**ECOLES NORMALES SUPERIEURES**
**CONCOURS D'ADMISSION 2023**

**VENDREDI 21 AVRIL 2023**
**14h00 - 18h00**
**FILIERE MPI - EPREUVE n° 10**
**INFO - FONDAMENTALE (ULSR)**

Durée : 4 heures
L'utilisation des calculatrices n'est pas autorisée pour cette épreuve

# Épreuve d'informatique fondamentale<br>Concision et ambiguïté

Le sujet porte sur les automates finis, dont la définition est rappelée dans le préambule. Le sujet s'intéresse à la propriété d'ambiguïté des automates finis, propriété également définie dans le préambule. La première partie lie les notions de déterminisme et d'ambiguïté d'un automate fini en utilisant la notion de miroir d'un automate. La deuxième partie amène à définir un algorithme qui teste si un automate est ambigu et vous demande de déterminer sa complexité asymptotique. La troisième et la quatrième partie étudient la concision des automates non ambigus et ambigus.

Les parties 1 et 2 sont indépendantes ; il est conseillé de les traiter en premier car les parties 3 et 4 en dépendent. Il est permis d'admettre les réponses à certaines questions pour répondre aux suivantes.

## Préambule

On note $\Sigma$ un *alphabet* fini, c'est à dire un ensemble fini de symboles appelés *lettres*. Un *mot* sur $\Sigma$ est une suite finie de lettres. On note $\Sigma^n$ l'ensemble des mots de longueur $n$ sur $\Sigma$ et $\Sigma^*$ l'ensemble de tous les mots sur $\Sigma$. Soient $u, v$ deux mots sur $\Sigma$, on note $u \cdot v$ la *concaténation* de $u$ et $v$. Un *langage* sur $\Sigma$ est un sous-ensemble de $\Sigma^*$.

Un automate sur $\Sigma$ est un tuple $\mathcal{A} = (Q, T, I, F)$ où :

— $Q$ est un ensemble fini de symboles appelés *états* ;

— $T \subseteq Q \times \Sigma \times Q$ est appelé ensemble des *transitions* ;

— $I \subseteq$ est l'ensemble des états *initiaux* ;

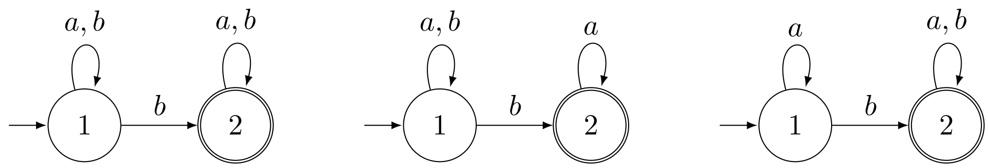
— $F \subseteq Q$ est l'ensemble des états *finaux*.

La Figure 1 représente graphiquement trois automates. Les symboles dans $Q$ sont encerclés, avec deux cercles pour les symboles dans $F$. Une transition $(q, a, q') \in T$ est représentée par une flèche étiquetée par $a \in \Sigma$, allant de l'état source $q$ à l'état destination $q'$. Les états initiaux sont indiqués par une flèche sans état source.

Un calcul de $\mathcal{A}$ sur un mot $w = a_0 \dots a_{n-1} \in \Sigma^*$ est une suite finie d'états $q_0 \dots q_n \in Q^*$ telle que $q_0 \in I$ et pour tout $i < n$, $(q_i, a_i, q_{i+1}) \in T$. Un tel calcul est dit *acceptant* si $q_n \in F$. On dit alors que $\mathcal{A}$ accepte $w$. Le langage de $\mathcal{A}$, noté $L(\mathcal{A})$, est l'ensemble des mots acceptés par $\mathcal{A}$.

Un automate $\mathcal{A}$ est dit *déterministe* si $|I| \leq 1$ et pour tout $(q, a) \in Q \times \Sigma, |\{q' | (q, a, q') \in T\}| \leq 1$.

Un automate $\mathcal{A}$ est dit *complet* si $|I| \leq 1$ et pour tout $(q, a) \in Q \times \Sigma, |\{q' | (q, a, q') \in T\}| \geq 1$.




