Option Informatique en Spé MP et MP*

Devoir surveillé du mardi 12 décembre 2006

Résumé

Ce sujet comporte deux parties : un exercice de programmation et un problème.

L'exercice de programmation sera l'occasion de raisonner par induction structurelle, et de prouver le comportement d'un programme.

Le problème porte sur la détermination d'un majorant de la *longueur discriminante* de deux automates finis non équivalents (c'est-à-dire : qui ne reconnaissent pas le même langage). Pour ce faire, nous mettrons en œuvre des idées issues de l'algèbre linéaire. Ce sujet a été posé au concours Mines-Ponts en 2004.

Veuillez rédiger chaque partie sur une copie séparée.

Table des matières

1 Exercice de programmation

Question 1 Rédigez en Caml une fonction de signature :

```
filtre : ('a -> bool) -> 'a list -> 'a list
```

spécifiée comme suit : `filtre p l` dresse la liste (présentée dans un ordre quelconque) des éléments de ℓ qui satisfont le prédicat p. Par exemple, `filtre (fun x -> x>0) [-2;3;7;-1;0;11;-8]` doit rendre (à l'ordre près) la liste `[3;7;11]`.

Question 2 Rédigez en Caml une fonction de signature :

```
cherche_paires : 'a list -> 'a list
```

spécifiée comme suit : `cherche_paires l` dresse la liste (présentée dans un ordre quelconque) des éléments qui apparaissent *exactement* deux fois dans la liste ℓ. Par exemple, `cherche_paires [8;2;3;8;1;3;1;9;7]` doit rendre (à l'ordre près) la liste `[8;3;1]`.

▶ Nous définissons la fonction :

```
let rec castor c = match c with
 | (_,[]) | (_,[_]) -> c
 | (t,a::b::q) when a<=b -> let (t',r) = castor(t,b::q) in (t',a::r)
 | (t,a::b::q) -> let (_,r) = castor(t,a::q) in (true,b::r) ;;
```

Clairement, le type de `castor` est `(bool,'a list) -> (bool,'a list)`. Soient b un booléen, $\ell = (x_1, \dots, x_n)$ une liste, et (b', ℓ') le résultat rendu par `castor(b,l)`.

▶ Pour les trois questions suivantes, les preuves doivent se faire par induction structurelle.

Question 3 Montrez que ℓ' a même longueur que ℓ.

▶ Pour les trois questions suivantes, nous supposons $n \geqslant 1$; autrement dit, la liste ℓ n'est pas vide.

Question 4 Nous avons donc $\ell' = (y_1, \dots, y_n)$. Montrez que les listes ℓ et ℓ' contiennent les mêmes éléments, à l'ordre près.

Question 5 Montrez que y_n est le plus grand élément de ℓ.

Question 6 Caractérisez les listes ℓ pour lesquelles l'évaluation de `castor(false,l)` donne un couple dont la première composante est `true`.

Question 7 Nous définissons la fonction :

```
let rec pollux l =
 match castor(false,l) with
  | (false,r) -> r
  | (true,r) -> pollux r ;;
```

Quel est le résultat rendu par `pollux l` où `l` est une liste $(x_1, \dots, x_n)$?

Question 8 L'unité de coût est la comparaison entre deux éléments d'une liste. Exprimez le coût maximal de l'évaluation de `pollux l`, en fonction de la longueur n de la liste ℓ.

2 Longueur discriminante de deux automates finis

▶ Deux automates finis $\mathcal{A}$ et $\mathcal{A}'$ (déterministes ou non) sont équivalents s'ils reconnaissent le même langage. S'ils ne sont pas équivalents, alors il existe des mots qui sont reconnus par l'un et pas par l'autre ; un tel mot est un *séparateur* de $\mathcal{A}$ et $\mathcal{A}'$. La longueur discriminante de $\mathcal{A}$ et $\mathcal{A}'$ est la longueur minimale d'un séparateur. L'objet de ce problème est d'évaluer, par deux méthodes, un majorant de la longueur discriminante de $\mathcal{A}$ et $\mathcal{A}'$ en fonction des nombres d'états de $\mathcal{A}$ et $\mathcal{A}'$.

▶ Rappelons que tout automate ayant n états est équivalent à un automate déterministe complet ayant au plus 2^n états.

Approche naïve

Question 1 Soit $\mathcal{A}$ un automate, déterministe ou non déterministe, avec n états. Montrez que $L(A)$ est vide si et seulement s'il ne contient aucun mot de longueur inférieure ou égale à $n-1$.

