# Langages formels, calculabilité et complexité

## Magistère MMFAI

Examen du 3 mai 1999

durée 3 heures

*Il va sans dire que la rigueur des raisonnements, la clarté des explications, mais aussi la qualité de la présentation influeront sensiblement sur la note.*

## 1 Automate

Soit $\mathcal{A} = (Q,\Sigma,T,I,F)$ un automate fini avec $T \subseteq Q \times \Sigma \times Q$.

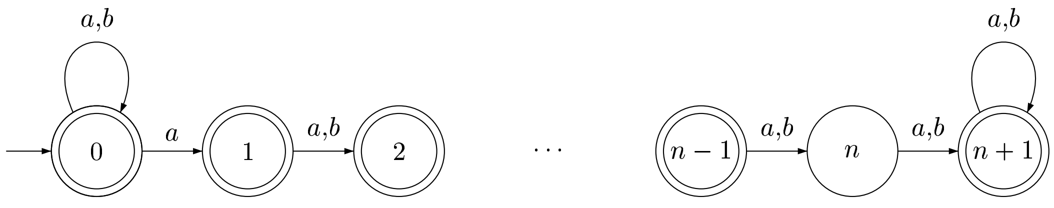
Un mot $u = u_1 \cdots u_n \in \Sigma^*$ est existentiellement accepté par l'automate $\mathcal{A}$ s'il existe un calcul sur $u$ partant d'un état initial et arrivant dans un état final. On note $L_\exists(\mathcal{A})$ l'ensemble des mots existentiellement acceptés par l'automate $\mathcal{A}$. Il s'agit du langage accepté par $\mathcal{A}$ lorsqu'on le considère comme un automate non déterministe.

Un mot $u = u_1 \cdots u_n \in \Sigma^*$ est universellement accepté par l'automate $\mathcal{A}$ si tout calcul sur $u$ issu d'un état initial arrive dans un état final :

$$q_0 \in I \quad \text{et} \quad q_0 \xrightarrow{u_1} q_1 \cdots q_{n-1} \xrightarrow{u_n} q_n \qquad \Longrightarrow \qquad q_n \in F.$$

On note $L_\forall(\mathcal{A})$ l'ensemble des mots universellement acceptés par l'automate $\mathcal{A}$.

**a)** Donner le langage $L_\forall(\mathcal{C})$ pour l'automate $\mathcal{C}$ défini par $Q = \{0,1,\ldots,n,n+1\}$, $\Sigma = \{a,b\}$, $I = \{0\}$, $F = Q \setminus \{n\}$ et les transitions ci-dessous :




**b)** Construire un automate $\mathcal{B}$ n'ayant pas plus d'états que l'automate $\mathcal{A}$ et tel que $L_\forall(\mathcal{A}) = \Sigma^* \setminus L_\exists(\mathcal{B})$.

**c)** Construire un automate déterministe $\mathcal{B}$ reconnaissant le langage $L_\forall(\mathcal{A})$.

**d)** Montrer que le langage $L_\forall(\mathcal{C})$ ne peut pas être reconnu par un automate déterministe ayant moins de $2^n$ états.

## 2 Décimales de $\sqrt[3]{5}$

Dans cet exercice, on pourra utiliser les fonctions primitives récursives et les schémas de construction de fonctions primitives récursives vus en cours.

**a)** Montrer que la fonction $f : \mathbb{N} \to \mathbb{N}$ définie par $f(n) = \lfloor 10^n \sqrt[3]{5} \rfloor$ est primitive récursive. On rappelle que $\lfloor x \rfloor$ représente la partie entière inférieure de $x$.

**b)** Montrer que la fonction $g : \mathbb{N} \to \mathbb{N}$ qui à un entier $n$ associe le $n^{\text{e}}$ chiffre après la virgule de $\sqrt[3]{5}$ est primitive récursive (si $n = 0$ on donne le chiffre avant la virgule).

## 3 2-SAT

Le problème 2-SAT est le suivant :

**Donnée :** Une formule $\varphi$ en forme normale conjonctive dont chaque clause contient au plus deux littéraux.

**Question :** $\varphi$ est-elle satisfaisable ?

On note $\overline{\text{2-SAT}}$ le problème complémentaire où l'on répond "oui" si $\varphi$ n'est pas satisfaisable.

Le problème ACC est le suivant :

**Données :** Un graphe orienté $G = (S,A)$ avec $S = \{s_1, \ldots, s_n\}$ et $A \subseteq S \times S$ et deux sommets $s_i, s_j \in S$. Le graphe $G$ est représenté par le nombre $n$ en binaire et une suite de $n^2$ bits décrivant la matrice d'adjacence de $G$.

**Question :** Existe-t-il un chemin de $s_i$ à $s_j$ dans le graphe $G$ ?

On note $\overline{\text{ACC}}$ le problème complémentaire où l'on répond "oui" s'il n'existe pas de chemin de $s_i$ à $s_j$ dans le graphe $G$.

On considère une formule $\varphi$ en forme normale conjonctive dont chaque clause contient au plus deux littéraux et utilisant les variables $x_1, \ldots, x_n$. On définit le graphe $G(\varphi) = (S,A)$ par

$$\begin{array}{rcl} S & = & \{x_1, \neg x_1, \ldots, x_n, \neg x_n\} \\ A & = & \{(\alpha,\beta) \mid (\neg\alpha \vee \beta) \text{ est une clause de } \varphi\}. \end{array}$$

Bien sûr, pour chaque variable $x$, on identifie $\neg\neg x$ et $x$ dans cette définition.

