

Examen – Modélisation et spécification Master Informatique

15 Décembre 2015

Durée : 2h30.

Documents autorisés : Une feuille A4 manuscrite recto-verso.

Rédaction : Il faut rendre deux copies : Les exercices 1, 2, 3 et 4 sont à rendre ensemble sur une même copie, et les exercices 5 et 6 sont à rendre sur une autre copie.

Exercice 1 :

Analyse de réseaux de Petri [2 points]

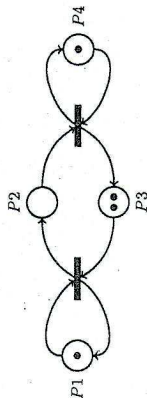


FIGURE 1 – Réseau de Petri RP_1

On considère le réseau de Petri RP_1 dessiné à la Figure 1.

1. Dessinez le graphe de marquages accessibles de ce réseau de Petri (que nous avons aussi appelé $S(RP_1)$ en cours).
2. Modifiez ce réseau de Petri pour que la transition prenant des jetons dans la place P_3 ne puisse en prendre que si la place P_2 est vide (Indice : Combien a-t-on au plus de jetons dans P_2 ?)

```
integer V1 := 0 // variables partagées
integer V2 := 0

-- Processus P1
loop forever :
p1 : /*section NC*/
p2 : V1 := 1;
p3 : if (V2!=0) then
p4 : goto p2;
p5 : else V2:=1;
p6 : if (V1!=1) then
p7 : if (V2!=1) then
p8 : goto p2;
p9 : /*section critique */
p10: V2:=0;
```

FIGURE 2

Exercice 2 :

On considère l'algorithme donnée par la Figure 2 pour deux processus P_1 et P_2 avec variables partagées.

1. Donnez une caractérisation du processus P_1 par un graphe de programmes PG_1 et donnez une caractérisation du processus P_2 par un graphe de programmes PG_2 .
2. Quelle est la formule LTL caractérisant l'exclusion mutuelle, i.e. le fait qu'il n'y a pas deux processus en même temps dans la section critique.
3. Le système de transitions étiqueté $TS(PG_1||PG_2)$ ne satisfait pas la propriété d'exclusion mutuelle. Donnez une exécution de $TS(PG_1||PG_2)$ pour laquelle P_1 et P_2 se retrouvent tous les deux en section critique.

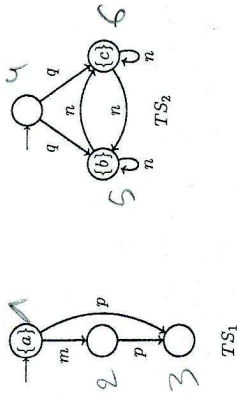


FIGURE 3 – Systèmes de transitions étiquetés TS_1 et TS_2

Exercice 3 :

On considère les systèmes de transitions étiquetés TS_1 et TS_2 donnés à la Figure 3 et la table de synchronisation T suivante (où Act_1 sont les actions de TS_1 et Act_2 les actions de TS_2) :

Act_1	Act_2
m	-
-	n
p	q
-	r

1. Dessinez le système de transitions étiqueté $(TS_1||TS_2)/r$.

2. Dites si les formules de LTL suivantes sont satisfaites ou non par le système $(TS_1||TS_2)/r$:

(a) $c \cup a$ (b) $FG(b \vee c)$ (c) $G(X(b \vee c))$ (d) $G(b \Rightarrow Xc)$

3. Dites si les formules de CTL suivantes sont satisfaites ou non par le système $(TS_1||TS_2)/r$:

(a) $AX(b \vee c)$ (b) $EX(EXc)$ (c) AFb (d) $EG(EXc)$

Exercice 4 :

Le père Noël dort dans son usine au Pôle Nord et il ne peut être réveillé que lorsque ses 5 elfes sont de retour de leurs vacances passées sur les plages d'une île tropicale dans le Pacifique Sud, ou par l'un des problèmes de fabrication de jouets. Le problème d'un unique elfe n'est jamais assez sérieux pour réveiller le père Noël (qui risquerait dans ce cas de ne jamais se reposer), par conséquent en cas d'ennui les elfes travaillent à trois. Quand le problème de trois elfes a été résolu, si d'autres elfes ont un problème ils doivent attendre de pouvoir former à leur tour un groupe de trois personnes et attendre le retour du père Noël pour que celui-ci résolve leur problème.

