Contrôle continu – Modélisation et spécification
Master Informatique

12 novembre 2018

Durée : 1h50.

Documents autorisés : Une feuille A4 manuscrite recto-verso.
Rendez deux copies. Une avec les exerices 1 et 2 et l'autre avec les exercices 3 et 4.

<u>Exercice</u> 1 : (3 points) *Analyse de systèmes de transitions*

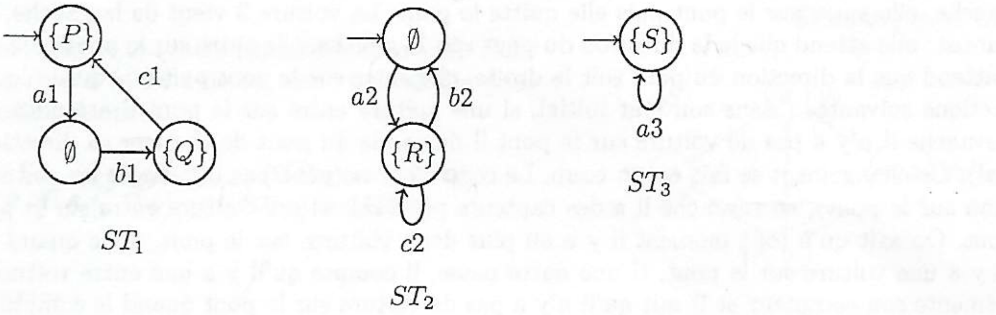


FIGURE 1 – Systèmes de transitions étiquetés ST_1, ST_2 et ST_3

1. On considère les systèmes de transitions ST_1 ,ST_2 et ST_3 donnés à la Figure 1. Dessinez le système de transitions $(ST_1||ST_2||ST_3)_T$ où T est la table de synchronisation suivante (Act_1 sont les actions de ST_1, Act_2 les actions de ST_2 et Act_3 les actions de ST_3) :

Act_1	Act_2	Act_3	
$a1$	$a2$	$-$	a
$-$	$c2$	$-$	c
$b1$	$-$	$a3$	b
$c1$	$b2$	$a3$	d

2. On considère les systèmes de transitions ST_3, ST_4 et ST_5 de la Figure 2. Donnez la table de synchronisation T' telle que $(ST_3||ST_4)_{T'}$ est égal à ST_5.

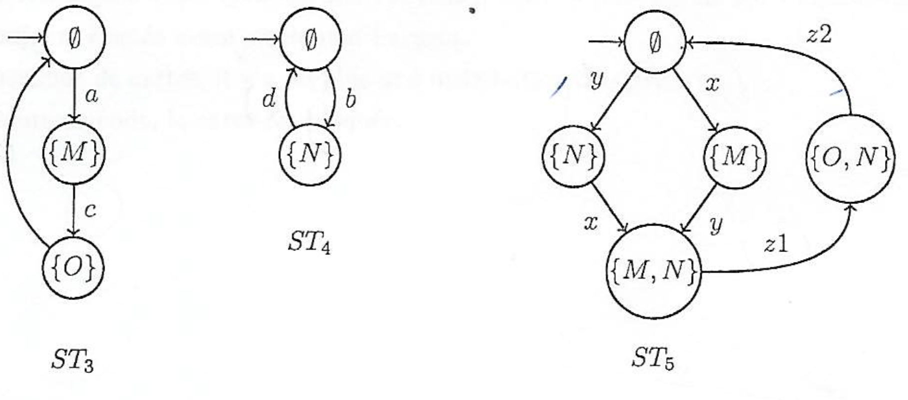


FIGURE 2 – Systèmes de transitions étiquetés ST_3, ST_4 et ST_5

<u>Exercice</u> 2 : (7 points) *Des voitures et un pont*

On considère un système avec des voitures, un pont et un contrôleur de pont unidirectionnel. Les voitures arrivent soit par la gauche du pont soit 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 voiture dessus. Nous voulons observer ici les événements décrits par les propositions atomiques suivantes :

