

Examen – Modélisation et spécification Master Informatique

18 Décembre 2017

Durée : 2h30.

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 [4 points]

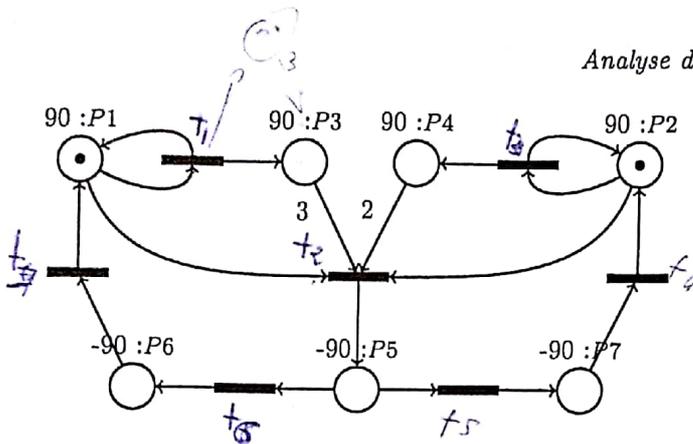


FIGURE 1 – Réseau de Petri RP_1

On considère le réseau de Petri représenté à la Figure 1.

- ✓ 1. Dans ce réseau existe-t-il une exécution permettant d'atteindre le marquage où la seule place avec un jeton est P_5 ? Justifiez votre réponse (sans justification la réponse est considérée comme fausse).
- ✓ 2. Dans ce réseau existe-t-il une exécution permettant d'atteindre le marquage où l'on a exactement un jeton dans P_6 et un dans P_7 et zéro jeton ailleurs? Justifiez votre réponse (sans justification la réponse est considérée comme fausse).
3. Modifiez votre réseau en ajoutant des places ou/et des transitions pour qu'à tout moment il y ait au plus 3 jetons dans P_3 et 2 dans P_4 . Pour ce qui concerne le reste du réseau, le comportement est inchangé.
- ✓ 4. Indiquez parmi les formules LTL fournies ci-dessous, lesquelles sont vérifiées par le système de transitions $ST(RP_1)$. Pour les formules LTL qui ne sont pas vérifiées, donnez une trace du système. (Rappel : $\#P$ représente le nombre de jetons dans la place P).
 - (a) $\neg((\#P_1 > 0) \cup (\#P_6 > 0))$
 - (b) $G((\#P_6 > 0) \Rightarrow (FG(\#P_1 > 0)))$

Exercice 2 :

Modélisation d'algorithmes [3 points]

On considère l'algorithme donné par la Figure 2 pour trois processus P, Q et R avec variables partagées.

1. Donnez une caractérisation du processus P par un graphe de programmes GP_P , une caractérisation du processus Q par un graphe de programmes GP_Q et une caractérisation du processus R par un graphe de programmes GP_R .
2. (a) Donnez la formule de LTL spécifiant qu'à aucun moment P et Q peuvent être en même temps dans leur section critique.
(b) Donnez la formule de LTL spécifiant qu'à tout moment si R1 vaut 1 alors P finit par accéder à sa section critique.
3. Le système de transitions $ST(GP_P ||| GP_Q ||| GP_R)$ vérifie-t-il ces propriétés? Justifiez votre réponse.

Exercice 3 :

La fête de fin d'année [3 points]

On souhaite organiser une fête de fin d'année pour les cinquante étudiants du département d'informatique sous la tutelle du directeur de département. Pour cette fête on dispose d'une salle. Au début la fête est lancée, chaque étudiant peut alors s'il le souhaite rentrer dans la salle tant que la fête n'est pas finie, et peut en sortir quand il veut.

```

int R1 = 0 // variables partagées
int R2 = 0
int T = 0

-- Processus P
while(true){
  R1 = 1;
  while(T!=1){}
  /*Section Critique*/
  R1 = 0;
  T = 0;
}

-- Processus Q
while(true){
  R2 = 1;
  while(T!=2){}
  /*Section Critique*/
  R2 = 0;
  T = 0;
}

-- Processus R
int U = 1;
while(true){
  while(R1+R2==0){}
  if(R1+R2==2){
    T = U;
    U = 3-U;
  } else if(R1==1){
    T = 1;
  }
  else{
    T = 2;
  }
}

