Modélisation et spécification – Master 2 LC

TD 4 : Réseaux de Petri

www.liafa.jussieu.fr/~sighirea/cours/modspec/

Algorithmes d'exclusion mutuelle

Exercice 1 : *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 2 : *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.

Modélisation des tampons

On veut modéliser avec des réseaux de Petri des tampons ayant différentes politiques de service. Vus depuis l'extérieur, ces réseaux auront une transition t_I pour l'entrée d'un élément, et une transition t_O pour la sortie d'un élément. On se limitera d'abord à des tampons qui peuvent contenir au plus 3 données.

1. Donner un modèle de type FIFO (First In First Out).
2. Donner un modèle de type LIFO (Last In First Out).
3. Les places sont-elles toutes de même nature ? Et les transitions ?
4. Comment modifier la capacité de ces tampons ?
5. Discuter des propriétés de vos modèles.

Réseaux de Petri colorés

Exercice 3 : *Système de facturation*[1]

[1] Cette description informelle a été proposée à la communauté des chercheurs intéressés par la modélisation afin de comparer les différentes méthodes de spécification et d'offrir

*I*ntroduction : soit la spécification informelle ci-dessous.

1. Le sujet doit facturer des ordres.
2. Facturer est de changer l'état d'un ordre (de l'état "en suspens" en "facturé").
3. Sur un ordre, on a une et seulement une référence à un produit commandé avec une certaine quantité. La quantité peut être différente d'un ordre à un autre.
4. La même référence peut être commandée sur plusieurs différents ordres.
5. L'état de l'ordre sera changé en "facturé" si la quantité commandée est au plus égale à la quantité qui est en stock selon la référence du produit commandé.

Vous devez considérer les deux cas suivants :

Cas 1 : Toutes les références commandées sont des références en stock. Les actions ou l'ensemble des ordres peuvent changer,

- en raison de l'entrée de nouveaux ordres ou d'ordres décommandés,
- en raison d'une nouvelle entrée de produits en stock.

Mais, nous ne devons pas tenir compte de ces entrées. Ceci signifie que vous ne recevrez pas deux entrée (ordres, entrées en stock). Les actions et l'ensemble d'ordres vous sont toujours donnés dans un état à jour.

Cas 2 : Vous devez tenir compte des entrées de :

- nouveaux ordres
- annulations des ordres
- entrées des produits en stock

Peut-être vous considériez que ce texte est inachevé. Le but de cet exercice est de savoir quelles questions sont soulevées par votre méthode de modélisation.

---

une introduction pratique aisée aux débutants du domaine.