DÉTECTION DE LA TERMINAISON

PR. DJAMEL EDDINE SAIDOUNI

EQUIPE CFSC, LABORATOIRE MISC

DÉPARTEMENT IFA, FACULTÉ NTIC

DJAMEL.SAIDOUNI@UNIV-CONSTANTINE2.DZ

0559082425

LE PROBLÈME

Etat logique d'un processus

- Actif
- Passif
  - Terminé
  - En attente
    - De message (le cas qui nous intéresse)
    - Autre

Remarques:

- L'interblocage est un cas particulier de terminaison (anormale)
- Seul un processus actif peut émettre un message de l'application.
- La seule façon de débloquer un processus en attente de message consiste à lui envoyer un message de l'application
- Etat d'un canal
  - Vide : pas de message en transfert
  - Non vide : existence de message en transfert

LE PROBLÈME (SUITE)

Terminaison:

- Une application est terminée s'il existe un instant **t** où on peut observer simultanément :
  - Tous les processus à l'état passif
  - Tous les canaux à l'état vide
- La terminaison est une propriété stable:
  - Tous les processus passifs, donc seul un message peut réveiller un processus, aucun processus n'enverra de message
  - Tous les canaux sont vides, donc aucun processus ne peut recevoir de message.

RÉFLEXIONS SUR L'ALGORITHME

Observer les processus:

- Visite de tous les processus de l'application (en séquentiel ou en parallèle)
    - Cas de la visite séquentielle: On envoie une sonde à P1
        - Tant que P1 est actif il conserve la sonde
        - Dès que P1 devient passif, il envoie la sonde à P2
        - …
        - La sonde arrive à Pn
        - Lorsque Pn devient passif: L'application n'est pas nécessairement terminée même si la sonde ne rencontre que des processus passifs; problème de la fausse terminaison

Validation d'un algorithme de détection de la terminaison

- Toute terminaison est effectivement détectée
- Pas de fausse terminaison: Si l'algorithme déclare « terminaison détectée » il faut montrer qu'il y a effectivement terminaison.

ALGORITHME DE MISRA

Hypothèses: Réseau quelconque **régulier**

Idée : Visite séquentielle des processus et des canaux. La sonde (ou marqueur) parcourt tous les canaux, ce qui implique qu'elle visite tous les processus.

On prédéfinit un circuit C de visite de tous les canaux.

Taille(C) = nombre de traversées d'arcs (longueur du circuit en nombre d'arcs)

Site ou Processus S_i :

- **État** : (actif, passif) := actif;
- **Couleur** : (blanc, noir) := blanc;
- **Marqueur-présent** : booléen := faux {sauf chez l'initiateur}
- **Valeur** : entier; /* le marqueur est évalué

PROCÉDURES DE L'ALGORITHME

Réception de (message,m) /*message de l'application

- Couleur := blanc;
- Etat := actif;

Attente de (message,m) /*message de l'application

- Etat := passif

Réception de (marqueur,l)

- Marqueur-présent := vrai;
- **Si** l = taille(C) **et** couleur = noir **alors** terminaison détectée

PROCÉDURES DE L'ALGORITHME

Emission de (marqueur,l)

*****{seulement si **marqueur-présent** et **état = passif**}

Si couleur = blanc **alors**

Valeur := 0

Sinon

Valeur := l + 1

Finsi

Emettre (marqueur, Valeur) à successeur sur C;

Couleur := noir;

Marqueur-présent := Faux;

VÉRIFICATION

Si l'application se termine, alors l'algorithme détecte la terminaison

- **Remarque:** Si l'application ne se termine jamais, l'algorithme ne se termine jamais. Cet algorithme doit détecter la terminaison, et non l'absence de terminaison.
- Un processus actif est toujours blanc alors qu'un processus passif est blanc ou noir.
- **Hypothèse:** Terminaison est équivalent à ce qu'il existe un instant t où tous les processus sont passifs et les canaux vides (de messages de l'application).
- Après l'instant t, le marqueur ne rencontre que des processus passifs. Il est donc automatiquement réémis par chaque processus (cf. condition de ***émettre marqueur***). Donc, après l'instant t, tous les processus deviennent noirs, et ils resteront noirs (puisqu'il n'y a plus de messages de l'application).

VÉRIFICATION (SUITE)

Si l'application se termine, alors l'algorithme détecte la terminaison

- Initialement, le marqueur est émis par un processus blanc (donc ***valeur = 0***)
- Chaque fois que le marqueur est émis par un blanc, ***valeur repasse à 0***.
- Chaque fois que le marqueur est émis par un noir, ***valeur*** est incrémentée.
- Donc ***valeur*** compte le nombre de processus noirs consécutifs rencontrés.
- Il existe nécessairement un instant où le marqueur rencontre le dernier blanc, donc dernière remise à zéro.
- Ensuite, ***valeur est incrémentée à chaque processus rencontré***, donc ***valeur atteindra taille(C)*** d'où la détection de la terminaison.

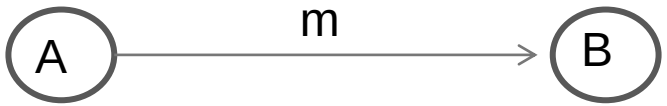
VÉRIFICATION (SUITE)

Si l'algorithme déclare *«terminaison détectée»* alors il y a effectivement terminaison.

- **Hypothèse:** terminaison détectée. Donc **valeur(marqueur) = taille(C).** Donc le marqueur à parcouru tout le circuit en ne rencontrant que des noirs. Donc il existe un instant **t** correspondant au dernier blanc rencontré (***instant de la dernière remise à zéro du marqueur***).
- Vérifions qu'à l'instant **t** l'application était terminée.
  - Vérifions qu'à l'instant **t** tous les processus sont passifs. Après l'instant **t**, tous les processus visités sont noirs. Ces processus ont-ils pu noircir après l'instant **t**? Non, puisque seule l'émission du marqueur peut faire noircir un processus, et dans ce cas le processus est blanc à l'arrivée du marqueur. Donc à l'instant **t** tous les processus sont noirs, donc **passifs**.
  - Vérifions qu'à l'instant **t** tous les canaux sont vides. Supposons qu'à l'instant **t** il y ait encore un message **m** en cours de transfert.

VÉRIFICATION (SUITE)

Supposons qu'à l'instant **t** il y ait encore un message **m** en cours de transfert.



A partir de l'instant **t**, le marqueur va parcourir tous les arcs, donc **AB** sera visité par le marqueur à l'instant **t' > t**. Le réseau étant régulier, **m** arrive donc sur le site **B** avant le marqueur (puisque **m** était sur le canal avant le marqueur), à l'instant **t" > t**. Donc **B** devient blanc lorsqu'il reçoit **m**, donc le marqueur rencontre un processus blanc à l'instant **t"'>t" > t**!! Donc à l'instant **t** tous les canaux sont vides.

EXPOSÉ (FACULTATIF)

Terminaison d’un calcul diffusant (Dijkstra-Scholten)