Théorie des langages

Grammaires

Jérôme Delobelle
jerome.delobelle@u-paris.fr

LIPADE - Université de Paris

Comment décrire un langage ?

Comment décrire un langage ?

- Description littéraire

  Ensemble des mots construis sur l'alphabet $\{a,b\}$*, de longueur paire*

Comment décrire un langage ?

- Description littéraire
  Ensemble des mots construis sur l'alphabet {a,b}, de longueur paire
- Enumération (écriture en extension)

$$L = \{\epsilon, aa, bb, ab, ba, aaaa, bbbb, aaab, baaa, \ldots\}$$

Comment décrire un langage ?

- Description littéraire

  Ensemble des mots construis sur l'alphabet {a,b}, de longueur paire

- Enumération (écriture en extension)

  $$L = \{\epsilon, aa, bb, ab, ba, aaaa, bbbb, aaab, baaa, \ldots\}$$

- Expression régulière

  $$((a+b)(a+b))^*$$

Comment décrire un langage ?

- Description littéraire

  Ensemble des mots construis sur l'alphabet {a,b}, de longueur paire

- Enumération (écriture en extension)

$$L = \{\epsilon, aa, bb, ab, ba, aaaa, bbbb, aaab, baaa, \ldots\}$$

- Expression régulière

$$((a+b)(a+b))^*$$

- Grammaire de réecriture

  Ensemble de règles pour générer les mots du langage

Comment décrire un langage ?

- Description littéraire

  Ensemble des mots construis sur l'alphabet $\{a,b\}$*, de longueur paire*

- Enumération (écriture en extension)

  $$L = \{\epsilon, aa, bb, ab, ba, aaaa, bbbb, aaab, baaa, \ldots\}$$

- Expression régulière

  $$((a+b)(a+b))^*$$

- Grammaire de réecriture

  Ensemble de règles pour générer les mots du langage

- Reconnaisseur (automates)

  Machine permettant de générer tous les mots du langage

Grammaires

1. Principe de base
2. Définitions
3. Langage généré par une grammaire
4. Arbres de dérivation
5. Types de grammaires

Principe de base

Principe de base

- Ensemble de règles pour générer les mots du langage
- Sous la forme de règles de réécriture
  - $\rightarrow$ Remplacer une séquence de symboles par une autre séquence
- Mots générés = mots obtenus à partir d'un symbole spécial appelé symbole de départ ou axiome

Exemple

- Considérons la phrase suivante :

  `La vieille dame regarde la petite fille`

- Peut-on construire une grammaire qui permette de générer cette phrase ?

Exemple

- Considérons la phrase suivante :

  `La vieille dame regarde la petite fille`

- Peut-on construire une grammaire qui permette de générer cette phrase ?
- Alphabet : $\Sigma = \{ \texttt{la, vieille, petite, dame, fille, regarde} \}$

Exemple

- Considérons la phrase suivante :

  `La vieille dame regarde la petite fille`

- Peut-on construire une grammaire qui permette de générer cette phrase ?
- Alphabet : $\Sigma = \{$ `la, vieille, petite, dame, fille, regarde`$\}$
- Structure de la phrase :
  - Un groupe sujet (article, adjectif, nom)
  - Un verbe
  - Un groupe complément d'objet (article, adjectif, nom)

Exemple

Règles de production

1. $\langle$Phrase$\rangle \rightarrow \langle$Sujet$\rangle\langle$Verbe$\rangle\langle$Complément$\rangle$
2. $\langle$Sujet$\rangle \rightarrow \langle$Groupe Nominal$\rangle$
3. $\langle$Complément$\rangle \rightarrow \langle$Groupe Nominal$\rangle$
4. $\langle$Groupe Nominal$\rangle \rightarrow \langle$Article$\rangle\langle$Nom$\rangle$
5. $\langle$Groupe Nominal$\rangle \rightarrow \langle$Article$\rangle\langle$Adjectif$\rangle\langle$Nom$\rangle$
6. $\langle$Article$\rangle \rightarrow$ `la`
7. $\langle$Nom$\rangle \rightarrow$ `dame` | `fille`
8. $\langle$Adjectif$\rangle \rightarrow$ `vieille` | `petite`
9. $\langle$Verbe$\rangle \rightarrow$ `regarde`

Définitions

Grammaire

Grammaire

Une **grammaire** est définie par un quadruplet $G = \langle V, \Sigma, P, S \rangle$ où

