

Algorithmique distribuée – Examen

INFO4, PolyTech’Grenoble

Tous les documents sont autorisés

Temps imparti : 2 heures

Tiers Temps : ne pas faire les questions 10, 11, 12 et 13.

Barème indicatif sur 20 points

Le barème est adapté de manière proportionnelle.

1 Circulation de jeton

On s’intéresse à un algorithme de circulation de jeton selon un parcours en profondeur d’abord dans un réseau quelconque.

Les hypothèses sont les suivantes :

- le réseau est bidirectionnel et connexe
- un noeud est distingué comme étant l’initiateur
- les noeuds ne sont pas identifiés, mais les canaux sont identifiés, de 1 à δ_p , le degré du noeud.

Algorithme 1 Algorithme pour un processus p

Variables

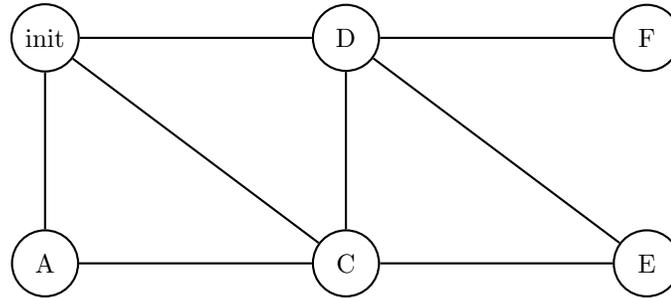
- 1: $parent_p$: un pointeur valant soit l’un des canaux issus de p , soit \perp (pour dire qu’il n’a pas encore de valeur), soit $self$ (valeur pour l’initiateur)
- 2: $children_p$: ensemble de canaux
- 3: $visited_p$: ensemble de canaux

Algorithme pour un processus p

- 4: $parent_p \leftarrow \perp$; $children_p \leftarrow \emptyset$; $visited_p \leftarrow \emptyset$
 - 5: **Si** p est initiateur **alors** $parent_p \leftarrow self$; soit $k \in \{1, \dots, \delta_p\}$ Envoyer $\langle T \rangle$ à k **Fin Si**
 - 6: **Pour toujours**
 - 7: Réceptionner $\langle M \rangle$ de k
 - 8: **Si** $M = T$ **alors**
 - 9: **Si** $parent_p = \perp$ **alors**
 - 10: $parent_p = k$; $visited_p = \{k\}$
 - 11: **Si** $|visited_p| = \delta_p$ **alors** Envoyer $\langle A, yes \rangle$ à $parent_p$
 - 12: **Sinon** soit $l \in \{1, \dots, \delta_p\} \setminus visited_p$ Envoyer $\langle T \rangle$ à l **FinSi**
 - 13: **Sinon** Envoyer $\langle A, no \rangle$ à k
 - 14: **Fin Si**
 - 15: **SinonSi** $M = A, b$ **alors**
 - 16: **Si** $b = yes$ **alors** $children_p = children_p \cup \{k\}$ **FinSi**
 - 17: $visited_p = visited_p \cup \{k\}$
 - 18: **Si** $|visited_p| = \delta_p$ et $parent_p \neq self$ **alors** Envoyer $\langle A, yes \rangle$ à $parent_p$
 - 19: **Sinon** soit $l \in \{1, \dots, \delta_p\} \setminus visited_p$ Envoyer $\langle T \rangle$ à l **FinSi**
 - 20: **Fin Si**
 - 21: **Fin Pour toujours**
-

La spécification qu’on veut satisfaire est qu’il y a exactement un jeton en circulation dans le réseau et que tout noeud reçoit le jeton en temps fini. Comme propriété supplémentaire, on assure que le chemin emprunté

par le jeton utilise un parcours en profondeur d'abord, l'arbre étant représenté dans l'algorithme ci-dessus par les variables *parent* et *children*.



Question 1 (2 points) Faire un exemple d'exécution de l'algorithme 1 sur la topologie précédente.

- L'algorithme termine-t-il ?
- Quel est l'arbre que vous obtenez ?
- Quels sont les messages qui matérialisent le jeton et sa circulation ?
- Quelles autres observations pouvez vous faire ?

Question 2 (1 point) Il y a exactement un jeton dans le réseau. Utilisez le code de l'algorithme pour obtenir simplement cette affirmation.

Question 3 (1 point) La première fois qu'un noeud reçoit un message, c'est un message de type T . Pourquoi ?

