \in A$ alors $Adj(s, s') = 1$ sinon $Adj(s, s') = 0$).

Question 2 Décrire informellement des procédures linéaires en fonction de la taille du graphe permettant de construire:

- $[\![\neg\phi]\!]$, à partir de $[\![\phi]\!]$,
- $[\![\phi_1 \vee \phi_2]\!]$, $[\![\phi_1 \wedge \phi_2]\!]$ à partir de $[\![\phi_1]\!]$ et $[\![\phi_2]\!]$,
- $[\![\exists\bigcirc\phi]\!]$ à partir de $[\![\phi]\!]$.

Question 3 Démontrer que, pour toute formule ϕ, on a $[\![\forall\bigcirc\phi]\!] = [\![\neg\exists\bigcirc\neg\phi]\!]$, $[\![\forall\Diamond\phi]\!] = [\![\neg\exists\Box\neg\phi]\!]$, et $[\![\forall\Box\phi]\!] = [\![\neg\exists\Diamond\neg\phi]\!]$.

Question 4 On pose $\exists\bigcirc^0\phi = \phi$ et pour tout $i \geq 0$, $\exists\bigcirc^{i+1}\phi = \exists\bigcirc\exists\bigcirc^i\phi$. Démontrer que $[\![\exists\Diamond\phi]\!] = \{s \in S \ : \ \exists k \geq 0.\ s \in [\![\exists\bigcirc^k\phi]\!]\}$.

Question 5 Donner un algorithme linéaire en fonction de la taille de G, permettant de calculer l'ensemble $[\![\exists\Diamond\phi]\!]$, étant donné l'ensemble $[\![\phi]\!]$ (l'algorithme construit $T_{\exists\Diamond\phi}$ en utilisant T_ϕ). En déduire un algorithme linéaire pour le calcul de l'ensemble $[\![\forall\Box\phi]\!]$ à partir de $[\![\phi]\!]$.

Question 6 Soit $G_\phi = (S_\phi, A_\phi)$ tel que $S_\phi = [\![\phi]\!]$ et $A_\phi = \{(s, s') \in A \mid s, s' \in S_\phi\}$. Montrer que pour tout $s \in S$, $s \in [\![\exists\Box\phi]\!]$ si et seulement s'il existe un chemin fini $s_0 \cdots s_i s_{i+1} \cdots s_j$ dans G_ϕ tel que $s_0 = s$, $0 \leq i < j$, et $s_i = s_j$ (c'est-à-dire, il existe un circuit accessible à partir de s_0 dans le graphe G_ϕ).

Question 7 Donner un algorithme linéaire en fonction de la taille de G, permettant de calculer l'ensemble $[\![\exists\Box\phi]\!]$, étant donné l'ensemble $[\![\phi]\!]$. En déduire un algorithme linéaire pour le calcul de l'ensemble $[\![\forall\Diamond\phi]\!]$ à partir de $[\![\phi]\!]$. (Indication: l'algorithme peut être défini sans construire explicitement une nouvelle représentation de G_ϕ).

Question 8 Donner un algorithme permettant de calculer ϕ pour toute formule ϕ. Donner la complexité de votre algorithme en fonction de la taille du graphe G et de la taille de la formule ϕ (nombre de ses sous-formules).