# Examen Théorie et pratique de la concurrence

Jeudi 24 mai 2012, 3h, aucun document autorisé

Ce sujet comporte quatre exercices. Le barème est donné à titre indicatif.

## Exercice 1 : CCS – 5 points (2/1/2)

1. Construire les systèmes de transitions étiquetés associés aux termes CCS suivants :
   - $P_0 \stackrel{\text{def}}{=} a \cdot (b \cdot \emptyset \ + \ c \cdot a \cdot \emptyset)$
   - $P_1 \stackrel{\text{def}}{=} P_0 \ + \ a \cdot P_1$
   - $P_2 \stackrel{\text{def}}{=} \bar{a} \cdot (\bar{a} \cdot P_2 \ + \ \bar{b} \cdot P_2)$
   - $P_3 \stackrel{\text{def}}{=} (P_1 \mid P_2)$
   - $P_5 \stackrel{\text{def}}{=} (P_1 \mid P_2) \backslash \{a, b\}$
2. Construire des processus CCS dont la sémantique correspond aux systèmes de transitions étiquetés (i) et (ii) de la figure 2 (préciser l'état de départ choisi).

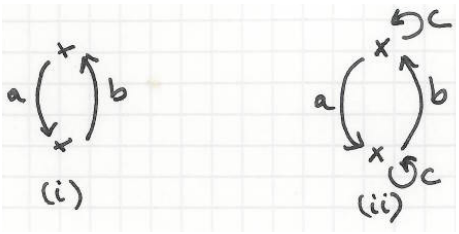


FIGURE 1 – STE de la question 2 (exercice 1)

3. Est-ce que les états $q_1$, $r_1$, $s_1$ et $t_1$ des systèmes de transitions de la figure 2 sont bisimilaires ? Justifier votre réponse. Lorsque les états ne sont pas bisimilaires, on donnera une formule de HML qui les distingue ou une stratégie pour l'attaquant (joueur 1) dans le jeu de bisimulation associé.

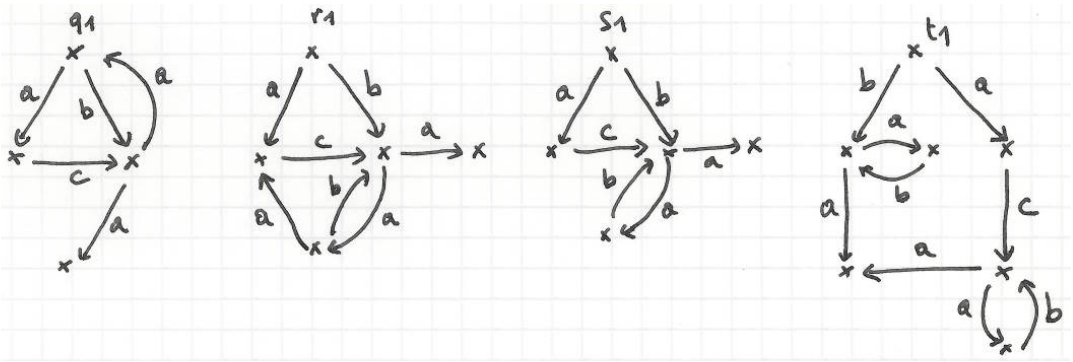


FIGURE 2 – STE de la question 3 (exercice 1)

### Exercice 2 : Exclusion mutuelle – 5 points (0,5/0,5/0,5/1/1/0,5)

1. Rappeler comment fonctionne l'algorithme de Peterson (voir l'annexe).
2. Maintenant, on considère la variante (appelée ensuite "algorithme K") suivante :

```
boolean B0 := False      // variables partagées
boolean B1 := False
int turn0 := 1
int turn1 := 1
```

```
-- Processus P0
   int local0 = 0 // var locale
loop forever :
p1: section NC
p2: B0 := True
p3: local0 := turn1
p4: turn0 := local0
p5: await (B1 == False OR
                 local0 != turn1)
p6: section critique
p7: B0:=False
```

```
-- Processus P1
   int local1 = 0 // var locale
loop forever :
p1: section NC
p2: B1 := True
p3: local1 := 1-turn0
p4: turn1 := local1
p5: await (B0 == False OR
                 local1 == turn0)
p6: section critique
p7: B1:=False
```

   (a) Comparer l'algorithme K et celui de Peterson du point de vue du type de variables partagées utilisées (multi/single-reader, multi/single-writer).

