Université Denis Diderot Paris 7
UFR d'informatique
M2-algorithmique répartie

Examen d'algorithmique répartie
(25 mars 2013)
(documents manuscrits autorisés)
3 pages

Préambule : Π est un ensemble de processus communiquant par envoi-réception de messages. La communication est asynchrone sans pertes (les message arrivent toujours à leur destinataire, et le temps entre l'émission et la réception est arbitraire).

$G = <\Pi, A>$ est le graphe (orienté) de communication : les sommets de G sont les processus de Π et (i,j) est une arc de G si et seulement si il existe un lien de communication du processus i vers le processus j (ainsi i peut envoyer un message à j si et seulement si (i,j) est un arc de G). On supposera toujours que G est un graphe fortement connexe (entre deux processus quelconques i et j il existe un chemin de liens de communication de i à j). On supposera aussi que les processus sont sans défaillance.

Dans un *algorithme d'élection de leader*, chaque processus p a une variable locale $etat_p$ initialisée à $\perp$, l'algorithme d'élection de leader doit (1) terminer pour chaque processus et (2) quand cet algorithme termine un seul processus p_0 aura sa variable $etat_{p_0}$ égale à ELU tous les autres ayant leur variable $etat_p$ égale à $BATTU$.

Le but de l'exercice est de montrer que, essentiellement, avoir un leader unique et avoir des identités uniques connues par les processus revient au même.

Exercice 1.— On considère un système de n processus $\Pi = \{p_0, \cdots, p_{n-1}\}$. On suppose dans cet exercice que chaque processus a une identité unique $id(p_i)$ qu'il connaît et qui est un entier.

1. On suppose que G est un anneau unidirectionnel : (p_i, p_j) est un lien de communication si et seulement si $j = (i+1) \bmod n$. Proposer un algorithme d'élection de leader dans ce cas. (On supposera que pour chaque processus i, $sendNext(m)$ permet au processus i d'envoyer un message au processus suivant sur l''anneau. On décrira le code exécuté par chacun des processus).
2. On suppose maintenant que le graphe de communication G est quelconque (mais fortement connexe) et que l'on dispose d'un algorithme de vague V, pour lequel tout processus peut être initiateur et tous les processus décident.
   (a) En supposant qu'un seul processus p_{init} prend l'initiative de lancer la vague, comment peut-on utiliser cet algorithme de vague pour réaliser une élection de leader ? (Pour cela on pourra ajouter des informations aux messages émis par l'algorithme de vague, et des codes particuliers que les processus peuvent exécuter sur réception des messages de l'algorithme de vague. On pourra aussi choisir comme leader le processus ayant l'identité la plus petite).
   (b) Que peut-on faire si dans l'algorithme de vague considéré V, plusieurs processus prennent l'initiative de lancer une vague ?
   (c) Que peut-on faire, si contrairement à ce qui est supposé, dans l'algorithme de vague considéré tous le processus ne décident pas ?
   (d) Déduire des questions précédentes que dès que les processus connaissent leur identité et que le graphe de communication est non orienté (les liens de communication sont "bi-directionnels" : il y a un lien de communication de i à j si et seulement si il y a aussi un lien de communication de j à i) alors il est possible de réaliser une élection de leader (on pourra utiliser l'existence de l'algorithme de probe/echo).

Exercice 2.— Dans cet exercice les processus ne connaissent pas leurs identités, ils ne connaissent pas non plus le nombre de processus.

Les processus sont, cependant, capables de distinguer entre leurs divers liens de communication. Plus précisément chaque processus p a une variable LI_p qui donne l'ensemble des liens entrants de p et une variable LO_p qui donne l'ensemble des liens sortants de p. Si $LO_p = \{x_1, \ldots, x_m\}$ alors chaque x_i correspond à un lien sortant de communication de p vers un autre processus (x_i correspond à une arc de G dont l'origine est p). De même, si $LI_p = \{y_1, \ldots, y_l\}$ alors chaque y_i correspond à un lien de communication entrant de p (y_i correspond à une arc de G dirigé vers p).