```

FIGURE 2

Quand un étudiant sort de la fête, il ne peut plus y rentrer et il est juste content (les étudiants qui attendent encore pour rentrer dans la fête ne sont pas contents). Si la fête est finie, les étudiants en attente ne peuvent plus rentrer. Le nombre d'étudiants dans la salle n'est pas limité (mais comme il n'y a que 50 étudiants, on sait qu'il n'y aura jamais plus de 50 étudiants dans la salle). Le directeur du département peut quand à lui rentrer dans la salle quand il le veut, si il n'y a aucun étudiant dans la salle (pour contrôler que tout est en ordre) ou si il y a plus de trente étudiants dans la salle, dans ce dernier cas il met fin à la fête. Lorsque le directeur est dans la salle, aucun étudiant ne peut rentrer mais ceux dans la salle peuvent sortir. De plus le directeur ne peut sortir de la salle que si celle-ci est vide (il pourra y rentrer de nouveau ensuite en respectant les conditions données ci-dessus).

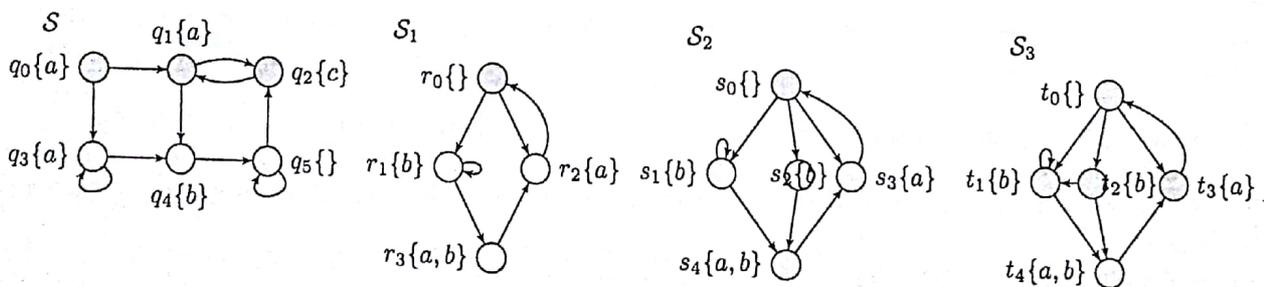
1. Proposez une modélisation par un réseau de Petri de ce système.
2. Donnez la formule CTL précisant qu'il existe une exécution où au bout d'un moment il existe un chemin au cours duquel le directeur est toujours dans la salle et où les étudiants sont toujours contents. NB : Pour cette formule CTL, 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).

Exercice 4 :

Vérification de formules de CTL et LTL [3 points]

Dans cet exercice, les deux premières questions portent sur le système S de la figure ci-dessous, et les deux dernières questions portent sur les systèmes S_1 , S_2 et S_3 :

1. On considère le système de transition S de la figure ci-dessous (les propositions atomiques sont entre accolades) et la formule CTL $\Phi = EG(EXa \wedge EXb)$. Indiquer pour chaque état les sous-formules de Φ vérifiées par l'état. Justifier vos réponses. Indiquer aussi la valeur de vérité des formules $Ea U$, $AG(EFb)$ et $Aa U b$ à l'état q_0 .
2. Dire si les formules LTL suivantes $\Psi_1 = GFa$ et $\Psi_2 = G\neg b \vee GF\neg a$ sont vraies pour le système S en supposant que l'état initial est l'état q_0 . Justifier vos réponses.



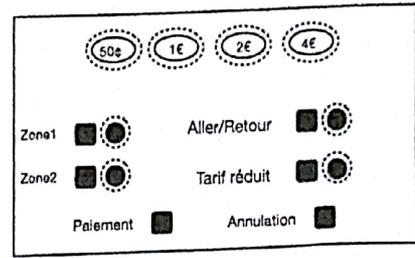
3. Peut-on distinguer S_1 et S_2 , S_1 et S_3 , et S_2 et S_3 avec CTL? Justifier vos réponses. NB : On suppose que les états initiaux sont q_0 , r_0 et s_0 .
4. Même question avec LTL. Justifier vos réponses.

Exercice 5 :

Spécification en logique temporelle [7 points]

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

- deux boutons zone 1 et zone 2 et deux voyants associés à ces boutons. Propositions atomiques¹ : $B_{z1}, B_{z2}, V_{z1}, V_{z2}$.
- un bouton Tarif réduit, un bouton Aller/Retour et leurs voyants. Propositions atomiques : $B_{tr}, B_{ar}, V_{tr}, V_{ar}$.
- un bouton Paiement et un Annulation. Propositions : B_{payer} et B_{an} .
- 4 voyants pour indiquer la somme à payer : 50c, 1€, 2€ et 4€. Propositions : $V_{p,5}, V_{p,1}, V_{p,2}$ et $V_{p,4}$.



