Master FIIL Paris-Sud
Master 2

Software Model Checking
Année 2018-2019

Examen

22 février 2019
Durée : 2h

Exercice 1.

Nous considérons un système de gestion de données utilisateur (typiquement, une interface web connectée à une base de donnée SQL). Dans ce type de système, l'interface va typiquement récupérer des données utilisateur et les stocker dans une variable donnée du code (`read`), traiter ces données de manière sensible – typiquement une requête SQL (`treat`), ou *sanitizer* ces données, c'est-à-dire les transformer de manière à ce qu'elles ne provoquent pas de problème lors d'un traitement sensible (`sanitize`).

On se concentre sur une variable particulière du code, et les opérations de `read`, `treat` et `sanitize` sur cette variable. On suppose que cette variable ne sert à rien d'autre (*variable d'échange*). Notre spécification haut niveau est que :

« (1) traiter des données non sanitizées provoque une erreur ; (2) utiliser des données non définies provoque une erreur ; (3) toute donnée récupérée doit finir par être traitée. »

Questions :

1. Quelle partie de la spécification ci-dessus est de la sûreté, quelle partie est de la vivacité ?
2. Dessinez un automate de Büchi (complet) pour la spécification complète.
3. Donnez une formule LTL pour la spécification complète.

Exercice 2.

Questions

1. Quelle est la différence entre les logiques LTL, CTL, CTL* du point de vue des opérateurs permis ?
2. Expliquez brièvement le principe des algorithmes de model checking de LTL et CTL vus en cours ($\approx$ 10 lignes par algorithme).
3. Donnez les automates de Büchi représentant $\mathbf{GF}p$ et $\mathbf{FG}p$.

Exercice 3 : Le protocole MESI

Dans cet exercice, nous allons modéliser le protocole MESI qui permet d'assurer la cohérence des données entre les caches d'un micro-processeur. Dans ce protocole, un cache peut être dans quatre états : `Modified`, `Exclusive`, `Shared` ou `Invalid`.

- `Modified`. Indique que les données qui sont dans le cache sont « sales » : elles ont été modifiées par rapport aux données qui sont en mémoire centrale. Il n'y a pas cohérence. La mémoire centrale doit être mise à jour avec la version locale avant que d'autres puissent la lire.
- `Exclusive`. Indique que seul ce cache possède une valeur « propre », c'est-à-dire identique à la valeur correspondante en mémoire centrale. On dit que la cohérence est assurée.
- `Shared`. Indique que les données qui sont dans le cache sont aussi potentiellement présentes dans d'autres caches. La cohérence est assurée dans cet état.
- `Invalid`. Indique que les données qui sont dans le cache ne sont pas valides.

Les changements d'état des caches en cas de requêtes de lectures et d'écritures sont les suivantes :

- Si un processeur effectue une requête de lecture.
  — Si son cache est dans un état autre que `Invalid` alors l'état reste inchangé.
  — Si l'état de son cache est `Invalid` :
    1. Son cache passe en `Exclusive` *ssi* tous les autres caches sont dans l'état `Invalid`.
    2. Sinon, son cache passe en `Shared` et, *simultanément*, les caches des processeurs dans l'état `Modified` ou `Exclusive` passent en `Shared`.
- Si un processeur effectue une requête d'écriture.
  — Si son cache est en `Invalid` ou `Shared` alors il passe en `Modified` et, *simultanément*, tous les autres caches passent en `Invalid`.
  — Si son cache est en `Exclusive`, alors il passe en `Modified`.
  — Sinon, son cache est inchangé.

Questions :

1. Modéliser en Cubicle le fonctionnement du protocole MESI en supposant que les caches sont tous `Invalid` dans l'état initial.
2. Donner une formule `unsafe` qui permet de vérifier qu'il n'est pas possible d'avoir simultanément deux caches dans l'état `Modified`.
3. Comme vu en cours, dessiner (sous forme d'un arbre) l'exécution de l'algorithme d'atteignabillté en partant de la formule `unsafe` ci-dessus, et en prenant soin d'expliquer les tests SMT de point-fixe.
4. Montrer, en donnant une suite de transitions depuis l'état initial, que le protocole est *unsafe* si un cache peut passer dans l'état `Exclusive` sans vérifier que tous les autres caches sont dans l'état `Invalid`.