

TD 6 - Parcours dans un graphe quelconque

1 Parcours par circulation d'un jeton

Hypothèses : Graphe quelconque avec un initiateur. Canaux de communication asynchrones et fiables.

Principe : Un initiateur crée un jeton qui circule dans le graphe. Lorsque i reçoit le jeton pour la première fois, il désigne comme son père l'émetteur du jeton. Lorsque i reçoit le jeton, pour la première fois ou non, il l'envoie à un de ses voisins privé de son père et à qui il n'a pas déjà envoyé le jeton. Si i a déjà envoyé le jeton à tous ses voisins privé de son père, il l'envoie alors à son père. Finalement, on obtient un arbre enraciné en l'initiateur. Cet arbre n'a aucune propriété particulière.

Algorithm 1 - Parcours par circulation d'un jeton

———— **Connaissance** ————

$vois_i$: ensemble des voisins du site i

———— **Variables** ————

$pere_i$: identifiant de site, initialisé à $ndef$

$utilise_i$: tableau de booléen (indiqué par l'identifiant des voisins), initialisé à $Faux$

———— **Algorithme** ————

Initialement \rightarrow

$pere_i \leftarrow i$

Choisir $j_0 \in vois_i$

$utilise_i[j_0] \leftarrow Vrai$

Envoyer *Jeton()* à j_0

\triangleright Un seul site

Sur réception de *Jeton()* de j_0 \rightarrow

if $pere_i = ndef$ **then**

$pere_i \leftarrow j_0$

if $\forall j \in vois_i, utilise_i[j] = Vrai$ **then**

STOP

else

if $\exists j \in vois_i, j \neq pere_i \wedge \neg utilise_i[j]$ **then**

Choisir $j \in vois_i \setminus \{pere_i\} \wedge \neg utilise_i[j]$

$utilise_i[j] \leftarrow Vrai$

Envoyer *Jeton()* à j

else

$utilise_i[pere_i] \leftarrow Vrai$

Envoyer *Jeton()* à $pere_i$

Exercice 1. – Calcul de la taille dans un graphe identifié de topologie quelconque –

Donner un algorithme distribué utilisant un parcours et qui permet à un initiateur de déterminer la taille d'un système distribué. La taille d'un système distribué est définie comme le nombre de sites appartenant à ce système. Initialement, un site ne connaît que ses voisins immédiats.

Exercice 2. – Parcours avec plusieurs initiateurs –

On suppose ici que le système distribué est muni de plusieurs initiateurs. Donner un algorithme distribué basé sur le parcours qui permet de construire un arbre couvrant du réseau enraciné en l'un des initiateurs. L'idée est que chaque initiateur lance un parcours, en créant un jeton qu'il étiquette de son identifiant. Il y a donc plusieurs jetons qui vont parcourir le graphe et lorsque "deux parcours se rencontrent", c'est celui qui a le plus grand identifiant qui "gagne".

2 Propagation d'information avec retour

(PIF pour "Propagation of Information with Feedback")

Hypothèses : Graphe quelconque avec un initiateur. Canaux de communication asynchrones et fiables.

Principe : Il y a un initiateur qui initie le PIF. Il envoie un jeton à tous ses voisins, puis attend le jeton de tous ses voisins. Voici ce que fait un site non initiateur : à la première réception du jeton, il désigne comme son père, le site émetteur de ce jeton, puis il diffuse ce jeton à tous ses voisins privé de son père. Enfin, il attend que tous ses voisins privé de son père lui renvoient le jeton. Alors, le site transmet le jeton à son père. Cet algorithme effectue un PIF et construit en même temps un arbre couvrant.

Algorithm 2 Algorithme de PIF, local au site i

———— **Connaissance** ————

$vois_i$: ensemble des voisins du site i

———— **Variables** ————

$nbRecu_i$: nombre de messages reçus, initialisé à 0

$pere_i$: identifiant de site, initialisé à $ndef$

$init_i \in \{Vrai, Faux\}$ initialisé à Faux

———— **Algorithme** ————

Initialement \rightarrow /* Sur un site exactement */

$init_i \leftarrow Vrai$

for all $j \in vois_i$ **do**

Envoyer $Jeton()$ à j

Sur réception de $Jeton()$ **de** $k \rightarrow$

$nbRecu_i ++$

if $init_i$ **then**

if $nbRecu_i = |vois_i|$ **then**

Décider ; **STOP_GLOBAL**

else

if $pere_i = ndef$ **then**

$pere_i \leftarrow k$

for all $j \in vois_i \setminus \{k\}$ **do**

Envoyer $Jeton()$ à j

if $nbRecu_i = |vois_i|$ **then**

Envoyer $Jeton()$ à $pere_i$

Exercice 3. – Election de leader dans un graphe identifié de topologie quelconque –

On souhaite écrire un algorithme distribué d'élection de leader dans un système distribué identifié, de sorte que le leader soit le site d'identifiant maximal. On suppose qu'il existe un unique initiateur dans le réseau (non nécessairement le site d'identifiant maximal).

1. En se basant sur l'algorithme du PIF ci-dessus, écrire un algorithme qui effectue cette élection et tel que, à la fin de l'exécution, l'initiateur du PIF connaisse l'identifiant du leader.
2. Ajouter le nécessaire à votre algorithme afin qu'à la fin de l'exécution, tous les sites connaissent l'identifiant du leader.