

# Ingénierie des Protocoles : Examen

Stéphane Devismes

## Résumé

Lisez attentivement ce qui suit :

- Aucun document n'est autorisé.
- Le barême est donné à titre indicatif uniquement. Il est susceptible de changer !

## 1 Algorithmique non tolérante aux fautes : élection de leader dans un réseau en arbre (10 points)

1. Rappelez la spécification de l'élection de leader. (1 point)
2. Donnez un algorithme d'élection de leader fonctionnant sous les hypothèses suivantes (3 points) :
  - Processus asynchrones avec identité unique.
  - Canaux FIFO asynchrones étiquetés de 1 à  $\delta_p$  pour tout processus  $p$ .
  - Pas de faute.
  - Topologie en arbre avec au moins deux nœuds.
  - Multi-initiateurs.

Vous justifierez votre démarche en donnant l'idée intuitive de votre algorithme. (1 point)

**Indice :** en utilisant la circulation de jeton vue en cours, tout algorithme pour réseaux en anneau peut être « émulé » sur une topologie en arbre.

3. Prouvez la correction de votre algorithme. (3 points)
4. Justifiez la complexité de votre algorithme. (1 point)
5. Exhibez un pire des cas. (1 point)

## 2 Autostabilisation (10 points)

Le problème du *coloriage* consiste à assigner une couleur (un nombre) à chaque processus en respectant la contrainte suivante : chaque pair de processus voisins a une couleur différente.

En supposant un démon central, il existe une solution autostabilisante simple dans le modèle à états utilisant  $\Delta + 1$  couleurs (les couleurs sont numérotées de 0 à  $\Delta$ ) où  $\Delta$  est le degré de réseau : si un processus à un voisin de même couleur, il choisit une couleur (*e.g.* la plus petite) parmi celles qui ne sont pas utilisées par ses voisins.

1. Ecrivez un algorithme *autostabilisant silencieux* utilisant  $\Delta+1$  couleurs pour résoudre le problème du coloriage dans le modèle à états en supposant un *démon central inévitabile*. Le réseau sera supposé *anonyme et bidirectionnel*. (2 points)
2. Prouvez la stabilisation de votre algorithme (fermeture, convergence, correction). (3 points)
3. Justifiez la complexité en nombre de pas de calculs de votre algorithme. (1 point)
4. Est-ce votre algorithme fonctionne aussi sous l'hypothèse d'un démon localement central? (Justifiez) (1 point)

5. Montrez qu'il existe des topologies où tout algorithme utilisant moins de  $\Delta + 1$  états (*i.e.*, couleurs) ne fonctionne pas. **(1 point)**
6. Montrez qu'il n'existe pas d'algorithme (déterministe) autostabilisant de coloriage fonctionnant dans un réseau anonyme et bidirectionnel sous l'hypothèse d'un démon *distribué inéquitable*. **(2 points)**

### 3 Bonus : consensus (3 points)

Il existe un algorithme déterministe de consensus tolérant  $f$  pannes franches initiales (*i.e.*, tout processus est soit correct soit ne fait jamais de pas de calcul) avec  $2.f < n$  ( $n$  est le nombre de processus) fonctionnant sous les hypothèses suivantes :

- Topologie complète.
- Processus asynchrones avec identité unique.
- Liens asynchrones et fiables.

Construisez un algorithme déterministe tolérant  $f$  pannes franches initiales (avec  $2.f < n$ ) qui résout l'élection de leader sous les mêmes hypothèses. Vous pourrez supposer l'existence d'un algorithme de consensus, et donc l'utiliser. Expliquez le fonctionnement de votre algorithme.