Université Paris Diderot Année 2011–2012

Théorie et pratique de la concurrence

Contrôle Continu

Exercice 1 [2 points] Soit une fonction f pour laquelle il existe une valeur i telle que $f(i) = 0$. Les algorithmes concurrents ci-dessous cherchent une telle valeur i (appelée aussi le point zéro) en divisant l'espace de recherche en deux parties : les points zéro positifs $i \geq 0$ et les points zéro négatifs $i < 0$. Un algorithme est correct si, pour toutes les exécutions, les deux processus terminent après que l'un d'entre eux a trouvé un point zéro.

L'algorithme suivant est-il correct, justifiez votre réponse.

Algorithme Zéro

```
boolean found:=false
```

```
process P
    integer i:=0;
p1: while not found
p2:   i:=i+1
p3:   if (f(i)==0)
p4:     found:=true
```

```
process Q
    integer j:=1;
q1: while not found
q2:   j:=j-1
q3:   if (f(j)==0)
q4:     found:=true
```

Exercice 2 [4 points] On considère le programme concurrent suivant :

```
int n=0; // variable partagée
```

```
process Q
q1: n:=n+1
q2: n:=n+1
q3: skip
```

```
process P
  loop forever:
  p1: if(n<2) {
  p2:   print(n)}
  p3: else print('_')
```

1. Construisez le diagramme d'états de cet algorithme.
2. Indiquez les exécutions qui affichent les mots suivants (donnés sous forme d'expressions régulières) : `012'_'*`; `002'_'*` et `0'_'*`.
3. Pour capturer le fait qu'une certaine valeur v de n est imprimée pendant l'exécution, on définit le prédicat suivant :

$$print(v) \ := \ (n = v) \wedge p2 \wedge \bigcirc p1$$

Écrivez les formules LTL correspondant aux propriétés suivantes :

(a) "2 doit apparaître dans l'affichage"

(b) "2 apparaît au plus une fois"

(c) "2 apparaît une infinité de fois"

(d) "2 ne peut apparaître qu'après 0"

4. Pour chaque propriété ci-dessus, indiquez si le programme concurrent la vérifie ou non.

Exercice 3 [5 points] Soit la solution suivante pour l'implémentation d'une section critique :

Algorithme 1

```
boolean wantp = False, wantq = False; // variables partagées
```

```
Process P
  loop forever:
  p1: section NC
  p2: await (wantq==false)
  p3: wantp:=True
  p4: section critique
  p5: wantp:=False
```

```
Process P
  loop forever:
  p1: section NC
  p2: await (wantp==false)
  p3: wantq:=True
  p4: section critique
  p5: wantq:=False
```

1. Construisez le diagramme d'états de cet algorithme
2. Quelles sont les propriétés vérifiées par cet algorithme (exclusion mutuelle, absence de famine, absence d'interblocage) ?

Exercice 4 [5 points] Considérons l'Algorithme 2 pour l'implémentation d'une section critique :

Algorithme 2

```
int wantP = 0, wantQ = 0; // variable partagée
```

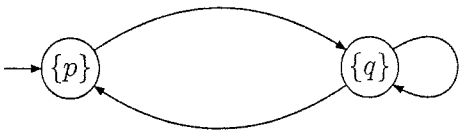
```
Process P
  loop forever:
    if (wantQ = 1)
        wantP := -1
    else
        wantP := -1
    await (wantP != wantQ)
    section critique
    wantP := 0
```

```
Process Q
  loop forever:
    if (wantP = -1)
        wantQ := 1
    else
        wantQ := -1
    await (wantQ == wantP)
    section critique
    wantQ := 0
```

1. Numérotez les lignes de cet algorithme dans l'idée de pouvoir construire ensuite le diagramme d'états
2. Exprimez à l'aide d'une formule LTL la propriété d'exclusion mutuelle
3. Exprimez à l'aide d'une formule LTL la propriété d'absence de famine
4. En utilisant le diagramme d'états, indiquez si cet algorithme vérifie la propriété d'exclusion mutuelle (dans le cas d'une réponse négative, seulement une partie du diagramme d'états exhibant le comportement incorrect est demandée).

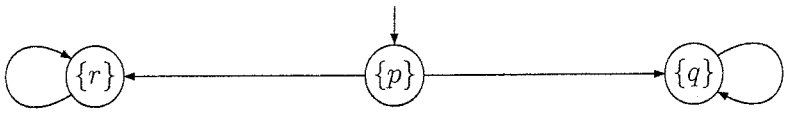
Exercice 5 [4 points] Dans les différents exemples donnés ci-dessous, parmi la liste des propriétés LTL fournies, donnez celles que vérifient le diagramme d'états correspondant.

1.



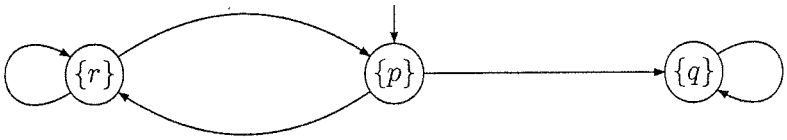
(a) $\Box p$
(b) $\Box(p \Rightarrow \bigcirc q)$
(c) $(\Box\Diamond p) \Rightarrow (\Box\Diamond q)$

2.



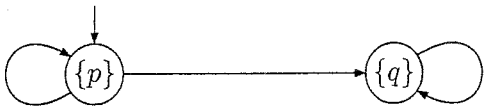
(a) $\Box(\bigcirc(q \vee r))$
(b) $\Box(p \Rightarrow \Diamond q)$
(c) $\Diamond(q \vee r)$

3.



(a) $\bigcirc(q \vee r)$
(b) $\Box(p \Rightarrow \Diamond q)$
(c) $\Box(p \Rightarrow \bigcirc(q \vee r))$

4.



(a) p
(b) $\Box(p \vee q)$
(c) $\Box(p \Rightarrow \Diamond q)$