Option Informatique en Spé MP et MP*

Devoir surveillé du mardi 13 décembre 2005

Résumé

Sujet constitué de trois exercices totalement indépendants. Les deux premiers portent sur les langages et permettront de vérifier vos connaissances sur les automates finis, les expressions rationnelles et le lemme de l'étoile. Le troisième exercice propose diverses questions de combinatoire, d'algorithmique et de programmation, en prenant comme objet d'étude le jeu *Sudoku*.

Veuillez rédiger chaque partie sur une copie séparée.

Table des matières

1 Autour du lemme de l'étoile

▶ Dans cet exercice, Σ désigne l'alphabet $\{a,b\}$. Nous nous intéressons au langage L, ensemble des mots de la forme $a^{e_1}ba^{e_2}b\dots a^{e_p}b$, où p est un nombre premier, et les e_k sont des naturels non nuls. Par exemple, $abaab$ et $aabaababaabab$ appartiennent à L ; en revanche, ni $aaabba$ ni $aababababaaab$ n'appartiennent à L. Rappel : 1 n'est pas un nombre premier.

Question 1 Énumérez les mots de L de longueur 6. On ne vous demande pas de preuve, juste une liste, présentée dans un ordre ≪raisonnable≫.

Question 2 Notons φ le morphisme de Σ^* défini par $\varphi(a) = \varphi(b) = a$. Décrivez simplement le langage $\varphi(\mathsf{L})$; bien entendu, vous donnerez une preuve à l'appui de votre affirmation.

Question 3 Peut-on utiliser le lemme de l'étoile pour montrer que L est rationnel ?

Question 4 Exhibez un langage rationnel M *très simple*, tel que $\mathsf{L} \cap \mathsf{M}$ ne soit pas rationnel.

Question 5 Qu'en déduisez-vous, concernant le langage L ?

Question 6 $\star\star$ Peut-on montrer que L n'est pas rationnel, en utilisant le lemme de l'étoile ?

2 Fibonacci revient !

▶ Dans cet exercice, Σ désigne l'alphabet $\{a,b\}$. Nous nous intéressons au langage F, ensemble des mots u vérifiant la propriété suivante : toute occurrence de b dans u est suivie d'au moins deux lettres, dont la deuxième est un a. Par exemple, les mots $abaa$ et $baaabbaa$ appartiennent à F ; en revanche, les mots ab, ba et $babaa$ n'appartiennent pas à F.

Question 1 Énumérez les mots de F de longueur au plus 4. On ne vous demande pas de preuve, juste une liste, présentée dans un ordre ≪raisonnable≫.

▶ Rappel : pour montrer qu'un automate $\mathcal{A}$ reconnaît un certain langage F, il faut prouver :

1. que tout mot de F est reconnu par $\mathcal{A}$;
2. que tout mot reconnu par $\mathcal{A}$ appartient à F.

Question 2 Construisez un automate fini $\mathcal{A}$ reconnaissant F. Il serait souhaitable que $\mathcal{A}$ ne comporte pas plus de cinq états.

▶ Rappel : pour montrer qu'une expression rationnelle e décrit un certain langage F, il faut prouver :

1. que tout mot de F est décrit par e ;
2. que tout mot décrit par e appartient à F.

Question 3 Donnez une expression rationnelle du langage F.

▶ Un mot sur l'alphabet Σ sera représenté par une `lettre list`, le type `lettre` étant défini comme suit :

```
type lettre = A | B ;;
```

Par exemple, le mot $ababb$ sera représenté par la liste `[A;B;A;B;B]`.

Question 4 • Rédigez en Caml une fonction de signature :

```
dans_L : lettre list -> bool
```

spécifiée comme suit : `dans_L u` indique si le mot u appartient à F. Objectif : cinq lignes. Vous n'utiliserez pas de gardes (construction `when...`).

▶ Notons n_k le nombre de mots de F de longueur k. Nous avons donc $n_0 = n_1 = n_2 = 1$, $n_3 = 2$ et $n_4 = 4$.

Question 5 Montrez que la suite $(n_k)_{k\in\mathbb{N}}$ vérifie une relation de récurrence linéaire que vous expliciterez.

Question 6 Avec cette relation et les valeurs précédemment citées, dressez un tableau donnant les valeurs de n_5 à n_{10}. Vous ne ferez pas apparaître les calculs sur votre copie.

▶ La suite de Fibonacci est définie par les valeurs $F_0 = 0$ et $F_1 = 1$ et la relation $F_{k+2} = F_{k+1} + F_k$ pour tout $k \in \mathbb{N}$. Rappel : $F_n = \dfrac{\varphi^n - (-1/\varphi)^n}{\sqrt{5}}$, où $\varphi = \dfrac{1+\sqrt{5}}{2}$ est le nombre d'or.

Question 7 Pour $k \in \mathbb{N}$, prouvez l'inégalité $n_k \geqslant F_k$.

Question 8 Existe-t-il un réel $\alpha > 0$ tel que $n_k = \mathcal{O}(k^\alpha)$ lorsque k tend vers l'infini ?

Question 9 Exhibez un réel $\beta > 1$ tel que $\beta^k = \mathcal{O}(n_k)$ lorsque k tend vers l'infini.

Question 10 Pour $k \geqslant 0$, prouvez l'égalité $n_{k+2} + n_k = F_{k+3}$.

Question 11 Exhibez un réel $\gamma > 1$ tel que $n_k = \mathcal{O}(\gamma^k)$ lorsque k tend vers l'infini.

Question 12 Le suspense s'éternise ! Résolvez l'équation caractéristique associée à la relation de récurrence établie à la question 5, et déduisez-en l'expression *exacte* de n_k. Bien entendu, vous ferez intervenir φ.