— `V1D` : la voiture 1 est sur le pont et vient de la droite
— `V1G` : la voiture 1 est sur le pont et vient de la gauche
— `V2D` : la voiture 2 est sur le pont et vient de la droite
— `V2G` : la voiture 2 est sur le pont et vient de la gauche
— `D` : le pont laisse passer les voitures venant de la droite
— `G` : le pont laisse passer les voitures venant de la gauche

Le but de cet exercice est de proposer un modèle pour ce système en utilisant les systèmes de transitions et leur synchronisation. Nous allons donc donner une description informelle des différentes entités composant le système. Tout d'abord le pont, il laisse passer soit les voitures venant de droite, soit les voitures venant de gauche (mais pas les deux en même temps). Au début, il laisse passer les voitures venant de gauche. Si le contrôleur lui demande de changer de direction, il change de direction. Les voitures peuvent aussi vérifier la direction du pont. On aura également deux voitures. La voiture 1 vient de la droite, et fait en boucle les actions suivantes : elle attend que le la direction du pont soit la droite, elle entre sur le pont puis elle quitte le pont, et ensuite elle attend que la direction du pont soit la gauche, elle entre sur le pont puis elle quitte le pont. La voiture 2 vient de la gauche, et fait en boucle les actions suivantes : elle attend que le la direction du pont soit la gauche, elle entre sur le pont puis elle quitte le pont, et ensuite elle attend que la direction du pont soit la droite, elle entre sur le pont puis elle quitte le pont. Le contrôleur, lui, fait les actions suivantes : dans son état initial, si une voiture entre sur le pont il attendra que la voiture soit sortie, si en revanche il n'y a pas de voiture sur le pont il demande au pont de changer sa direction (et revient dans son état initial). Ce changement se fait en un coup. Le contrôleur ne peut pas interroger les voitures (pour savoir si elles sont ou non sur le pont), en revanche il a des capteurs pour voir si une voiture entre sur le pont ou si une voiture quitte le pont. On sait qu'à tout moment il y a au plus deux voitures sur le pont, donc quand une voiture passe il compte qu'il y a une voiture sur le pont, si une autre passe, il compte qu'il y a une autre voiture et quand la voiture sort il décrémente son compteur et il sait qu'il n'y a pas de voiture sur le pont quand le compteur vaut 0. Au début il n'y a bien entendu pas de voiture sur le pont.

1. Modélisez avec le système de transitions S_P le comportement du pont.
2. Modélisez avec les systèmes de transitions S_{V1} et S_{V2} le comportement des voitures 1 et 2.
3. Donnez le système de transitions S_C du contrôleur
4. Donnez la table de synchronisation du système global.
5. Exprimez en LTL (en utilisant les propositions atomiques fournies ci-dessus) :
   (a) Jamais la voiture 1 et la voiture 2 sont sur le pont en même temps en venant de deux directions opposées
   (b) À chaque fois que la voiture 1 est sur le pont, la direction du pont correspond à la direction d'où elle vient
   (c) Les deux voitures ne sont jamais sur le pont en même temps
6. Votre système vérifie-t-il la dernière propriété ? Justifez votre réponse.

Rappel : Syntaxe et sémantique de LTL. On considère des formules de LTL (sans passé) dont la syntaxe est donnée par la grammaire :

$$\phi ::= P \mid \phi \vee \psi \mid \phi \wedge \psi \mid \neg\phi \mid \mathbf{X}\phi \mid \phi\,\mathbf{U}\,\psi$$

où P appartient à un ensemble de propositions atomiques AP. On utilise aussi les abréviations $\mathbf{F}$ (pour $\top\,\mathbf{U}\,\_$), $\mathbf{G}$ (pour $\neg\mathbf{F}\neg$) et $\mathbf{W}$ (défini par : $a\,\mathbf{W}\,b = (a\,\mathbf{U}\,b) \vee (\mathbf{G}a)$).