- V est un alphabet
- $\Sigma \subseteq V$ est l'ensemble des symboles terminaux
- $V \setminus \Sigma$ est l'ensemble des symboles non terminaux
- $S \in V \setminus \Sigma$ est le symbole de départ ou axiome
- $P \subseteq (V^+ \times V^*)$ est l'ensemble (fini) de règles de production

Grammaire

Une **grammaire** est définie par un quadruplet $G = \langle V, \Sigma, P, S \rangle$ où

- V est un alphabet
- $\Sigma \subseteq V$ est l'ensemble des symboles terminaux
- $V \setminus \Sigma$ est l'ensemble des symboles non terminaux
- $S \in V \setminus \Sigma$ est le symbole de départ ou axiome
- $P \subseteq (V^+ \times V^*)$ est l'ensemble (fini) de règles de production

Notations :

- Σ : lettres minuscules
- $V \setminus \Sigma$: lettre majuscules
- Règles de production : $\alpha \rightarrow \beta$
    - Signification intuitive : l'élément $\alpha \in V^+$ peut être remplacé par $\beta \in V^*$

Grammaire : exemple

Soit $G_1 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{:=, a, b, c, +, *,), (\}$
- $V \setminus \Sigma = \{S, I, E\}$
- Axiome S
- 8 règles de production :

$$
\begin{array}{ll}
S \to I := E & I \to a \\
I \to b & I \to c \\
E \to E + E & E \to E * E \\
E \to (E) & E \to I
\end{array}
$$

Grammaire : exemple

Soit $G_1 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{:=, a, b, c, +, *,), (\}$
- $V \setminus \Sigma = \{S, I, E\}$
- Axiome S
- 8 règles de production :

$$\begin{array}{ll} S \rightarrow I := E & I \rightarrow a \\ I \rightarrow b & I \rightarrow c \\ E \rightarrow E + E & E \rightarrow E * E \\ E \rightarrow (E) & E \rightarrow I \end{array}$$

On peut aussi écrire $I \rightarrow a|b|c$.

Grammaire : exemple

Soit $G_2 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production :
  - $S \rightarrow aSa$
  - $S \rightarrow SbS$
  - $S \rightarrow \epsilon$

Grammaire : exemple

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

Langage généré par une grammaire

Problématique(s)

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

Problématique(s)

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

Est-ce que le mot *aaaabaaaa* peut être généré à partir de G_3 ?

Problématique(s)

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

Est-ce que le mot *aaaaabaaaa* peut être généré à partir de G_3 ?

- Problème 1 : Comment savoir si un mot est généré ou non par une grammaire ?

Problématique(s)

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

Est-ce que le mot *aaaaabaaaa* peut être généré à partir de G_3 ?

- Problème 1 : Comment savoir si un mot est généré ou non par une grammaire ?
- Problème 2 : Quel est le langage généré par une grammaire ?

Dérivation

Dérivation en une étape

Soit $G = \langle V, \Sigma, P, S\rangle$, $u \in V^+$ et $v \in V^*$. G permet de **dériver** v de u **en une étape**, noté $u \underset{G}{\longrightarrow} v$, si et seulement si

- $u = xu'y$ (u peut être décomposé en x, u' et y ; x et y peuvent être vides)
- $v = xv'y$ (v peut être décomposé en x, v' et y)
- $u' \to v'$ est dans P

Dérivation

Dérivation en une étape

Soit $G = \langle V, \Sigma, P, S \rangle$, $u \in V^+$ et $v \in V^*$. G permet de **dériver** v de u **en une étape**, noté $u \underset{G}{\longrightarrow} v$, si et seulement si

- $u = xu'y$ (u peut être décomposé en x, u' et y ; x et y peuvent être vides)
- $v = xv'y$ (v peut être décomposé en x, v' et y)
- $u' \rightarrow v'$ est dans P

Dérivation en plusieurs étapes

G permet de **dériver** v de u **en plusieurs étapes**, noté $u \overset{*}{\underset{G}{\longrightarrow}} v$, si et seulement si $\exists k \geq 0$ et $\exists v_0, \ldots, v_k \in V^*$ tels que

- $u = v_0$
- $v = v_k$
- $v_i \underset{G}{\longrightarrow} v_{i+1}$ pour $0 \leq i < k$

Langage généré par une grammaire

Mots générés par une grammaire

Soit $G = \langle V, \Sigma, P, S \rangle$.

Les **mots générés par** G sont les mots $v \in \Sigma^*$ (symboles terminaux) qui peuvent être dérivés à partir de l'axiome : $S \xrightarrow[G]{*} v$.

