

# Théorie et pratique de la concurrence – Master 1 II

## TD 2 : Sections critiques (modèles)

### Exercice 1:

*Algorithme de Manna et Pnueli*

Soit la solution suivante pour l'implémentation d'une section critique (Manna-Pnueli) :

```
int wantP = 0, wantQ = 0; // variable partagée
```

```
-- Processus P                                -- Processus Q
loop forever:                                  loop forever:
  section NC                                    section NC
  if (wantQ = -1)                                if (wantP = -1)
    wantP := -1                                    wantQ := 1
  else
    wantP := 1
  await wantP != wantQ
  section critique
  wantP := 0
                                                else
                                                wantQ := -1
                                                await wantP != -wantQ
                                                section critique
                                                wantQ := 0
```

1. Construisez le diagramme d'états de cet algorithme.
2. En utilisant ce diagramme, indiquez quelles sont les propriétés de correction (exclusion mutuelle, absence d'inter-blocage, absence de famine, attente bornée) satisfaites par cet algorithme.
3. Est-il possible de modifier cet algorithme afin de satisfaire les propriétés ci-dessus non satisfaites ?

### Exercice 2:

*Mutex avec instructions atomiques*

Supposons qu'une machine fournit l'instruction atomique suivante :

```
flip(int lock) :
  atomic { lock = (lock+1) % 2; /* inverse le lock */
          return (lock) }      /* renvoie sa nouvelle valeur */
```

Un programmeur propose la solution suivante pour l'implémentation d'une section critique pour deux processus :

```
int lock = 0; /* variable partagée */
-- Processus i: /* i est soit 1 soit 2 */
loop forever :
  section NC

  while (flip(lock) != 1) :
    await lock == 0
  section critique
  lock = 0
```

1. Expliquer-lui pourquoi sa solution ne marche pas en lui montrant une trace d'exécution dans laquelle les deux processus se retrouvent en même temps dans leur section critique.
2. Il vous propose de garder sa solution mais de changer dans la machine l'instruction atomique `flip` : au lieu d'une addition modulo 2, utiliser une addition modulo 3.  
Cette nouvelle solution est-elle correcte pour deux processus ?  
Justifier votre réponse (si la réponse est positive donner un invariant, sinon exhiber une trace d'exécution).
3. Pour cette dernière solution, commenter informellement les propriétés d'absence d'inter-blocage et de famine ainsi que celle d'attente bornée.