

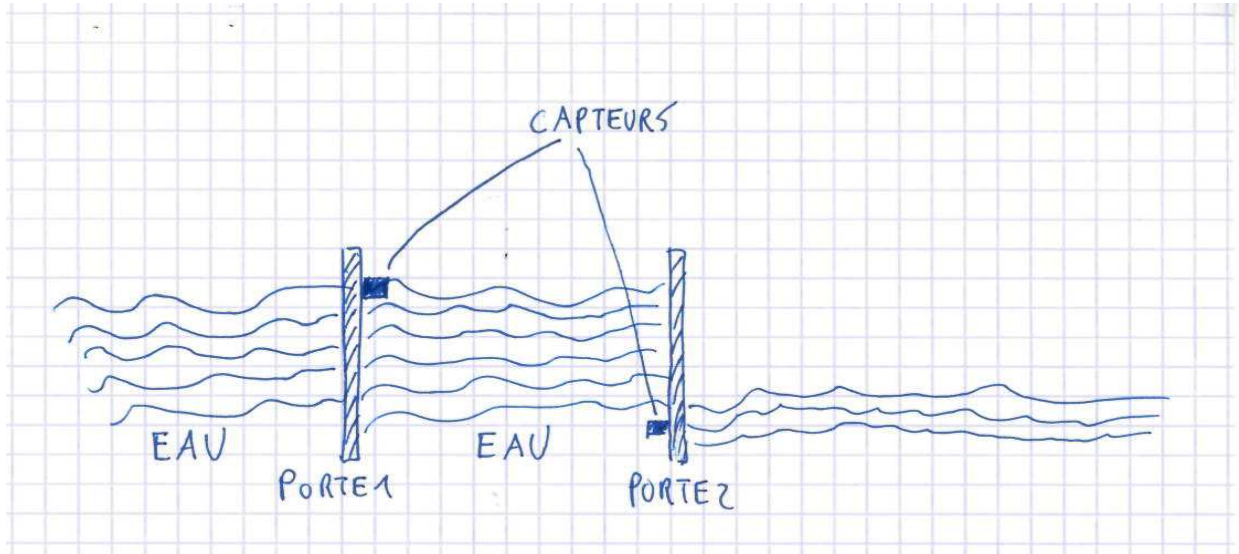
Modélisation et spécification – Master 2 Informatique

TD : Systèmes de transitions

Une écluse

Exercice 1 :

Modélisation d'une écluse



On souhaite modéliser le comportement d'une écluse pour réguler le niveau d'eau sur un canal. L'écluse est composée de deux portes (P1 et P2) et d'un système pour évacuer l'eau et mesurer le niveau des eaux. Les portes sont soit ouvertes, soit fermées et on peut en commander l'ouverture ou la fermeture. Au début la porte 1 est ouverte et la porte 2 est fermée. En ce qui concerne, l'écluse on peut soit la remplir d'eau ou la vider et on peut tester grâce à deux capteurs, si le niveau d'eau est en bas (quand l'écluse a été vidée) ou en haut (quand l'écluse est pleine). L'écluse peut donc être dans quatre états possibles, soit elle est pleine, soit elle est vide, soit elle est en train de se vider, soit elle est en train de se remplir et pour passer de l'état en train de se vider à vide il s'agit d'une action interne de l'écluse, de même pour passer de l'état en train de se remplir à pleine. On a de plus un contrôleur qui en boucle fait les actions suivantes : ferme la porte 1, commande que l'écluse se vide, attend que l'écluse se soit vidée, ouvre la porte 2, ferme la porte 2, commande que l'écluse se remplisse, attend que l'écluse se soit remplie, ouvre la porte 1.

Nous allons modéliser le système global de l'écluse par un ST. Pour cela il vous est demandé de choisir vous même les actions et les propositions atomiques qui font sens.

1. Proposez une modélisation par Système de Transitions (ST) des deux portes par deux ST.
2. Proposez une modélisation par ST de l'écluse (sans les portes).
3. Proposez une modélisation par ST du contrôleur.
4. Donnez la table de synchronisation du système global comprenant l'écluse, les deux portes et le contrôleur.
5. Donnez la formule LTL qui permet de vérifier que la porte 1 est ouverte si et seulement si le niveau de l'eau est haut et que la porte 2 est ouverte si et seulement si le niveau de l'eau est bas.