$(a)\mathcal{A}_1$ $(b)\mathcal{A}_2$ $(c)\mathcal{A}_3$

**FIGURE 1** - Trois automates reconnaissant le même langage

Soient $w \in \Sigma^*$ et $\mathcal{A}$ un automate. Le mot $w$ est dit *ambigu* pour $\mathcal{A}$ s'il existe deux calculs acceptants $\rho$ et $\rho'$ de $\mathcal{A}$ sur $w$ avec $\rho \neq \rho'$. On définit $d_{\mathcal{A}}(w)$, le *degré d'ambiguïté* de $w$ dans $\mathcal{A}$, comme étant le nombre de calculs acceptants différents de $\mathcal{A}$ sur $w$. Ainsi, $w$ est ambigu pour $\mathcal{A}$ si et seulement si $d_{\mathcal{A}}(w) > 1$. On note $\mathrm{Amb}(\mathcal{A})$ l'ensemble des mots ambigus pour $\mathcal{A}$. L'automate $\mathcal{A}$ est dit *ambigu* si $\mathrm{Amb}(\mathcal{A}) \neq 0$.

Ce sujet s'intéresse tout particulièrement aux automates *non ambigus.*

# 1 Déterminisme et ambiguïté

**Question 1.1.** Les trois automates de la Figure 1 acceptent le même langage. Donnez-en, sans justification, une description intuitive.

**Question 1.2.** Pour chacun des automates de la Figure 1, dites s'il est déterministe ou non déterministe. Justifiez vos affirmations.

**Question 1.3.** Pour l'automate $\mathcal{A}_1$ de la Figure 1, calculez $\mathrm{Amb}(\mathcal{A}_1)$ et déduisez en si $\mathcal{A}_1$ est ambigu ou non. Faites de même pour les automates $\mathcal{A}_2$ et $\mathcal{A}_3$.

Soit $\mathcal{A}$ un automate. On note $\tilde{\mathcal{A}} = (\tilde{Q}, \tilde{T}, \tilde{I}, \tilde{F})$ l'*automate miroir* de $\mathcal{A}$, défini par :

- $\tilde{Q} = Q$ ;
- $\tilde{I} = F$ ;
- $\tilde{F} = I$ ;
- $\tilde{T} = \{(t, a, s) | (s, a, t) \in T\}$.

Un automate est dit *co-déterministe* si son automate miroir est déterministe.

**Question 1.4.** Soit $\mathcal{A}$ un automate, $w \in L(\mathcal{A})$ un mot accepté par $\mathcal{A}$ et $q_0 \ldots q_n$ un calcul acceptant de $\mathcal{A}$ sur $w$. Montrez qu'il existe un mot $\tilde{w}$ tel que $q_n ... q_0$ soit un calcul acceptant de $\tilde{A}$ sur $\tilde{w}$.

**Question 1.5.** Montrez qu'un automate $\mathcal{A}$ est ambigu si et seulement si $\tilde{A}$ est ambigu.

**Question 1.6.** Montrez que si un automate $\mathcal{A}$ est déterministe, alors $\mathcal{A}$ n'est pas ambigu.

**Question 1.7.** Montrez que si un automate $\mathcal{A}$ est co-déterministe, alors $\mathcal{A}$ n'est pas ambigu.

**Question 1.8.** Pour chacune des questions suivantes, donnez un automate $\mathcal{A}$ ayant au plus 4 états et respectant les propriétés demandées. Justifiez vos réponses.

(*i*) $\mathcal{A}$ est non ambigu mais ni déterministe, ni co-déterministe ;

(*ii*) $L(\mathcal{A})$ est infini et $\mathrm{Amb}(\mathcal{A}) = L(\mathcal{A})$ ;

(*iii*) $\mathrm{Amb}(\mathcal{A})$ est infini, et $\mathrm{Amb}(\mathcal{A}) \neq L(\mathcal{A})$.

# 2 Test d'ambiguïté

Le but de cette partie est d'obtenir un algorithme qui teste si un automate est ambigu et de déterminer sa complexité asymptotique.

Dans cette partie, $\mathcal{A}$ est un automate $(Q, T, I, F)$ tel que $L(\mathcal{A}) \neq \emptyset$.

