Université Paris Diderot Année 2012–2013

Examen Théorie et Pratique de la Concurrence

Mercredi 15 Mai 2013. Durée : 2 heures.
Document autorisé : une feuille manuscrite A4 recto-verso

Ce sujet comporte cinq exercices.

Exercice 1 - LTL

1. Donnez des formules LTL pour exprimer les propriétés suivantes (utilisant les propositions atomiques p, q et r) :
   (a) Tant que q n'apparaît pas, p est présent.
   (b) p n'apparaît jamais.
   (c) q apparaît au bout de deux "coups" et ensuite il n'apparaît plus jamais.
   (d) q apparaît à un moment dans le futur et strictement après p apparaît aussi dans le futur.
2. On considère le système de transitions étiqueté représenté à la Figure 1. Pour ce système (dont l'état initial est pointé par une fléche), indiquez si il satisfait ou non les formules suivantes (en justifiant brièvement votre réponse) :
   (a) $\bigcirc q$
   (b) $\Diamond r$
   (c) $(p \mathcal{U} q) \wedge (\Box \neg q)$
   (d) $p \mathcal{U} (q \mathcal{U} r)$

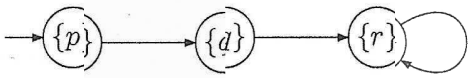

FIGURE 1 –

3. Pour chacune des formules LTL suivantes, donnez un système de transitions étiqueté qui vérifie la formule (les propositions atomiques sont p, q et r) :
   (a) $p \wedge q$
   (b) $\Box \neg p$
   (c) $(p \vee q) \mathcal{U} r$
   (d) $(\bigcirc q) \wedge (\Box r)$

Exercice 2 - Analyse d'algorithme

```
                boolean B1 := False      // variable partagées
                boolean B2 := False
                int k := 1

-- Processus P1                          -- Processus P2
loop forever :                           loop forever :
p1: section NC                           q1: section NC
p2: B1 := True                           q2: B2 := True
p3: while k!=1 do                        q3: while k!=2 do
p4:    while B2==True do                 q4:    while B1==True do
p5:       skip;                          q5:       skip;
       end while                                end while
p6:    k:=1                              q6:    k:=2
    end while                                end while
p7: section critique                     q7: section critique
p8: B1:=False                            q8: B2:=False
```

FIGURE 2 – Algorithme de Hyman

```
                int C2 := 0      // variables partagées
                int C1 := 0
                int k := 1

-- Processus P1                          -- Processus P2
loop forever :                           loop forever :
p1:  section NC                          q1:  section NC
p2:  C1 := 1                             q2:  C2:=1
p3:  if k==1 then                        q3:  if k==2 then
p4:     goto p7                          q4:     goto q7
     end if                                   end if
p5:  if C2!=0 then                       q5:  if C1!=0 then
p6:     goto p3                          q6:     goto q3
     end if                                   end if
p7:  C1:=2                               q7:  C2:=2
p8:  if C2==2 then                       q8:  if C1==2 then
p9:     goto p2;                         q9:     goto q2;
     end if                                   end if
p10: k:=1                                q10: k:=2
p11: section critique                    q11: section critique
p12: k:=2                                q12: k:=1
p13: C1:=0                               q13: C2:=0
```

FIGURE 3 – Algorithme de Knuth

1. Quelle forme ont les états du diagramme d'états de l'algorithme d'Hyman (Figure 2) ? Dessinez le début de ce diagramme d'états (ne dessinez pas plus que 10 états).
2. Donnez la formule LTL qui caractérise la propriété d'absence d'exclusion mutuelle pour l'algorithme d'Hyman (Figure 2).
3. L'algorithme d'Hyman (Figure 2) ne vérifie pas la propriété d'exclusion mutuelle. Donnez une trace du système qui mette en évidence ce fait (i.e. un exemple d'exécution à la fin de laquelle les deux processus se retrouvent en même temps en section critique).
4. On s'intéresse maintenant à l'algorithme de Knuth (Figure 3). Pour cet algorithme montrez que les points suivants sont des invariants :
   (a) `p11` $\Rightarrow$ `(C1==2)`
   (b) `q11` $\Rightarrow$ `(C2==2)`

   En déduire que l'algorithme de Knuth respecte la propriété d'exclusion mutuelle (on pourra raisonner par contradiction en supposant que la propriété est fausse).

