# Option Informatique en Spé MP et MP*

## Devoir surveillé du mardi 7 décembre 2004

**Résumé**

À partir du modèle classique de l'automate fini (déterministe ou non), de nombreuses variantes ont été étudiées : automate avec bande de sortie (sur laquelle on peut écrire un mot), automate boustrophédon (la tête de lecture peut revenir en arrière) ... Dans un automate *pondéré*, chaque transition est étiquetée par la lettre lue et par un poids ; le poids d'un calcul est la somme des poids des transitions constituant ce calcul.

Nous nous intéressons dans le premier problème à un modèle particulier d'automate pondéré : les poids appartiennent au groupe $(\mathbb{Z}, +)$ ; un mot est reconnu s'il est l'étiquette d'un calcul de poids nul commençant dans l'état initial. Nous verrons que la famille des langages reconnus par ces automates possède certaines des propriétés des langages rationnels, mais pas toutes. Les idées ont été trouvées dans un chapitre du livre *Jewels are Forever*, recueil d'articles écrits en l'honneur d'Arto Salomaa pour son soixante-cinquième anniversaire.

Le deuxième problème demande la définition par induction structurelle puis la programmation en Caml de quelques fonctions définies sur la famille des langages rationnels sur un alphabet fixé.

*Veuillez rédiger chaque partie sur une copie séparée.*

# Table des matières

# 1 Automates finis pondérés, d'après Vesa Halava et Tero Harju

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

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

▶ Un langage $L$ sera dit *propre* lorsqu'il ne contient pas le mot vide $\varepsilon$.

▶ Nous dirons qu'un automate fini non déterministe $\mathcal{A} = (Q, \delta, I, F)$ est *réduit* s'il vérifie les conditions suivantes :

- $\mathcal{A}'$ possède un seul état initial $i$ ;
- $\mathcal{A}'$ possède un seul état final $f$, distinct de $i$ ;
- aucune transition de $\mathcal{A}'$ ne mène à $i$ ;
- aucune transition de $\mathcal{A}'$ ne part de $f$.

▶ Il est interdit d'utiliser des automates à transitions instantanées (dites aussi $\varepsilon$-transitions). Ceci s'applique aux questions 1 et 5.

***Question 1*** Soit $L$ un langage propre, reconnu par un automate fini non déterministe $\mathcal{A} = (Q, \delta, I, F)$. Montrez que $L$ peut être reconnu par un automate fini non déterministe réduit $\mathcal{A}'$. Vous donnerez une preuve constructive, en expliquant précisément comment déduire $\mathcal{A}'$ de $\mathcal{A}$.

▶ Fixons un alphabet $\Sigma$ contenant au moins les deux lettres $a$ et $b$. Soit $G$ un groupe noté multiplicativement. Un $G$-automate est un triplet $\mathcal{A} = (Q, i, \delta)$ où :

- $Q$ est un ensemble fini non vide, dont les éléments sont les *états* de $\mathcal{A}$ ;
- $i \in Q$ est l'*état initial* ;
- $\delta \subset Q \times \Sigma \times G \times Q$ est la *table des transitions.*

▶ Nous noterons $q \xrightarrow{(x,g)} q'$ lorsque $(q, x, g, q')$ est une transition de $\mathcal{A}$. Ceci revient à associer à $\mathcal{A}$ un graphe orienté étiqueté, dont les sommets sont les éléments de $Q$, et les étiquettes des arcs sont des éléments de $\Sigma \times G$. Notez bien que, dans un $G$-automate, il n'y a pas de transitions instantanées.

▶ Un *calcul* de $\mathcal{A}$ de longueur $n \geqslant 1$ est une suite $(q_{j-1}, x_j, g_j, q_j)_{1 \leqslant j \leqslant n}$ de transitions. L'*étiquette* de ce calcul est le mot $x = x_1 x_2 \ldots x_n$, son *poids* est $g = g_1 g_2 \ldots g_n$. Un calcul est *réussi* s'il commence dans l'état $i$, et si son poids est le neutre de $G$ ; notez que le mot vide ne peut pas être reconnu, puisque nous ne considérons que des calculs de longueur non nulle. Le *langage reconnu* par $\mathcal{A}$ est l'ensemble des calculs réussis de $\mathcal{A}$.

***Question 2*** Dans cette question, $G$ est le groupe $(\mathbb{Z}, +)$. Notons $M_{\text{eq}}$ le langage reconnu par le $G$-automate $\mathcal{A}_{\text{eq}}$ représenté à la figure 1. Quel est ce langage bien connu ?

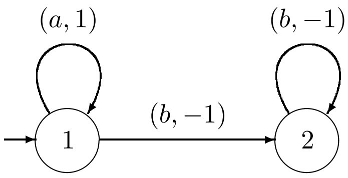


Figure 1: l'automate $\mathcal{A}_{\text{eq}}$

▶ Le résultat de la question précédente montre que la classe des $G$-automates reconnaît des langages non rationnels.

