# Chapitre 3 : Grammaires

Vincent Guisse
vincent.guisse@esisar.grenoble-inp.fr

Grenoble INP - ESISAR
3e année IR & C

22 mars 2022

## Exemple : en français

Les phrases :

| | | | |
|---|---|---|---|
| le chat | regarde | la souris | . |
| la situation | nécessite | le calme | . |
| le courage | montre | le chemin | . |

Sont générées par la **grammaire** suivante, avec l'ensemble de **symboles terminaux** $\Sigma = \{$a,b,. . .,y,z,é,., $\}$ et l'ensemble de **symboles non terminaux** $\{P_h, V_t, G_n, N_m, N_f\}$, en prenant $P_h$ comme **symbole de départ** :

$$
\begin{array}{lcl}
P_h & \rightarrow & G_n\ V_t\ G_n\ . \\
G_n & \rightarrow & \text{le}\ N_m \mid \text{la}\ N_f \\
V_t & \rightarrow & \text{regarde}|\text{nécessite}|\text{montre} \\
N_m & \rightarrow & \text{chat}|\text{calme}|\text{courage}|\text{chemin} \\
N_f & \rightarrow & \text{souris}|\text{situation}
\end{array}
$$

En effet cette grammaire permet **générer** les mots suivants sur l'alphabet $\Sigma$, par **dérivations** (notées $\Rightarrow$) successives :

$$\begin{aligned} P_h &\Rightarrow G_n\ V_t\ G_n\ . \\ &\Rightarrow \text{le}\ N_m\ V_t\ G_n\ . \\ &\Rightarrow \text{le}\ N_m\ \text{regarde}\ G_n\ . \\ &\Rightarrow \text{le chat regarde}\ G_n\ . \\ &\Rightarrow^2 \text{le chat regarde la souris.} \end{aligned}$$

Ce qu'on résume en écrivant : $P_h \Rightarrow^*$ le chat regarde la souris..

## Questions

1. Quel est le langage des mots générés par cette grammaire ?
2. Quel est le cardinal de cet ensemble ?
3. Comment modifier cette grammaire pour générer un langage infini ?

## Une grammaire qui génère un langage infini

1. On ajoute la virgule dans les non terminaux : $\Sigma' = \Sigma \cup \{,\}$
2. On renomme $P_h$ en $G_v$, et on retire le point final ($G_v$ n'est pas le symbole de départ qui reste $P_h$)
3. On utilise de la récursivité en mettant le non terminal $P_h$ à gauche et à droite d'une **règle**, cela donne :

$$
\begin{array}{lcl}
P_h & \rightarrow & Gv. \,|\, Gv,\ P_h \\
G_v & \rightarrow & G_n\ V_t\ G_n \\
G_n & \rightarrow & \text{le } N_m \,|\, \text{la } N_f \\
V_t & \rightarrow & \text{regarde}|\text{nécessite}|\text{montre} \\
N_m & \rightarrow & \text{chat}|\text{calme}|\text{courage}|\text{chemin} \\
N_f & \rightarrow & \text{souris}|\text{situation}
\end{array}
$$

## Question : Combien vaut $n$ ?

$P_h \Rightarrow^n$ Le chat regarde la souris, la situation nécessite le calme, la souris montre le courage.

## Exercice

Pour $\Sigma = \{a, b\}$ et l'ensemble de non terminaux $\{S\}$,

- Montrer que le langage généré par la grammaire ci dessous n'est pas régulier :

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow \varepsilon \end{aligned}$$

- Proposer une grammaire qui génère les palindromes sur $\Sigma = \{0, 1\}$.

