

I – Miroir d'un langage non-contextuel

On note $\widetilde{m} = m_n \dots m_1$ le miroir d'un mot $m = m_1 \dots m_n$ et $\widetilde{L} = \{\widetilde{m} \mid m \in L\}$ le miroir d'un langage L .

Soit L un langage non-contextuel. Montrer que $\widetilde{L}$ est non-contextuel.

II – Trouver une grammaire

Montrer que chacun des langages suivants est non-contextuel en donnant une grammaire et en justifiant que son langage est exactement celui demandé. Tous les exposants sont des entiers naturels.

1. $L_1 = (ab)^*$.
2. $L_2 = \{a^n b^p \mid n \geq p\}$.
3. $L_3 =$ représentations en base 2 des multiples de 3.
4. $L_4 = \{a^i b^j c^k \mid i = j + k\}$.

III – Trouver le langage engendré

Déterminer les langages engendrés par les grammaires suivantes, d'axiome S , en prouvant les deux inclusions. On note $|w|_a$ le nombre d'occurrences de a dans w .

1.
$$\begin{aligned} S &\rightarrow X \mid Y \\ X &\rightarrow aX \mid aZ \\ Y &\rightarrow Yb \mid Zb \\ Z &\rightarrow \varepsilon \mid aZb \end{aligned}$$
2.
$$\begin{aligned} S &\rightarrow 0A1 \mid \varepsilon \\ A &\rightarrow 1S0 \mid \varepsilon \end{aligned}$$
3.
$$\begin{aligned} S &\rightarrow X \mid XaS \\ X &\rightarrow aXbX \mid bXaX \mid \varepsilon \end{aligned}$$
4.
$$\begin{aligned} S &\rightarrow X \mid Y \\ X &\rightarrow Z0X \mid Z0Z \\ Y &\rightarrow Z1Y \mid Z1Z \\ Z &\rightarrow \varepsilon \mid 1Z0Z \mid 0Z1Z \end{aligned}$$

IV – Lemme d'Ogden

On admet le lemme de pompage suivant pour les langages non-contextuels : si L est non-contextuel, il existe un entier $p \geq 1$ tel que tout mot $t \in L$ de taille au moins p s'écrit $t = uvwxy$, avec :

- $|vwx| \leq p$
- $vx \neq \varepsilon$
- $\forall i \in \mathbb{N}, uv^iwx^iy \in L$

On considère $L_1 = \{a^n b^n c^k \mid n, k \in \mathbb{N}\}$ et $L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

1. Montrer que L_1 est non-contextuel.
2. Montrer que L_2 n'est pas non-contextuel.
3. Montrer que les langages non-contextuels sont stables par union. En déduire qu'ils ne sont stables ni par intersection ni par passage au complémentaire. Les complémentaires sont pris dans $\{a, b, c\}^*$.

V – Ambiguïté et langages réguliers

CCINP 2023

On considère la grammaire non-contextuelle G sur $\Sigma = \{a, b\}$, d'axiome S , donnée par $S \rightarrow SaS \mid b$.

1. Cette grammaire est-elle ambiguë ? Justifier.
2. Déterminer, sans preuve pour cette question, le langage L engendré par G . Quelle est la plus petite classe, parmi les langages réguliers et non-contextuels, à laquelle L appartient ?
3. Prouver que $L = L(G)$.
4. Donner une grammaire non ambiguë engendrant L et justifier sa non-ambiguïté.
5. Montrer que tout langage régulier peut être engendré par une grammaire non-contextuelle non ambiguë.

VI – Forme normale de Chomsky et reconnaissance

Cet exercice introduit un prolongement du cours. Une grammaire $G = (\Sigma, V, R, S)$ est en forme normale de Chomsky si ses règles sont $X \rightarrow YZ$, avec $Y, Z \in V$, ou $X \rightarrow a$, avec $a \in \Sigma$. On autorise aussi $S \rightarrow \varepsilon$, à condition que S n'apparaisse dans aucun membre droit.

1. Soit G une grammaire non-contextuelle telle que $\varepsilon \notin L(G)$. Montrer qu'il existe une grammaire en forme normale de Chomsky engendrant $L(G)$. Commencer par supprimer les règles d'effacement $X \rightarrow \varepsilon$, puis les règles unitaires $X \rightarrow Y$.
2. Donner une grammaire en forme normale de Chomsky pour $\{a^n b^n \mid n \geq 1\}$. Combien d'applications de règles comporte une dérivation terminale d'un mot de taille $n \geq 1$ dans une grammaire en forme normale de Chomsky sans règle d'effacement ?

- On fixe une grammaire en forme normale de Chomsky sans règle d'effacement, et un mot $w = w_0 \dots w_{n-1}$, avec $n \geq 1$. Pour $0 \leq i < j \leq n$, noter $T(i, j)$ l'ensemble des variables engendrant le facteur $w[i : j]$. Donner une récurrence permettant de calculer ces ensembles par tailles croissantes. En déduire un algorithme de reconnaissance et justifier ses complexités pour une grammaire fixée.
- Prolongement : montrer que l'intersection d'un langage non-contextuel avec un langage régulier est non-contextuel. Utiliser un automate déterministe et une grammaire en forme normale de Chomsky pour les mots non vides. Une variable (p, X, q) devra engendrer les mots produits par X qui mènent l'automate de p à q .

VII – Mots de Dyck : grammaires et dénombrement

Sur $\Sigma = \{a, b\}$, un mot w est un mot de Dyck, ou mot bien parenthésé, si :

- il contient autant de a que de b ;
- chacun de ses préfixes contient au moins autant de a que de b .

On note D le langage de ces mots et $h(m) = |m|_a - |m|_b$ la hauteur d'un mot m . La lettre a représente une parenthèse ouvrante et b une parenthèse fermante.

- Montrer que D n'est pas régulier.
- Montrer que tout mot de Dyck non vide se décompose de manière unique sous la forme $aubv$, où $u, v \in D$.
- Soit G la grammaire $S \rightarrow SS \mid aSb \mid \varepsilon$. Montrer que $L(G) = D$.
- Montrer que G est ambiguë.
- Donner une grammaire non ambiguë engendrant D .
- Donner une bijection entre les mots de Dyck et les arbres binaires stricts ordonnés, dont chaque nœud possède zéro ou deux fils. Préciser le lien entre la taille du mot et les nombres de nœuds internes et de feuilles.
- Soit C_n le nombre de mots de Dyck de taille $2n$. Donner C_0 et une relation de récurrence. Calculer C_1, C_2, C_3, C_4 .
- Après le cours de mathématiques sur les séries entières, montrer que $C_n = \frac{1}{n+1} \binom{2n}{n}$ en utilisant la série génératrice $F(x) = \sum_{n=0}^{\infty} C_n x^n$.
- Variante sans séries entières : parmi les mots contenant n lettres a et n lettres b , transformer ceux qui ne sont pas de Dyck en échangeant a et b dans leur premier préfixe de hauteur -1 . Montrer que cette transformation est une bijection avec les mots contenant $n+1$ lettres a et $n-1$ lettres b , puis retrouver la formule de C_n .
- On autorise plusieurs sortes de parenthèses, par exemple $()$, $[]$ et $\{\}$. Décrire un algorithme en temps linéaire pour reconnaître les mots bien parenthésés. Pourquoi un simple compteur ne suffit-il plus ?
- Décrire un algorithme en temps linéaire pour trouver la taille du plus long facteur bien parenthésé d'un mot sur $\{a, b\}$. La réponse vaut zéro si aucun facteur non vide ne convient.