

# Analyseur syntaxique

## Rapport de projet VISI\_201

Porteries Tristan

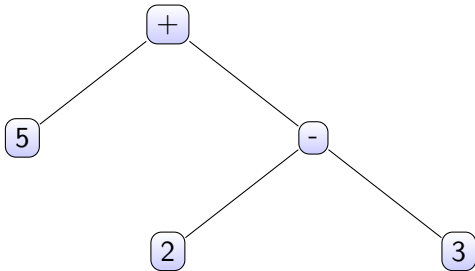
12 avril 2018

- ① Grammaires non-contextuelles
- ② Hiérarchie de Chomsky
- ③ Analyseur lexical
  - Automates finis
- ④ Analyseur syntaxique
  - Backus-Naur Form
  - Automates à piles
- Analyseur LL
- Analyseur LR
- ⑤ **Arbre abstrait**
  - Arbre de dérivation
  - Grammaire attribuée
    - Attributs synthétisés
    - Attributs hérités
- ⑥ **Implémentation**
  - Usage

Objectifs d'un analyseur syntaxique :

- déterminer l'existence d'une chaîne selon une grammaire ;
- reporter des erreurs ;
- construire un arbre abstrait de syntaxe.

5 + 2 - 3



# Sommaire

- 1 Grammaires non-contextuelles
- 2 Hiérarchie de Chomsky
- 3 Analyseur lexical
- 4 Analyseur syntaxique
- 5 Arbre abstrait
- 6 Implémentation

$$G = (V, A, S, P)$$

- $V$  : ensemble des non-terminaux.
- $A$  : ensemble des terminaux.
- $S$  : non-terminal axiome,  $S \in V$ .
- $P$  : ensemble des productions,  $P \subset V \times (A \cup V)^*$ .

grammaire ambiguës

$$S \rightarrow aSb$$

$$S \rightarrow \epsilon$$

Une dérivation d'un non-terminal selon une production mène à une proto-phrased ou phrase.

aabb

$$S \rightarrow aSb$$

$$S \rightarrow aaSbb$$

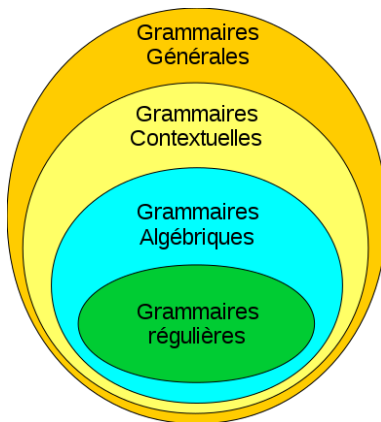
$$S \rightarrow aabb$$

$$S \xrightarrow{*} aabb$$



# Sommaire

- 1 Grammaires non-contextuelles
- 2 Hiérarchie de Chomsky
- 3 Analyseur lexical
- 4 Analyseur syntaxique
- 5 Arbre abstrait
- 6 Implémentation



Grammaires contextuelles	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Grammaires algébriques	$A \rightarrow \gamma$
Grammaire régulières	$A \rightarrow aB, A \rightarrow a$

# Sommaire

- Automates finis

1 Grammaires non-contextuelles

2 Hiérarchie de Chomsky

3 Analyseur lexical

4 Analyseur syntaxique

5 Arbre abstrait

6 Implémentation

Analyse de la chaîne d'entrée pour reconnaître des lexèmes.  
 Utilisation d'expression pour décrire des grammaires linéaire.  
 Soit  $x$  et  $y$  appartenant au langage :

Opération	Notation	Exemple
Concaténation	$xy$	$\{ab\}$
Union	$x y$	$\{a, b\}$
Étoile Kleene	$(x y)^*$	$\{\epsilon, a, b, ab, ba, aa, bb, \dots\}$

Limitation :  $a^n b^n$

Théorème de Kleene : l'ensemble des langages rationnels sur un alphabet  $A$  est exactement l'ensemble des langages sur  $A$  reconnaissables par automate fini.

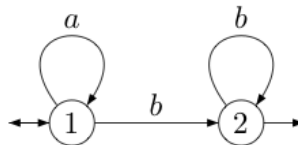


FIGURE – Automate reconnaissant le langage  $a^*b^*$

# Sommaire

- 1 Grammaires non-contextuelles
- 2 Hiérarchie de Chomsky
- 3 Analyseur lexical
- 4 Analyseur syntaxique
  - Backus-Naur Form
  - Automates à piles
  - Analyseur LL
  - Analyseur LR
- 5 Arbre abstrait
- 6 Implémentation

Description de règles d'analyse (productions) :

$$A \rightarrow aAb$$
$$\langle A \rangle ::= a \langle A \rangle b$$





Dérivation de l'axiome à partir de la gauche, si les dérivations mènent à la phrase d'entrée : accepter la phrase.

Si  $S \xrightarrow{*} \omega$  accepter  $\omega$

$$S \rightarrow aAb$$

$$S \rightarrow \epsilon$$

$$\omega = aabb$$

- ①  $S$
- ②  $aAb$
- ③  $aaAbb$
- ④  $aabb$

Contrainte de récursivité gauche des productions.

$$A \rightarrow A\alpha$$

$$A \rightarrow \beta$$

Suppression de la récursivité gauche :

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A'$$

```
initialiser une pile contenant S;  
soit C le premier lexème;  
tant que la pile est non-vide faire  
|   soit T le sommet de la pile;  
|   si T est non-terminal alors  
|   |   choisir une production P;  
|   |   dépiler le sommet de la pile;  
|   |   empiler la partie droite de P;  
|   sinon  
|   |   si  $T = C$  alors  
|   |   |   dépiler le sommet de la pile;  
|   |   |   passer C au lexème suivant;  
|   |   fin  
|   fin  
fin
```

### Algorithme 1 : Algorithme LL à pile

Pile	Entrée	Opération
$S$	aabb	
$aSb$	aabb	dériver $S \rightarrow aSb$
$Sb$	abb	valider $a$
$aSbb$	abb	dériver $S \rightarrow aSb$
$Sbb$	bb	valider $a$
$bb$	bb	dériver $S \rightarrow \epsilon$
$b$	b	valider $b$
		valider $b$

Comment choisir la production à dériver ?

Utilisation d'un symbole de prévision : analyseur LL(1).

Construction d'une table associant non-terminal, lexème à une production.

Limitation : seul les grammaires où toutes les productions d'un non-terminal n'ont pas les mêmes terminaux préfixes.

$$A \rightarrow a\gamma$$

$$A \rightarrow b\gamma$$



# Sommaire

- 1 Grammaires non-contextuelles
- 2 Hiérarchie de Chomsky
- 3 Analyseur lexical
- 4 Analyseur syntaxique
- 5 Arbre abstrait**
  - Arbre de dérivation
  - Grammaire attribuée
- 6 Implémentation













# Sommaire

- 1 Grammaires non-contextuelles
- 2 Hiérarchie de Chomsky
- 3 Analyseur lexical
- 4 Analyseur syntaxique
- 5 Arbre abstrait
- 6 Implémentation**
  - Usage