Question 2 Soit $\mathcal{A}$ un automate déterministe complet ayant n états. Donnez un automate ayant aussi n états qui reconnaît le complémentaire $\overline{L(A)}$ de $L(A)$ dans Σ^*. Vous justifierez votre réponse.

Question 3 Soient $\mathcal{A}$ et $\mathcal{A}'$ deux automates utilisant le même alphabet Σ avec respectivement n et n' états. Donnez un automate ayant $n \times n'$ états qui reconnaît $L(A) \cap L(A')$. Vous justifierez votre réponse.

Question 4 Soient $\mathcal{A}$ et $\mathcal{A}'$ deux automates déterministes complets utilisant le même alphabet Σ avec respectivement n et n' états. Montrez que si $\mathcal{A}$ et $\mathcal{A}'$ ne sont pas équivalents, il existe un mot de longueur au plus $n \times n' - 1$ qui est reconnu par l'un et non par l'autre, i.e. la longueur discriminante de $\mathcal{A}$ et $\mathcal{A}'$ est inférieure ou égale à $n \times n' - 1$.

Question 5 En déduire un majorant pour la longueur discriminante de deux automates non équivalents quelconques avec n et n' états respectivement.

▶ L'objectif de cette partie est de démontrer que la longueur discriminante de deux automates déterministes non équivalents, ayant respectivement n et n' états, est inférieure ou égale à $n + n' - 1$.

Question 6 Montrez sur un exemple, avec un alphabet à une seule lettre, qu'il existe des automates déterministes $\mathcal{A}$ et $\mathcal{A}'$ non équivalents et qui reconnaissent les mêmes mots de longueur strictement inférieure à $n + n' - 1$, où n (resp. n') désigne le nombre d'états de $\mathcal{A}$ (resp. de $\mathcal{A}'$). Indication : les automates sont déterministes, mais rien ne les oblige à être complets !

Approche plus fine

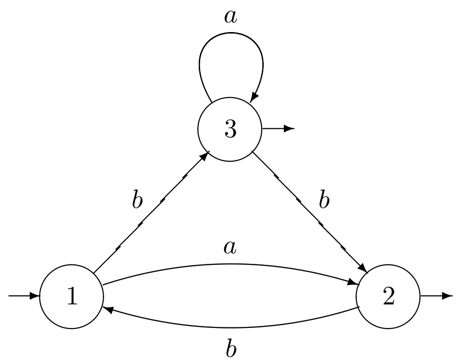
▶ Soit $A = (\Sigma, Q, \delta, i, F)$ un automate déterministe à n états. Sans perte de généralité, nous supposons que $Q = [\![1, n]\!]$ et $i = 1$. Notons $(e_1, e_2, \dots, e_n)$ la base canonique de l'espace vectoriel $\mathbb{R}^n$. Pour $i \in [\![1, n]\!]$, le vecteur e_j est donc le vecteur de $\mathbb{R}^n$ dont toutes les composantes sont nulles sauf la j-ime qui vaut 1. Notons $\mathbf{0}$ le vecteur nul de $\mathbb{R}^n$. Pour chaque lettre a de l'alphabet Σ, définissons une application linéaire φ_a de $\mathbb{R}^n$ dans $\mathbb{R}^n$ comme suit : pour j et k appartenant à $[\![1, n]\!]$, $\varphi_a(e_j) = e_k$ si $\delta(j, a) = k$; et $\varphi_a(e_j) = \mathbf{0}$ sinon. L'existence et l'unicité de l'application φ_a découlent du fait que $\mathcal{A}$ est déterministe et de la linéarité de φ_a. Soit $m = a_1 a_2 \dots ak$ est un mot non vide de Σ^* ; nous noterons $\varphi_m = \varphi_{a_k} \circ \dots \circ \varphi_{a_1}$. En particulier, φ_ε est l'identité de $\mathbb{R}^n$. Nous noterons $z = \displaystyle\sum_{f \in F} e_f$. Enfin, le produit scalaire des deux vecteurs u et v de $\mathbb{R}^n$ sera noté $u \cdot v$.