## Grammaire BNF (Bachus-Naur Form) des nombres en virgule flottante :

<nombre> ::= <entier> | <entier>.<partie_decimale>

<entier> ::= 0 | <entier_non_nul>

<entier_non_nul> ::= <chiffre_non_nul> | <entier_non_nul><chiffre>

<partie_decimale> ::= <chiffre_non_nul> | <chiffre><partie_decimale>

<chiffre> ::= 0 | <chiffre_non_nul>

<chiffre_non_nul> ::= 1 | 2 | ... | 9

## BNF if en langage C :

<structure_if> ::= if "(" <condition> ")" <instruction> ;

<structure_if> ::= if "(" <condition> ")" "{" <code> "}"

<code> ::= <instruction> ;

<code> ::= <code> ; <instruction>

# Définition formelle d'un grammaire



## Définition

Une grammaire est un quadruplet $(V, \Sigma, S, \mathcal{R})$ où

- $V$ est un ensemble fini de symboles,
- $\Sigma \subset V$ est un ensemble fini de symboles terminaux (alphabet terminal), les éléments de $V \setminus \Sigma$ sont appelé les symboles non terminaux,
- $S \in V \setminus \Sigma$ est le symbole de départ (axiome, ou encore symbole initial),
- $\mathcal{R} \subset V^* \times V^*$ est une relation finie de $V^*$ et :
  - $(w_1, w_2) \in \mathcal{R}$ est noté $w_1 \to w_2$
  - si $w_1 \to w_2$, alors le mot $w_1$ contient au moins un non terminal

L'usage est d'utiliser des majuscules pour les non terminaux, et des minuscules pour les terminaux.

# Définition du langage généré par $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$



## Définition : dérivations

- On dit que $v \in V^*$ est dérivable en une étape à partir de $u \in V^*$ par la grammaire $\mathcal{G}$, et on note $u \underset{\mathcal{G}}{\Rightarrow} v$ si et seulement si :

$$\begin{cases} u = xu'y \\ v = xv'y \end{cases} \quad \text{avec } u' \to v', \text{ c'est à dire } (u', v') \in \mathcal{R}.$$

- On note $\underset{G}{\Rightarrow}^*$ la clôture réflexive transitive de la relation de dérivation en une étape.
- $w \in \Sigma^*$ est généré par la grammaire $G$ ssi $S \underset{G}{\Rightarrow}^* w$.

## Définition : langage généré par une grammaire

Le langage généré par une grammaire est l'ensemble des mots qu'elle génère.

## Exercices

1. On considère la grammaire : $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$ où :
   - $V = \{S, A\} \cup \Sigma$
   - $\Sigma = \{a, b, 0\}$
   - $\mathcal{R} = \{S \to aSa|bSb|A; A \to 0A|\varepsilon\}$

   a) Peut-on dériver le mot $aba^2 0^3 a^2 ba$ ?
   b) Quel est le langage généré par $\mathcal{G}$ ?
2. $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$ où $V = \{S, A\} \cup \Sigma$, $\Sigma = \{a, b, c, d\}$ et $\mathcal{R} = \{S \to aA|c; A \to Sb|d\}$.
   Déterminer $\mathcal{L}(\mathcal{G})$
3. $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$ où $V = \{S, A, B\} \cup \Sigma$, $\Sigma = \{0, 1, 2\}$ et $R = \{S \to 0SAB|\varepsilon, BA \to AB, 0A \to 01, 1A \to 11, 1B \to 12, 2B \to 22\}$.
   Montrer que $0^3 1^3 2^3 \in \mathcal{L}(G)$.
4. Déterminer une grammaire qui génère $\{0^{2n}1^n$ où $n \geqslant 1\}$

## Classification des grammaires

La classification des grammaires, définie en 1957 par Noam CHOMSKY, distingue les quatre classes suivantes, par inclusion croissante :

- **type 0** : Toutes les grammaires sont de type 0.
- **type 1** : Grammaires contextuelles : les règles sont de la forme : $\alpha \to \beta$ avec $|\alpha| \leqslant |\beta|$ ou $\alpha = S$ et $\beta = \varepsilon$.
- **type 2** : Grammaires hors-contexte : les règles sont de la forme : $A \to w$ où $A \in V \setminus \Sigma$ et $w \in V^*$
  Le membre de gauche de chaque règle est constitué d'un seul symbole non terminal.
- **type 3** : Grammaires régulières :
  - à droite : les règles sont de la forme $A \to aB$ ou $A \to a$ avec $(A, B) \in V \setminus \Sigma$ et $a \in \Sigma \cup \{\varepsilon\}$,
  - à gauche : les règles sont de la forme $A \to Ba$ ou $A \to a$ avec $(A, B) \in V \setminus \Sigma$ et $a \in \Sigma \cup \{\varepsilon\}$.