Langage généré par une grammaire

Mots générés par une grammaire

Soit $G = \langle V, \Sigma, P, S \rangle$.

Les **mots générés par** G sont les mots $v \in \Sigma^*$ (symboles terminaux) qui peuvent être dérivés à partir de l'axiome : $S \xrightarrow[G]{*} v$.

Langage généré par une grammaire

Soit $G = \langle V, \Sigma, P, S \rangle$.

Le **langage généré par** G, noté $\mathcal{L}(G)$ est l'ensemble des mots générés par G.

$$\mathcal{L}(G) = \{v \in \Sigma^* | S \xrightarrow[G]{*} v\}$$

Langage généré par une grammaire

Remarque :

- Une grammaire définit un seul langage
- Par contre, un même langage peut être généré par plusieurs grammaires différentes

Langage généré par une grammaire

Remarque :

- Une grammaire définit un seul langage
- Par contre, un même langage peut être généré par plusieurs grammaires différentes

Equivalence entre deux grammaires

Deux grammaires G et G' sont équivalentes si et seulement si elles génèrent le même langage.

$$\mathcal{L}(G) = \mathcal{L}(G')$$

Grammaire : exemple

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

Grammaire : exemple

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \to aS|aSbS|\epsilon$

On veut montrer que le mot $aaba \in \mathcal{L}(G)$

Grammaire : exemple

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

On veut montrer que le mot $aaba \in \mathcal{L}(G)$

$$\underline{S} \underset{G_3}{\longrightarrow} aSb\underline{S} \underset{G_3}{\longrightarrow} aSba\underline{S} \underset{G_3}{\longrightarrow} a\underline{S}ba\epsilon \underset{G_3}{\longrightarrow} aa\underline{S}ba \underset{G_3}{\longrightarrow} aa\epsilon ba \underset{G_3}{\longrightarrow} aaba$$

Grammaire : exemple

Soit $G_3 = \langle V, \Sigma, P, S \rangle$ avec

- $\Sigma = \{a, b\}$
- $V \setminus \Sigma = \{S\}$
- Axiome S
- 3 règles de production : $S \rightarrow aS|aSbS|\epsilon$

On veut montrer que le mot $aaba \in \mathcal{L}(G)$

$$\underline{S} \xrightarrow[G_3]{} aSb\underline{S} \xrightarrow[G_3]{} aSba\underline{S} \xrightarrow[G_3]{} a\underline{S}ba\epsilon \xrightarrow[G_3]{} aa\underline{S}ba \xrightarrow[G_3]{} aa\epsilon ba \xrightarrow[G_3]{} aaba$$

Seconde dérivation possible :

$$\underline{S} \xrightarrow[G_3]{} a\underline{S}bS \xrightarrow[G_3]{} aa\underline{S}bS \xrightarrow[G_3]{} aa\epsilon b\underline{S} \xrightarrow[G_3]{} aaba\epsilon \xrightarrow[G_3]{} aaba$$

Dérivation la plus à gauche

Dérivation la plus à gauche (LPG)

Soit $G = \langle V, \Sigma, P, S \rangle$, et $w \in \Sigma^*$

$S \xrightarrow[G]{*} w$ est une **dérivation la plus à gauche (LPG)** si, à chaque étape de la dérivation, c'est la variable la plus à gauche qui est dérivée. Donc, si $\exists w_1, \ldots, w_n$ tels que

- $S = w_0 \xrightarrow[G]{} w_1 \xrightarrow[G]{} \ldots \xrightarrow[G]{} w_n \xrightarrow[G]{} w_{n+1} = w$, et
- $\forall i$, $0 \leq i \leq n$
  - $w_i = u_i A_i v_i$,
  - $w_{i+1} = u_i \alpha_i v_i$ et
  - $A_i \rightarrow \alpha_i$

alors $u_i \in \Sigma^*$ (u_i est un symbole terminal, et ne contient donc pas de variable).