   (b) Comment s'énoncent l'exclusion mutuelle et l'absence de famine en LTL pour l'algorithme K ?

3. Analyse :

   (a) Expliquer comment fonctionne l'algorithme K (on pourra le comparer à l'algorithme de Peterson).

   (b) Montrer qu'il n'y a pas d'interblocage.

   (c) Sous hypothèse d'équité entre processus et en supposant que toute section critique termine et conduit à l'exécution du post-protocole, montrer que l'algorithme vérifie l'absence de famine.

   (d) **bonus :** Montrer que l'algorithme vérifie l'exclusion mutuelle.

### Exercice 3 : Sémaphores – 5 points (2,5/2,5)

Un problème de bus... Nous considérons $N$ passagers et un bus ayant $C$ places (avec $C < N$). Le comportement du bus est le suivant :

(a) Il attend que $C$ passagers soient montés.

(b) Il part.

(c) Il attend que les $C$ passagers soient descendus du bus pour aller de nouveau au point (a).

Le comportement d'un passager est le suivant :

(a') Il essaie de monter dans le bus : si il y a encore de la place, il peut monter sinon il doit attendre.

(b') Une fois monté dans le bus, il fait le voyage dans le bus.

(c') Il descend du bus et retourne en (a').

On souhaite représenter ce système avec des processus concurrents (on suppose qu'il y a plusieurs portes et donc que plusieurs passagers peuvent monter ou descendre en même temps).

1. Proposer un algorithme utilisant **des** sémaphores pour décrire le comportement du processus BUS ainsi que celui d'un processus PASSAGER.
2. Expliquer le changement de comportement du système selon que les sémaphores utilisés dans votre solution sont forts ou faibles.

**Indication :** Voilà une façon possible d'aborder le problème :
- Les passagers pourront à l'aide d'un compteur partagé se compter de façon à savoir combien de passagers sont montés dans le bus et combien sont descendus. (Attention à protéger l'accès à ce compteur partagé grâce à un sémaphore binaire).
- Le dernier passager à remplir le bus réveillera le bus pour lui signaler qu'il est plein.
- Le dernier passager à descendre du bus réveillera le bus pour lui signaler qu'il est vide.

**Exercice 4 : Simulation – 5 points (0,5/1/1/1/1/0,5)**

Dans cet exercice, on suppose donnés deux systèmes de transitions étiquetés $\mathcal{S}_1 = (S_1, \mathsf{Act}, \rightarrow_1)$ et $\mathcal{S}_2 = (S_2, \mathsf{Act}, \rightarrow_2)$.
On définit la notion de **simulation :** on dit qu'une relation binaire $\mathcal{R} \subseteq S_1 \times S_2$ est une **simulation** si et seulement si : pour tous sommets $s$ et $t$ tels que $s\mathcal{R}t$, et pour tout $\alpha \in \mathsf{Act}$, si il existe $s \xrightarrow{\alpha}_1 s'$ alors il existe $t \xrightarrow{\alpha}_2 t'$ tel que $s'\mathcal{R}t'$.

On dit que $s$ est "simulé" par $t$ ssi il existe une simulation $\mathcal{R}$ telle que $s\mathcal{R}t$. On le note $s \sqsubseteq t$.

1. Expliquer (informellement) ce que signifie $s \sqsubseteq t$.
2. Etant donnés les systèmes de transitions de la figure 3, a-t-on les propriétés suivantes :

$$q_0 \sqsubseteq r_0? \quad q_0 \sqsubseteq s_0? \quad s_0 \sqsubseteq q_0? \quad r_0 \sqsubseteq q_0? \quad s_0 \sqsubseteq r_0? \quad r_0 \sqsubseteq s_0?$$

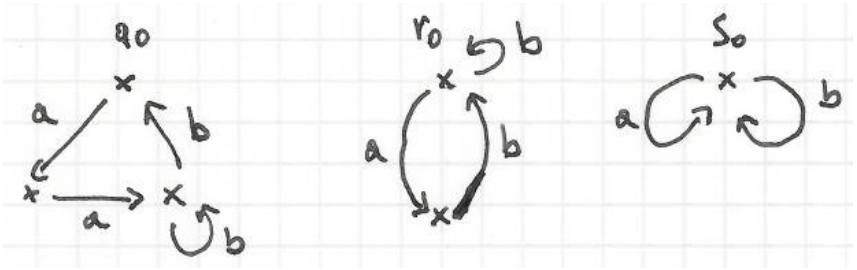


FIGURE 3 – Exemples de STE (exercice 2)

