Examen – Modélisation et spécification
Master Informatique

17 Décembre 2018

Durée : 3h.
Documents autorisés : Une feuille A4 manuscrite recto-verso.

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

Exercice 1 : *Analyse de réseaux de Petri* **[2 points]**

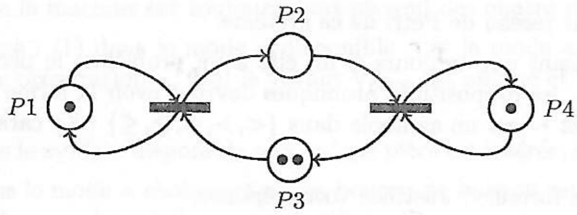


FIGURE 1 – Réseau de Petri RP_1

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 $TS(RP_1)$ en cours).
2. Modifiez en ajoutant des places et/ou des transitions ce réseau de Petri pour que la transition prenant des jetons dans la place $P3$ ne puisse en prendre que si la place $P2$ est vide.

Exercice 2 : *Faire une file* **[3 points]**

On considère les trois systèmes de transitions étiqueté donnés à la Figure 2 qui représentent trois 'tampons' à une place dans chacun desquels l'action put_i sert à insérer une donnée et l'action get_i sert à prendre une donnée. Par exemple, pour le système PI_1 le plus à gauche, l'état avec l'étiquette $\{V1\}$ représente que le tampon est vide et lorsque l'on fait put_1 on met une donnée et le système devient plein (étiquette $\{P1\}$).

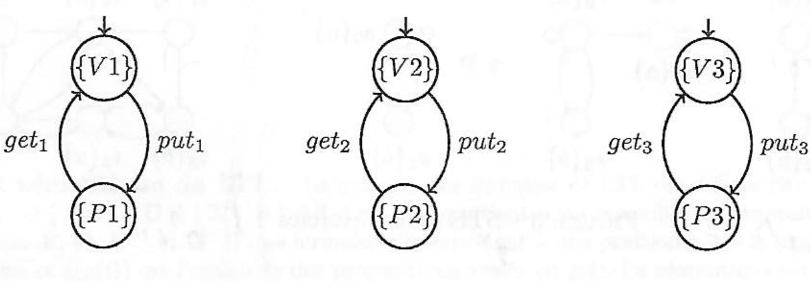


FIGURE 2 – Systèmes de transitions étiquetés PI_1, PI_2 et PI_3

1. On souhaite synchroniser ces trois systèmes de façon à obtenir le comportement d'une file à trois places. Ainsi les actions put_1 et get_3 pourront être faites toute seule de l'extérieure et les données circuleront du tampon 1 vers le tampon 2 puis du tampon 2 vers le tampon 3. Donnez la table de synchronisation T de façon à ce que le système $(PI_1|||PI_2|||PI_3)_T$ se comporte comme une file de capacité 3.
2. Dessinez une partie du système $(PI_1|||PI_2|||PI_3)_T$?
3. Donnez la formule LTL qui permet d'exprimer le fait que toujours si le système 2 est plein et au coup d'après il est vide alors le système 3 est vide est au coup d'après il est plein.

Exercice 3 : *Modélisation par réseaux de Petri* [**5 points**]

Le père Noël dort dans son usine au Pôle Nord et il ne peut être réveillé que lorsque ses neuf rênes sont de retour de leurs vacances passées sur les plages d'une île tropicale dans le Pacifique Sud, ou par certains elfes qui ont 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 réveillent le père Noël par groupe de 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. Le père Noël ne résoud le problème des elfes que si tous les rénes ne sont pas là (c'est-à-dire qu'au moins un rêne est encore en vacances) sinon il part en tournée.

On veut représenter ce « système » par un réseau de Petri. On vous donne les informations suivantes. On supposera qu'il y a 9 rennes 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 cerf 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 rennes sont en vacances et les elfes travaillent. Vous ne devez pas faire une place par renne ou par elfe mais exploiter les comportements similaires.

1. Donnez une représentation par réseau de Petri de ce système.
2. Donnez la formule LTL précisant que toujours si un elfe a un problème, le père Noël finira par le résoudre. **NB :** Pour cette formule LTL, les propositions atomiques devront avoir la forme $\#p \sim a$ où p est le nom d'une place, a est un entier positif et $\sim$ est un symbole dans $\{<, >, =, \geq, \leq\}$ ($\#p$ caractérisant le nombre de jetons dans la place p).
3. Votre système vérifie-t-il cette formule ? Justifiez votre réponse.

Exercice 4 : *Logiques temporelles CTL et LTL* [**4 points**]

Dans cet exercice, on considère les quatre systèmes de transitions étiquetés de la figure 3 : l'état initial de $\mathcal{S}_1$ est q_0, celui de $\mathcal{S}_2$ est r_0, celui de $\mathcal{S}_3$ est s_0 et celui de $\mathcal{S}_3$ est t_0.