Arbres de dérivation

Arbres de dérivation pour une grammaire hors contexte

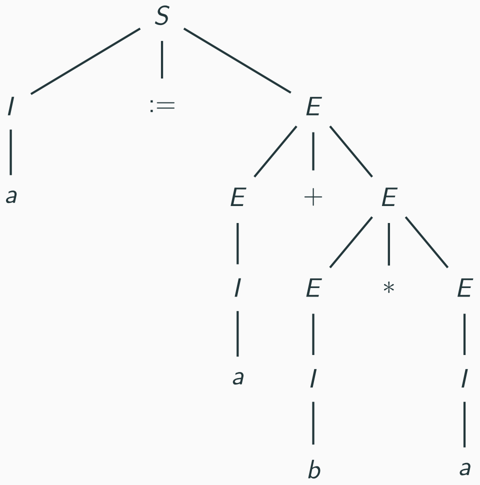
Arbres de dérivation

Soit $G = \langle V, \Sigma, P, S \rangle$ une grammaire *hors contexte*. Un arbre D est un **arbre de dérivation** pour un mot w à partir de l'axiome S si :

- La racine de D est étiquetée par S (l'axiome)
- Les feuilles de D sont étiquetées par des éléments de $\Sigma \cup \epsilon$ (symboles terminaux)
- L'étiquette d'une feuille est le mot vide seulement si la feuille est fille unique
- Les nœuds de D qui ne sont pas des feuilles sont étiquetés par un symbole non terminal ($V \setminus \Sigma$)
- Pour tout nœud, si Y est l'étiquette du nœud, et si $Z_1, \ldots, Z_n$ sont les nœuds de ses fils, **dans cet ordre**, alors $Y \rightarrow Z_1 \ldots Z_n$ est une règle
- Le mot des feuilles de D, c'est-à-dire le mot obtenu en concaténant les étiquettes des feuilles de la gauche vers la droite, est le mot w

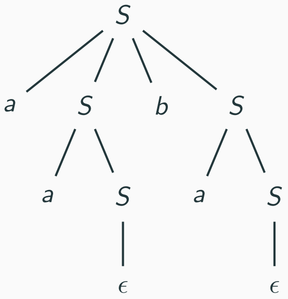
Arbres de dérivation : exemple

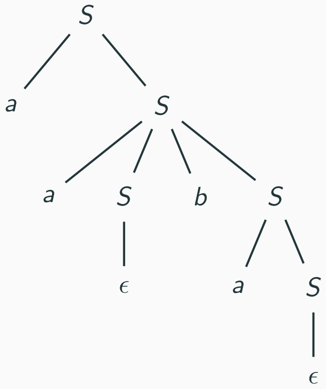
$a := a + b * a \in \mathcal{L}(G_1)$



Arbres de dérivation : exemple

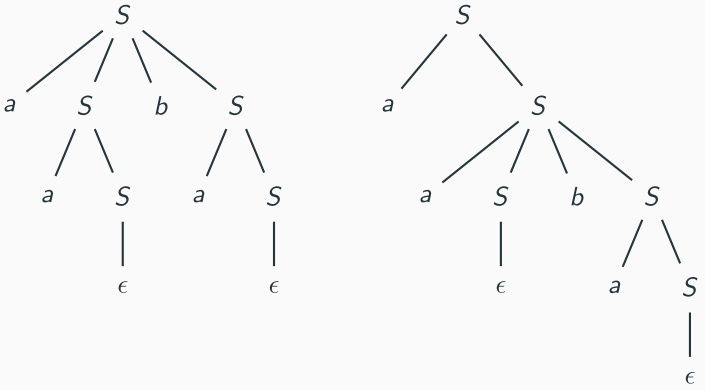
$aaba \in \mathcal{L}(G_3)$. Il existe deux arbres de dérivation différents pour ce mot.





Arbres de dérivation : exemple

$aaba \in \mathcal{L}(G_3)$. Il existe deux arbres de dérivation différents pour ce mot.



La grammaire est **ambigüe**.

Grammaires ambigües

Ambiguité

Une grammaire G est **ambigüe** s'il existe un mot de $\mathcal{L}(G)$ qui a au moins deux dérivations LPG à partir de S (et donc deux arbres de dérivation).

Dans le cas contraire G est non ambigüe.

Grammaires ambigües

Ambiguité

Une grammaire G est **ambigüe** s'il existe un mot de $\mathcal{L}(G)$ qui a au moins deux dérivations LPG à partir de S (et donc deux arbres de dérivation).

Dans le cas contraire G est non ambigüe.

G_1 est ambigüe.

Grammaires ambigües

Théorème

Etant donné une grammaire G, un mot est généré par G ($S \xrightarrow[G]{*} w$) si et seulement si il existe un arbre de dérivation qui génère w.

Grammaires ambigües

Théorème

Etant donné une grammaire G, un mot est généré par G ($S \xrightarrow[G]{*} w$) si et seulement si il existe un arbre de dérivation qui génère w.

