Modélisation et spécification – Master 2 Informatique
TD 1 : Systèmes de transitions étiquetés, (bi)simulation

Ta pon avec acquitte ent

En cours, vous avez modélisé et spécifié l'exemple d'un tampon à une place qui reçoit des données par un *port* étiqueté par `put` et qui envoie ces données par un port étiqueté `get`. Dans cet exercice on vous demande de modéliser un tampon à une place qui, pour chaque donnée reçue, il envoie un acquittement à l'émetteur. Plus précisément, en plus des ports `get` et `put`, ce tampon comporte :

- **un port `ackput` sur lequel il reçoit l'acquittement et**
- **un port `ackget` sur lequel il envoie l'acquittement reçu.**

Exercice 1 : *Premier essai*

Une implementation possible de ce type de tampon reçoit et envoie la donnée, puis reçoit et envoie l'acquittement.

1. **Donner une modélisation de cette implementation.**
2. **Comparer son comportement avec celui du tampon à une place simple en utilisant les di□érentes relations étudiées en cours : inclusion de traces, simulation, etc.**
3. **Refaire le point précédent si les ports `ackput` et `ackget` sont renommés vers l'action interne τ.**
4. **On compose en parallèle n copies du modèle obtenu tel que les ports `get` et `ackput` communiquent avec les ports `put` respectivement `ackget` du voisin de gauche.**
   Donner une trace du système ainsi obtenu.
   Est-ce une trace d'un tampon à n places avec acquittement ? (Un tel tampon acquitte chaque donnée reçue.)

Exercice 2 : *Deuxième essai*

Une autre implémentation du tampon avec acquittement envoie l'acquittement sur `ackget` immédiatement après avoir reçu la donnée sur le port `put`, ensuite il renvoie la donnée sur `get` et attend son acquittement sur `ackput`.

1. **Donner une modélisation de cette implémentation.**
2. **Comparer son comportement avec celui du tampon à une place simple en utilisant les di□érentes relations étudiées en cours quand les ports `ackput` et `ackget` sont renommés vers τ.**
3. **On compose en parallèle n copies du modèle obtenu selon les indications de l'exercice précédent.**
   Donner une trace du système ainsi obtenu.
   Est-ce une trace d'un tampon à n places avec acquittement ?

n autre exe pl e

Exercice 3 : *L'atelier de Milner*

Deux personnes partagent deux outils — un marteau et une tenaille — pour fabriquer des objets en partant des composantes simples. Chaque objet est manufacturé en introduisant un bâton dans un block. On appelle la paire bâton et block une tache ; les taches arrivent en séquence sur un tapis roulant d'entrée et, après leur fabrication, les objets sont déposés sur un autre tapis roulant, celui de départ.

Une personne répète les actions suivantes : il prend une tache sur le tapis et en fonction de sa di□culté il utilise aucun outil (si tache facile), le marteau (si tache di□cile), ou un des outils disponible (sinon) ; puis il dépose l'objet sur le tapis de départ.

Un outil peut être pris ou déposé.

1. **Donner un modèle STE pour une personne.**

2. **Donner un modèle STE pour chaque outil. Essayer de réutiliser au maximum vos modèles.**
3. **Donner un modèle complet de l'atelier.**
4. **Si on observe que les actions de prise d'une tache et de dépôt d'un objet, donner une spécification de cet atelier.**
5. **Donner le modèle STE d'une personne très forte qui est capable d'e□ectuer sa tache sans utiliser les outils.**
6. **De point de vue de l'extérieur, la personne forte est-elle "équivalente" à la personne qui utilise les outils ? Préciser la relation d'équivalence que vous avez utilisée.**

Al gorith e d,exd usion utud l e

Exercice 4 : *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 }
19
20 process P2() {
21   while (true) {
22     non_critique_2;
23     wantP2 = 1;
24     while (wantP1 == 1) {
25       if (turn == 1) {
26         wantP2 = 0;
27         while (turn != 2);
28         wantP2 = 1;
29       }
30     }
31     critique_2;
32     turn = 1;
33     wantP2 = 0;
34   }
35 }
```

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 e□ectuées 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. **En considérant que les seules actions observables sont non_critique_i et critique_i ($i \in 1..2$), est-ce qu'on peut donner une spécification sous forme de STE du comportement correcte de cet algorithme ?**
5. **Quelle est la relation qui doit être utilisée pour comparer le STE modèle avec le STE spécification ?**