
Notes de cours prises pendant les Cours/TD/TP autorisées.

Questions de cours

Question 1. *Donnez les principales caractéristiques du modèle LOCAL.*

SOLUTION. [3 pts |3] Il s'agit d'un modèle synchrone, où tous les processeurs, qui possèdent un identifiant, démarrent en même temps, et les messages sont de taille illimitée.

Question 2. *Même question pour le modèle b-CONGEST.*

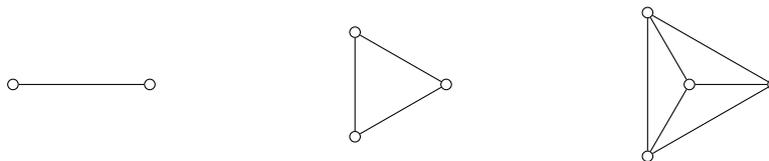
SOLUTION. [3 pts |6] Comme le modèle LOCAL sauf que les messages sont ici limités à b bits. Le modèle CONGEST est identique au modèle $O(\log n)$ -CONGEST, soit avec $b = O(\log n)$.

Détection

Question 3. *Dans le cours, nous avons étudié des algorithmes distribués de détection de cycles de longueur 3 ou 4. De manière générale en algorithmique distribué, pour un graphe H donné, définissez le problème de la « détection d'un graphe H » dans un graphe G .*

SOLUTION. [3 pts |9] Chaque sommet u possède une variable locale booléenne $\text{DETECT}(u)$ qui à la fin doit vérifier : Le graphe G sur lequel l'algorithme distribué s'exécute contient H comme sous-graphe si et seulement s'il existe un sommet u avec $\text{DETECT}(u) = \text{vrai}$.

Dans la suite on notera K_t le graphe composé de t sommets qui sont tous voisins deux-à-deux. Ci-dessous des illustrations du graphe K_t , pour $t = 2, 3, 4$. Notez que le degré de chaque sommet de K_t est $t - 1$.



Dans la suite, on cherche à détecter dans un graphe G une certaine configuration qu'on appellera « K_4 -enraciné ». Pour qu'elle existe, il doit exister dans G au moins un sommet, appelé *racine*, avec exactement trois voisins qui de plus sont deux-à-deux voisins. Dit autrement, G possède un K_4 -enraciné si et seulement s'il possède un cycle de longueur trois, disons $u - v - w$, et un sommet racine r qui a pour voisins u, v et w .

Notez que la racine r joue un rôle potentiellement différent des autres, car si les quatre sommets r, u, v, w forment un graphe K_4 et donc sont de degré au moins trois, r doit impérativement être de degré exactement trois.

Question 4. *Construisez un graphe G contenant un K_4 mais pas de K_4 -enraciné.*

SOLUTION. [3 pts |12] Il faut au moins un K_4 où tous les sommets du k_4 soient de degré > 3 . On peut prendre $G = K_5$ par exemple ou encore G obtenu à partir d'un K_4 en ajoutant un sommet pendant (feuille) à chacun des sommets.

Pour un sommet u d'un graphe G à n sommets, on notera $\text{ID}(u)$ l'identifiant du sommet u , un entier unique de $\{0, \dots, n-1\}$. Comme vu en cours, on suppose que les algorithmes distribués à développer peuvent utiliser les instructions de communication suivantes (supposées être exécutées par un sommet u arbitraire) :

- $\text{SEND}(v, M)$: qui permet d'envoyer au voisin v de u le message M .
- $\text{RECEIVE}(v)$: qui donne le message envoyé depuis le voisin v de u .

Dans les instructions ci-dessus, on pourra confondre v et $\text{ID}(v)$. Pour simplifier, on suppose qu'on dispose également des deux extensions suivantes (qui ne nécessitent pas la connaissance de v) :

- $\text{SENDALL}(M)$: qui permet d'envoyer à tous les voisins de u le message M .
- RECEIVEALL : qui donne l'ensemble des couples (v, M_v) , M_v étant le message envoyé par v .

Vous pourrez également utiliser l'instructions NEWPULSE permettant d'indiquer dans l'algorithme le démarrage d'une nouvelle ronde. Notez que contrairement aux instructions $\text{SEND}/\text{RECEIVE}$ de base, les extensions n'ont pas besoin de connaître explicitement le voisinage du sommet u . Au démarrage de l'algorithme, chaque sommet u supposera connu n et $\text{ID}(u)$. En particulier, u ne connaît ni son degré (le nombre de ses voisins) si son voisinage (la liste de ses voisins).

Question 5. *Construisez dans le modèle LOCAL un algorithme distribué pour un graphe G , permettant de détecter dans G , via une variable DETECT , un K_4 -enraciné. Précisez la complexité en nombre de rondes de votre algorithme.*

SOLUTION. [4 pts |16] L'algorithme, exécuté par chaque sommet u , prend deux rondes : (1) Chaque sommet envoie et récupère ses voisins ; et (2) chaque sommet échange ses listes de voisins puis effectue la vérification.