***Question 3*** Proposez un $Z$-automate reconnaissant le langage $M_{\text{diff}} = \{a^p b^q \mid p, q \in \mathbb{N} \text{ et } p \neq q\}$.

***Question 4*** Soient $G$ un groupe non réduit à son neutre $e$, et $L$ un langage rationnel propre reconnu par un automate $\mathcal{A} = (Q, \delta, I, F)$. Montrez que $L$ peut être reconnu par un $G$-automate $\mathcal{B}$. *Indications* : utilisez le résultat de la question 1 ; choisissez un élément un élément $g$ de $G$ autre que $e$ et donnez aux transitions un poids égal à $e$, $g$ ou $g^{-1}$.

***Question 5*** Soient $G$ un groupe fini et $\mathcal{A}$ un $G$-automate. Montrez que le langage $L$ reconnu par $\mathcal{A}$ est rationnel ; vous construirez un automate fini non déterministe qui reconnaît $L$.

► Dans la suite, nous fixons $G = (\mathbb{Z}, +)$ ; un $\mathbb{Z}$-langage est un langage reconnu par un $\mathbb{Z}$-automate. Nous allons établir quelques propriétés de la classe des $\mathbb{Z}$-langages.

***Question 6*** Soient $L_1$ et $L_2$ deux $\mathbb{Z}$-langages. Montrez que $L_1 \cup L_2$ est également un $\mathbb{Z}$-langage.

***Question 7*** Soient $L$ un $\mathbb{Z}$-langage et $R$ un langage rationnel propre. Montrez que $R \cdot L = \{uv \mid u \in R \text{ et } v \in L\}$ est également un $\mathbb{Z}$-langage.

► On montrerait de même que $L \cdot R$ est un $\mathbb{Z}$-langage.

***Question 8*** Soient $R$ un langage rationnel et $L$ un $\mathbb{Z}$-langage. Montrez que $R \cap L$ est également un $\mathbb{Z}$-langage.

► Nous allons montrer que la famille des $\mathbb{Z}$-langages n'est pas close par intersection. Définissons trois langages : $L_1 = \{a^n b^n \mid n \geqslant 1\}$, $L_2 = \{b^n c^n \mid n \geqslant 1\}$ et $M = \{a^n b^n c^n \mid n \geqslant 1\}$.

***Question 9*** Exhibez deux langages rationnels $R_1$ et $R_2$ tels que $M = (L_1 \cdot R_1) \cap (R_2 \cdot L2)$.

***Question 10*** Soit $u \in a^* b^* c^*$. Montrez qu'il n'existe aucune décomposition $u = pqr$ vérifiant $q \neq \varepsilon$ et $pq^k r \in M$ pour tout $k \in \mathbb{N}$.

***Question 11*** Soit $u \in a^* b^* c^*$. Montrez qu'il n'existe aucune décomposition $u = pqrst$ vérifiant $q \neq \varepsilon$, $s \neq \varepsilon$ et $pq^k rs^k t \in M$ pour tout $k \in \mathbb{N}$.

► Dans la suite, $\mathcal{A}$ désigne un $\mathbb{Z}$-automate reconnaissant un $\mathbb{Z}$-langage infini. Soit $(q_{j-1}, x_j, g_j, q_j)_{1 \leqslant j \leqslant n}$ un calcul de $\mathcal{A}$. Nous dirons que ce calcul contient un *cycle* s'il existe deux indices $k$ et $\ell$ tels que $0 \leqslant k < \ell \leqslant n$ et $q_k = q_\ell$.

***Question 12*** Justifiez l'affirmation suivante : il existe au moins un calcul réussi de $\mathcal{A}$ contenant au moins un cycle.

***Question 13*** Montrez que s'il existe un calcul réussi de $\mathcal{A}$ contenant au moins un cycle de poids nul, alors cet automate ne peut reconnaître $M$. Indication : utilisez le résultat de la question 10.

► Supposons maintenant que $\mathcal{A}$ ne contient aucun cycle de poids nul.

***Question 14*** Montrez qu'il existe un calcul réussi de $\mathcal{A}$ contenant au moins un cycle de poids strictement positif.

► Par raison de symétrie, il existe un calcul réussi de $\mathcal{A}$ contenant au moins un cycle de poids strictement négatif.