Langage ambigüe

Un langage est **ambigüe de façon inhérente** si toutes les grammaires qui l'engendrent sont ambigües.

On dira qu'un langage est **non ambigüe** s'il n'est pas ambigüe de façon inhérente.

Grammaires ambigües

Théorème

Etant donné une grammaire G, un mot est généré par G ($S \xrightarrow[G]{*} w$) si et seulement si il existe un arbre de dérivation qui génère w.

Langage ambigüe

Un langage est **ambigüe de façon inhérente** si toutes les grammaires qui l'engendrent sont ambigües.

On dira qu'un langage est **non ambigüe** s'il n'est pas ambigüe de façon inhérente.

Lemme de Parikh

Il existe au moins un langage ambigüe de façon inhérente.

$$L = \{a^p b^q c^r | p = q \text{ ou } q = r \text{ avec } p, q, r \geq 1\}$$

Types de grammaires

Type de règles

Classement des grammaires en fonction de leurs règles de production :

Type de règles

Classement des grammaires en fonction de leurs règles de production :

1. Règles régulières à gauche de la forme :

$$\begin{array}{llll} A \rightarrow Ba & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

Type de règles

Classement des grammaires en fonction de leurs règles de production :

1. Règles régulières à gauche de la forme :

$$\begin{array}{llll} A \rightarrow Ba & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

2. Règles régulières à droite de la forme :

$$\begin{array}{llll} A \rightarrow aB & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

Type de règles

Classement des grammaires en fonction de leurs règles de production :

1. Règles régulières à gauche de la forme :

$$\begin{array}{llll} A \rightarrow Ba & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

2. Règles régulières à droite de la forme :

$$\begin{array}{llll} A \rightarrow aB & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

3. Règles hors-contexte de la forme :

$$A \rightarrow \beta, \text{ où } A \in V \setminus \Sigma, \ \beta \in V^*$$

Type de règles

Classement des grammaires en fonction de leurs règles de production :

1. Règles régulières à gauche de la forme :

$$\begin{array}{llll} A \to Ba & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \to a & & a \in \Sigma & \text{terminaux} \end{array}$$

2. Règles régulières à droite de la forme :

$$\begin{array}{llll} A \to aB & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \to a & & a \in \Sigma & \text{terminaux} \end{array}$$

3. Règles hors-contexte de la forme :

$$A \to \beta, \text{ où } A \in V \setminus \Sigma, \ \beta \in V^*$$

4. Règles contextuelles de la forme :

$$\alpha \to \beta, \text{ où } |\alpha| \leq |\beta| \text{ si } \beta \neq \epsilon, \ \alpha \in V^+, \ \beta \in V^*$$

Type de règles

Classement des grammaires en fonction de leurs règles de production :

1. Règles régulières à gauche de la forme :

$$\begin{array}{llll} A \rightarrow Ba & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

2. Règles régulières à droite de la forme :

$$\begin{array}{llll} A \rightarrow aB & \text{où} & A, B \in V \setminus \Sigma & \text{non terminaux} \\ A \rightarrow a & & a \in \Sigma & \text{terminaux} \end{array}$$

3. Règles hors-contexte de la forme :

$$A \rightarrow \beta, \text{ où } A \in V \setminus \Sigma, \ \beta \in V^*$$

4. Règles contextuelles de la forme :

$$\alpha \rightarrow \beta, \text{ où } |\alpha| \leq |\beta| \text{ si } \beta \neq \epsilon, \ \alpha \in V^+, \ \beta \in V^*$$

5. Règles sans restriction de la forme :

$$\alpha \rightarrow \beta, \text{ où } |\alpha| \geq 1$$

Types de grammaires – Hiérarchie de CHOMSKY

Une grammaire est :

- De Type 3 ou régulière (ou linéaire à gauche ou à droite) si
  - **toutes** ses règles de production sont régulières à gauche, ou
  - **toutes** ses règles de production sont régulières à droite.
- De Type 2 ou hors-contexte (Context-Free) si **toutes** ses règles de production sont hors-contexte
- De Type 1 ou contextuelles (ou sensibles au contrôle) si **toutes** ses règles de production sont contextuelles
- De Type 0 ou sans restriction si **toutes** ses règles de production sont sans restriction