## Typologie des langages

A chaque type de grammaire est associé un type de langage :

- les grammaires de type 0 permettent de générer tous les langages décidables, c'est à dire tous les langages qui peuvent être reconnus par une machine de Turing.
- Les langages contextuels sont les langages générés par une grammaire contextuelle.
- Les langages hors-contexte sont les langages générés par un grammaire hors-contexte.
- On va démontrer que les langages réguliers sont les langages générés par une grammaire régulière.

**Remarque :** la typologie des langages est inclusive et toutes les inclusions sont strictes.

## Exemples

1. Dans les cas suivants déterminez le type de grammaire $(V, \Sigma, S, \mathcal{R})$ et le langage généré.
   1. $\Sigma = \{b, c\}$, $V = \{S\} \cup \Sigma$ et $\mathcal{R} = \{S \to bS \mid cc\}$.
   2. $\Sigma = \{0, 1\}$, $V = \{S\} \cup \Sigma$ et $\mathcal{R} = \{S \to 0S \mid 1S \mid 0\}$.
   3. $\Sigma = \{a, b, 0\}$, $V = \{S, A\} \cup \Sigma$ et $\mathcal{R} = \{S \to aSa \mid bSb \mid A; \quad A \to 0A \mid \varepsilon\}$.
2. On considère la grammaire $G = (V, \Sigma, S, \mathcal{R})$ avec $\Sigma = \{a, b, c\}$, $V = \{S, A, B\} \cup \Sigma$ et $\mathcal{R} = \{S \to aSAB \mid \varepsilon; \quad aA \to ab; \quad bB \to bc; \quad cA \to AB; \quad bA \to bb; \quad cB \to cc\}$.
   1. Quel est le type de $G$ ?
   2. Écrire la dérivation qui partant de l'axiome, applique 2 fois la première règle et une fois la seconde.
   3. Poursuivre la dérivation jusqu'à obtenir une chaîne de terminaux.
   4. En raisonnant par récurrence, déterminer le langage généré.

## Théorème

Un langage est régulier si et seulement si il est généré par une grammaire régulière à droite (resp. à gauche).

## Preuve de $\Rightarrow$

Si un langage est régulier, alors il est reconnu par un AFN $\mathcal{A} = (Q, \Sigma, \delta, q_i, F)$. Soit la grammaire $\mathcal{G} = (Q \cup \Sigma, \Sigma, q_i, \mathcal{R})$ où

$$\mathcal{R} = \{q \to wq' \,|\, (q, w, q') \in \delta\} \cup \{q \to \varepsilon \,|\, q \in F\}$$

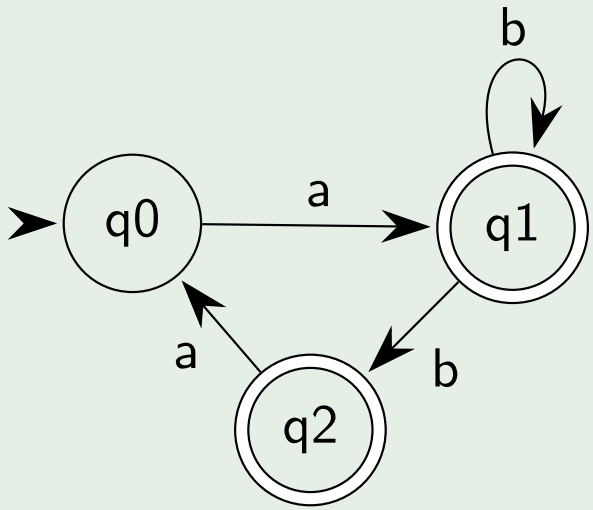
Alors, pour $q, q' \in Q$ et $w \in \Sigma^*$ on a :

$$(q, w) \vdash_{\mathcal{A}}^{*} (q', \varepsilon) \Leftrightarrow q \underset{\mathcal{G}}{\Rightarrow}^{*} wq'$$

et la conclusion, en prenant $q = q_i$, par le choix des règles faisant disparaître un non terminal.

