

Systemes de transitions - Modelisation TLA

Durée 1h45 - Documents autorisés

13 avril 2012

1 Questions de cours (4 points)

Soit S un ensemble fini d'entiers naturels. Répondre aux questions suivantes en respectant la syntaxe TLA.

1. Donner une expression représentant le sous-ensemble de S restreint aux nombres ≥ 10 .
2. Donner un prédicat qui teste si S contient au moins un entier pair.
3. Soit x une variable. Écrire une action exécutable seulement si x est dans S , et qui dans ce cas change x pour une autre valeur de S .
4. Soit $f \in [S \rightarrow Nat]$. Écrire une action qui transforme f en une fonction où les valeurs d'indices pairs sont doublées, les autres restant inchangées.

2 Exercice (4 points)

Soit le module `Test.tla` ci-dessous définissant le système de transitions `Spec`. Les propriétés suivantes, exprimées en logique LTL ou CTL, sont-elles vérifiées (donner une justification informelle) ?

- | | |
|---|-------------------------------|
| 1. $\exists \diamond (y > 10)$ | 5. $\forall \diamond (x = y)$ |
| 2. $\exists \square (y = 0)$ | 6. $\diamond (y > 10)$ |
| 3. $\forall \square \exists \diamond (y = 0)$ | 7. $\square (y > 0)$ |
| 4. $\exists \square \forall \diamond (x + y = 1)$ | 8. $\square \diamond (x = 0)$ |

2.1 Module fourni : `Test.tla`

```
----- MODULE examen11_test -----  
EXTENDS Naturals  
VARIABLES x, y  
-----  
TypeInvariant  $\triangleq \wedge x \in Nat \wedge y \in Nat$   
-----  
SwitchX  $\triangleq \wedge x' = 1 - x \wedge \text{UNCHANGED } y$   
IncrY  $\triangleq \wedge x = 1 \wedge y' = y + 1 \wedge \text{UNCHANGED } x$   
DecrY  $\triangleq \wedge y > 0 \wedge y' = y - 1 \wedge \text{UNCHANGED } x$   
  
Fairness  $\triangleq \text{WF}_{\langle x, y \rangle}(\text{SwitchX}) \wedge \text{WF}_{\langle x, y \rangle}(\text{IncrY}) \wedge \text{WF}_{\langle x, y \rangle}(\text{DecrY})$   
Spec  $\triangleq \wedge x = 1 \wedge y = 0 \wedge \square [\text{SwitchX} \vee \text{IncrY} \vee \text{DecrY}]_{\langle x, y \rangle} \wedge \text{Fairness}$   
-----
```

3 Problème des Lecteurs/Rédacteurs (12 points)

On souhaite modéliser et vérifier le problème des lecteurs/rédacteurs. Soit un ensemble fini de processus qui cherchent à accéder à une ressource commune, soit en tant que lecteur, soit en tant que rédacteur. Un processus rédacteur doit être seul à accéder à la ressource, tandis qu'au contraire plusieurs processus lecteurs peuvent se la partager. Un processus ne peut pas être à la fois lecteur et rédacteur. Un squelette de module TLA `Lectred.tla` est fourni en 3.6.

3.1 Module complet

1. Définir le prédicat de transitions `Next` qui représente toutes les transitions possibles.
2. Définir la propriété `Spec` qui décrit le système de transitions (sans équité).

3.2 Exclusion entre processus

1. Définir une propriété `Exclusion` exprimant les contraintes d'accès à la ressource des lecteurs/rédacteurs qui doivent être toujours vérifiées.
2. Le squelette proposé ne vérifie pas cette propriété. Modifier uniquement les actions afin de respecter cette propriété.
3. Justifier la difficulté d'implanter de façon répartie une telle solution.

3.3 Accès à la ressource

1. Définir une propriété `Vivant` qui exprime le fait qu'il y a toujours au moins un processus demandeur qui finira par accéder à la ressource.
2. Définir une propriété `Acces` d'accès garanti à la ressource pour tout processus demandeur.
3. Ajouter en justifiant l'équité faible minimale sur les actions, sans laquelle la propriété `Vivant` est trivialement fausse.

3.4 Analyse des exécutions

Malgré l'équité faible, l'accès à la ressource n'est pas garanti. On s'intéresse au graphe d'exécution du système. Pour diminuer la complexité, on considère $N = 3$ processus dont les deux premiers ne cherchent qu'à lire, et le troisième qu'à écrire. De plus, le processus 0 fait toujours sa demande de lecture avant le processus 1. De même, le processus 0 obtient toujours l'accès avant le processus 1 et libère toujours après. Ces différentes contraintes sont résumées dans la figure 1. On admettra que les contraintes imposées aux processus lecteurs permettent tout de même de représenter le cas général.

1. Dessiner le graphe d'exécution du système de transitions en suivant les contraintes proposées (15 états).
2. Expliquer comment vérifier l'absence d'interblocage par un examen du graphe.
3. Expliquer comment vérifier la propriété `Vivant` par un examen du graphe.
4. Proposer une exécution cyclique de la forme : $s_0 \rightarrow \dots s_{n-1} \rightarrow (s_n \rightarrow \dots \rightarrow s_{n+p})^\omega$ pour laquelle un des processus demandeurs n'accède jamais à la ressource, même en supposant l'équité forte sur les actions. Justifier votre réponse.

