

Master Ingénierie Informatique  
Université Paris 7 - UFR Informatique

Modélisation et Spécification : Examen

14 Décembre 2009

Durée : 2 heures

NB : Seuls les documents manuscrits sont autorisés. Toute réponse doit être justifiée.

Exercice 1

Question 1 Exprimer en CTL les propriétés suivantes :

1. A chaque fois que  $P$  est vraie, il est inévitable que  $Q$  soit vraie dans le futur
- 2. A chaque fois que  $P$  est vraie, il est possible que  $Q$  soit vraie dans le futur
- 3. Il est possible que  $P$  soit vraie dans le futur, et qu'à partir de cet instant  $Q$  devienne inévitable.

Dire quelles sont les propriétés parmi les trois ci-dessus qui peuvent être exprimée en LTL (donner son expression en LTL quand cela est possible).

Question 2 Les équivalences ci-dessous entre formules LTL sont-elles vraies : *JUSTIFIER*

1.  $(\diamond P) \wedge (\diamond Q) \equiv \diamond(P \wedge \diamond Q)$
2.  $(\diamond P) \vee (\diamond Q) \equiv \diamond(P \vee Q)$
3.  $(\square P) \vee (\square Q) \equiv \square(P \vee Q)$
4.  $(\square P) \wedge (\square Q) \equiv \square(P \wedge Q)$
5.  $(\bigcirc P) \wedge (\diamond P) \equiv \bigcirc \diamond P$
6.  $(\bigcirc P) \wedge (\diamond P) \equiv \diamond P$
7.  $(\square \diamond P) \wedge (\square \diamond Q) \equiv \square \diamond(P \wedge Q)$
8.  $(\diamond \square P) \wedge (\diamond \square Q) \equiv \diamond \square(P \wedge Q)$

Exercice 2

Soit le système à un compteur (variable entière) ayant les transitions suivantes :

$$t1 : x \leq 8 : x := x + 6$$

$$t2 : 2 \leq x : x := x - 3$$

On veut montrer que, partant de la valeur  $x = 0$ , la valeur de  $x$  reste toujours dans l'intervalle  $[0, 12]$ , c'est-à-dire, que le système satisfait la propriété :

$$x = 0 \Rightarrow \square(0 \leq x \leq 12) \quad (1)$$

*faire  
calcul*

**Question 1 :** Soit  $\phi$  la formule  $0 \leq x \leq 12$ . Calculer  $\widetilde{pre}(\phi) = \neg(pre(\neg\phi))$  avec la méthode vue en cours. A-t-on  $\phi \Rightarrow \widetilde{pre}(\phi)$  ?

**Question 2 :** Soit alors  $\phi_1 = \phi \cap \widetilde{pre}(\phi)$ . A-t-on alors  $\phi_1 \Rightarrow \widetilde{pre}(\phi_1)$  ? Sinon, proposer une formule  $\phi'$  telle que  $\phi' \Rightarrow \widetilde{pre}(\phi')$  et  $\phi' \Rightarrow \phi$ .

*argumenter  
calcul pas obligatoire*

**Exercice 3** *Chaque peut demander les ressources dans l'ordre qui lui plaît  $\rightarrow$  non bloq.*

On considère deux processus (identiques)  $P_1$  et  $P_2$  qui utilisent durant leurs exécutions une ressource partagée  $R$ . Le comportement de chacun des processus est le suivant : Initialement le processus  $P_i$  est dans l'état *Attente<sub>i</sub>*. Il peut quitter cet état en demandant l'acquisition de  $R$ . Après avoir obtenu la ressource, le processus passe à l'état *Utilisation<sub>i</sub>*. Il peut ensuite quitter cet état en libérant la ressource  $R$  et il se remet alors dans l'état *Attente<sub>i</sub>*. L'utilisation de la ressource  $R$  par les deux processus  $P_1$  et  $P_2$  doit se faire en exclusion mutuelle. (c'est-à-dire que les deux processus ne peuvent utiliser en même temps la ressource).

**Question 1 :** Modéliser par un réseau de Petri avec priorités le système constitué des deux processus  $P_1$  et  $P_2$  : Modéliser chacun des processus et donner un ordre (partiel) de priorité entre les transitions du réseau garantissant l'exclusion mutuelle (c'est-à-dire le fait que les deux processus ne peuvent être simultanément dans *Utilisation<sub>1</sub>* et dans *Utilisation<sub>2</sub>*).

On suppose maintenant que les processus utilisent deux ressources  $R_1$  et  $R_2$ , et que maintenant le comportement de chacun des processus est le suivant : Le processus  $P_i$  quitte l'état *Attente<sub>i</sub>* en demandant l'acquisition des deux ressources dans un ordre *quelconque*, c'est-à-dire, soit en demandant  $R_1$  puis  $R_2$ , soit  $R_2$  puis  $R_1$ . Après avoir obtenu les deux ressources, le processus passe à l'état *Utilisation<sub>i</sub>*. Il peut ensuite quitter cet état en libérant les deux ressources encore une fois dans un ordre *quelconque*. Après avoir libéré les deux ressources, il se remet dans l'état *Attente<sub>i</sub>*.

**Question 2 :** Modéliser à l'aide d'un réseau de Petri (sans priorités) le système constitué des deux processus et des deux ressources : Modéliser chacun des processus et chacune des ressources, et les mettre en parallèle de manière à assurer l'exclusion mutuelle (c'est-à-dire, comme dans la question 1, le fait que les deux processus ne peuvent être simultanément dans *Utilisation<sub>1</sub>* et dans *Utilisation<sub>2</sub>*). Montrer que le système peut se bloquer, c'est-à-dire atteindre un état à partir duquel aucune action n'est possible.

**Question 3 :** Proposer une modélisation du système par un réseau de Petri avec priorités dans laquelle il n'y a pas de blocage : Modéliser chacun des processus et proposer un ordre partiel de priorité entre transitions permettant d'éviter le blocage (et garantissant aussi l'exclusion mutuelle).