Université Paris Diderot
M2 IAP
Algorithmes Répartis

Devoir à rendre pour le 20 mars sur Moodle ou lors du cours. A faire en monôme ou binôme

Exercice 1.—

$\Pi = \{p_1, p_2, \dots, p_n\}$. On dispose d'un graphe non orienté $G = \langle \Pi, V \rangle$ connexe. On suppose que chaque processus connaît son identité mais ne connaît pas G ni la taille de G . Chaque processus peut identifier ses voisins par ses "ports" de communication. On rappelle Figure 1 le code de l'algorithme de vague Echo sur un graphe non orienté avec un unique initiateur qui est aussi l'unique décideur.

1. On suppose qu'il y a un seul processus initiateur. Donner le code d'une adaptation de l'algorithme d'Echo qui permet à l'initiateur lorsqu'il décide de calculer la taille du graphe.
2. On suppose qu'il y a 2 processus initiateurs p et q qui exécute l'algorithme précédent.
 - (a) Existe-t-il des exécutions où p ne décide pas?
 - (b) Lorsque p (resp. q) décide avec n_p (resp. n_q) comme taille du graphe, que vaut $n_p + n_q$?

Exercice 2.— On considère n processus anonymes sur un graphe G non-orienté et connexe. Chaque noeud dispose d'un générateur de nombre aléatoire. Les processus connaissent n .

Donner un algorithme probabiliste Las Vegas d'élection de leader i.e tous les processus ont une variable LEADER initialisé à indéfini. Pour chaque processus (1) s'il termine, il y a un unique processus qui a sa variable LEADER à gagné les autres processus ont leurs variables LEADER à perdu, et (2) il termine avec une probabilité de terminaison 1.

```

Variable   rec_p : integer init 0
           pere_p:processus init <>

Pour un initiateur
begin
  for all q dans Voisin_p faire
    send(<token>) to q
  while (rec_p<#Voisin_p)
  do   receive(<token>);
      rec_p ++
  decide
end

Pour un non initiateur
begin
  receive(<token>) from q
  pere_p=q
  rec_p ++
  for all q dans Voisin_p et q <> pere_p faire
    send(<token>) to q
  while (rec_p<#Voisin_p)
  do   receive(<token>);
      rec_p ++
  send(<token>) to pere_p
end

```

Figure 1: Algorithme d’Echo. Code du processus p