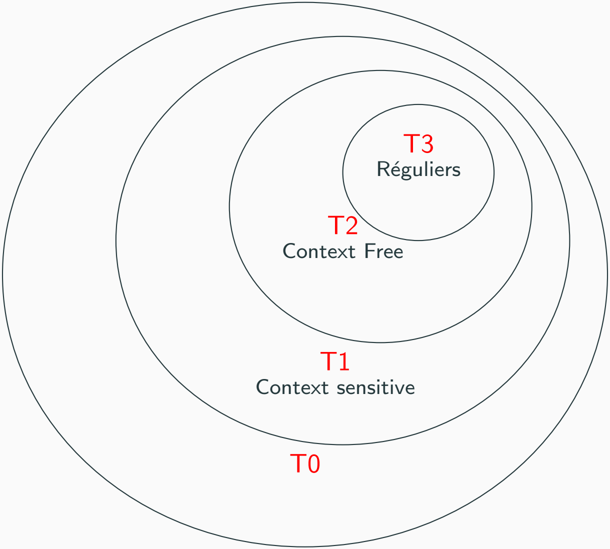
Lien grammaire / langage

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

- les grammaires de Type 3 génèrent les langages réguliers,
- les grammaires de Type 2 génèrent les langages hors-contexte (ou algébrique),
- les grammaires de Type 1 génèrent les langages contextuels,
- les grammaires de Type 0 génèrent les langages "décidables"
- sinon ce sont des langages "indécidables".

Ces langages sont ordonnés par inclusion.

Hiérarchie de CHOMSKY



Exemple de langages réguliers

- $L = \{a^n b^p | n, p \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S \rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S, R\}$
  - Règles de production :
    - $S \rightarrow \epsilon | aS | bR$
    - $R \rightarrow \epsilon | bR$

Exemple de langages réguliers

- $L = \{a^n b^p | n, p \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S\rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S, R\}$
  - Règles de production :
    - $S \rightarrow \epsilon | aS | bR$
    - $R \rightarrow \epsilon | bR$
  - L peut aussi d'écrire sous forme d'expression régulière : $L = a^* b^*$

Exemple de langages réguliers

- $L = \{a^n b^p | n, p \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S\rangle$ avec
    - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S, R\}$
    - Règles de production :
        - $S \to \epsilon | aS | bR$
        - $R \to \epsilon | bR$
    - L peut aussi d'écrire sous forme d'expression régulière : $L = a^* b^*$
- $L = \{(a + b)^n | n \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S\rangle$ avec
    - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S\}$
    - Règles de production :
        - $S \to \epsilon | aS | bS$

Exemple de langages réguliers

- $L = \{a^n b^p | n, p \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S \rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S, R\}$
  - Règles de production :
    - $S \to \epsilon | aS | bR$
    - $R \to \epsilon | bR$
  - L peut aussi d'écrire sous forme d'expression régulière : $L = a^* b^*$
- $L = \{(a + b)^n | n \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S \rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S\}$
  - Règles de production :
    - $S \to \epsilon | aS | bS$
  - L peut aussi d'écrire sous forme d'expression régulière : $L = (a + b)^*$

Exemple de langages hors-contexte

- $L = \{a^n b^n | n \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S \rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S\}$
  - $S \rightarrow \epsilon | aSb$

Exemple de langages hors-contexte

- $L = \{a^n b^n | n \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S\rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S\}$
  - $S \rightarrow \epsilon | aSb$
- $L = \{w\tilde{w} | w \in \Sigma^*\}$ (ensemble des palindromes de longueur paire sur Σ) ; $G = \langle V, \Sigma, P, S\rangle$ avec
  - $\Sigma = \{a, b\}$; $V \setminus \Sigma = \{S\}$
  - $S \rightarrow \epsilon | aSa | bSb$

Exemple de langages contextuels

- $L = \{a^n b^n c^n | n \in \mathbb{N}\}$; $G = \langle V, \Sigma, P, S \rangle$ avec
  - $\Sigma = \{a, b, c\}$; $V \setminus \Sigma = \{S, A, B, C\}$
  - Règles de production :
    - $S \to \epsilon | A$
    - $A \to aABC | aBC$
    - $CB \to BC$
    - $aB \to ab$
    - $bB \to bb$
    - $bC \to bc$
    - $cC \to cc$

Forme normale d'une grammaire

Rappel : une grammaire hors-contexte (ou algébrique) générant un langage n'est pas nécessairement unique !

Forme normale d'une grammaire

Rappel : une grammaire hors-contexte (ou algébrique) générant un langage n'est pas nécessairement unique !

