**L2/L3 MASS** **Année universitaire 2012–2013**
**Université Paris 1**

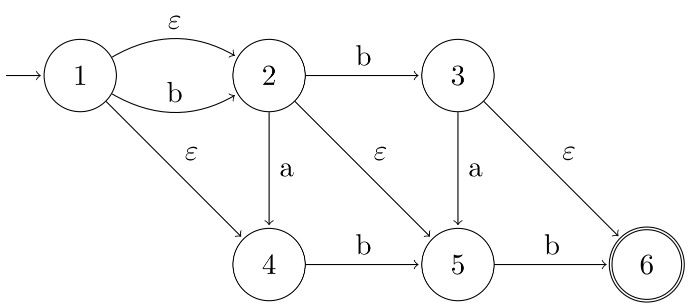
**Partiel de mathématiques pour l'informatique**
**Mardi 23 avril – Durée 3 heures**

Les exercices sont indépendants. Au sein d'un exercice, il est conseillé de traiter les questions dans l'ordre. Chacune des trois parties de l'énoncé (automates, logique propositionnelle et preuves de programmes) compte pour environ un tiers de la note. Les réponses devront être soigneusement argumentées et justifiées. Les documents imprimés ou manuscrits sont autorisés. L'utilisation de calculatrices, téléphones portables ou tout autre appareil électronique, est interdite.

# 1 Automates

### Exercice 1

On considère l'automate asynchrone suivant :




**Question 1** Donner un exemple de mot reconnu par l'automate et le plus court possible.

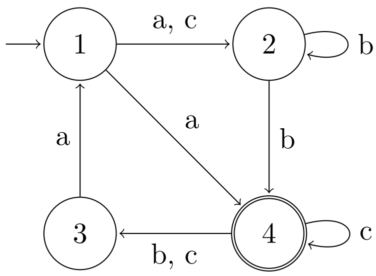
**Question 2** Donner un exemple de mot reconnu par l'automate et le plus long possible.

**Question 3** Supprimer les $\varepsilon$-transitions de l'automate en utilisant l'algorithme classique (attention, on ne demande pas un automate simplifié équivalent mais bien l'automate sans $\varepsilon$-transitions obtenu par l'application de l'algorithme de suppression, même s'il comporte des transitions et/ou des états inutiles).

**Question 4** Donner une version déterministe de l'automate obtenu.

### Exercice 2

On considère l'automate suivant :




**Question 1** Donner le mot le plus court reconnu par l'automate.

**Question 2** Donner un mot reconnu par l'automate selon un chemin passant au moins une fois par chaque état (on considère que l'état initial est toujours parcouru).

**Question 3** Déterminer un langage $L$ de taille infinie (c'est-à-dire contenant une infinité de mots) dont les mots sont reconnus par l'automate. On ne demande pas de déterminer le langage reconnu *exactement* par l'automate mais seulement une partie infinie de ce dernier. La réponse devra être justifiée.

**Question 4** Donner une version déterministe de l'automate.

## 2 Logique propositionnelle

### Exercice 3

Étudier la nature de la formule suivante en utilisant l'algorithme de Quine sans les équivalences notables générales (on utilisera seulement les équivalences associées à l'algorithme de Quine qui font toutes intervenir $\top$ et $\bot$) :

$$(((a \Rightarrow (c \vee d)) \wedge (b \Rightarrow (c \vee a)) \wedge (d \Rightarrow c)) \Rightarrow (b \Rightarrow c))$$

### Exercice 4

Étudier la nature de la formule suivante en utilisant le calcul des séquents sans les équivalences notables (on utilisera donc uniquement les 8 règles du calcul des séquents, ainsi que les axiomes associés).

$$((a \vee b) \wedge (b \Rightarrow c) \wedge (a \Rightarrow d)) \Rightarrow (c \vee d)$$