## 2.1 Une construction utile

En utilisant $\mathcal{A}$, on définit l'automate $\widehat{\mathcal{A}} = (\widehat{Q}, \widehat{T}, \widehat{I}, \widehat{F})$ comme suit :

— $\widehat{Q} = Q \times Q \times \{0, 1\}$ ;

— $\widehat{I} = \{(i, i, 0) | i \in I\} \cup \{(i, i', 1) | i \in I, i' \in I, i \neq i'\}$ ;

— $\widehat{F} = \{(f, f', 1) | f \in F, f' \in F\}$ ;

— $\widehat{T} = T_1 \cup T_2 \cup T_3$ avec :

- $T_1 = \{((s, s, 0), a, (t, t, 0)) | (s, a, t) \in T\}$ ;
- $T_2 = \{((s, s, 0), a, (t, t', 1)) | (s, a, t) \in T, (s, a, t') \in T, t \neq t'\}$ ;
- $T_3 = \{((s, s', 1), a, (t, t', 1)) | (s, a, t) \in T, (s', a, t') \in T\}$.

**Question 2.1.** Soit $\mathcal{A}_1$ le premier automate de la figure Figure 1. Construisez l'automate $\widehat{\mathcal{A}_1}$. Il est inutile de faire figurer les états qui ne sont pas accessibles à partir d'un état initial. Donnez, sans justification, le langage $L(\widehat{\mathcal{A}_1})$.

**Question 2.2.** Soient $w \in \Sigma^*$ un mot et $\rho = (q_0, q'_0, b_0) \dots (q_n, q'_n, b_n)$ un calcul de $\widehat{\mathcal{A}}$ sur $w$. On pose $\mu = q_0 \dots q_n$ et $\mu' = q'_0 \dots q'_n$.

(*i*) Montrez que $\mu$ et $\mu'$ sont des calculs de $\mathcal{A}$ sur $w$.

(*ii*) Montrez que $b_n = 0$ si et seulement si $\mu = \mu'$.

(*iii*) Montrez que, si $\rho$ est acceptant, alors $\mu$ et $\mu'$ sont acceptants.

(*iv*) Montrez qu'il existe un calcul $\rho$ tel que $\mu$ et $\mu'$ sont acceptants mais $\rho$ ne l'est pas.

**Question 2.3.** Montrez que $L(\widehat{\mathcal{A}}) = \mathrm{Amb}(\mathcal{A})$.

## 2.2 L'algorithme

Pour deux fonctions $f, g : \mathbb{N} \to \mathbb{N}$, on dit que $g$ est une borne asymptotique de $f$, et on le note par $f \in \mathcal{O}(g)$, s'il existe deux constantes strictement positives $n_0 \in \mathbb{N}$ et $c \in \mathbb{N}$ telles que pour tout $n \geq n_0$, $f(n) \leq c \times g(n)$. Cette définition se généralise naturellement à des fonctions $f$ et $g$ avec plusieurs paramètres.

**Question 2.4.** Pour chaque ensemble $\widehat{Q}, \widehat{T}, \widehat{I}$ et $\widehat{F}$, donnez une borne asymptotique au nombre d'éléments qu'il contient en fonction des tailles de $Q, T, I$ et $F$.

**Question 2.5.** Donnez un algorithme qui a comme entrée $\mathcal{A}$ et comme sortie $\widehat{\mathcal{A}}$ en précisant :

($i$) quelle structure de données classique (matrice d'adjacence ou liste d'adjacence) est utilisée pour représenter $\mathcal{A}$ et $\widehat{\mathcal{A}}$,

($ii$) une borne asymptotique de la complexité en temps d'exécution de cet algorithme en fonction de la somme des tailles des ensembles $Q, T, I$ et $F$.

**Question 2.6.** Décrivez une méthode permettant de tester si $\mathcal{A}$ est ambigu en utilisant l'automate $\widehat{\mathcal{A}}$. Donnez une borne asymptotique de la complexité en temps de cette méthode en fonction de la somme des tailles des ensembles $Q, T, I$ et $F$.

## 2.3 Généralisation

Soit $k$ un entier strictement positif.

**Question 2.7.** Soit $w \in L(\mathcal{A})$ un mot de longueur $k$.

($i$) Donnez, en fonction de $k$ et $|Q|$, une borne supérieure sur le degré d'ambiguïté de $w$ dans $\mathcal{A}$.

($ii$) Donnez un automate $\mathcal{A}$ et un mot de longueur $k$ pour lesquels la borne supérieure indiquée ci-dessus est atteinte.

**Question 2.8.** On pose $\mathrm{Amb}_{\geq k}(\mathcal{A}) = \{w \in L(\mathcal{A}) | d_{\mathcal{A}}(w) \geq k\}$. Montrez que $\mathrm{Amb}_{\geq k}(\mathcal{A})$ est régulier.