**a)** Donner le graphe $G(\varphi)$ pour $\varphi = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3)$.

**b)** Montrer que s'il y a un chemin de $\alpha$ à $\beta$ dans $G(\varphi)$ alors il y a aussi un chemin de $\neg\beta$ à $\neg\alpha$.

**c)** Montrer que $\varphi$ n'est pas satisfaisable si et seulement si il existe une variable $x$ et un circuit de $G(\varphi)$ contenant à la fois $x$ et $\neg x$, i.e., s'il y a un chemin de $x$ à $\neg x$ et un chemin de $\neg x$ à $x$ dans $G(\varphi)$.

**d)** En déduire que le problème 2-SAT est dans la classe **NL**.

A partir d'une instance $G$ du problème $\overline{\text{ACC}}$, on définit la formule $\varphi(G)$ utilisant les variables $x_1, \ldots, x_n$ par

$$\varphi(G) = (x_1) \wedge (\neg x_n) \wedge \bigwedge_{(i,j) \in A} (\neg x_i \vee x_j).$$

**e)** Montrer que le sommet $s_n$ n'est pas accessible à partir du sommet $s_1$ dans le graphe $G$ si et seulement si la formule $\varphi(G)$ est satisfaisable.

**f)** Décrire une MT qui réalise la construction de la formule $\varphi(G)$ et travaille en espace logarithmique.

**g)** On rappelle que si $\mathcal{C}$ est une classe de complexité alors un langage $K \subset \Sigma^*$ est dans la classe co-$\mathcal{C}$ si et seulement si son complémentaire $\overline{K} = \Sigma^* \setminus K$ est dans la classe $\mathcal{C}$. Montrer que si $K$ est $\mathcal{C}$-complet, alors $\overline{K}$ est co-$\mathcal{C}$-complet.

**h)** Montrer que le problème $\overline{\text{ACC}}$ est **NL**-complet. Que peut-on en déduire sur le problème 2-SAT ?

# 4 Langages compatibles

Un alphabet $A$ est ordonné s'il est muni d'une relation d'ordre total. Dans tout ce problème, on considère des alphabets ordonnés. Un langage $L \subseteq A^*$ est compatible s'il vérifie la propriété suivante pour tous $u,v \in A^*$ et $x,y \in A$ :

$$x < y \quad \text{et} \quad uyv \in L \qquad \Longrightarrow \qquad \exists w \in A^*, \quad uxw \in L.$$

**a)** On considère un langage rationnel $L \subseteq A^*$ et son automate minimal complet $\mathcal{A} = (Q,A,\delta,i,F)$. On rappelle que $\mathcal{A}$ admet au plus un état puits (noté $\pi$) à partir duquel aucun mot n'est reconnu. Trouver une caractérisation sur l'automate $\mathcal{A}$ du fait que $L$ est compatible.

**b)** En déduire que l'on peut décider le problème de la compatibilité pour les langages rationnels.

**Donnée :** Une expression rationnelle E sur un alphabet ordonné $A$.
**Question :** Le langage $L(E)$ est-il compatible ?

On considère $k$ couples de mots $(u_1,v_1),\dots,(u_k,v_k)$ sur un alphabet $A$.

**c)** Donner une grammaire linéaire qui engendre le langage $M$ sur l'alphabet $B = A \cup \{x_1,\dots,x_k,\sharp\}$ défini par

$$M = \{x_{i_1} \cdots x_{i_n} \sharp v_{i_n} \cdots v_{i_1} \mid n \geq 1 \text{ et } i_1,\dots,i_n \in \{1,\dots,k\}\}.$$

**d)** Montrer que $M$ n'est pas rationnel si il existe $i \in \{1,\dots,k\}$ tel que $v_i \neq \varepsilon$.

**e)** Montrer que le langage $P$ défini sur l'alphabet $B$ par

$$P = \{x_{i_1} \cdots x_{i_n} \sharp w \mid w \in A^*, w \neq u_{i_n} \cdots u_{i_1}, n \geq 1 \text{ et } i_1,\dots,i_n \in \{1,\dots,k\}\}.$$

est algébrique.

On considère deux nouvelles lettres $y,z$ et l'alphabet $C = B \cup \{y,z\}$ avec un ordre total vérifiant $y < z$ et $b < y$ pour tout $b \in B$. On pose $L = B^* \cup Mz \cup Py$.

**f)** Montrer que $L$ est compatible si et seulement si $M \subseteq P$.

**g)** Que peut-on en déduire pour le problème de la compatibilité pour les langages algébriques ?

**Donnée :** Une grammaire $G = (V,P,S)$ sur un alphabet ordonné $A$.
**Question :** Le langage $L_G(S)$ est-il compatible ?

On rappelle que le problème de Post ci-dessous est indécidable.

**Données :** Des couples de mots $(u_1,v_1),\dots,(u_k,v_k)$ sur un alphabet $A$.
**Question :** Existe-t-il $n \geq 1$ et $i_1,\dots,i_n \in \{1,\dots,k\}$ tels que $u_{i_1} \cdots u_{i_n} = v_{i_1} \cdots v_{i_n}$ ?