

L3 d'Informatique

EXAMEN TERMINAL EI61

Algorithmique avancée

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage *après* avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

Tout document autorisé
Durée : 2 heures

Les deux parties sont à rédiger sur des copies séparées.

1 Algorithmique répartie

Les algorithmes de parcours de réseau, éventuellement associés à la construction d'un arbre couvrant, sont souvent utilisés comme base d'autres algorithmes répartis. Un tel algorithme peut également fournir un "sous-produit" intéressant, comme par exemple des informations de routage. En effet, si un site lance un parcours et que chaque site mémorise dans une variable *Pere* l'identité du site qui lui a envoyé le parcours, alors chaque site saura que, pour envoyer un message à l'initiateur, il lui suffit d'envoyer ce message à *Pere* (à charge pour ce dernier de faire de même, et ainsi de suite afin de remonter jusqu'à l'initiateur).

Précisons les hypothèses sur les réseaux utilisés dans ce texte :

Le réseau est constitué de processus ou sites communicants par messages. Ces processus ont des identités, deux à deux distinctes, et chacun connaît les identités de ses voisins immédiats. On parlera en fait de successeurs au lieu de voisins, car les lignes de communications sont unidirectionnelles pour ce qui concerne les messages utilisés dans les algorithmes ci-dessous (d'autres messages, utilisant les informations de routage ainsi créées, pourront circuler dans l'autre sens, mais on ne s'en préoccupe pas ici). Les parcours considérés seront toujours lancés par un site particulier, appelé initiateur. On supposera pour simplifier que cet initiateur est connu de tous (ce qui évitera de transmettre son identité dans les messages). On suppose de plus que les lignes sont fiables (les messages ne sont ni perdus, ni altérés, ni dé-séquencés).

Un algorithme de parcours parallèle (sans acquittements) peut alors s'écrire ainsi :

Chaque site possède :

une constante *Successeurs* (identités des sites successeurs)
une variable booléenne *Dejavu*, initialisée à faux
une variable de type identité de site, *Pere*, initialisée à nil.

Initiateur :

Sur décision de lancer le parcours

Dejavu = vrai

envoyer *Parcours* à tous les éléments de *Successeurs*

```
Tout site :
sur réception de Parcours venant de j
si DeJaVu = faux alors
    DeJaVu = vrai
    Pere = j
    envoyer Parcours à tous les éléments de Successeurs
```

1.1 Calcul et mémorisation des longueurs

On suppose de plus que toute liaison de communication est munie d'une longueur. Il s'agit d'une valeur entière positive qui modélise par exemple le temps de transmission d'un message. Cette notion de longueur s'étend aux chemins constitués par une succession de liaisons par simple addition des longueurs de chaque liaison. On suppose que chaque site connaît les longueurs des liaisons vers ses successeurs immédiats.

Question 1.1 *Ajouter à l'algorithme ci-dessus les constantes, variables, paramètres, instructions nécessaires pour que chaque site calcule et mémorise la longueur du chemin qui lui permet d'atteindre l'initiateur via les sites mémorisés dans les variables Pere.*

1.2 Mémorisation des plus courts chemins

Dans un réseau quelconque, le chemin allant d'un site à l'initiateur n'est en général pas unique. Les différents chemins peuvent avoir des longueurs différentes. On appelle distance d'un site à l'initiateur la plus petite longueur des chemins allant à l'initiateur. L'algorithme vu ci-dessus ne permet pas à chaque site de connaître sa distance à l'initiateur, parce que le contrôle effectué sur le parcours est trop rudimentaire : on prend en compte le premier message qui arrive, et on rejette les autres.

Question 1.2 *Modifier l'algorithme pour que chaque site puisse connaître et mémoriser sa distance à l'initiateur, et que les variables Pere mémorisent de manière répartie les plus courts chemins (au lieu du premier chemin trouvé).*

1.3 Terminaison de l'algorithme

Dans la question précédente, on a dû modifier le mécanisme du contrôle du parcours standard vu en cours et rappelé au début de la section.

Question 1.3 *Montrer que ce nouveau mode de contrôle assure encore que l'algorithme se termine (i.e. ne comporte qu'un nombre fini d'événements).*

Indication : on pourra raisonner sur l'évolution de la valeur du vecteur (inconnus, totaldist, messages) où :

- inconnus est le nombre de sites ne connaissant pas encore de chemin (donc de longueur) vers l'initiateur (ce sont les sites n'ayant pas encore reçu de message)
- totaldist est le total des distances connues à l'initiateur (par tous les sites ayant reçu au moins un message)
- messages est le nombre de messages de parcours émis et non encore traités.

2 PARTIE DE JULIEN CLEMENT