Etant donnés un STE $\mathcal{S}$ et son état initial s_0, on dit qu'une formule ϕ de CTL est vraie pour $\mathcal{S}$ (noté $\mathcal{S} \models \phi$) si et seulement si on a $s_0 \models \phi$. Et pour une formule ϕ de LTL, on dit que ϕ est vraie pour $\mathcal{S}$ (noté $\mathcal{S} \models \phi$) si et seulement si on a $\rho \models \phi$ pour toute exécution ρ qui part de s_0. On rappelle aussi qu'une logique L distingue deux modèles $\mathcal{S}$ et $\mathcal{S}'$ si et seulement si il existe une formule ϕ de L telle que $\mathcal{S} \models \phi$ et $\mathcal{S}' \not\models \phi$.

1. Pouvez-vous distinguer $\mathcal{S}_1$ et $\mathcal{S}_2$ avec CTL ? Et avec LTL ?
2. Pouvez-vous distinguer $\mathcal{S}_2$ et $\mathcal{S}_3$ avec CTL ? Et avec LTL ?
3. Pouvez-vous distinguer $\mathcal{S}_3$ et $\mathcal{S}_4$ avec CTL ? Et avec LTL ?
4. En déduire si $\mathcal{S}_1$ et $\mathcal{S}_4$ sont distinguables ou non par CTL et LTL.

Justifier vos réponses.

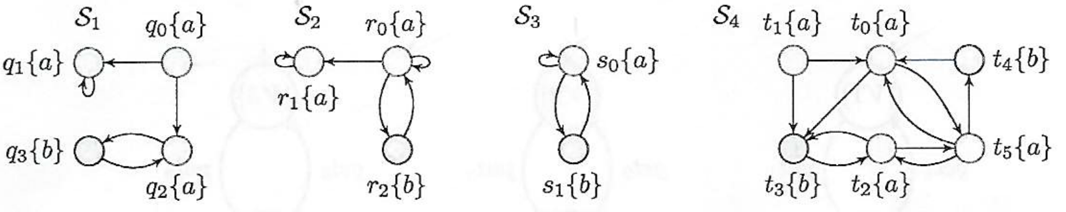


FIGURE 3 – STE pour l'exercice 4.

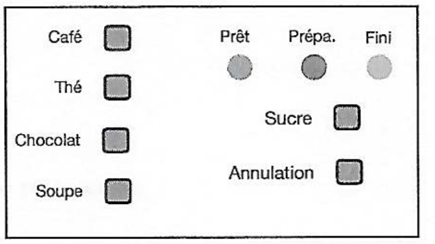
Exercice 5 : *Spécification en logique temporelle* [**6 points**]

On s'intéresse ici à la spécification en LTL d'un distributeur de boisson. Le tableau de bord de la machine (voir le dessin ci-contre) comporte :

- Quatre boutons Café, Thé, Chocolat, et Soupe pour choisir une boisson. Les propositions atomiques[1] associées sont : $\mathsf{B}_{\text{café}}, \mathsf{B}_{\text{thé}}, \mathsf{B}_{\text{choco}}, \mathsf{B}_{\text{soupe}}$.
- Un bouton Annulation. Sa propositions atomique est : B_{an}.
- Un bouton Sucre. Sa propositions atomique est : $\mathsf{B}_{\text{sucre}}$.
- Trois voyants Prêt, Prépa et Fini. Leurs propositions sont : $\mathsf{V}_{\text{prêt}}, \mathsf{V}_{\text{prépa}}$ et V_{fini}.

1. Comme d'habitude, on associe aux boutons des propositions atomiques : la proposition d'un bouton (resp. voyant) est vraie ssi le bouton est enfoncé (resp. le voyant est allumé).

La machine peut se trouver dans quatre modes différents : (1) disponible (proposition M_{dispo}), (2) choix d'une boisson (proposition M_{choix}), (3) en cours de préparation d'une boisson (proposition $M_{\text{prépa}}$), et (4) en fin de transaction (c'est-à-dire en attente de la prise du gobelet par l'utilisateur) (proposition M_{fin}).



Le comportement attendu de la machine est le suivant : lorsqu'un usager souhaite une boisson, il commence par mettre une pièce de 50¢ (la proposition $P_{50¢}$ est alors vraie), puis il choisit sa boisson en appuyant sur le bouton correspondant, puis la machine prépare la boisson et ensuite attend que le gobelet soit retiré pour revenir dans le mode de départ. Entre le moment où la pièce est insérée et le choix de la boisson, l'utilisateur a la possibilité d'utiliser le bouton sucre (B_{sucre}) pour ajouter du sucre. Appuyer sur le bouton <u>Annulation</u> après l'insertion d'une pièce et avant le choix de la boisson permet de revenir en mode disponible et rend la pièce.