On veut représenter ce « système » par un réseau de Petri. On vous donne les informations suivantes. On suppose qu'il y a 9 remes et 5 elfes. Quand le père Noël ne dort pas, soit il fait sa tournée de Noël soit il résout un problème. Un elfe suit le comportement suivant en boucle : il travaille, il signale un problème au père Noël puis il attend que le problème ait été résolu pour retravailler. Un elfe a le comportement suivant en boucle : il est en vacances, il est au pôle nord, il est en tournée. Dans l'état initial, le père Noël dort, les remes sont en vacances et les elfes travaillent. Vous ne devez pas faire une place par reme ou par elfe mais exploiter les comportements similaires.

Exercice 5 :

1. On considère les deux modèles de la figure 4. Peut-on les distinguer (en partant de l'état initial) avec CTL ? Et avec LTL ? Justifiez votre réponse. Pour distinguer on cherche une formule satisfait par un modèle et pas par l'autre.



FIGURE 4 – Systèmes de transitions étiquetés S_1 et S_2

2. Même question avec les deux modèles S'_1 et S'_2 de la Figure 5.

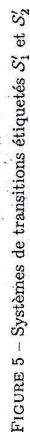


FIGURE 6 - Tableau de bord du lave-linge.

Exercice 6 :

Exercice 6 : *Spécification en logique temporelle*, [6 points]
On s'intéresse ici à la spécification en LTL d'un lave-linge. Le tableau de bord de la machine comporte un voyant lumineux « Fin », un sélecteur de programme avec deux positions « Prog 1 » et « Prog 2 », et deux boutons « start » et « stop ». La figure 6 représente ce tableau de bord.

On considère les propositions atomiques suivantes : V_{fin} est vrai lorsque le voyant « fin » est allumé, $progl$ est vrai lorsque le sélecteur de programme est sur « Prog 1 » (et $progl2$ est vrai lorsqu'il est sur « Prog 2 »). Estat est vrai au moment où le bouton « start » est enfoncé, et Estop est vrai lorsque c'est le bouton « stop » qui est enfoncé.

La machine peut se trouver dans trois modes différents : marche, arrêt, ou pause. A chacun de ces modes on associe une proposition atomique : marche, arrêt, pause. La figure 7 contient trois exemples d'exécutions de la machine qui peuvent être utiles pour la question 5.

1. Ecrivez une formule exprimant que le voyant « Fin » est allumé si la machine est en mode « arrêt ».
2. Ecrivez une formule spécifiant qu'à tout moment un programme (et un seul) est sélectionné.
3. Ecrivez une formule qui spécifie qu'à chaque instant, la machine est dans un et un seul des trois modes.
4. On utilise le bouton « start » pour démarrer la machine lorsqu'elle est à l'arrêt ou en pause. On utilise le bouton « stop » pour faire une pause si la machine est en marche, et pour passer en mode arrêt si elle est en mode pause. Appuyer sur « start » lorsque la machine est en marche ou appuyer sur « stop » lorsqu'elle est à l'arrêt, ne change pas de mode. Ecrivez des formules de LTL pour toutes ces propriétés :
- (a) Lorsqu'on appuie sur le bouton « start » et que la machine est à l'arrêt ou en pause, alors la machine passe en mode « marche » au prochain état.
- (b) Lorsqu'on appuie sur le bouton « stop » et que la machine est en marche, alors la machine passe en mode « pause » au prochain état.
- (c) Lorsqu'on appuie sur le bouton « stop » et que la machine est en pause, alors la machine passe en mode « arrêt » au prochain état.
- (d) Appuyer sur « start » en mode « marche » n'a pas d'effet sur le mode. Idem pour « stop » en mode arrêt.
5. On va, à présent, décrire les cycles de lavage. Pour cela, on va utiliser les propositions atomiques suivantes : pré-lavage, lavage, rincage, essorage.