3. On définit le fragment "existentiel" de la logique HML (noté $HML_e$) avec la grammaire suivante :

$$HML_e \ni \phi, \psi ::= \top \mid \phi \wedge \psi \mid \phi \vee \psi \mid \langle\alpha\rangle\phi$$

avec $\alpha \in \mathsf{Act}$. L'ensemble $HML_e$ contient donc les formules de HML construites sans négation, sans $\bot$ et sans modalité $[\alpha]$.
De la même manière, on définit le fragment "universel" ($HML_u$) avec la grammaire suivante : $HML_u \ni \phi, \psi ::= \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid [\alpha]\phi$

(a) Montrer que pour deux états $s_1 \in S_1$ et $s_2 \in S_2$ tels que $s_1 \sqsubseteq s_2$, alors pour toute formule $\phi \in HML_e$, on a : $s_1 \models \phi$ implique $s_2 \models \phi$.

(b) En déduire que si $s_1 \sqsubseteq s_2$, alors pour toute formule $\phi \in HML_u$, on a : $s_2 \models \phi$ implique $s_1 \models \phi$.

4. Adapter à la simulation le jeu à deux joueurs défini en cours pour caractériser la bisimulation.
5. Montrer que si le STE $\mathcal{S}_2$ est à branchement fini, on a :

$$\Big(\forall \phi \in HML_e, s_1 \models \phi \text{ implique } s_2 \models \phi\Big) \text{ implique } s_1 \sqsubseteq s_2$$

---

# Annexe

## Algorithme de Peterson

On rappelle ici l'algorithme de Peterson vu en cours. On considère deux processus $P_0$ et $P_1$.

```
            boolean D0 := False      // variables partagées
            boolean D1 := False
            int turn := 0
-- Processus P0                          -- Processus P1
loop forever :                           loop forever :
p1: section NC                           q1: section NC
p2: D0 := True                           q2: D1 := True
p3: turn := 1                            q3: turn := 0
p4: await (D1==False  OR turn==0)        q4: await (D0==False  OR turn==1)
p5: section critique                     q5: section critique
p6: D0:=False                            q6: D1:=False
```

## Sémantique de CCS

On rappelle la sémantique de CCS. Avec les conventions habituelles : $\mathcal{A}$ est un ensemble de noms de ports, $\mathcal{L} = \mathcal{A} \cup \bar{\mathcal{A}}$, $\mathsf{Act} = \mathcal{L} \cup \{\tau\}$, et on suppose : $\alpha \in \mathsf{Act}$, $a \in \mathcal{L}$, $L \subseteq \mathcal{L}$, et $f$ est une fonction de renommage.

$$\text{(Act)}\ \frac{}{\alpha \cdot P \xrightarrow{\alpha} P} \qquad \text{(Som } j\text{)}\ \frac{P_j \xrightarrow{\alpha} P'_j \quad \text{avec } j \in I}{\sum_{i\in I} P_i \xrightarrow{\alpha} P'_j} \qquad \text{(Const)}\ \frac{P \xrightarrow{\alpha} P' \quad \text{avec } K \stackrel{def}{=} P}{K \xrightarrow{\alpha} P'}$$

$$\text{(Com 1)}\ \frac{P \xrightarrow{\alpha} P'}{P|R \xrightarrow{\alpha} P'|R} \qquad \text{(Com 2)}\ \frac{P \xrightarrow{\alpha} P'}{R|P \xrightarrow{\alpha} R|P'} \qquad \text{(Com 3)}\ \frac{P \xrightarrow{a} P' \quad R \xrightarrow{\bar{a}} R'}{P|R \xrightarrow{\tau} P'|R'}$$

$$\text{(Res)}\ \frac{P \xrightarrow{\alpha} P' \quad \text{avec } \alpha, \bar{\alpha} \notin L}{P\backslash L \xrightarrow{a} P'\backslash L} \qquad \text{(Ren)}\ \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$$

## Sémantique de HML

On rappelle la sémantique des deux opérateurs $\langle\alpha\rangle$ et $[\alpha]$ avec $\alpha \in \mathsf{Act}$ :
- $s \models \langle\alpha\rangle\phi$ si et seulement si il existe $s \xrightarrow{\alpha} s'$ avec $s' \models \phi$.
- $s \models [\alpha]\phi$ si et seulement si pour toute transition $s \xrightarrow{\alpha} s'$, on a $s' \models \phi$.