▶ Sur l'exemple de la figure, nous avons $\varphi_a(e_1) = e_2$, $\varphi_a(e_2) = \mathbf{0}$, $\varphi_a(e_3) = e_3$, $\varphi_b(e_1) = e_3$, $\varphi_b(e_2) = e_1$, $\varphi_b(e_3) = e_2$ et $z = (0, 1, 1)$. Nous avons (entr'autres) les égalités $\varphi_{abb}(e_1) = e_3$, $\varphi_{ba}(e_3) = \mathbf{0}$ et $\varphi_{ba}(e_2) = e_2$.

Question 7 Soient $m \in \Sigma^*$ et $j, k \in [\![1, n]\!]$. Montrez qu'il existe un calcul d'origine j, d'extrémité k et d'étiquette m si et seulement si $\varphi_m(e_j) = e_k$.

Question 8 Soit $m \in \Sigma^*$. Donnez les valeurs possibles du produit scalaire $\varphi_m(e_1) \cdot z$ et indiquez une condition nécessaire et suffisante portant sur ce produit scalaire pour que m soit un mot reconnu par $\mathcal{A}$.

▶ Dans la suite du problème, nous considérons, en plus de $\mathcal{A}$, un deuxième automate déterministe $\mathcal{A}' = (Q', \delta', i', F')$; n' est le nombre d'états de $\mathcal{A}'$; les vecteurs de la base canonique de $\mathbb{R}^{n'}$ sont notés e'_j ; la fonction φ'_m est définie comme φ_m, en remplaçant les e_j par les e'_j ; et $z' = \displaystyle\sum_{f \in F'} e'_f$.



► Soient u et v deux vecteurs appartenant respectivement à $\mathbb{R}^n$ et $\mathbb{R}^{n'}$. Nous noterons $(u;v)$ le vecteur w de $\mathbb{R}^{n+n'}$ obtenu en concaténant u et v ; plus précisément, le vecteur w est défini par $w_i = u_i$ si $1 \leqslant i \leqslant n$ et $w_i = v_{i-n}$ si $n+1 \leqslant i \leqslant n+n'$. Par exemple, si $u = (0,1,0)$ et $v = (1,2)$, alors $(u;v)$ est le vecteur $(0,1,0,1,2)$. Nous noterons $E = (e_1; -e'_1)$; notez bien le signe ≪moins≫.

► À tout mot $m \in \Sigma^*$, nous associons l'application linéaire Φ_m de $\mathbb{R}^n$ dans $\mathbb{R}^n$ définie par $\Phi_m\big(u;u'\big) = \big(\varphi_m(u); \varphi'_m(u')\big)$ pour tout $u \in \mathbb{R}^n$ et pour tout $u' \in \mathbb{R}^{n'}$; nous admettrons la linéarité de Φ_m. Nous noterons $Z = (z; z')$.

Question 9 Soit $m \in \Sigma^*$; indiquez les valeurs possibles du produit scalaire $\Phi_m(E) \cdot Z$ et, selon ces valeurs, précisez l'appartenance de m à $L(A)$ et $L(A')$.

► Soit $k \in \mathbb{N}$; nous notons V_k le sous-espace vectoriel de $\mathbb{R}^{n+n'}$ engendré par la famille des vecteurs $\Phi_m(E)$ où m décrit l'ensemble des mots de longueur au plus k sur l'alphabet Σ. En particulier, V_0 est engendré par le vecteur $(e_1; -e'_1)$.

Question 10 Pour $k \geqslant 0$, montrez que V_k est contenu dans V_{k+1}.

Question 11 Soient $m \in \Sigma^*$ et $a \in \Sigma$. Montrez l'égalité $\Phi_{ma} = \Phi_a \circ \Phi_m$.

Question 12 Soit $w \in V_k$ et $a \in \Sigma$; montrez que $\Phi_a(w)$ appartient à V_{k+1}.

Question 13 Supposons qu'il existe $k \geqslant 0$ tel que $V_k = V_{k+1}$; montrez l'égalité $V_{k+2} = V_{k+1}$.

Question 14 Montrez qu'il existe un entier $h \leqslant n + n' - 1$ tel que $V_k = V_h$ pour tout $k \geqslant h$.

Question 15 Montrez que si $\mathcal{A}$ et $\mathcal{A}'$ ne sont pas équivalents, il existe un mot de longueur inférieure ou égale à $n+n'-1$ qui est accepté par l'un et pas par l'autre. Ceci prouve que $n+n'-1$ est un majorant de la longueur discriminante de $\mathcal{A}$ et $\mathcal{A}'$.

Question 16 En déduire un majorant de la longueur discriminante de deux automates quelconques ayant n et n' états respectivement.

FIN