Ronde 1 :

- $\text{SENDALL}(u)$
- $R_u := \bigcup_{v \in N(u)} \text{RECEIVE}(v)$ NB : $R_u = \{v : v \in N(u)\} = N(u)$

Ronde 2 :

- $\text{SENDALL}(R_u)$
- $S := \text{RECEIVEALL}$ NB : $S = \{(v, R_v) : v \in N(u)\}$
- $\text{DETECT} := \text{vrai}$ ssi $|R_u| = 3$ et $\forall (v, R_v) \in S, R_u \setminus \{v\} \subseteq R_v$

Question 6. *En supposant que le nombre de sommets n de G vérifie $n \leq 2^b$, construisez dans le modèle b -CONGEST un algorithme distribué permettant de détecter dans G un K_4 -enraciné. Précisez la complexité en nombre de rondes de votre algorithme.*

SOLUTION. [4 pts |20] On peut procéder de la même manière, mais en transmettant seulement b bits à la fois, comme fait dans le cours, avec un identifiant par messages puisque $n \leq 2^b$. Le deuxième SENDALL coûte alors $\text{deg}(u)$ messages dans le modèle b -CONGEST, soit un total de n rondes au plus car $\text{deg}(u) \leq n-1$.

On pouvait cependant faire beaucoup mieux, avec seulement quatre rondes, comme ceci : (1) la racine r (un sommet de degré trois susceptible d'être la racine du K_4 -enraciné) transmet à chaque voisin v les identifiants des deux autres voisins (ce qui prend deux rondes dans le modèle b -CONGEST) ; (2) chaque sommet v vérifie s'il est bien connecté aux deux sommets en questions grâce à R_v et transmet sa réponse à r ; (3) r vérifie que le test reçu par ses trois voisins est vrai pour chacun (il effectue un ET logique). Cela fait 4 rondes de communication en tout, sachant que la dernière ronde n'utilise que des messages d'1 bit.

Coloration

Dans tout l'exercice, on considèrera des algorithmes distribués s'exécutant dans le modèle LOCAL. Pour un sous-ensemble $W \subseteq V(G)$ de sommets de G , on notera $\text{ID}(W) = \{\text{ID}(u) : u \in W\}$ l'ensemble des identifiants des sommets de W . Pour tout sous-ensemble $X \subset \mathbb{N}$, on notera $\text{FIRSTFREE}(X) = \min(\mathbb{N} \setminus X)$, c'est-à-dire le plus petit entier naturel qui n'est pas dans X . Par exemple, $\text{FIRSTFREE}(\{0, 1, 2, 4, 8\}) = 3$ et $\text{FIRSTFREE}(\{0\}) = 1$.

Question 7. Montrez que $\text{FIRSTFREE}(X) \leq |X|$ où $|X|$ représente le nombre d'éléments de X .

SOLUTION. [3 pts |23] Si $\text{FIRSTFREE}(X) = k$, alors X contient au moins tous les entiers de $[0, k[$, soit au moins k entiers. Donc $|X| \geq k = \text{FIRSTFREE}(X)$.

On considère l'algorithme A consistant, pour chaque sommet u d'un graphe G , à calculer en parallèle $\text{FIRSTFREE}(\text{ID}(N(u)))$, et on notera $c(u)$ cet entier. Rappelons que $N(u)$ représente l'ensemble des voisins de u dans G .

Question 8. Écrivez l'algorithme A et donnez le nombre de rondes.

SOLUTION. [3 pts |26] Code exécuté par un sommet u quelconque :

- $\text{SENDALL}(\text{ID}(u))$
- $R_u := \bigcup_{v \in N(u)} \text{RECEIVE}(v)$ NB : $R_u = \{\text{ID}(v) : v \in N(u)\}$
- $c(u) := \text{FIRSTFREE}(R_u)$

ce qui prend une seule ronde.

Question 9. Donnez un graphe connexe G à n sommets, et les identifiants pour ses sommets, où l'application de A ne produit que deux valeurs différentes pour la variable c . Quelles sont ces valeurs ?

SOLUTION. [2 pts |28] On peut considérer le graphe K_n avec les sommets numérotés de 0 à $n - 1$ où toutes les arêtes incidentes à 0 ont été supprimées sauf l'arête 0 - 1. Tous les sommets vont se colorier en 0 (car ils sont pas adjacents à 0) sauf le sommet 1 qui lui va se colorier 1. Un cycle ou un chemin à n sommets, ou encore une étoile à $n - 1$ feuilles dont le centre est numéroté 0 ou 1, produisent le même résultat.

NB : Un exemple de graphe G de taille fixée, comme un cycle à quatre sommets, ou une clique à deux sommets, ne constitue pas un exemple de graphe G à n sommets.

Question 10. Donnez un graphe connexe G à n sommets, et les identifiants pour ses sommets, où l'application de A produit $c(u) = \text{ID}(u)$ pour chaque sommet u .

SOLUTION. [2 pts |30] On peut considérer le graphe K_n avec les sommets numérotés de 0 à $n - 1$. Le sommet u va se colorier $c(u) = u$ puisque $N(u) = \{0, \dots, n - 1\} \setminus \{u\}$.