# Langages formels, calculabilité et complexité

## Magistère MMFAI

Examen du 5 février 2004

durée 3 heures

*Les documents sont autorisés.*

*Toutes les réponses devront être correctement justifiées.*

*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 Conjugués

Soit $\Sigma$ un alphabet fini. Deux mots $w,w' \in \Sigma^*$ sont *conjugués* s'il existe deux mots $u,v \in \Sigma^*$ tels que $w = uv$ et $w' = vu$. Dans la suite, on note $C(w)$ l'ensemble des conjugués du mot $w \in \Sigma^*$. De même, pour un langage $L \subseteq \Sigma^*$, on note $C(L) = \bigcup_{w \in L} C(w)$ l'ensemble des conjugués des mots de $L$.

**a)** Donner $C(aabaab)$ et $C(\{a^n b^n \mid n \geq 0\})$.

**b)** Montrer que si $L$ est un langage rationnel alors $C(L)$ l'est aussi.
Indication : on pourra considérer un automate $\mathcal{A} = (Q,\delta,I,F)$ qui reconnaît $L$ et utiliser les langages $L_{p,q} = \{w \in \Sigma^* \mid p \xrightarrow{w} q\}$ pour $p,q \in Q$.

**c)** Soient $K,L \subseteq \Sigma^*$ deux langages. Comparer $C(K \cap L)$ et $C(K) \cap C(L)$. Soit $\# \notin \Sigma$. Montrer que $C(\#K \cap \#L) = C(\#K) \cap C(\#L)$.

**d)** Soient $A$ et $B$ deux alphabets, $L \subseteq A^*$ et $h : A^* \to B^*$ un morphisme. Comparer $h(C(L))$ et $C(h(L))$. Montrer que si $h$ est alphabétique (i.e. $h(a) \in B \cup \{\varepsilon\}$ pour tout $a \in A$) alors $h(C(L)) = C(h(L))$.

Soit $A = \{a_1, \ldots, a_n\}$, $\overline{A} = \{\overline{a}_1, \ldots, \overline{a}_n\}$ et $\Sigma = A \cup \overline{A}$. Pour $w \in A^*$, on note $\overline{w}$ le mot obtenu en remplaçant chaque lettre $a_i \in A$ par $\overline{a}_i$. On rappelle que le langage de Dyck $D_n^*$ est engendré par la grammaire

$$S \longrightarrow \varepsilon + \sum_{i=1}^{n} a_i S \overline{a}_i S$$

Par ailleurs, si $w \in (\Sigma \cup \{\#\})^*$, on note $\rho(w)$ le réduit de $w$ obtenu par effacements successifs de facteurs $a_i \overline{a}_i$ $(1 \leq i \leq n)$. Le langage de Dyck $D_n^*$ est aussi l'ensemble des mots $w \in \Sigma^*$ tels que $\rho(w) = \varepsilon$.

**e)** Montrer que $\rho(C(\#D_n^*)) = \{\widetilde{\overline{\alpha}} \# \alpha \mid \alpha \in A^*\}$, où $\widetilde{w}$ dénote le miroir du mot $w$.

**f)** Montrer que le langage $C(\#D_n^*)$ est algébrique.

**g)** Montrer que si $L$ est un langage algébrique alors $C(L)$ l'est aussi.
Indication : Utiliser le théorème de Chomsky-Schützenberger.

## 2 Fonctions récursives

**a)** Montrer que le prédicat *être premier* est primitif récursif.

**b)** On considère la fonction $f : \mathbb{N} \to \mathbb{N}$ qui à un entier $n$ associe le $(n+1)^{\mathrm{e}}$ nombre premier : $f(0) = 2$, $f(1) = 3$, $f(2) = 5$, .... Montrer que $f$ est primitive récursive.

## 3 Indécidabilité

**a)** Soit $\Sigma$ un alphabet fini. Donner un morphisme injectif $f : \Sigma^* \to \{0,1\}^*$ tel que aucun mot de $f(\Sigma^*)$ ne comporte trois 1 consécutifs ou trois 0 consécutifs.

Si $M$ est une machine de Turing (MT), on note $c(M)$ le codage usuel de la machine $M$ sur l'alphabet $\{0,1,\#\}$.

**b)** Soit $K \subseteq \{0,1,\#\}^*$ l'ensemble des mots de la forme $c(M)$ où $M$ est une MT qui possède au moins un calcul écrivant trois caractères identiques et consécutifs sur sa bande. Montrer que le langage $K$ est indécidable.

## 4 Complexité : mots croisés

On considère le problème des mots croisés défini comme suit :

**Données :** Un entier $n$, un ensemble $N \subseteq \{1,\dots,n\}^2$ de cases noires et un ensemble fini $D \subseteq \Sigma^*$, appelé dictionnaire, de mots sur un alphabet $\Sigma$, .

**Question :** Existe-t-il une application $f : \{1,\dots,n\}^2 \setminus N \to \Sigma$ telle que tout mot maximal $f(i,j)\cdots f(i,j{+}p)$ sur une ligne $i$ soit dans $D$ et tout mot maximal $f(i,j)\cdots f(i{+}p,j)$ sur une colonne $j$ soit dans $D$ ?

Le mot $f(i,j)\cdots f(i,j+p)$ est maximal si ($j = 1$ ou $(i,j-1) \in N$) et ($j + p = n$ ou $(i,j+p+1) \in N$). La définition est similaire pour les colonnes.

**a)** Préciser le codage des données utilisé et montrer que le problème des mots croisés est dans **NP**.

**b)** Montrer que le problème des mots croisés est **NP**-complet.
Indication : On pourra réduire le problème SAT au problème des mots croisés. Plus précisément, si $\varphi$ est une formule en forme normale conjonctive comportant $n$ clauses et $m$ variables (montrer qu'on peut supposer $m + 1 < n$), on pourra construire une grille de mots croisés de taille $n$ avec $N = \{m + 2,\dots,n\} \times \{1,\dots,n\}$. On pourra choisir comme alphabet $\Sigma = \{a_1,\dots,a_n\} \cup \{\top,\bot\}$ et construire un dictionnaire

$$D \subseteq \{a_1 \cdots a_n, \top^n, \bot^n\} \cup \{\top,\bot\}^m \cdot \{a_1,\dots,a_n\}$$

de telle sorte que si une application $f$ est solution du problème des mots croisés alors l'assignation $\sigma$ définie par $\sigma(x_i) = f(i,1)$ satisfait la formule $\varphi$.

Si vous suivez cette indication, commencez par définir précisément le dictionnaire puis montrez que l'application qui à la formule $\varphi$ associe l'instance $(n,N,D)$ est bien une réduction. Enfin n'oubliez pas de montrer que cette réduction peut se calculer en espace logarithmique ou en temps polynomial avec une représentation appropriée du dictionnaire.