$G_1 = \langle V, \Sigma, P, S \rangle$ avec $\Sigma = \{a, b\}$,
$V \setminus \Sigma = \{S, S_1, S_2, S_3, S_4, A, A_1, B, B_1\}$
Règles de production P :

- $S \to S_1$
- $S_1 \to S_2$
- $S_2 \to S_3 | S_4$
- $S_3 \to aA$
- $S_4 \to bB$
- $A \to aA_1$
- $A_1 \to a | \epsilon$
- $B \to bB_1$
- $B_1 \to b | \epsilon$

$G_2 = \langle V, \Sigma, P, S \rangle$ avec $\Sigma = \{a, b\}$ et
$V \setminus \Sigma = \{S, A, B\}$
Règles de production P :

- $S \to aaA | bbB$
- $A \to a | \epsilon$
- $B \to b | \epsilon$

Forme normale d'une grammaire

Rappel : une grammaire hors-contexte (ou algébrique) générant un langage n'est pas nécessairement unique !

$G_1 = \langle V, \Sigma, P, S \rangle$ avec $\Sigma = \{a, b\}$,
$V \backslash \Sigma = \{S, S_1, S_2, S_3, S_4, A, A_1, B, B_1\}$
Règles de production P :

- $S \rightarrow S_1$
- $S_1 \rightarrow S_2$
- $S_2 \rightarrow S_3 | S_4$
- $S_3 \rightarrow aA$
- $S_4 \rightarrow bB$
- $A \rightarrow aA_1$
- $A_1 \rightarrow a|\epsilon$
- $B \rightarrow bB_1$
- $B_1 \rightarrow b|\epsilon$

$G_2 = \langle V, \Sigma, P, S \rangle$ avec $\Sigma = \{a, b\}$ et
$V \setminus \Sigma = \{S, A, B\}$
Règles de production P :

- $S \rightarrow aaA|bbB$
- $A \rightarrow a|\epsilon$
- $B \rightarrow b|\epsilon$

$\mathcal{L}(G_1) = \mathcal{L}(G_2) = \{aa, aaa, bb, bbb\}$

Forme normale de Chomsky

Definition

Une grammaire hors-contexte $G = \langle V, \Sigma, P, S \rangle$ est sous **forme normale de Chomsky** si les règles de production de G sont toutes de l'une des formes suivantes :

- $A \to BC$ avec $A \in V \backslash \Sigma$ et $B, C \in VV \backslash \Sigma \cup \{S\}$,
- $A \to a$ avec $A \in V \backslash \Sigma$ et $a \in \Sigma$.

Forme normale de Chomsky

Definition

Une grammaire hors-contexte $G = \langle V, \Sigma, P, S \rangle$ est sous **forme normale de Chomsky** si les règles de production de G sont toutes de l'une des formes suivantes :

- $A \to BC$ avec $A \in V \backslash \Sigma$ et $B, C \in VV \backslash \Sigma \cup \{S\}$,
- $A \to a$ avec $A \in V \backslash \Sigma$ et $a \in \Sigma$.

Théorème

Soit $G = \langle V, \Sigma, P, S \rangle$ une grammaire hors-contexte. Il est possible de construire de manière efficace une grammaire équivalente G' sous forme normale de Chomsky.

Algorithme pour transformer une grammaire hors-contexte en une grammaire équivalente écrite sous forme normale de Chomsky

Pour les règles ne respectant pas la forme normale de Chomsky :

1. Remplacer chaque paire de symboles non-terminaux par un seul symbole non-terminal (et sa règle correspondante), jusqu'à ce que toutes les règles soient binaires.
$$S \rightarrow aBC \quad \Longrightarrow \quad S \rightarrow AC; A \rightarrow aB$$
2. Remplacer les symboles terminaux par des symboles terminaux + ajouter des règles $A \rightarrow a$ pour compenser,
$$S \rightarrow aB \quad \Longrightarrow \quad S \rightarrow AB; A \rightarrow a$$
3. Remplacer les règles avec un seul symbole non-terminal par ce qui peut être produit par ce symbole non-terminal,
$$S \rightarrow A; A \rightarrow aBC \quad \Longrightarrow \quad S \rightarrow aBC$$

Exemple de transformation en forme normale de Chomsky

$S \rightarrow Aa|Bb$
$A \rightarrow A'$
$A' \rightarrow a|b$
$B \rightarrow CDc$
$C \rightarrow c$
$D \rightarrow d$

Exemple de transformation en forme normale de Chomsky

1. Remplacer chaque paire de symboles non-terminaux par un seul symbole non-terminal (et sa règle correspondante), jusqu'à ce que toutes les règles soient binaires.