Un processus p peut envoyer un message m sur le lien x_i élément de LO_p (respectivement recevoir un message m' sur un lien y_j élément de LI_p), dans ce cas le message sera envoyé (respectivement reçu) sur le lien correspondant à x_i (respectivement y_j) : si x_i correspond au lien (p, q), alors q, recevra ce message m sur l'élément de LI_q correspondant au lien (p, q) (respectivement si y_j correspond au lien (r, p), alors r a émis ce message m' sur l'élément de $L0_r$ correspondant au lien (r, p)).

1. On suppose dans cette question que le graphe de communication est un anneau uni-directionnel (cf question 1 de l'exercice 1), mais les processus ne connaissent pas leur identité.

   Par ailleurs, on suppose qu'un processus particulier q est distingué parmi les processus : il exécute un code différent des autres. On dira que q est le seul "initiateur" de l'algorithme.

   (a) Ecrire un code pour le processus initiateur q et un code pour les autres processus qui permettra à q de compter le nombre de processus sur l'anneau. (q pourra générer un "jeton" qui circule sur l'anneau et qui compte le nombre de processus)

   (b) En supposant que q, le processus initiateur, connaisse le nombre n de processus sur l'anneau, écrire un code pour q et un code pour les autres processus qui permette d'attribuer à chaque processus un entier unique dans $\{0, \cdots, n-1\}$.

   (c) Déduire des questions précédentes que si l'on disposait d'un algorithme d'élection de leader sur un anneau anonyme, alors chaque processus pourrait obtenir une identité unique prise dans l'ensemble $\{0, \cdots, n-1\}$. En déduire aussi que dans un anneau de processus anonymes si on peut réaliser une élection de leader, on peut exécuter tout algorithme défini pour un anneau non-anonyme.

2. On suppose maintenant que les liens de communication sont "bi-directionnels" : s'il y a un lien de communication du processus i au processus j il y a aussi un lien de communication de j à i. Ainsi pour tout processus p, les liens entrants sont aussi des liens sortants (et vice-versa) : $LI_p = LO_p$.

   (a) Comme dans la question précédente, on suppose que q est un processus distingué qui exécute un code différent des autres.

      i. Ecrire simplement l'algorithme de probe/echo avec q comme seul initiateur dans ce cadre où les processus ne connaissent pas leur identité mais uniquement les liens de communication (variables LI_p et LO_p).

      ii. (Cette question est une question d'algorithmique séquentielle). On suppose que l'on a un arbre orienté de racine r : pour chaque noeud m, $pere_m$ est le père de m (nil si m est la racine de l'arbre), et $Fils_m$ est la liste des fils de m ($Fils_m$ est vide si m est une feuille).

         A. A chaque noeud m est associé un tableau d'entier $TssArbre_m$ indexé par les fils de m. Décrire (de façon relativement informelle) un algorithme (séquentiel) qui assurera que pour tout noeud m, $TssArbre_m[i]$ sera égal au nombre de noeuds dans le sous-arbre de m de racine i. (On remarquera que la somme des $TssArbre_m[i]$ pour $i \in Fils_m$ est la taille du sous-arbre de racine m, et, en particulier, le nombre total de noeuds de l'arbre est égal à la somme des $TssArbre_r[i]$ pour i fils de r plus 1).

         B. On suppose que $TssArbre$ vérifie la condition précédente, décrire un algorithme permettant d'associéer à chaque noeud de l'arbre un entier différent dans l'ensemble $\{1, \cdots, n\}$ où n est le nombre de noeuds de l'arbre. (Pour cela, le père m de chaque noeud pourra associer à chacun de ses fils f des sous-intervalles disjoints de taille $TssArbre_m[f]$)

      iii. En supposant qu'il y a un seul processus distingué q, écrire un code pour q et un code pour les autres processus permettant à q de compter le nombre de processus. (On pourra utiliser l'algorithme de probe/echo qui construit un arbre et les questions précédentes).

iv. En supposant qu'il n'y a qu'un seul processus distingué q et que ce dernier connaît le nombre n de processus, écrire un code pour q et un code pour les autre processus qui attribue à chaque processus un nombre dans l'ensemble $\{0, \cdots, n-1\}$. (On pourra aussi s'inspirer de l'algorithme de probe/echo et des question précédentes)

v. Déduire des questions précédentes que dans le cas d'un graphe de communication avec des liens bidirectionnels, l'existence d'un leader unique dans le système permet d'exécuter tout algorithme défini pour un tel graphe de communication pour lequel chaque processus a une identité unique.