## 3 Preuves de programmes

### Exercice 5

On étudie le programme suivant :

prog

```
s ← 0
k ← 1
if ( n < 0 ) {
  m ← -n
} else {
  m ← n
}
while ( k <= m ) {
  u ← 1
  l ← 1
  while  ( l <= p ) {
    u ← u * k
    l ← l + 1
  }
  s ← s + u
  k ← k +1
}
```

On cherche à montrer que ce programme calcule $\sum_{i=1}^{|n|} i^p$, avec la convention classique que pour tout $l \geq 0$, $\sum_{i=l+1}^{l} i^p = 0$. On cherche plus précisément à montrer que le triplet suivant est totalement correct :

$$\langle p \geq 0 \rangle \ \texttt{prog} \ \left\langle s = \sum_{i=1}^{|n|} i^p \right\rangle.$$

Dans les questions qui suivent, on peut toujours supposer que les résultats des questions précédentes ont été démontrés pour prouver ce qui est demandé.

On commence par traiter le problème de la valeur absolue.

**Question 1** Montrer que le triplet $\langle\rangle \ \texttt{prog}_{3\to 7} \ \langle m = |n| \rangle$ est totalement correct. On rappelle que $\texttt{prog}_{i\to j}$ désigne les lignes $i$ à $j$ (incluses) du programme $\texttt{prog}$.

On étudie ensuite la boucle interne. Pour ce faire, on introduit la formule suivante :

$$J \equiv (l \leq p+1) \wedge (u = k^{l-1})$$

**Question 2** Montrer que le triplet $\langle (p \geq 0) \wedge (k > 0) \rangle \ \texttt{prog}_{9\to 10} \ \langle J \rangle$ est totalement correct. Que dire de la condition $k > 0$ ?

On définit $W$ comme le terme $p - l + 1$.

**Question 3** Montrer que le triplet suivant est totalement correct :

$$\left\langle (l \leq p) \wedge (u = k^{l-1}) \wedge (p - l + 1 = z) \right\rangle \ \texttt{prog}_{12\to 13} \ \langle J \wedge (p - l + 1 < z) \rangle$$

**Question 4** Déduire des deux résultats précédents que le triplet suivant est totalement correct :

$$\langle (p \geq 0) \wedge (k > 0) \rangle \ \texttt{prog}_{9\to 14} \ \langle u = k^p \rangle$$

On détaillera avec précision le raisonnement utilisé (comme pour toutes les questions !).

Pour traiter la boucle externe, on introduit la formule suivante :

$$L \equiv (p \geq 0) \wedge (0 < k \leq m+1) \wedge (s = \sum_{i=1}^{k-1} i^p)$$

**Question 5** Montrer que le triplet suivant est totalement correct

$$\langle p \geq 0 \rangle \ \texttt{prog}_{1\to 7} \ \langle L \rangle$$

**Question 6** En choisissant un terme $V$ adapté, montrer que le triplet suivant est totalement correct

$$\left\langle (p \geq 0) \wedge (0 < k \leq m) \wedge (s = \sum_{i=1}^{k-1} i^p) \wedge (V = y) \right\rangle \ \texttt{prog}_{9\to 16} \ \langle L \wedge (V < y) \wedge (V \geq 0) \rangle$$

Pour simplifier la preuve, on admettra que le triplet suivant est totalement correct :

$$\langle (p \geq 0) \wedge (k > 0) \rangle \ \texttt{prog}_{9\to 14} \ \langle (u = k^p) \wedge (p \geq 0) \wedge (k > 0) \rangle$$

**Question 7** Déduire des résultats précédents que le triplet suivant est totalement correct :

$$\langle L \rangle \ \texttt{prog}_{8\to 17} \ \left\langle s = \sum_{i=1}^{m} i^p \right\rangle$$

**Question 8** Montrer finalement que le triplet $\langle p \geq 0 \rangle \ \texttt{prog} \ \left\langle s = \sum_{i=1}^{|n|} i^p \right\rangle$ est totalement correct.