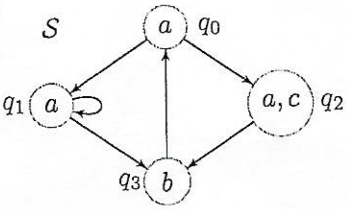
Ces formules s'interprètent sur une exécution ρ d'un STE : $\rho(i)$ est le i-ème état, ρ^i est le i-ème suffixe (commençant en $\rho(i)$) et $\ell(\rho(i))$ est l'ensemble des propositions vraies en $\rho(i)$.
La sémantique est notamment définie par :

1. $\rho \models P$ ssi $P \in \ell(\rho(0))$,
2. $\rho \models \mathbf{X}\phi$ ssi $\rho^1 \models \phi$
3. $\rho \models \phi\,\mathbf{U}\,\psi$ ssi $\exists j \geq 0$ t.q. $(\rho^j \models \psi) \wedge (\forall k \in [0, j[$, on a : $\rho^k \models \phi$)

Sémantique de LTL

Exercice 3 : (4 points)

1. On considère le STE S de la figure ci-contre (q_0 est l'état initial). Pour chacune des formules suivantes, dire si la formule est vraie pour S (*ie* pour toutes ses exécutions). Justifier les réponses.



$a\,\mathbf{U}\,b \qquad \neg(a\,\mathbf{U}\,b) \qquad \mathbf{G}\,\mathbf{F}\,b \qquad \mathbf{G}(a \vee b) \qquad (\mathbf{GF}\,b) \Rightarrow (\mathbf{GF}\,c)$

$(\mathbf{FG}\neg a) \Rightarrow (a\,\mathbf{U}\,b) \qquad (\mathbf{F}\,c) \Rightarrow (a\,\mathbf{U}\,b) \qquad (\mathbf{GF}\,c) \Rightarrow (\mathbf{GF}\,b)$

2. Comparer les formules ci-dessous :

$(\mathbf{FF}\,a) \stackrel{?}{\equiv} (\mathbf{F}\,a) \qquad (\mathbf{F}\,(a \wedge b)) \stackrel{?}{\equiv} ((\mathbf{F}\,a) \wedge (\mathbf{F}\,b)) \qquad (\mathbf{F}(a \wedge \mathbf{X}b)) \stackrel{?}{\equiv} (\mathbf{F}(a \wedge \mathbf{F}b)) \qquad (\mathbf{GF}(a \vee b)) \stackrel{?}{\equiv} ((\mathbf{GF}\,a) \vee (\mathbf{GF}\,b))$

Ecrire des formules LTL

Exercice 4 : (6 points)

On souhaite écrire des propriétés pour un distributeur de billets en logique temporelle LTL (avec ou sans passé). On introduit les propositions atomiques nécessaires au fur et à mesure. On y ajoute la proposition la proposition `CIn` qui caractérise les états où on vient d'insérer une nouvelle carte dans la machine (donc le début d'une session).

1. Après l'insertion d'une carte, le code sera demandé (proposition `Code?`).
2. Après une demande de code, on arrivera dans un état où le code est accepté (proposition `CodeOK`), ou alors dans un état où le code est refusé (proposition `CodeKO`).
3. Lorsque la carte est éjectée (proposition `COut`), un bip est émis (proposition `Bip`).
4. Si le bouton annulation est utilisé (proposition `Cancel`), alors soit la carte sera éjectée , soit la carte sera bloquée (proposition `CRetain`).
5. Si la distribution d'argent a lieu (proposition `Distrib`), alors le code donné par l'utilisateur a été accepté.
6. La carte est toujours éjectée avant de donner l'argent.
7. Entre deux insertions de cartes, il y a au plus une distribution d'argent.
8. Après trois erreurs de code, la carte est bloquée.