

Théorie de la complexité – Feuille de cours intégré n° 2

Grammaires et langages hors-contexte

Nous avons vu que certains langages, pourtant très simples, ne sont pas réguliers. Par exemple sur l'alphabet binaire $\{0, 1\}$, un automate fini ne peut pas tester si son mot d'entrée contient autant de 0 que de 1. Il est pourtant simple d'écrire un programme dans son langage favori qui réalise ce test.

Dans cette partie, nous abordons des langages plus généraux que les langages réguliers : les langages hors-contexte. Ils sont engendrés par des formalismes permettant de simuler des appels récursifs simples.

Definition - Grammaires Hors-Contexte.

Une **grammaire hors-contexte** est un tuple $G = (\Sigma, V, S, R)$ qui définit un langage de mots.

- $\Sigma = \{a, b, c, \dots\}$ est un alphabet, dont les lettres sont appelées **terminaux**. La grammaire définit un langage sur l'alphabet Σ .
- $V = \{X, Y, Z, \dots\}$ est un ensemble fini de **variables**, aussi appelées **non-terminaux**.
- $S \in V$ est la **variable de départ**.
- R est un ensemble de **règles** de réécriture. Chaque règle s'écrit de la façon suivante :

$$X \rightarrow \alpha \quad \text{avec } X \in V \text{ et } \alpha \in (V \cup \Sigma)^*.$$

Par exemple $H = (\{a, b\}, \{X\}, X, \{X \rightarrow ab, X \rightarrow aXb\})$ est une grammaire hors-contexte. Par commodité, on regroupe souvent les règles qui ont le même membre gauche, en séparant tous les membres droits possibles par une barre verticale $|$. Par exemple, la grammaire ci-dessus peut s'écrire $X \rightarrow ab \mid aXb$.

À toute grammaire hors-contexte $G = (\Sigma, V, S, R)$, on associe un langage $\mathcal{L}(G) \subseteq \Sigma^*$ de la façon suivante. Soient α, β deux mots dans $(\Sigma \cup V)^*$ (c'est-à-dire que leurs lettres sont soit des variables, soit des terminaux). On note,

$$\alpha \rightarrow \beta$$

s'il existe un symbole X dans α et une règle $(X \rightarrow \gamma) \in R$ tels que β s'obtient en remplaçant cette occurrence de X par γ dans α . Autrement dit, on a $\alpha \rightarrow \beta$ si on peut écrire $\alpha = \alpha_1 X \alpha_2$ et $\beta = \alpha_1 \gamma \alpha_2$, où $X \rightarrow \gamma$ est une règle. On note $\alpha \xrightarrow{*} \beta$ s'il existe $\alpha_0, \dots, \alpha_n$ tels que $\alpha = \alpha_0 \rightarrow \alpha_1 \rightarrow \dots \rightarrow \alpha_n = \beta$. On dit dans ce cas que α **se dérive** en β en n pas (ce qu'on note aussi $\alpha \xrightarrow{n} \beta$ si on veut préciser la longueur n de la dérivation). Si $n = 0$, la dérivation est vide et on obtient $\alpha = \beta$.

On peut maintenant définir le **langage engendré** par la grammaire G , noté $\mathcal{L}(G) \subseteq \Sigma^*$:

$$\mathcal{L}(G) = \{w \in \Sigma^* \mid S \xrightarrow{*} w\}.$$

Enfin, on dit qu'un langage L est **hors-contexte** s'il existe une grammaire hors-contexte qui engendre ce langage, autrement dit, s'il existe une grammaire G telle que $L = \mathcal{L}(G)$.

Compréhension de la définition et applications directes

Exercice 1 – ☆ – Application de la définition.

1. Quel est le langage engendré par la grammaire H ci-dessus? Démontrer votre réponse.
2. Donner une grammaire engendrant le langage $\{a^n b^n c^m \mid n \geq 0 \text{ et } m \geq 0\}$.