Exercice 3 - Conception d'algorithme concurrent

On considère le système suivant fait de trois types de processus P, Q et R et de deux variables partagées A et B. Les accès aux variables A et B se font en exclusion mutuelle (c'est à dire qu'il ne peut pas y avoir deux processus manipulant la variable A en même temps, ni deux processus manipulant la variable B en même temps). Pour chacun des processus, le comportement est le suivant :

1. Un processus de type P prend l'accès à la variable A, exécute sa tâche (en faisant `modify(A)` par exemple), libère la variable et recommence.
2. Un processus de type Q demande l'accès aux variables A et B, exécute sa tâche (en faisant `modify(A)` et `modify(B)` par exemple), libère les variables et recommence.
3. Un processus de type R attend à chaque fois avant de commencer qu'un processus de type P ET qu'un processus de type Q aient exécuté leur tâche pour s'exécuter, il prend ensuite l'accès à la variable B, exécute sa tâche (en faisant `modify(B)` par exemple), libère la variable et retourne dans son état initial où il attend de nouveau qu'un processus de type P et qu'un processus de type Q aient exécuté leur tâche.

Le nombre de processus de type P, Q et R n'est pas borné.

1. En utilisant des sémaphores, proposez une implémentation en langage algorithmique abstrait (c'est-à-dire comme vu en cours) des processus de type P, Q et R.
2. Quelles sont les différences sur votre implémentation selon que vous utilisiez des sémaphores forts ou des sémaphores faibles ?

Exercice 4 CCS

On considère une pile pouvant contenir deux éléments. Cette pile peut-être représentée par le processus CCS suivant :

- $Pile \stackrel{def}{=} push.Pile_1$
- $Pile_1 \stackrel{def}{=} push.Pile_2 + pop.Pile$
- $Pile_2 \stackrel{def}{=} pop.Pile_1$

1. Dessinez le système de transitions étiqueté correspondant au processus $Pile$.

2. Si à l'place d'une pile à deux éléments, nous voulions une pile à trois éléments, quelle serait la définition du processus *Pile* ?
3. On considère les processus CCS *Producteur* et *Consommateur* décrit de la façon suivante :
   - $Pr : Producteur \stackrel{def}{=} \overline{push}.Producteur$
   - $Co : Consommateur \stackrel{def}{=} \overline{pop}.Consommateur$

   Dessinez le diagramme d'états du processus $((Producteur|Consommateur)|Pile)$. Quelle est la différence avec le processus $((Producteur|Consommateur)|Pile) \setminus \{push, pop\}$?

Exercice 5 Bisimulation

Pour chacune des paires de systèmes de transitions étiquetés des Figures 4 et 5, dîtes si les deux systèmes sont bisimilaires ou non. Si il y a bisimulation, donnez une relation de bisimulation, sinon donnez la formule de HML permettant de distinguer les deux systèmes.

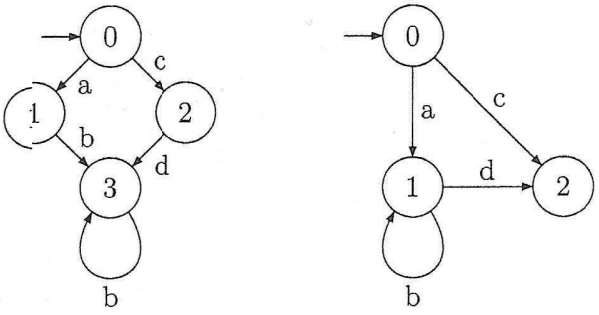

FIGURE 4 – STE

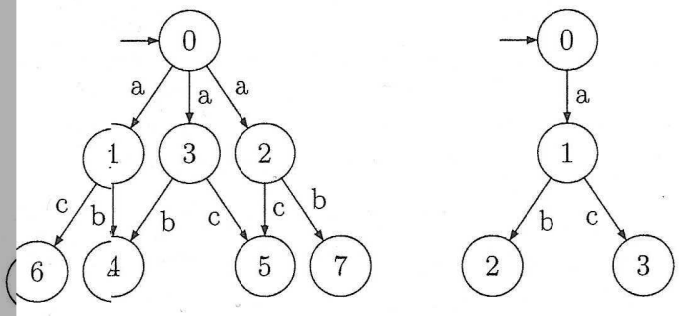

FIGURE 5 – STE