

Examen – Modélisation et spécification
 Master Informatique
 Partie Modélisation

17 Décembre 2020

Durée : 1h.

Documents autorisés : Une feuille A4 manuscrite recto-verso.

Rédaction : Il faut rendre les exercices de cette partie sur une copie différente de la partie Spécification

Exercice 1 :

Analyse de réseaux de Petri [3 points]

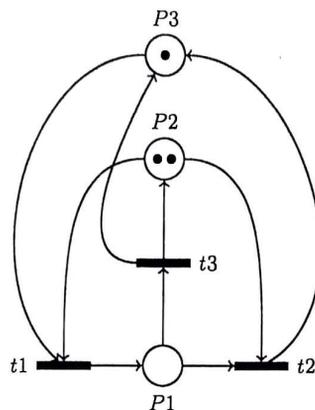


FIGURE 1 – Réseau de Petri RP_1

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

1. Dessiner le graphe des marquages accessibles de ce réseau de Petri (que nous avons aussi appelé $ST(RP_1)$ en cours).
2. Modifier le réseau de Petri (en ajoutant des arcs, des places ou des transitions) pour que lorsque l'on franchisse la transition t_2 on revienne au marquage initial.

Exercice 2 :

Modélisation par système de transitions [4 points]

On considère un système fait d'une structure de données, que nous appellerons boîte, qui peut contenir au plus deux données, et de trois utilisateurs qui utilisent la boîte de façon concurrente. Un utilisateur peut mettre une donnée dans la boîte, prendre une donnée dans la boîte, tester si la boîte est vide ou pleine (c'est-à-dire qu'elle contient deux données). Si un utilisateur demande à mettre une donnée dans la boîte et que celle-ci est pleine, alors la boîte va dans un état d'erreur, de même si un utilisateur essaie de prendre une donnée dans la boîte quand celle-ci est vide, alors la boîte va dans un état d'erreur. Par conséquent, les actions mettre une donnée dans la boîte/prendre une donnée dans la boîte ne bloquent jamais l'utilisateur mais soit elles se passent correctement (et l'état de la boîte change en conséquence) soit elles amènent la boîte dans un état d'erreur à partir duquel elle ne peut rien faire. Au début la boîte est vide. Le comportement des trois utilisateurs est le suivant :

- (a) L'utilisateur 1 teste si la boîte est pleine, si ce n'est pas le cas il met une donnée dans la boîte et sinon il reste dans son état. Ensuite si il a réussi à mettre une donnée, il teste de nouveau si la boîte est pleine, et si ce n'est pas le cas il met une deuxième donnée dans la boîte et sinon il ne change pas d'état. Ensuite si il a réussi à mettre une deuxième donnée il s'arrête.
- (b) L'utilisateur 2 teste si la boîte est pleine, si ce n'est pas le cas il met une donnée dans la boîte et sinon il reste dans son état. Ensuite si il a réussi à mettre une donnée dans la boîte, il s'arrête.
- (c) L'utilisateur 3 fait en boucle les actions suivantes : il teste si la boîte est vide, si ce n'est pas le cas, il prend la donnée et revient dans son état initial, sinon il ne change pas d'état.

On souhaite modéliser ce système par un système de transitions. Pour cela on considèrera l'alphabet suivant pour les actions :

- put : mettre une donnée dans boîte
- get : prendre une donnée dans la boîte
- full : test positif si la boîte est pleine
- nfull : test positif si la boîte n'est pas pleine
- empty : test positif si la boîte est vide
- nempty : test positif si la boîte n'est pas vide

On aura aussi une seule proposition atomique **Err** pour indiquer que la boîte est dans un état d'erreur.

1. Donner une modélisation sous forme de système de transitions de la boîte.
2. Donner une modélisation des trois utilisateurs par trois systèmes de transitions.
3. En déduire une modélisation pour tout le système en donnant la table de synchronisation.
4. Le système obtenu vérifie-t-il la formule LTL $G(\neg \text{Err})$? Justifier la réponse.

Exercice 3 :

Système concurrent partagé [3 points]

Dans une petite entreprise de conception de logiciels, il y a deux administrateurs système et quatre développeurs qui peuvent chacun se connecter au serveur principal pour faire des opérations. On suppose que chacune de ces personnes a le comportement suivant en boucle : elle essaie de se connecter, puis si elle y arrive, elle travaille et finalement elle se déconnecte pour revenir à son état initial. Les règles de l'entreprise imposent que si un administrateur est connecté alors aucune autre personne n'est connectée.

1. Proposer une modélisation de cette entreprise par un réseau de Petri en ajoutant l'hypothèse que si un développeur est connecté alors aucune autre personne n'est connectée. On devra bien entendu distinguer les développeurs des administrateurs.
2. On change maintenant la règle de la question précédente en autorisant au plus deux développeurs à être connectés en même temps. Proposer une nouvelle modélisation respectant cette dernière hypothèse.

Examen – Modélisation et spécification

Master Informatique

Partie Spécification

17 décembre 2020

Durée totale : 1h.

Documents autorisés : Une feuille A4 manuscrite recto-verso.

Rédaction : Il faut rendre deux copies : une pour les trois exercices de la partie spécification et une pour les exercices de la partie modélisation.

Exercice 1 :

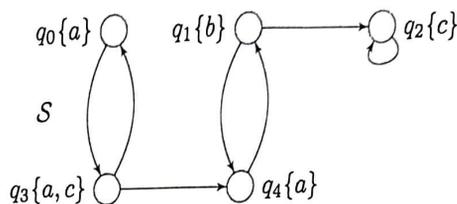
Interpréter des formules de CTL et LTL [3 points]

On considère le système de transition S de la figure ci-contre (les propositions atomiques sont entre accolades) où q_0 est l'état initial.

- Donner la valeur de vérité des formules de LTL listées ci-dessous pour les exécutions de S (depuis q_0) :

$$a \text{ U } b \quad a \text{ W } b \quad G(a \Rightarrow (X(a \vee b)))$$

$$(G \text{ F } b) \Rightarrow (G \text{ F } c) \quad (\neg G \text{ F } a) \Rightarrow (F G c)$$



- Donner la valeur de vérité des formules de CTL listées ci-dessous pour l'état q_0 . Pour cela, on demande la valeur de vérité de toutes les sous-formules pour tous les états de S :

$$E a \text{ U } b \quad AG(E a \text{ U } b) \quad AG(b \Rightarrow EX c) \quad AG(c \Rightarrow AG c) \quad AG(b \Rightarrow AX c)$$

Exercice 2 :

Exprimer des propriétés en LTL [3 points]

Ecrire en LTL les propriétés ci-dessous :

- Si il y a un jour un a et un jour un b , il y aura aussi un jour un c .
- Tout a est précédé par un b .
- Si il y a infiniment souvent a , il y a un nombre fini de b .
- Tout a est suivi un jour par un c et entre les deux, b est toujours vrai.

Exercice 3 :

CTL et bisimulation [4 points]

On considère les 4 modèles S_1 , S_2 et S_3 et S_4 ci-dessous. On suppose que les états initiaux sont q_0 , r_0 , s_0 et t_0 . Peut-on distinguer ces états avec CTL? Justifier votre réponse en donnant, pour chaque paire d'états initiaux, une formule de CTL qui les distingue ou une bisimulation qui les relie.