## Exemple : une grammaire régulière pour un automate

Déterminez une grammaire qui génère le langage reconnu par l'automate ci-contre :



## Preuve de $\Leftarrow$

Réciproquement, soit $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$ une grammaire régulière.Alors l'AFN $\mathcal{A} = (V \setminus \Sigma \cup \{f\}, \Sigma, \delta, S, \{f\})$ avec :

- $f$ un nouvel état
- $\delta = \{(A, w, B) \,|\, A \to wB \in \mathcal{R}\} \cup \{(A, w, f) \,|\, A \to w \in \mathcal{R}\}$

accepte le langage généré par $\mathcal{G}$.

# Exercices



## Exercice 1

Les langages générés par les grammaires suivantes sont-ils réguliers ?
Le symbole de départ est $S$.

(1)

$$\begin{array}{lcl} S & \to & ABC \\ A & \to & aAB \\ A & \to & \varepsilon \\ B & \to & b \\ C & \to & cC \\ C & \to & \varepsilon \end{array}$$

(2)

$$\begin{array}{lcl} S & \to & AC \\ A & \to & abA \\ bA & \to & bC \\ C & \to & cC \\ C & \to & \varepsilon \end{array}$$

## Exercice 2

Soit un langage $L$ généré par une grammaire $G$ dont toutes les règles ont une des formes suivantes :

$$A \to wB,\ A \to Bw,\ A \to w$$

où $A, B \in V - \Sigma$ et $w \in \Sigma^*$. Le langage $L$ est-il obligatoirement régulier ?

## Arbre de dérivation : idée

Soit $\mathcal{G}$ un grammaire hors-contexte. On va représenter graphiquement une dérivation d'un mot généré par $\mathcal{G}$ en utilisant une arborescence ordonnée, appelée un arbre de dérivation ou arbre d'analyse.

- La racine de cet arbre sera le symbole de départ,
- les nœuds intermédiaires des non terminaux,
- les feuilles de l'arbre des terminaux.

## Exemple

Soit la grammaire $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$ où $\Sigma = \{a, b\}$, $V = \{S, A\} \cup \Sigma$ et $\mathcal{R} = \{S \to aSa \mid A; \quad A \to bA \mid \varepsilon\}$.
Déterminer l'arbre de $a^2b^3a^2$.

On parle aussi d'**arbre d'analyse** ou d'**arbre syntaxique** pour désigner la même notion.

## Arbre de dérivation : définition

On considère une grammaire hors contexte $\mathcal{G} = (V, \Sigma, S, \mathcal{R})$.

- Soit $A$ un non terminal et $w$ un élément de $V^*$ tel que $A \underset{\mathcal{G}}{\Rightarrow}^* w$. L'arbre de dérivation associé à $A \underset{\mathcal{G}}{\Rightarrow}^* w$ est l'arbre dont les nœuds intérieurs sont étiquetés par des non-terminaux, dont les feuilles sont étiquetées par des éléments de $V$, ayant pour racine $A$, dont la frontière (concaténation des feuilles de gauche à droite) est $w$ et tel que pour tout nœud intérieur $B$, si $C_1, \ldots, C_k \in V$ sont les fils de $B$ dans cet ordre, alors $B \to C_1 \ldots C_k \in \mathcal{R}$.Si $w \in \Sigma^*$, il s'agit d'un arbre de dérivation totale (sinon partielle).
- Si $w \in L(G)$, on appelle arbre d'analyse du mot $w$ (pour la grammaire $G$) tout arbre de dérivation (totale) associé à $S \underset{\mathcal{G}}{\Rightarrow}^* w$. On dit que $w$ est généré par cet arbre de dérivation.

## Définition : Grammaire ambiguë

Un mot généré par une grammaire est ambigu s'il admet au moins deux arbres de dérivation distincts pour cette grammaire.
Une grammaire est ambiguë si elle génère au moins un mot ambigu.

## Exemple

Soit la grammaire $\mathcal{G} = (V, \Sigma, E, \mathcal{R})$ où $\Sigma = \{a, b, +, *\}$,
$V = \{E, A\} \cup \Sigma$ et $\mathcal{R} = \{E \to E + E \mid E * E \mid (E) \mid a \mid b\}$.

1. $\mathcal{G}$ est-elle ambiguë ?
2. Le langage généré par $\mathcal{G}$ peut-il être généré par une grammaire non ambiguë ?