La machine peut se trouver dans trois modes différents : (1) en attente d'une transaction (proposition P_{init}), (2) en cours d'utilisation (proposition P_{actif}), et (3) en fin de transaction (proposition P_{fin}).

Le comportement attendu de la machine est le suivant : lorsqu'un usager commence à l'utiliser (en appuyant sur un bouton de zone, de tarif réduit ou d'aller/retour), la machine passe du mode "attente" au mode "en cours d'utilisation". Ensuite, l'usager peut choisir sa zone de destination, son tarif et son type de trajet. Il peut le faire dans n'importe quel ordre et même changer d'avis : si il sélectionne la zone 1, il peut toujours choisir plus tard la zone 2 (et c'est celle-ci qui sera retenue), choisir le tarif réduit en appuyant sur le bouton, puis finalement le plein tarif (en reappuyant sur le bouton), etc. Ensuite lorsque l'usager a fait son choix, il appuie sur le bouton "paiement" qui fait passer la machine en mode "fin de transaction" pour procéder au paiement et le tarif s'affiche alors avec un des 4 voyants de prix décrits précédemment, et le distributeur revient en mode attente après le paiement et l'impression du ticket. Notons enfin qu'à tout moment, appuyer sur le bouton "annulation" fait revenir le distributeur en mode d'attente.

Écrire les formules suivantes :

1. Une formule qui spécifie que dans le mode "attente", tous les voyants sont toujours éteints.
2. Une formule exprimant que la machine est toujours dans un seul des trois modes.
3. Une formule qui spécifie les changements de modes dus à l'action des boutons :
 - dans l'état d'attente, appuyer sur un bouton de zone, de tarif ou de trajet fait passer dans le mode "en cours d'utilisation" ;
 - dans l'état "en cours d'utilisation", lorsqu'on appuie sur paiement, on passe dans le mode "fin de transaction" si une zone de destination a été choisie (sans effet sinon) ;
 - dans tous les modes, appuyer sur annulation fait passer la machine dans le mode d'attente.
4. Une formule qui spécifie le cycle des modes : le comportement de la machine consiste perpétuellement à passer du mode "attente" au mode "en cours d'utilisation" puis soit au mode "attente", soit au mode "fin de transaction" puis "attente". De plus, on spécifiera que le mode "attente" est rencontré infiniment souvent.
5. Une formule qui décrit le comportement des boutons et voyants de zone : appuyer sur le bouton d'une zone allume (à l'état suivant) le voyant associé et éteint (ou laisse éteint) celui de l'autre zone. Le voyant d'une zone reste allumé jusqu'au retour dans le mode d'attente ou jusqu'à ce que l'autre zone soit sélectionnée. Et bien sûr, le voyant ne s'allume pas sans raison.
6. Une formule qui décrit le comportement des boutons et voyants de tarif et de trajet : appuyer sur un de ces boutons, change l'état du voyant dans l'état suivant (il s'allume ou s'éteint en fonction de son état antérieur). Le voyant reste allumé tant qu'il n'est pas éteint par une pression sur le bouton et tant que la machine n'est pas en mode d'attente. Et il ne s'allume pas sans raison.
7. Une formule qui décrit les quatre voyants qui indiquent le prix à payer. Lorsqu'on passe dans le mode "fin de transaction", la machine doit indiquer le prix à payer en fonction des choix de l'usager (ie l'état des voyants), et l'indiquer avec les voyants de prix. Il faut donc spécifier que dans le mode "fin de transaction", seul le voyant de prix correct est allumé, et que tous sont éteints en dehors de ce mode. On donne ci dessous la grille tarifaire en vigueur :

	Plein tarif		Tarif réduit	
	Aller	Aller/retour	Aller	Aller/retour
Zone 1	1€	2€	50c	1€
Zone 2	2€	4€	1€	2€

8. On considère deux nouvelles propositions : $P_{payé}$ qui est vraie lorsque le monnayeur a reçu la somme demandée de la part de l'usager, et P_{ticket} qui est vraie lorsque le ticket est imprimé. Ecrire une formule indiquant que dans le mode "fin de transaction", lorsque l'usager a payé le prix du ticket, alors le ticket est imprimé.

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).