$S \rightarrow Aa \vert Bb$	$S \rightarrow Aa \vert Bb$
$A \rightarrow A'$	$A \rightarrow A'$
$A' \rightarrow a \vert b$	$A' \rightarrow a \vert b$
$B \rightarrow CDc$	$B \rightarrow CE$
$C \rightarrow c$	$E \rightarrow Dc$
$D \rightarrow d$	$C \rightarrow c$
	$D \rightarrow d$

Exemple de transformation en forme normale de Chomsky

2. Remplacer les symboles terminaux par des symboles terminaux + ajouter des règles $A \to a$ pour compenser.

$S \to Aa \mid Bb$	$S \to Aa \mid Bb$	$S \to AF \mid BG$
$A \to A'$	$A \to A'$	$F \to a$
$A' \to a \mid b$	$A' \to a \mid b$	$G \to b$
$B \to CDc$	$B \to CE$	$A \to A'$
$C \to c$	$E \to Dc$	$A' \to a \mid b$
$D \to d$	$C \to c$	$B \to CE$
	$D \to d$	$E \to DH$
		$H \to c$
		$C \to c$
		$D \to d$

Exemple de transformation en forme normale de Chomsky

3. Remplacer les règles avec un seul symbole non-terminal par ce qui peut être produit par ce symbole non-terminal.

$S \rightarrow Aa|Bb$
$A \rightarrow A'$
$A' \rightarrow a|b$
$B \rightarrow CDc$
$C \rightarrow c$
$D \rightarrow d$

$S \rightarrow Aa|Bb$
$A \rightarrow A'$
$A' \rightarrow a|b$
$B \rightarrow CE$
$E \rightarrow Dc$
$C \rightarrow c$
$D \rightarrow d$

$S \rightarrow AF|BG$
$F \rightarrow a$
$G \rightarrow b$
$A \rightarrow A'$
$A' \rightarrow a|b$
$B \rightarrow CE$
$E \rightarrow DH$
$H \rightarrow c$
$C \rightarrow c$
$D \rightarrow d$

Exemple de transformation en forme normale de Chomsky

Grammaire écrite sous la forme normale de Chomsky

$$\begin{aligned} S &\rightarrow Aa|Bb \\ A &\rightarrow A' \\ A' &\rightarrow a|b \\ B &\rightarrow CDc \\ C &\rightarrow c \\ D &\rightarrow d \end{aligned}$$

$$\begin{aligned} S &\rightarrow Aa|Bb \\ A &\rightarrow A' \\ A' &\rightarrow a|b \\ B &\rightarrow CE \\ E &\rightarrow Dc \\ C &\rightarrow c \\ D &\rightarrow d \end{aligned}$$

$$\begin{aligned} S &\rightarrow AF|BG \\ F &\rightarrow a \\ G &\rightarrow b \\ A &\rightarrow A' \\ A' &\rightarrow a|b \\ B &\rightarrow CE \\ E &\rightarrow DH \\ H &\rightarrow c \\ C &\rightarrow c \\ D &\rightarrow d \end{aligned}$$

$$\begin{aligned} S &\rightarrow AF|BG \\ F &\rightarrow a \\ G &\rightarrow b \\ \color{red}{A} &\color{red}{\rightarrow a|b} \\ B &\rightarrow CE \\ E &\rightarrow DH \\ H &\rightarrow c \\ C &\rightarrow c \\ D &\rightarrow d \end{aligned}$$

Forme normale de Greibach

Definition

Une grammaire hors-contexte $G = \langle V, \Sigma, P, S\rangle$ est sous **forme normale de Greibach** si les règles de production de G sont toutes de l'une des formes suivantes :

- $A \to aA_1 \dots A_n$ avec $A \in V \backslash \Sigma$, $a \in \Sigma$ et $A_i \in VV \backslash \Sigma \cup \{S\}$,
- $A \to a$ avec $A \in V \backslash \Sigma$ et $a \in \Sigma$

Grammaires vs Reconnaisseurs

- Une grammaire d'un langage L permet de générer tous les mots appartenant à L
- Un reconnaisseur pour un langage L est un programme qui détermine si le mot w appartient à L ou non
- Pour chaque classe de grammaire, il existe une classe de reconnaisseurs qui définit la même classe de langages

Type de grammaire	**Type de reconnaisseur**
Grammaire régulière	Automate fini
Grammaire hors-contexte	Automate à pile
Grammaire contextuelle	Linear Bounded Automaton
Grammaire sans restriction	Machine de Turing

Hiérarchie de CHOMSKY