

Algorithmique distribuée – Devoir à la maison (contrôle continu 1)

INFO4, PolyTech’Grenoble

1 Modalités

Le DM est à commencer à partir du **lundi 13 février 2025** et à rendre **au plus tard le 13 mars 2025 à 20 heures**. Il est à réaliser *en binôme exclusivement*, durant le temps libre. Le rendu s’effectuera selon les modalités précisées par l’enseignant de TD.

Travail à rendre. Il vous est demandé un document où vous ferez figurer les réponses aux questions, hormis le code développé. Le document devra aussi contenir, de façon *synthétique*, les choix d’implémentation que vous aurez pu faire, en les argumentant, et les tests fonctionnels que vous avez réalisés (cas au limite, exécution de toutes les portions du code...). Vous incluez dans une archive `apd-dm-nom1-nom2.{tgz,tar.gz,zip}` :

- (a) le document au format `apd-dm-nom1-nom2.pdf`,
- (b) les sources de votre réalisation : n’inclure que le projet, c’est-à-dire le sous-répertoire du répertoire `sinalgo/src/projects`, que vous aurez ajouté.

Barème : le barème annoncé dans les questions n’est qu’indicatif.

2 Sujet

Le but de ce devoir à la maison est d’entraîner vos capacités d’analyse d’un algorithme donné (*cf.* pseudo-code ci-après).

Hypothèses : les processus et les communications sont asynchrones. Chaque processus est identifié avec un identifiant unique et chaque processus est initiateur. La topologie est un anneau bidirectionnel : chaque processus distingue son voisin droit et son voisin gauche et peut leur envoyer des messages ; l’orientation est consistante. Il n’y a pas de fautes.

Travail à faire :

Question 1 (5 points) Implémenter cet algorithme dans `sinalgo` et le simuler.

Question 2 (2 points) La variable *resultat* représente le résultat booléen calculé par l’algorithme. Que fait cet algorithme ? Spécifier le proprement en termes de propriétés de sûreté et de vivacité.

Question 3 (2 points) Mettre en place des tests permettant de valider la propriété de sûreté.

Question 4 (8 points) Cette question vous guide pour comprendre pourquoi l’algorithme est correct.

- (a) Que représentent les variables ℓ et n ? Constaté que ℓ est de la forme 2^i : on dit que i est le numéro de la phase courante.
- (b) Au début de la phase i , un processus envoie deux messages $\langle ASK, id, 2^i - 1 \rangle$ à ses voisins de gauche et droite. A la fin de la même phase, un processus est considéré comme non éliminé lors de la phase s’il a reçu deux messages $\langle REPLY, id \rangle$ de ses voisins de gauche et de droite. Que dire de l’identité d’un tel processus relativement aux processus qui l’entourent ? Justifier.
- (c) L’algorithme commence à la phase numéro $i = 0$ (puisque ℓ est initialisé à 1). Termine-t-il ? Et si oui, pour quelle valeur de i ?
- (d) Justifier alors la spécification en expliquant pourquoi les propriétés de sûreté et de vivacité sont satisfaites.

Question 5 (2 points) L’algorithme a une complexité en $O(n \log n)$ messages, où n est le nombre de processus du réseau. Expliquer pourquoi.

Question 6 BONUS Expliquer pourquoi l’algorithme a une complexité en temps de l’ordre de $O(n)$ rondes.

Question 7 (1 point) Synthèse et conclusion : pourquoi cet algorithme est-il intéressant ?

Variables

- 1: $id \in \mathbb{Z}$ (identifiant unique du processus)
- 2: $resultat \in \{Vrai, Faux\}$
- 3: $\ell, n \in \mathbb{N}$

Algorithme pour tout processus

```
4:  $\ell \leftarrow 1$ 
5:  $n \leftarrow 0$ 
6:  $resultat \leftarrow faux$ 
7: Envoyer  $\langle ASK, Id, \ell - 1 \rangle$  à  $D$ 
8: Envoyer  $\langle ASK, Id, \ell - 1 \rangle$  à  $G$ 
9: Pour toujours
10:   Réceptionner  $\langle Type, T \rangle$  de  $N$ 
11:   Si  $Type = ASK$  alors
12:     on note  $T = id', \ell'$ 
13:     Si  $id' = id$  alors
14:        $resultat \leftarrow vrai$ 
15:     SinonSi  $\ell' > 0$  alors
16:       Si  $id' > id$  alors
17:         Envoyer  $\langle ASK, id', \ell' - 1 \rangle$  à  $N'$  ( $N' \neq N$ )
18:       Fin Si
19:     Sinon ( $\ell' = 0$ )
20:       Si  $id' > id$  alors
21:         Envoyer  $\langle REPLY, id' \rangle$  à  $N$ 
22:       Fin Si
23:     Fin Si
24:   SinonSi  $Type = REPLY$  alors
25:     on note  $T = id'$ 
26:     Si  $id' \neq id$  alors
27:       Envoyer  $\langle REPLY, id' \rangle$  à  $N'$  ( $N' \neq N$ )
28:     SinonSi  $id' = id$  alors
29:        $n \leftarrow n + 1$ 
30:       Si  $n = 2$  alors
31:          $n \leftarrow 0$ 
32:          $\ell \leftarrow \ell * 2$ 
33:         Envoyer  $\langle ASK, id, \ell - 1 \rangle$  à  $D$ 
34:         Envoyer  $\langle ASK, id, \ell - 1 \rangle$  à  $G$ 
35:       Fin Si
36:     Fin Si
37:   Fin Si
38: Fin Pour toujours
```