***Question 15*** $\star\star\star$ Montrez qu'il existe un calcul réussi de $\mathcal{A}$ contenant deux cycles dont les poids sont de signes opposés. *Vous pourrez admettre ce résultat, au besoin.*

***Question 16*** Montrez qu'il existe un calcul réussi de $\mathcal{A}$ contenant deux cycles de poids opposés.

► Soit $\ell$ la longueur de ce calcul, et $(q_i)_{0 \leqslant i \leqslant \ell}$ la suite des états traversés. Notons $\alpha$ et $\omega$ (resp. $\alpha'$ et $\omega'$) les indices de début et de fin d'un cycle de poids positif (resp. négatif) apparaissant dans ce calcul. Nous avons donc $\alpha < \omega$ et $\alpha' < \omega'$ ; et on ne peut avoir à la fois $\alpha = \alpha'$ et $\omega = \omega'$.

***Question 17*** Montrez que ces deux cycles ne sont ni imbriqués ni enchevêtrés ; on a donc $\alpha' \leqslant \omega$ ou $\omega' \leqslant \alpha$.

***Question 18*** En utilisant les résultats des questions 9 à 17, montrez qu'aucun $\mathbb{Z}$-automate ne peut reconnaître le langage $M$.

► Le *complément* d'un $\mathbb{Z}$-langage $L$ est $\Sigma^+ \setminus L$ : il faut exclure le mot-vide, puisque celui-ci ne peut être reconnu par un $\mathbb{Z}$-automate.

***Question 19*** La famille des $\mathbb{Z}$-langages est-elle close par complémentation ?

## 2 Longueur minimale d'un mot d'un langage rationnel

▶ Nous nous intéressons au problème suivant : étant donné un langage rationnel $L$ non vide sur un alphabet $\Sigma$, déterminer la longueur minimale d'un mot de $L$. Nous envisagerons deux situations : celle où $L$ est décrit par une expression rationnelle ; celle où $L$ est reconnu par un automate fini. L'algorithme devra détecter le cas où $L$ est vide.

▶ Nous étudions la première situation ; l'ensemble des expressions rationnelles sur $\Sigma$ sera noté $\mathcal{ER}(\Sigma)$. Le langage décrit par l'expression rationnelle $e$ sera noté $\mathcal{L}_{ER}(e)$.

***Question 1*** Définissez par induction structurelle une fonction $\lambda : \mathcal{ER}(\Sigma) \mapsto \{\mathbf{v}, \mathbf{f}\}$ spécifiée comme suit :

- $\lambda(e) = \mathbf{v}$ si $\mathcal{L}_{ER}(e)$ est vide ;
- $\lambda(e) = \mathbf{f}$ dans le cas contraire.

▶ Supposons défini le type Caml suivant :

```
type 'a exprat = Vide | Epsilon | Lettre of 'a
 | Union of 'a exprat * 'a exprat
 | Produit of 'a exprat * 'a exprat | Etoile of 'a exprat ;;
```

***Question 2*** Rédigez en Caml une fonction de signature :

```
lambda : 'a exprat -> bool
```

spécifiée comme suit : `lambda e` calcule $\lambda(e)$. Objectif : sept lignes.

***Question 3*** Définissez par induction structurelle une fonction $\mu : \mathcal{ER}(\Sigma) \mapsto \mathbb{N} \cup \{-\infty\}$ spécifiée comme suit :

- $\mu(e)$ est la longueur minimale d'un mot de $\mathcal{L}_{ER}(e)$ si ce langage n'est pas vide ;
- $\mu(e) = -\infty$ sinon.

▶ Rappel : à tout type `'a`, Caml associe le type `'a option`, qu'il définit comme suit :

```
type 'a option = Some 'a | None ;;
```

Ce mécanisme permet donc d'ajouter à un ensemble $E$ un élément dont on est certain qu'il n'appartient pas à $E$.

***Question 4*** Rédigez en Caml une fonction de signature :

```
mu : 'a exprat -> int option
```

spécifiée comme suit : si $e$ décrit un langage non vide, alors `mu e` rend `Some n`, où $n = \mu(e)$ ; sinon, `mu e` rend `None`. Objectif : treize lignes.

▶ Nous supposons désormais que $L$ est reconnu par un automate fini déterministe $\mathcal{A} = (Q, \delta, i, F)$.

***Question 5*** Rappelez le principe de l'algorithme de parcours en largeur du graphe d'un automate fini exposé en cours.

***Question 6*** En vous inspirant de cet algorithme, proposez un algorithme de calcul de $\lambda(L)$.

***Question 7*** Dans le même esprit, proposez un algorithme de calcul de $\mu(L)$.

***Question 8*** Proposez un algorithme déterminant un mot de $L$ de longueur $\mu(L)$, lorsque ce langage n'est pas vide.

***Question 9*** Proposez un algorithme déterminant l'ensemble des mots de $L$ de longueur $\mu(L)$, lorsque ce langage n'est pas vide. Que pouvez-vous dire de son coût dans le pire des cas ?

**FIN**