Aix*Marseille université

Domaine Sciences et Technologies
L3 Informatique

Introduction à la vérification : Projet

Année 2023-24

Un protocole d'exclusion mutuelle

Ce projet sera évalué à l'aide des fichiers Promela que vous fournirez, accompagnés d'un fichier au format pdf qui contiendra vos explications textuelles, éventuellement complétées par des graphiques.

On étudie dans ce projet un nouveau protocole d'exclusion mutuelle. Ce protocole est décrit d'abord pour deux processus `P` et `Q`. Il utilise une variable partagée, nommée `S`, qui peut prendre 5 valeurs différentes : `Z, P, Q, PQ, QP`.

Le protocole est décrit par le pseudo-code suivant :

type switch = {Z, P, Q, PQ, QP} switch S := Z	
p	q
while true do	while true do
p1: non-critical section	q1: non-critical section
p2: if	q2: if
:: S=Z → S:=P;	:: S=Z → S:=Q;
:: S=Q → S:=QP;	:: S=P → S:=PQ;
:: else → skip	:: else → skip
p3: await (S=P or S=PQ)	q3: await (S=Q or S=QP)
p4: critical section	q4: critical section
p5: if	q5: if
:: S=P → S:=Z;	:: S=Q → S:=Z;
:: S=PQ → S:=Q;	:: S=QP → S:=P;
:: else → skip	:: else → skip

Figure 1 – Un protocole d'exclusion mutuelle.

Notez qu'ici l'attente dans les lignes p3 et q3 est bloquante (comme dans Promela), et que les tests et actions des commandes p2, q2, p5 et q5 sont réalisés de manière atomique. Ainsi, les tests sur `S` et les mises à jour consécutives sont réalisés sans interruption.

1. Modéliser ce protocole en Promela. Vous ferez des simulations de votre modèle pour vous assurer de son bon fonctionnement.

2. Ajouter une variable partagée permettant de compter le nombre de processus en section critique. Vérifier ensuite à l'aide de Spin la propriété d'exclusion mutuelle.

3. On souhaite à présent vérifier que lorsqu'un processus fait une requête, celle-ci finit par être satisfaite. Pour cela, nous étudierons seulement un des deux processus, la situation étant symétrique pour le second. Quelle est la formule LTL permettant de s'assurer de cette propriété ? En intégrant des `#define`, intégrer cette formule sous la forme d'un processus `never`. Vérifier à l'aide de Spin que la propriété est vérifiée.

4. On souhaite à présent généraliser ce protocole à un nombre quelconque de processus, que l'on note N. Vous utiliserez à nouveau un `#define` pour définir cet entier dans votre modèle, et le laisser comme un paramètre.

   Pour passer à ce cadre paramétré, on remplace la variable `S` par un tableau partagé de taille N. On supposera que les identifiants des processus commencent à 1. Le tableau permettra de stocker les différentes demandes d'accès à la section critique des processus. Ainsi, si le système contient 5 processus, et que les processus 2, 1 et 4 ont fait des requêtes, dans cet ordre, le tableau contiendra $[2, 1, 4, 0, 0]$. Notez que chaque processus peut faire au plus une requête, donc un tableau de taille N suffit. Dans ce modèle, les commandes p2,q2 (requête) et p5,q5 (relâche) sont modifiées de la façon suivante :
   - lorsqu'un processus fait une requête, il ajoute son identifiant à la place du premier 0 dans le tableau. Par exemple, dans la situation précédente, si le processus 3 fait une requête, le nouveau contenu du tableau est $[2, 1, 4, 3, 0]$.
   - lorsqu'un processus relâche sa requête, il met à 0 la première case du tableau, et décale toutes les cases non nulles d'un indice vers la gauche. Par exemple, dans la situation précédente, si le processus 2 relâche sa requête, le nouveau contenu du tableau est $[1, 4, 0, 0, 0]$.

   Créer un nouveau modèle Promela représentant cette version du protocole.

5. Ajouter une variable permettant de compter le nombre de processus en section critique, et vérifier à l'aide de Spin que la propriété d'exclusion mutuelle est vérifiée, pour $N = 2$.

6. Refaire la vérification, pour des valeurs croissantes de N : $3, 4, 5 \ldots$ Que constatez-vous ? Vous pourrez utiliser l'option `-m` de `./pan` pour considérer des profondeurs d'exploration plus grandes.