

Modélisation et spécification – Master 2 LC

TD 4 : Réseaux de Petri

Peter Habermehl (www.liafa.jussieu.fr/~haberm/cours/modspec/)

Modélisation

Exercice 1 :

Feu de signalisation

On souhaite modéliser deux feux de signalisation. Chaque feu a trois couleurs (verte, jaune, rouge). Les couleurs sont changées de rouge à verte à jaune à rouge est ainsi de suite. On veut que les deux feux ne soient pas rouges en même temps. Modéliser ce système avec les réseaux de Petri.

Algorithmes d'exclusion mutuelle

Exercice 2 :

Premier essai

Les deux programmes suivants coopèrent dans le but d'assurer l'exclusion mutuelle à leurs section critiques.

```
boolean c1, c2 = true;
process P1() {
  while (true) {
    non_critique_1;
    c1 = false;
    while (c2 == false) ;
    critique_1;
    c1 = true;
  }
}
process P2() {
  while (true) {
    non_critique_2;
    c2 = false;
    while (c1 == false) ;
    critique_2;
    c2 = true;
  }
}
```

1. Modéliser avec des réseaux de Petri le programme ci-dessus. Utiliser des places pour modéliser les points de contrôle des processus et les états des variables partagées.
2. En utilisant le graphe de marquage, montrer qu'il existe une possibilité de blocage dans cet algorithme.

Exercice 3 :

Algorithme de Dekker

Soit l'algorithme d'exclusion mutuelle suivant :

```
1 bit wantP1 = 0, wantP2 = 0;
2 int turn = 1; // 1 pour P1, 2 pour P2
3 process P1() {
4   while (true) {
5     non_critique_1;
```

```

6      wantP1 = 1;
7      while (wantP2 == 1) {
8          if (turn == 2) {
9              wantP1 = 0;
10             while (turn != 1);
11             wantP1 = 1;
12         }
13     }
14     critique_1;
15     turn = 2;
16     wantP1 = 0;
17 }
18 }

```

1. Modéliser cet algorithme avec des réseaux de Petri.
2. Montrer, en utilisant le graphe de marquage, que la propriété d'exclusion mutuelle est assurée.

Exercice 4 :

Les philosophes affamés

Voir wikipedia. Il y a 4 philosophes qui se trouvent autour d'une table. Chaque philosophe a devant lui une assiette de spaghetti. Directement à gauche de chaque assiette se trouve une baguette pour manger (Il y a donc 4 baguettes). Un philosophe fait deux choses : penser et manger. Pour manger il a besoin des deux baguettes qui sont à côté de son assiette. Chaque philosophe agit de la façon suivante : Il pense, ensuite quand il a envie de manger il prend d'abord la baguette à sa gauche, *ensuite* la baguette à sa droite. Quand il termine de manger, il rend *en même temps* les deux baguettes et il pense, etc.

- Modéliser avec un réseau de Petri ce système.
- Donner le graphe de marquage.
- Que constatez vous ?
- Comment éviter ce problème ?
- Est-ce que votre solution marche aussi avec 5 philosophes ?