L2/L3 MASS/MIAGE **Année universitaire 2011–2012**
Université Paris 1

Examen de mathématiques pour l'informatique
Mercredi 13 juin – Durée 2 heures
Sujet 1

Les exercices sont indépendants. Au sein d'un exercice, il est conseillé de traiter les questions dans l'ordre. 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.

Exercice 1

On considère le programme suivant dont les paramètres sont le nombre réel x et l'entier n (on rappelle que `t` étant un entier, `t % 2` désigne le reste de la division entière de `t` par 2) :

prog

```
1  y ← 1
2  s ← 0
3  p ← 0
4  while ( p < n ) {
5    y ← y * x
6    p ← p + 1
7    if ( p % 2 == 0 ) {
8       s ← s + y
9    } else {
10      s ← s - y
11   }
12 }
```

On cherche à montrer que le triplet suivant est vrai

$$\langle n \geq 0 \wedge x > 0 \rangle \ \texttt{prog} \ \left\langle s = \sum_{k=1}^{n} (-1)^k x^k \right\rangle,$$

avec la convention que $\sum_{k=1}^{n}(-1)^k x^k = 0$ quand $n = 0$. On note V l'expression $n - p$ et I le prédicat $(p \leq n) \wedge (s = \sum_{k=1}^{p}(-1)^k x^k) \wedge (y = x^p)$.

On rappelle que le but de l'exercice n'est pas d'exécuter le programme « à la main » mais d'établir une preuve mathématique de la correction du calcul effectué. Les réponses doivent donc s'appuyer sur les règles d'inférence valables pour les triplets de Hoare.

Question 1 Montrer que le triplet suivant est vrai :

$$\langle n \geq 0 \wedge x > 0 \rangle \ \texttt{prog}_{1\to 3} \ \langle I \rangle,$$

où $\texttt{prog}_{i\to j}$ désigne les lignes i à j (incluses) du programme `prog` (ici, les trois premières lignes).

Question 2 Montrer que le triplet suivant est vrai :

$$\left\langle p < n \wedge s = \sum_{k=1}^{p} (-1)^k x^k \wedge y = x^p \right\rangle \ \texttt{prog}_{5\to 6} \ \left\langle p \leq n \wedge s = \sum_{k=1}^{p-1} (-1)^k x^k \wedge y = x^p \right\rangle.$$

Question 3 Montrer que le triplet suivant est vrai :

$$\left\langle p \leq n \wedge s = \sum_{k=1}^{p-1} (-1)^k x^k \wedge y = x^p \right\rangle \ \texttt{prog}_{7\to 11} \ \left\langle p \leq n \wedge s = \sum_{k=1}^{p} (-1)^k x^k \wedge y = x^p \right\rangle.$$

Question 4 Montrer que le triplet suivant est vrai :

$$\langle V = t \rangle \ \texttt{prog}_{5\to 11} \ \langle V < t \rangle.$$

Question 5 En utilisant les résultats précédents, montrer que le triplet suivant est vrai :

$$\langle I \rangle \ \texttt{prog}_{4\to 12} \ \left\langle s = \sum_{k=1}^{n} (-1)^k x^k \right\rangle.$$

Question 6 Déduire des analyses précédentes que le triplet suivant est vrai :

$$\langle n \geq 0 \wedge x > 0 \rangle \ \texttt{prog} \ \left\langle s = \sum_{k=1}^{n} (-1)^k x^k \right\rangle.$$

Exercice 2

On considère la fonction récursive suivante :

```
1  bidule(n,p) {
2    if(n == 0) {
3      return p
4    } else {
5      z ← bidule(n - 1, p + 1)
6      k ← 0
7      while (k < n) {
8        z ← z + k
9        k ← k + 1
10     }
11     return z
12   }
13 }
```

Question 1 Expliquer en détail pourquoi la valeur de `p` n'a pas d'influence sur le temps de calcul de `bidule(n,p)`.

Question 2 Calculer le nombre *exact* d'additions réalisées par les lignes 6 à 10 du programme en fonction de `n` (supposé positif ou nul).

Question 3 On note $T(\texttt{n})$ le nombre *exact* d'additions réalisées par l'appel à `bidule(n,p)` (pour `n` positif ou nul). Établir une relation de récurrence entre $T(\texttt{n})$ et $T(\texttt{n-1})$ en justifiant la réponse. Pour les besoins de l'exercice, on considérera qu'une soustraction est une forme particulière d'addition.

Question 4 Donner la valeur de $T(\texttt{n})$ sous forme d'une expression arithmétique simple (sans récurrence) et en déduire l'ordre de grandeur (en Θ) du temps de calcul de `bidule(n,p)`.

Exercice 3

Question 1 Dessiner un automate *non déterministe* reconnaissant le langage suivant

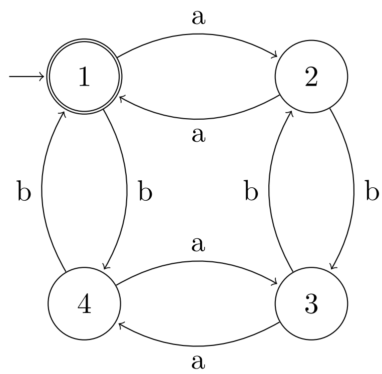
$$L = (b*)\Big(\big(b(a*)c\big)\Big|\big(a(c*)b\big)\Big)(a+),$$

défini sur l'alphabet $\Sigma = \{a,b,c\}$.

Question 2 Donner une version *déterministe* de l'automate précédent.

Exercice 4

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



On rappelle que X_i désigne le langage reconnu par l'automate en considérant i comme l'état initial. Dans l'automate ci-dessus, X_1 est donc le véritable langage reconnu.

Question 1 L'automate est-il déterministe ? Est-il complet ?

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

Question 3 Donner un exemple de mot dont la reconnaissance entraîne le passage par tous les états de l'automate (au moins une fois par état).

Question 4 Déterminer les équations satisfaites par les $(X_i)_{1\leq i\leq 4}$.

Question 5 Par des substitutions, exprimer X_1 et X_3 uniquement en fonction de X_1 et X_3.

Question 6 Appliquer le lemme de Arden à X_3 pour l'exprimer uniquement en fonction de X_1. On rappelle que le lemme de Arden indique que l'unique solution de $X = \alpha X|\beta$ est $X = \alpha^*\beta$ si $\varepsilon \notin \alpha$.

Question 7 En appliquant le lemme de Arden une nouvelle fois, déterminer le langage reconnu par l'automate.