Boîte avec accusé de réception

Dans ces exercices on vous demande de modéliser une boîte à une place que l'on peut remplir et vider et qui dispose également de deux boutons un pour envoyer un accusé de réception lorsque l'on a vidé la boîte et un autre pour signaler que l'on a reçu l'accusé de réception (à pousser avant de la remplir de nouveau). Plus précisément, en plus des actions **remplir** et **vider**, ce tampon comporte :

- une action **envoie-acc** pour l'envoi de l'accusé de réception
- une action **recoit-acc** pour la réception de l'accusé de réception.

Exercice 2 :

Premier essai

Une implémentation possible de ce type de boîte est rempli puis vider, puis attend l'envoi de l'accusé de réception et ensuite la réception de cet accusé.

1. Donner une modélisation de cette implémentation.
2. On compose en parallèle 2 copies du modèle obtenu tel que les actions **vider** et **recoit-acc** de la première boîte communiquent avec les actions **remplir** respectivement **envoie-acc** du voisin de droite. Pensez-vous que le modèle obtenu corresponde à une boîte avec deux places avec accusé de réception ?

Exercice 3 :

Deuxième essai

Une autre implémentation attend la réception de l'accusé de réception avec **recoit-acc** immédiatement après avoir été rempli, ensuite elle attend d'être vidée et attend son accusé de réception avec **envoie-acc**.

1. Donner une modélisation de cette implémentation.
2. On compose en parallèle 3 copies du modèle obtenu selon les indications de l'exercice précédent. Qu'en pensez-vous par rapport au modèle de l'exercice précédent ?

Exclusion mutuelle

Exercice 4 :

Manipulation de verrous

On considère un système fait de deux processus $P1$ et $P2$ et deux verrous $V1$ et $V2$ de type mutex.

Chaque verrou a deux états, soit pris, soit libre.

Le processus $P1$ réalise au sein d'une boucle infinie répétitive les actions suivantes : il teste si le verrou $V1$ est libre, si c'est le cas, il le prend, puis il prend le verrou $V2$, il réalise alors sa tâche et ensuite il libère $V2$ puis $V1$; si en revanche le verrou $V1$ n'est pas libre, alors il prend le verrou $V2$ puis le verrou $V1$, il réalise sa tâche puis il libère $V2$ et $V1$. (On rappelle que lorsque l'on dit que $P1$ prend le verrou $V2$ cela veut dire qu'il le prend si il est libre, et que sinon il attend que le verrou soit libre et lui soit donné).

Le processus $P2$ réalise quant à lui au sein d'une boucle infinie répétitive les actions suivantes : il prend le verrou $V2$ puis il prend le verrou $V1$, il réalise sa tâche et ensuite il libère $V1$ puis $V2$.

On souhaite modéliser ce système par un système de transitions. Pour cela on considèrera l'alphabet suivant pour les actions :

- $S-i$ (pour i valant 1 et 2) : prend le verrou V_i
- $V-i$ (pour i valant 1 et 2) : libère le verrou V_i
- $L-i$ (pour i valant 1 et 2) : teste positif si le verrou V_i est libre
- $P-i$ (pour i valant 1 et 2) : teste positif si le verrou V_i est pris
- $T-i$ (pour i valant 1 et 2) : P_i fait sa tâche

On supposera de plus que dans l'état initial les deux verrous sont libres.

1. Donner une modélisation sous forme de système de transitions du verrou $V1$.
2. Déduire une modélisation du verrou $V2$.
3. Donner une modélisation du processus $P1$.
4. Donner une modélisation du processus $P2$.
5. En déduire une modélisation pour tout le système.
6. Existe-t-il une trace dans le système obtenu amenant à un état sans successeur (c'est-à-dire un *deadlock*) ?

Exercice 5 :

Un algorithme simple

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

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

1. Proposer des ST S_{D1} et S_{D2} pour modéliser les variables D1 et D2.
2. Proposez un ST S_P pour modéliser le comportement du processus P.
3. À partir du ST précédent, proposer un ST pour modéliser le comportement du processus Q.
4. Donnez la définition du ST 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 ST permettant d'arriver à cette situation problématique.