**Question 2.9.** On pose $\mathrm{Amb}_{k}(\mathcal{A}) = \{w \in L(\mathcal{A}) | d_{\mathcal{A}}(w) = k\}$. Montrez que $\mathrm{Amb}_{k}(\mathcal{A})$ est régulier.

## 3 Concision des automates non ambigus

Le but de cette partie est de prouver que les automates non ambigus peuvent être exponentiellement plus concis que leurs équivalents déterministes et complets.

Dans toute cette partie, on fixe l'alphabet $\Sigma = \{a, b\}$. Pour $n \geq 1$ un entier, on pose $L_n = \{w_1 \cdot b \cdot w_2 | w_1, w_2 \in \Sigma^*, |w_2| = n - 1\}$, le langage des mots dont la $n$-ième lettre en partant de la fin est un $b$.

**Question 3.1.** Donnez un automate non ambigu acceptant $L_3$, puis donnez un automate déterministe et complet acceptant $L_3$.

**Question 3.2.** Montrez que pour tout $n \geq 1$ entier, il existe un automate non ambigu $\mathcal{A}$ avec $n + 1$ états tel que $L(\mathcal{A}) = L_n$.

**Question 3.3.** Montrez que pour tout $n \geq 1$ entier, il existe un automate déterministe et complet $\mathcal{B}$ avec $2^n$ états tel que $L(\mathcal{B}) = L_n$.

On veut à présent prouver que tout automate déterministe et complet acceptant $L_n$ a au moins $2^n$ états.

**Question 3.4.** Soit $\mathcal{B}$ un automate déterministe et complet. Soit $w \in \Sigma^*$. Montrez qu'il existe un unique calcul, non nécessairement acceptant, de $\mathcal{B}$ sur $w$.

Soit $\mathcal{B}$ un automate déterministe et complet et $w \in \Sigma^*$. On note $q_w$ l'état atteint par l'unique calcul de $\mathcal{B}$ sur $w$, c'est-à-dire l'état $q_m$ tel que $q_0 \ldots q_m$ est un calcul de $\mathcal{B}$ sur $w$.

**Question 3.5.** Soit $n \in \mathbb{N}$ et $\mathcal{B}$ un automate déterministe et complet reconnaissant $L_n$. Soient $w$ et $w'$ deux mots de $\Sigma^*$ de longueur $n$. Montrez que $q_w = q_{w'}$ si et seulement si $w = w'$.

**Question 3.6.** Soit $n \in \mathbb{N}$. Montrez que tout automate déterministe et complet reconnaissant $L_n$ a au moins $2^n$ états.

## 4 Concision des automates ambigus

Le but de cette partie est de prouver que les automates ambigus peuvent être exponentiellement plus concis que leurs équivalents non ambigus.

Pour $n \in \mathbb{N}$, on pose $\Sigma_n$ un alphabet à $n$ lettres et $K_n$ le langage des mots sur $\Sigma_n^*$ dont au moins une lettre apparaît au moins deux fois, c'est-à-dire :

$$K_n = \{w_1 \cdot x \cdot w_2 \cdot x \cdot w_3 | w_1, w_2, w_3 \in \Sigma_n^*, x \in \Sigma_n\}$$

**Question 4.1.** Pour $\Sigma_3 = \{a, b, c\}$

$(i)$ donnez un automate acceptant $K_3$ et ayant au plus 5 états,

$(ii)$ donnez un automate non ambigu acceptant $K_3$ et ayant au plus 9 états.

**Question 4.2.** Montrez que pour tout $n \in \mathbb{N}$, il existe un automate avec au plus $n + 2$ états acceptant $K_n$.

**Question 4.3.** Montrez que pour tout $n \in \mathbb{N}$, il existe un automate non ambigu avec au plus $2^n + 1$ états acceptant $K_n$.

On veut à présent prouver que tout automate non ambigu acceptant $K_n$ a au moins $2^n + 1$ états. Jusqu'à la fin de cette partie, on fixe un entier $n \geq 2$ et un automate non ambigu $\mathcal{A} = (Q, T, I, F)$ acceptant $K_n$.

Soient $u$ et $v$ deux mots de $\Sigma_n^*$. Lorsque $u \cdot v \in K_n$, on note $q_{u,v}$ l'état de $\mathcal{A}$ atteint après avoir lu $u$ lors de l'unique calcul acceptant de $\mathcal{A}$ sur $u \cdot v$. Autrement dit, soit $\ell = |u|$ et $q_0 \ldots q_m$ l'unique calcul acceptant de $\mathcal{A}$ sur $u \cdot v$, alors $q_{u,b} = q_\ell$.