Question 4 (1 point) Que se passe-t-il quant un message de type T est envoyé à un noeud déjà raccroché à l'arbre (c'est-à-dire qu'il a déjà positionné sa variable *parent*) ? Dédurre une propriété importante du graphe engendré par les variables *parent*.

Question 5 (2 points) En cours d'exécution de l'algorithme, une étape consiste à, depuis un noeud donné (qui peut être l'initiateur comme un autre noeud), construire par envoi de message de type T une chaîne la plus longue possible sans créer de cycle.

Compléter et justifier cette explication en vous appuyant sur les numéros de ligne de l'algorithme.

Question 6 (2 points) En cours d'exécution de l'algorithme, considérer une telle chaîne et on suppose qu'elle est complète, c'est-à-dire qu'elle ne peut plus être allongée. On nomme f le dernier noeud ajouté dans cette chaîne : il est, par définition, entouré de noeuds déjà atteints par la circulation de jeton (c'est-à-dire qu'ils ont positionné leur variable *parent*).

Décrire ce qu'exécute f jusqu'à sa terminaison locale.

Question 7 (2 points) En cours d'exécution de l'algorithme, considérer de nouveau une telle chaîne complète. On nomme p le noeud le plus éloigné du début de la chaîne et qui a un voisin, q non encore atteint par la circulation de jeton (c'est-à-dire qu'il n'a pas positionné sa variable *parent*).

Montrer qu'en temps fini q va être atteint par la circulation de jeton.

Question 8 (1 point) Que peut on conclure sur la propriété de vivacité de la spécification et pourquoi ?

Question 9 (2 points) Combien de messages sont échangés par l'algorithme ? Justifiez.

Question 10 (2 points) On suppose que l’algorithme précédent a fini de s’exécuter. Les variables *children* et *parent* des noeuds peuvent maintenant être utilisées pour écrire un autre algorithme. On vous demande d’écrire un algorithme,

- que l’initiateur démarre
- et qui termine;
- quand l’algorithme termine, l’initiateur doit avoir une valeur qui contient le nombre de noeuds du réseau et une autre avec le nombre de feuilles de l’arbre construit.

Question 11 (1 point) On a supposé que l’algorithme 1 avait terminé son exécution avant d’exécuter celui de la question précédente. Peut-on détecter facilement la terminaison de l’algorithme 1? Comment modifier le code de l’algorithme 1 pour que les 2 algorithmes s’enchaînent?

2 Variante sur l’algorithme de propagation avec retour

On veut écrire une variante de l’algorithme de propagation avec retour étudié en cours de façon à ce qu’il puisse envoyer plusieurs données. On se place exactement dans les mêmes hypothèses que pendant le cours.

Pour se fixer les idées, on suppose maintenant que la donnée d est découpée en trois parties d_1 , d_2 , d_3 et chaque partie va être envoyée séparément dans le réseau. Tous les noeuds doivent accuser réception des trois parties avant que l’initiateur ne puisse décider.

Cela signifie que votre algorithme devra gérer non plus une vague mais trois vagues, une par donnée. Les messages doivent donc porter l’information de leur vague (sur le type de donnée envoyée ou accusée). On propose d’utiliser les 6 types de messages suivants : Brd_i , ACK_i , $i = 1, 2, 3$.

Question 12 (1 point) Précisez les variables locales de votre algorithme.

Question 13 (3 points) Compléter l’algorithme suivant de façon à répondre à la description ci-dessus.

Comment gérez-vous les accusés de réception?

Comment est prise la décision par l’initiateur?

Algorithme 2 Algorithme à compléter

Algorithme pour le processus initiateur

- 1: ... *initialisation des variables* ...
- 2:
- 3: **Pour toujours**
- 4: Réceptionner $\langle M \rangle$ de q
- 5: // M est de la forme $\langle Brd_i, di \rangle$ ou $\langle ACK_i \rangle$
- 6: ... *traitement* ...
- 7: **Fin Pour toujours**

Algorithme pour les processus non-initiateurs

- 8: ... *initialisation des variables* ...
 - 9:
 - 10: **Pour toujours**
 - 11: Réceptionner $\langle M \rangle$ de q
 - 12: // M est de la forme $\langle Brd_i, di \rangle$ ou $\langle ACK_i \rangle$
 - 13: ... *traitement* ...
 - 14: **Fin Pour toujours**
-