3 Autour du casse-tête de l'été 2005

▶ *Sudoku* est un casse-tête qui nous vient, paraît-il, de l'Orient mystérieux … L'objectif est de remplir avec les chiffres 1 à 9 les cases vides d'un tableau comme celui de gauche, tout en respectant la règle suivante : dans chaque ligne, chaque colonne et chaque «bloc», les chiffres 1 à 9 doivent apparaître une fois (et, donc, une seule).

1	3					8		
			5	3			2	
				8				6
		6				5		
	7	3				6	8	
		9				4		
4				7				
	6			5	2			
		7					4	1

0	**1**	**2**
3	**4**	**5**
6	**7**	**8**

1	3	5	2	4	6	8	9	7
6	9	8	5	3	7	1	2	4
7	2	4	1	8	9	3	5	6
8	4	6	7	2	3	5	1	9
5	7	3	9	1	4	6	8	2
2	1	9	8	6	5	4	7	3
4	8	2	3	7	1	9	6	5
9	6	1	4	5	2	7	3	8
3	5	7	6	9	8	2	4	1

Le tableau de droite donne une solution de ce casse-tête ; on peut montrer qu'il n'y en a pas d'autre.

▶ Désormais, nous appelerons *sudoku d'ordre 3* (ou tout simplement *sudoku*) une grille comme celle de droite. Notez qu'un sudoku d'ordre n contient n^4 cases.

▶ Les lignes sont numérotées de 0 à 8 (en allant du haut vers le bas) et les colonnes de 0 à 8 (en allant de la gauche vers la droite), conformément à la tradition Caml. Les neuf blocs sont numérotés de 0 à 8, selon le petit tableau central.

Question 1 Montrez que la case située ligne 1, colonne 0 doit contenir un 6.

Un peu de combinatoire

Question 2 Combien existe-t-il de sudokus d'ordre 2 ?

Question 3 Notons $N_s(3)$ le nombre de sudokus d'ordre 3. 9^{81} est un majorant très grossier de $N_s(3)$. Proposez, preuve à l'appui, un majorant de $N_s(3)$ inférieur à 10^{35}.

Un peu de programmation

▶ Un sudoku sera représenté par un objet `m` de type `int vect vect`. Ainsi, un fragment de programme construisant le sudoku de notre exemple pourrait comporter (entr'autres) les lignes suivantes :

```
let m = make_matrix 9 9 0 ;;
m.(0).(0) <- 1 ;;
...
m.(5).(2) <- 9 ;;
...
m.(8).(9) <- 1 ;;
```

Question 4 • Rédigez en Caml une fonction de signature :

```
flat : 'a list list -> 'a list
```

spécifiée comme suit : `flat l` «aplatit» la liste de listes ℓ ; ainsi, `flat [[2;7];[9;1;3];[3];[];[5;2]];;` rendra la liste `[2;7;9;1;3;3;5;2]`. Objectif : trois lignes.

▶ Pour les questions suivantes, on ne vous demande pas de tester la validité des arguments : g, d, h et b seront censés appartenir à l'intervalle $[\![0, 8]\!]$ et vérifier $g \leqslant d$ et $h \leqslant b$; m se devra d'être une brave matrice carrée d'ordre 9.

Question 5 • Rédigez en Caml une fonction de signature :

```
list_of_bande : int -> int -> int -> 'a vect vect -> 'a list
```

spécifiée comme suit : `list_of_bande h g d m` construit la liste (présentée dans un ordre quelconque) des $m_{h,j}$ où j décrit $[\![g, d]\!]$. Objectif : deux lignes ; vous pouvez utiliser `intervalle`, à condition de rappeler sa définition.

Question 6 • Rédigez en Caml une fonction de signature :

```
list_of_zone : int -> int -> int -> int -> 'a vect vect -> 'a list
```

spécifiée comme suit : `list_of_zone h b g d m` construit la liste (présentée dans un ordre quelconque) des $m_{i,j}$ où (i, j) décrit $[\![h, b]\!] \times [\![g, d]\!]$. Objectif : deux lignes.

Question 7 • Rédigez en Caml une fonction de signature :

```
list_of_bloc : int -> 'a vect vect -> 'a list
```

spécifiée comme suit : `list_of_bloc k m` construit la liste (présentée dans un ordre quelconque) des $m_{i,j}$ où (i, j) décrit le bloc numéro k. Objectif : trois lignes.

Question 8 • Rédigez en Caml une fonction de signature :

```
bonne_liste : int list -> bool
```

spécifiée comme suit : `bonne_liste l` indique si la liste ℓ contient, à l'ordre près, les entiers de 1 à 9. Objectif : deux lignes, si possible !

Question 9 • Rédigez en Caml une fonction de signature :

```
bon_sudoku : int vect vect -> bool
```

spécifiée comme suit : `bon_sudoku m` indique si le tableau m contient effectivement un sudoku. Objectif : sept lignes.

Questions pour les 5/2 (enfin, pas uniquement)

Question 10 • Montrez que la résolution d'un sudoku peut être réalisée avec un parcours en profondeur d'un arbre.

▶ Un *graphe non orienté* est un couple $G = (S, A)$ où S est l'ensemble (fini) des *sommets* du graphe, et A un ensemble de paires de sommets ; chaque paire est une *arête* du graphe. Nous dirons que les extrémités d'une arête sont des sommets *voisins*, et qu'ils sont *reliés* par l'arête. Un k-*coloriage* d'un graphe $G = (S, A)$ est une fonction $c : S \mapsto [\![1, k]\!]$ telle que $c(i) \neq c(j)$ lorsque i et j sont deux sommets voisins.

Question 11 • Montrez que résoudre un sudoku revient à colorier un graphe dont vous préciserez les sommets et les arêtes.

FIN