 **Exercice 2 – ☆ –** Soit Σ un alphabet. Une expression régulière sur l'alphabet Σ peut être vue comme un mot sur l'alphabet $\Sigma \cup \{\emptyset, (,), +, \cdot, *\}$. Montrer que le langage de toutes les expressions régulières est hors-contexte.

🎓 Exercice 3 – 🏠 ★ – Syntaxe des formules propositionnelles.

1. On considère l'ensemble des formules propositionnelles en une variable x , qui s'écrivent sur l'alphabet $\{\top, \perp, x, \vee, \wedge, \neg, (,)\}$. Montrer que ce langage est hors-contexte.
2. Même question pour l'ensemble des telles formules propositionnelles qui sont des tautologies.
3. Généraliser les questions précédentes aux formules ayant un nombre fini de variables.

🎓 Exercice 4 – ★ – Comptage On se place sur l'alphabet $\Sigma = \{a, b\}$.

1. Donner une grammaire engendrant les mots qui ont autant de a que de b , en le démontrant.
2. 🏠 Démontrer que la grammaire suivante engendre le langage des mots ayant deux fois plus de a que de b :
$$S \rightarrow \varepsilon \mid SS \mid aaSb \mid bSaa \mid aSbSa.$$

🎓 Exercice 5 – 🚫 ★ – Démontrer que tout langage régulier est aussi hors-contexte. On demande :

1. Une démonstration partant des automates,
2. Une démonstration partant des expressions régulières.
3. Montrer qu'il existe des langages hors-contextes qui ne sont pas réguliers.

🎓 Exercice 6 – 🏠 ★★ – Soit L un langage hors-contexte. Montrer qu'il est engendré par une grammaire **réduite**, c'est-à-dire une grammaire $G = (\Sigma, V, S, R)$ qui satisfait les propriétés suivantes :

1. Pour toute variable X , il existe $\alpha, \beta \in (\Sigma \cup V)^*$ tels que $S \xrightarrow{*} \alpha X \beta$.
2. Pour tout non-terminal X , il existe $w \in \Sigma^*$ tel que $X \xrightarrow{*} w$.

🎓 Exercice 7 – 🏠 ★★ – Soit L un langage hors-contexte. Montrer que $L \setminus \{\varepsilon\}$ est engendré par une grammaire réduite qui n'a pas de règle de la forme $X \rightarrow \varepsilon$, ni de la forme $X \rightarrow Y$, où X, Y sont des variables.

Propriétés de clôture

🎓 Exercice 8 – 🚫 ★★★ – Lemme de pompage simplifié. Soit L un langage hors-contexte. En utilisant l'exercice 7, montrer qu'il existe un entier N , dépendant de G , tel que tout mot de $\mathcal{L}(G)$ de longueur au moins N se décompose en $\alpha x \beta y \gamma$, avec x ou y non vide, et pour tout entier k , le mot $\alpha x^k \beta y^k \gamma$ est aussi dans $\mathcal{L}(G)$.

🎓 Exercice 9 – 🚫 ★★★ – Non-clôture par intersection et complément.

1. En utilisant l'exercice 8, montrer que le langage $\{a^n b^n c^n \mid n \geq 0\}$ n'est pas hors-contexte.
2. En utilisant la question précédente et la question 2 de l'exercice 1, montrer que la classe des langages hors-contexte n'est close **ni** par intersection, **ni** par complément.
3. Montrer que le complémentaire du langage $\{a^n b^n c^n \mid n \geq 0\}$ est hors-contexte.

Problèmes de décision

🎓 Exercice 10 – 🚫 ★★★ – Tests sur les grammaires. Décrire en français :

1. Un algorithme qui prend en entrée un mot $w \in \Sigma^*$ et une grammaire hors-contexte G et teste si $w \in \mathcal{L}(G)$,
2. Un algorithme qui prend en entrée une grammaire hors-contexte et teste si elle engendre le langage vide,
3. Un algorithme qui prend en entrée une grammaire hors-contexte et teste si elle engendre un langage infini.