(a) Spécifiez que pour tout état, au plus une des propositions atomiques pré-lavage, lavage, rinçage et essorage est vraie et que lorsque la machine est en mode « arrêt », aucune de ces propositions n'est vraie.

- b) Le programme de lavage 1 a la forme suivante : lavage, rinçage et essorage, puis se termine en passant en mode arrêt. Ce programme commence après que l'on ait appuyé sur le bouton « start » depuis le mode « arrêt » et lorsque le sélecteur est en position « Prog.1 ». On a alors une séquence d'états correspondant au lavage, puis des états correspondant au rinçage puis des états pour l'essorage et enfin le mode arrêt. Notez qu'à tout moment la séquence peut être interrompue par la mise à l'arrêt de la machine (conformément à la spécification déjà décrite).

Ecrivez une formule qui spécifie le déroulement de ce programme.

- (c) Même question pour le programme de lavage 2 qui a la forme suivante : pré-lavage, lavage, rinçage et essorage.

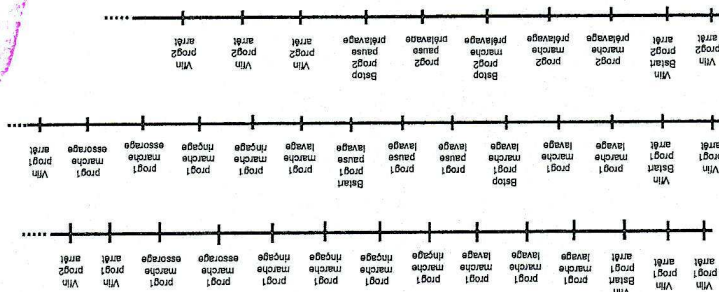


FIGURE 7 – Exemples d'exécutions pour le lave-linge.

Annexe

Sémantique de LTL. Les formules de LTL s'interprètent sur des traces infinies σ , $\sigma(i)$ est l'ensemble des propositions atomiques vraies à l'étape i . La sémantique (hors opération booléenne) est :

- $\sigma \models \phi$ ssi $\sigma, 0 \models \phi$
- $\sigma, i \models P$ ssi $P \in \sigma(i)$
- $\sigma, i \models \mathbf{X}\phi$ ssi $\sigma, i+1 \models \phi$
- $\sigma, i \models \bigcup \psi$ ssi $\exists j \geq i : (\sigma, j \models \psi) \wedge (\forall k \in [i, j] \text{ on a } \sigma, k \models \phi)$

On rappelle aussi les raccourcis : $\mathbf{F}\phi \stackrel{\text{def}}{=} \top \vee \phi$ et $\mathbf{G}\phi \stackrel{\text{def}}{=} \neg \mathbf{F}\neg\phi$.

Sémantique de CTL. Les formules de CTL s'interprètent sur des états de systèmes de transitions étiquetés. Soit

- $q \models P$ ssi $P \in \mathcal{L}(q)$
- $q \models \text{EX } \phi$ ssi $\exists q' \in \text{Post}(q)$ tel que $q' \models \phi$
- $q \models \text{AX } \phi$ ssi $\forall q' \in \text{Post}(q)$, on a $q' \models \phi$
- $q \models \text{EP } U$ ssi $\exists p \in \text{Exec}(q)$ telle que $\exists i \geq 0$ tel que $\rho(i) \models \psi$ et $\forall 0 \leq j < i$, on a ϕ .
- $q \models \text{AP } U$ ssi $\forall p \in \text{Exec}(q)$, $\exists i \geq 0$ tel que $\rho(i) \models \psi$ et $\forall 0 \leq j < i$, on a $\rho(j) \models \phi$.