

## Modélisation et spécification – Master 2 Informatique

### TD 4 : STE et Graphes de programmes

**Exercice 1 :**

*Des voitures et un pont*

On considère un système avec deux voitures, un pont et un contrôleur de pont unidirectionnel. Une voiture arrive par la gauche du pont et l'autre par la droite. Le pont ne laisse passer à chaque instant que des voitures venant de la gauche ou que des voitures venant de la droite. Le contrôleur peut changer la direction du pont si il n'y a pas de voitures dessus. On considérera l'ensemble des messages suivants pour notre modélisation :

- EG : la voiture gauche entre sur le pont
- QG : la voiture gauche quitte le pont
- ED : la voiture droite entre sur le pont
- QD : la voiture droite quitte le pont
- D : le pont signale qu'il laisse passer les voitures venant de la droite
- G : le pont signale qu'il laisse passer les voitures venant de la gauche
- C : le contrôleur change la direction du pont

On souhaite donner un modèle pour ce système en utilisant les STE (et leurs opérateurs de composition). Afin de vous aider voilà comment le comportement du pont, dont on suppose au début qu'il laisse passer les voitures venant de droite, pourrait être représenté.

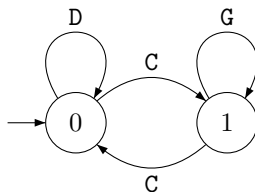


FIGURE 1 – STE  $S_P$  représentant le comportement du pont

1. Modélisez avec le STE  $S_D$  le comportement de la voiture venant de droite qui suit les règles suivantes, si la direction du pont est droite, elle entre sur le pont, puis quitte le pont.
2. À partir du STE précédent, donner le STE  $S_G$  de la voiture venant de gauche.
3. Donnez le STE  $S_C$  du contrôleur qui suit les règles suivantes : à tout moment si il n'y a pas de voiture sur le pont, il peut changer la direction du pont, si il y a une voiture sur le pont il doit attendre que celle-ci quitte le pont pour changer la direction du pont.
4. Donnez la définition du système entier  $S$  (en utilisant la synchronisation de systèmes avec les alphabets de synchronisation adéquates) et dessiner-le.
5. Donnez une formule LTL qui spécifie qu'on n'a jamais sur le pont une voiture droite et une voiture gauche. Votre système vérifie-t-il cette formule ?

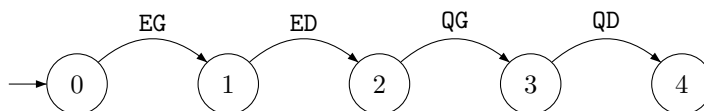


FIGURE 2 – STE  $S_3$

**Exercice 2 :**

*Un algorithme simple*

boolean D1:= false , D2 = false; // variables partagées

<pre> — Processus P while(true){   section non-critique;   D1:=true;   while(D2==true){}   section critique;   D1:=false; } </pre>	<pre> — Processus Q while(true){   section non-critique;   D2:=true;   while(D1==true){}   section critique;   D2:=false; } </pre>
--	--

1. Proposez des STE  $S_{D1}$  et  $S_{D2}$  pour modéliser les variables D1 et D2.
2. Proposez un STE  $S_P$  pour modéliser le comportement du processus  $P$ .
3. À partir du STE précédent, proposer un STE pour modéliser le comportement du processus  $Q$ .
4. Donnez la définition du STE caractérisant tout le système.
5. Est-il possible que deux processus se retrouvent en même instant en section critique? Justifiez votre réponse.
6. Quel est le problème que pose cet algorithme d'exclusion mutuelle? Donner une exécution du STE permettant d'arriver à cette situation problématique.

**Exercice 3 :**

*Algorithme de Dekker*

Soit l'algorithme d'exclusion mutuelle suivant :

```

bit wantP1 = 0, wantP2 = 0;
int turn = 1; // 1 pour P1, 2 pour P2

```

<pre> process P1() {   while (true) {     non_critique_1;     wantP1 = 1;     while (wantP2 == 1) {       if (turn == 2) {         wantP1 = 0;         while (turn != 1);         wantP1 = 1;       }     }     critique_1;     turn = 2;     wantP1 = 0;   } } </pre>	<pre> process P2() {   while (true) {     non_critique_2;     wantP2 = 1;     while (wantP1 == 1) {       if (turn == 1) {         wantP2 = 0;         while (turn != 2);         wantP2 = 1;       }     }     critique_2;     turn = 1;     wantP2 = 0;   } } </pre>
--	--

1. Comment peut-on modéliser les variables globales de cet algorithme en utilisant les systèmes de transitions? Indication : partez des leurs valeurs possibles et des opérations qui sont effectuées sur ces variables.
2. Donner un modèle de chaque processus de l'algorithme.
3. En utilisant les modèles déjà écrits, donner un modèle complet de cet algorithme.
4. Proposez maintenant un modèle de l'algorithme en utilisant des graphes de programmes.
5. Quelle est la différence entre le système de transitions étiquetés obtenu à partir du graphe de programmes de la question précédente et celui construit à la question 3.?
6. Exprimer à l'aide d'une formule LTL une propriété décrivant l'exclusion mutuelle. Cette formule est-elle satisfaite par votre système? Justifiez votre réponse.