Université Paris Diderot
M2 IAP - 2017-2018
Algorithmes Répartis

Devoir 2

Ce devoir est à rendre pour le 9 mars sur Moodle ou sur feuille lors du cours. Vous pouvez travailler en binôme.

Des réponses claires et précises sont attendues. Il est aussi inutile de recopier un livre (ou le devoir du voisin). Exprimez vos réponses avec vos propres mots!

Exercice 1.— On suppose que les processus sont organisés en anneau: un processus quelconque p peut communiquer avec son prédécesseur et son successeur. La taille de l'anneau, n, est *connue* de tous les processus, mais les processus ne connaissent pas leur identité. On rappelle que dans un algorithme d'élection tous les processus exécutent le *même* code.

1. Dans l'algorithme Figure 1 qui suit $Candidat_p$ est un booléen qui indique si p est candidat ou non. On suppose qu'initialement au moins un processus est candidat. $tirer()$ est une fonction de tirage aléatoire qui retourne $PILE$ ou $FACE$. On suppose que la probabilité de retourner $PILE$ est égale à la probabilité de retourner $FACE$: $Pr(PILE) = Pr(FACE) = \frac{1}{2}$. On suppose de plus que tous les tirages sont indépendants les uns des autres. Cet algorithme fonctionne en *rondes* dont la valeur est l'entier r (les échanges de messages entre processus ont lieu dans la même ronde pour tous)

```
for r := 0 to ∞ do
  if Candidat_p
  then
    V_p := tirer()
    send V_p to  Successeur_p
    receive  W from  Predecesseur_p
    if V_p = Pile ∧ W = Face then Candidat_p := False
  else
    receive W from Predecesseur_p
    send W to Successeur_p
  endif
```

Figure 1: Algorithme non-déterministe d'election.

   (a) Montrer que si, au début d'une ronde r, il y a au moins un candidat, à la fin de la ronde r il y aura aussi au moins un candidat.

(b) En supposant qu'au début de la ronde r il y a au moins deux candidats, si p est candidat au début de la ronde r quelle est la probabilité qu'à la fin de la ronde r p ne soit plus candidat?

(c) En déduire que la probabilité qu'à la ronde r il y ait plus d'un candidat tend vers 0 quand r tend vers l'infini.

(d) En utilisant cet algorithme transformer l'algorithme de la Figure 1 de façon à ce qu'il termine s'il n'y a plus qu'un seul candidat. En déduire que cet algorithme vérifie les propriétés suivantes: (1) s'il termine il ne reste plus qu'un seul processus candidat (2) la probabilité que cet algorithme termine est 1 (plus exactement la probabilité qu'il ne termine pas avant la ronde r tend vers 0 quand r tend vers l'infini).

(e) Donner un algorithme qui permet à chaque processus d'obtenir une identité (différente des identités des autres processus).

Exercice 2.— On dispose d'un réseau non orienté connexe. Les processus n'ont pas d'identité. Chaque processus dispose d'un ensemble $Voisin$ d'entiers qui identifient les ports avec lesquels il communique avec ses voisins. La taille du réseau n, est *connue* de tous les processus. En s'inspirant de l'algorithme d'Echo et de l'algorithme de l'algorithme de Itai et Rodeh, écrire un algorithme d'élection.