**Question 4.4.** Soient $u, u'$ et $v, v'$ quatre mots de $\Sigma_n^*$ tels que $u \cdot v \in K_n$ et $u' \cdot v' \in K_n$. On suppose que $q_{u,v} = q_{u',v'}$.

$(i)$ Montrez que $u \cdot v' \in K_n$ et $u' \cdot v \in K_n$.

$(ii)$ Montrez que $q_{u,v} = q_{u,v'} = q_{u',v} = q_{u',v'}$.

L'objectif à présent est de prouver que ces contraintes garantissent que $\mathcal{A}$ a au moins $2^n$ états.

Soit $s = (s_i)$ une suite de mots et $\alpha$ une lettre. On note $s \cdot \alpha$ la suite $(s_i \cdots \alpha)$, c'est-à-dire la suite obtenue en ajoutant $\alpha$ à la fin de chaque mot de $s$.

On fixe un ordre total $<$ sur $\Sigma_n$, et on note $\Sigma_n = \{\alpha_1, \ldots, \alpha_n\}$ avec pour tout $1 \leq i < j \leq n, \alpha_i < \alpha_j$. On construit une famille $s^0, \ldots, s^n$ de suites de mots de $\Sigma_n$ comme suit :

— $s^0$ est la suite contenant uniquement $\varepsilon$ ;

— Pour $0 < i \leq n, s^i = s^{i-1}, s^{i-1} \cdot \alpha_i$, la suite constituée de $s^{i-1}$, suivie de la suite $s^{i-1} \cdot \alpha_i$.

Finalement, on pose $s = s^n$. Par exemple, pour $n = 2, \Sigma_2 = \{a, b\}$ et $a < b$, la suite $s$ contient 4 éléments : $s_0 = \varepsilon, s_1 = a, s_2 = b, s_3 = ab$.

Finalement, on remarque que $s$ contient $2^n$ éléments et on pose $M_n$ la matrice de dimension $2^n \times 2^n$ à coefficients en $\mathbb{R}$ définie par :

$$M_n[i,j] = \begin{cases} 1 & \text{si } s_i \text{ et } s_j \text{ ont une lettre en commun} \\ 0 & \text{sinon} \end{cases}$$

Par exemple, $M_2$ est donnée ci-dessous :

$$M_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

**Question 4.5.** Montrez que $M_n$ peut s'écrire sous la forme :

$$M_n = \begin{pmatrix} M_{n-1} & M_{n-1} \\ M_{n-1} & \mathbf{1}_{n-1} \end{pmatrix}$$

où, pour $n \geq 1$, $\mathbf{1}_n$ est la matrice carrée de dimension $2^n \times 2^n$ dont tous les coefficients valent 1.

**Question 4.6.** Soient $i < 2^n$ et $j < 2^n$. Montrez que $M_n[i,j] = 1$ si et seulement si $s_i \cdot s_j \in K_n$.

À chaque état $q$ de $\mathcal{A}$, on associe le vecteur colonne $v_q$ de dimension $2^n$ défini par :

$$v_q[i] = \begin{cases} 1 & \text{s'il existe } j < 2^n \text{ tel que } q = q_{s_i, s_j} \\ 0 & \text{sinon} \end{cases}$$

**Question 4.7.** Montrez que l'ensemble de vecteurs $(v_q)_{q \in Q}$ est une famille génératrice du sous-espace vectoriel engendré par les vecteurs colonnes de $M_n$.
Indication : Soit $u_j$ la $j$-ème colonne de $M_n$. On pourra montrer que $u_j$ est une combinaison linéaire de vecteurs de $(v_q)_{q \in Q}$ en prouvant que :

$$u_j = \sum_{\substack{q \in Q \\ \exists i, q = q_{s_i, s_j}}} v_q$$

**Question 4.8.** Montrez que $M_n$ est de rang $2^n - 1$.

**Question 4.9.** En déduire que $\mathcal{A}$ contient au moins $2^n - 1$ états $q$ tels que $v_q \neq 0$.

**Question 4.10.** Montrez que si $v_q \neq 0$, alors $q \notin I$ et $q \notin F$.

**Question 4.11.** Conclure que $\mathcal{A}$ a au moins $2^n + 1$ états.