

Détection de la terminaison

Le but de ce TD est de décrire des algorithmes répartis pour détecter la terminaison d'applications réparties. On considère une application distribuée qui fonctionne selon le schéma suivant : un site initiateur unique noté P_1 commence « spontanément » à travailler. Ce site peut contacter à certains moments (par des messages de travail) les autres sites pour leur donner du travail. Chacun des sites alors réveillés peut à son tour déléguer du travail à des sites déjà réveillés ou à des sites encore inactifs.

Exercice : Le problème du gardien.

Les employés d'une entreprise travaillent dans un même immeuble. Tout employé qui travaille la nuit occupe un bureau allumé. Un employé peut quitter un bureau soit pour rentrer chez lui, soit pour continuer son travail dans un autre bureau. Quand tous les employés sont partis, le travail est dit terminé.

Un gardien a pour tâche de fermer l'immeuble et d'éteindre les lumières lorsque tous les employés ont fini leur travail. Décrire un algorithme que peut utiliser le gardien pour fermer l'immeuble. Les bureaux sont numérotés de 1 à n et les employés obéissent au règlement suivant :

1. tout employé qui quitte le travail doit éteindre son bureau.
2. tout employé qui quitte un bureau pour aller travailler dans un autre bureau doit laisser le premier bureau allumé.

Quelle est la règle du règlement qui n'est pas indispensable pour garantir un fonctionnement correct de l'algorithme utilisé par le gardien ?

Correction :

hypothèse : le gardien a une casquette.

En effet, la casquette joue le rôle de drapeau.

Description de l' algorithme (informelle) d'un tour de ronde.

- Si le gardien visite un bureau occupé (allumé) alors
il attend que l' employé quitte le bureau
- Si le gardien quitte un bureau occupé (allumé) alors
il éteint
il se met (ou reste) la tête nue.
il passe au bureau suivant.
- Si le gardien quitte un bureau éteint alors
il passe au bureau suivant.

Invariant : Si le contrôleur a visité tous les bureaux et si il a encore sa casquette, alors le travail est terminé.

Conclusion 1 Le contrôleur fait des tours de ronde tant qu' il termine ces tours de ronde la tête nue.

Propriété de l'algorithme

1. l' algorithme se termine car
 - Aucun bureau ne peut être rallumé lorsque tous les travailleurs sont partis (contrôleur ne rallume jamais un bureau).
 - Le contrôleur va faire un tour sup et il reviendra avec la casquette sur la tête.
2. l' algorithme est correct (même preuve).
3. la règle 1 n'est pas indispensable pour garantir un fonctionnement correct de l' algorithme utilisé par le gardien.

Exercice : Formalisation du problème du gardien.

Nous considérons que la topologie du réseau est un anneau unidirectionnel.

1. Transformer l'algorithme du gardien vu en TD afin qu'il puisse être utilisable. Expliquer en détail cette transformation.
2. Illustrer votre algorithme sur un anneau à cinq sommets.

Correction :

Variables locales :

- $actif : 1..N \rightarrow boolean$
- $a_jeton : 1..N \rightarrow boolean$
- $couleur : 1..N \rightarrow \{noir, blanc\}$
- $Messages : 1..N \rightarrow (1..N, Message, 1..N)^*$

Initialisation :

- $\forall i, actif[i] = false$
- $\forall i \neq 1, a_jeton[i] = false, a_jeton[1] = true$
- $\forall i, couleur[i] = blanc$
- $\forall i \neq 1, Messages[i] = \emptyset, Messages[1] = (n, < JETON, blanc >, 1),$

Protocole (légèrement modifié... :

émettre(s,M,d) →

$Messages[d] = Messages[d] \cup (s, < M >, d) ;$

recevoir(s,<Travail,M>,d) →

$actif[s]=true ;$

$couleur[s]=blanc ;$

Traiter (s, <Travail,M>,d) ;

(s,M, d) ∈ Messages[d] →

$Messages[d] = Messages[d] - (s, < M >, d) ;$

recevoir (s,<M>,d) ;

recevoir(d-1,<JETON,couleur>,d) ∧ actif[d]=false ∧ d ≠ 1 →

$couleur = \max (couleur, couleur[d]) ; // le blanc est le + fort$

émettre(d,<JETON,couleur>,d+1) ;

$a_jeton[d]=false ; couleur[d]=noir ;$

recevoir(d-1,<JETON,couleur>,d) ∧ actif[d]=false ∧ d = 1 ∧ (couleur[d]= blanc ∨ couleur= blanc) →

émettre(d,<JETON,noir>,d+1) ;

$a_jeton[d]=false ; couleur[d]=noir ;$

$\text{recevoir}(d-1, \langle \text{JETON}, \text{noir} \rangle, d) \wedge \text{couleur}[d] = \text{noir} \wedge \text{actif}[d] = \text{false} \wedge d = 1 \rightarrow$
Detection de la terminaison de l'application