Écrire les formules suivantes :

1. Une formule exprimant que la machine est toujours dans un seul des quatre modes.
2. Une formule qui spécifie que : (1) dans le mode « disponible » et le mode « choix », seul le voyant $V_{\text{prêt}}$ est allumé, (2) dans le mode « préparation », seul le voyant $V_{\text{prépa}}$ est allumé et (3) dans le mode « fin », seul le voyant V_{fini} est allumé.
3. Lorsque la machine est dans le mode « disponible » et qu'une pièce est insérée, elle passe dans le mode « choix ».
4. Lorsque la machine est dans le mode « choix » et qu'un bouton de boisson est enfoncé, elle passe dans le mode « prépa », mais si c'est le bouton <u>Annulation</u> qui est utilisé, elle repasse en mode « disponible »et la pièce est rendue (proposition $R_{50¢}$).
5. Lorsque la machine est dans le mode « fini » et que le gobelet est retiré (proposition G_{out}), elle passe en mode « disponible ».
6. Une formule qui spécifie le cycle des modes : le mode « disponible » est toujours suivi (après éventuellement plusieurs étapes...) par le mode « choix », le mode « choix » est toujours suivi par le mode « préparation » ou le mode « disponible », le mode « préparation » est toujours suivi par le mode « fini » et le mode « fini » est toujours suivi par le mode « disponible ». De plus, on spécifiera que le mode « disponible » est rencontré infiniment souvent.
7. Lorsque la machine est en mode « choix », une pression sur le bouton <u>Sucre</u> (proposition B_{sucre}) rend vraie (dans l'état suivant) la proposition $P_{\text{sucré}}$. Une seconde pression rend vraie la proposition $P_{\text{très sucré}}$ et fausse $P_{\text{sucré}}$, et une troisième pression revient dans la situation initiale où les deux propositions $P_{\text{sucré}}$ et $P_{\text{très sucré}}$ sont fausses. Cela permet à l'utilisateur de doser le sucre de trois manières (pas sucré, sucré, très sucré). Ecrire les formules correspondant à ce comportement.

Annexe

Rappel : Syntaxe et sémantique de LTL. La syntaxe des formules de LTL est définie par la grammaire suivante : $\phi ::= P \mid \phi \vee \psi \mid \phi \wedge \psi \mid \neg\phi \mid \mathbf{X}\phi \mid \phi\,\mathbf{U}\,\psi \mid \mathbf{X}^{-1}\phi \mid \phi\,\mathbf{S}\,\psi$ où P appartient à un ensemble de propositions atomiques AP. On utilise aussi les abréviations $\mathbf{F}$, $\mathbf{G}$, $\mathbf{F}^{-1}$ et $\mathbf{G}^{-1}$. Ces formules s'interprètent à une position $i \geq 0$ le long d'une exécution ρ d'un STE : $\rho(i)$ est le i-ème état et $\ell(\rho(i))$ est l'ensemble des propositions vraies en $\rho(i)$. La sémantique est définie par :

- $\rho, i \models P$ ssi $P \in \ell(\rho(i))$
- $\rho, i \models \mathbf{X}\phi$ ssi $\rho, i+1 \models \phi$
- $\rho, i \models \phi\,\mathbf{U}\,\psi$ ssi $\exists j \geq i$ t.q. $(\rho, j \models \psi) \wedge (\forall k \in [i, j[$, on a : $\rho, k \models \phi$)
- $\rho, i \models \mathbf{X}^{-1}\phi$ ssi $i > 0$ et $\rho, i-1 \models \phi$
- $\rho, i \models \phi\,\mathbf{S}\,\psi$ ssi $\exists j \leq i$ t.q.$(\rho, j \models \psi) \wedge (\forall k \in]j, i]$, on a : $\rho, k \models \phi$)

Sémantique de CTL. La syntaxe des formules de CTL est définie par la grammaire suivante : $\phi ::= P \mid \phi \vee \psi \mid \phi \wedge \psi \mid \neg\phi \mid \mathbf{EX}\phi \mid \mathbf{AX}\phi \mid \mathbf{E}\phi\,\mathbf{U}\,\psi \mid \mathbf{A}\phi\,\mathbf{U}\,\psi$. Les formules de CTL s'interprètent sur des états de systèmes de transitions étiquetés. Soit $\mathcal{S} = (Q, \mathsf{AP}, \rightarrow, L)$ et soit $q \in Q$, on a :

- $q \models P$ ssi $P \in L(q)$
- $q \models \mathbf{EX}\phi$ ssi $\exists q \rightarrow q'$ tel que $q' \models \phi$
- $q \models \mathbf{AX}\phi$ ssi $\forall q \rightarrow q'$, on a $q' \models \phi$
- $q \models \mathbf{E}\phi\,\mathbf{U}\,\psi$ ssi $\exists \rho \in \mathsf{Exec}(q)$ telle que $\exists i \geq 0$ tel que $\rho(i) \models \psi$ et $\forall 0 \leq j < i$, on a $\rho(j) \models \phi$.
- $q \models \mathbf{A}\phi\,\mathbf{U}\,\psi$ ssi $\forall \rho \in \mathsf{Exec}(q)$, $\exists i \geq 0$ tel que $\Big(\rho(i) \models \psi$ et $\forall 0 \leq j < i$, on a $\rho(j) \models \phi\Big)$.