L2/L3 MASS/MIAGE | **Année universitaire 2013–2014**
Université Paris 1

Examen de mathématiques pour l'informatique
Lundi 23 juin – Durée 2 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 ne sont pas autorisés. L'utilisation de calculatrices, téléphones portables ou tout autre appareil électronique, est interdite.

1 Automates

Exercice 1

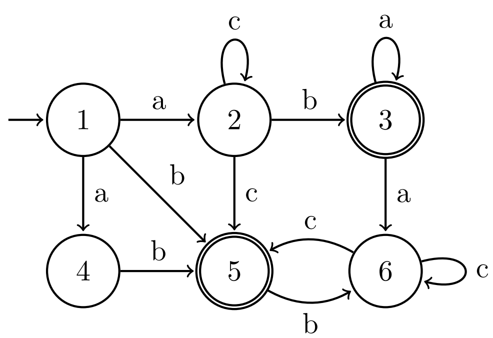
On étudie le langage rationnel $L = (ab|bc)(b*)c$ défini sur l'alphabet $\Sigma = \{a, b, c\}$.

Question 1 En utilisant l'algorithme de Thompson, déterminez un automate reconnaissant L. Attention, il ne faut pas simplifier l'automate obtenu !

Question 2 Dans l'automate obtenu, supprimez les ε-transitions, puis les états inutiles, en appliquant l'algorithme *backward*. Détaillez les deux étapes de façon claire.

Exercice 2

Soit l'automate suivant défini sur l'alphabet $\Sigma = \{a, b, c\}$:



Construisez une version déterministe de cet automate par la méthode de votre choix.

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 $\bot$) :

$$\big((c \vee a) \wedge (c \vee (\neg b))\big) \Rightarrow \big(b \vee ((\neg c) \vee (c \Rightarrow a))\big)$$

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) :

$$\Big(\big((d \Rightarrow c) \wedge b\big) \vee \big((d \Rightarrow a) \wedge (\neg b)\big)\Big) \Rightarrow \Big(d \Rightarrow \big((c \wedge b) \vee (a \wedge (\neg b))\big)\Big)$$

3 Preuves de programmes

Exercice 5

Soit le programme suivant :

prog

```
if (x > 0)
  x ← x - 1
  y ← y + 1
else
  y ← y + 2
end if
```

Les variables `x` et `y` sont supposées entières. Montrez que le triplet suivant est totalement correct :

$$\langle x > y \rangle \ \texttt{prog} \ \langle x > y - 2 \rangle$$

Exercice 6

Soit le programme suivant :

prog

```
x ← 0
while(x<y)
  x ← x + 1
end while
```

Dans ce programme `x` est une variable entière et `y` une variable réelle. On pose $I \equiv x < (y+1)$ et $V \equiv y + 1 - x$.

Question 1 Montrez que le triplet suivant est totalement correct :

$$\langle (y \geq 0) \rangle \ \texttt{x} \ \leftarrow \texttt{0} \ \langle I \wedge (y \geq 0) \rangle$$

Question 2 Montrez que le triplet suivant est totalement correct :

$$\langle I \wedge (V = n) \wedge (x < y) \wedge (y \geq 0) \rangle \ \texttt{x} \ \leftarrow \texttt{x + 1} \ \langle I \wedge (V = n - 1) \wedge (y \geq 0) \wedge (V \geq 0) \rangle$$

Question 3 Déduire des questions précédentes que le triplet suivant est totalement correct :

$$\langle (y \geq 0) \rangle \ \texttt{prog} \ \langle x \in [y, y+1[\rangle$$

Question 4 Que se passe-t-il si on ne fait plus l'hypothèse $y \geq 0$?