

*Université A.Mira, Bejaia*

*Faculté des Sciences Exactes*

*Département d'Informatique*

**Cours de Compilation**

# **Analyse Syntaxique**

*Mme D.Boulahrouz*  
*boukreda@hotmail.com*

# 1. Le rôle d'un Analyseur Syntaxique

## Définition

Un *analyseur syntaxique* reçoit une *suite d'unités lexicales* de l'analyseur lexical et *vérifie* que cette suite est engendrée par la *grammaire du langage*.

Le *but* d'un analyseur syntaxique est de *trouver la relation* entre les *unités lexicales* pour produire un *arbre syntaxique*.





## Le rôle d'un Analyseur Syntaxique est donc:

*En se basant sur une grammaire  $G$  du langage, déterminer Si pour une suite d'unités lexicales en entrée il existe ou non un **arbre de dérivation**★*

*Les grammaires les plus adaptées à l'analyse syntaxique sont les grammaires de type 2★ (non contextuelle).*

## 2. Grammaires non contextuelles

(à contexte libre -type2-)

### Définition

Une grammaire à contexte libre  $G = (T, N, S, P)$  consiste en:

1.  $T$ , un ensemble de *terminaux* (tokens).
2.  $N$ , un ensemble de *non-terminaux* (variables syntaxiques générées par productions).
3.  $S$ , désigne le non-terminal de départ (*axiome*).
4.  $P$ , un ensemble de *productions*. Chaque production est notée  $A \rightarrow \alpha$ , où  $A \in N$ ,  $\alpha \in (T \cup N)^*$



## 2. Grammaires non contextuelles

(à contexte libre -type2-)

Qu'est ce qu'un arbre de dérivation (syntaxique)?

*Un **arbre de dérivation** est une représentation graphique d'une séquence de dérivation de formes séquentielles.*

*La **racine** de l'arbre de dérivation est l'axiome, les **nœuds** représentent les **symboles non terminaux de la grammaire**, les **feuilles** représentent les **terminaux de la grammaire** (Tokens) et les **arcs** représentent les **étapes des dérivations**.*



## 2. Grammaires non contextuelles

(à contexte libre -type2-)



### Exemple

Soit la grammaire suivante:

$$E \rightarrow E + E \mid E * E \mid (E) \mid -E \mid id$$

*Est ce que la chaîne -(id + id) appartient à ce langage?*

*Oui, grâce à la dérivation suivante:*

$$E \rightarrow -E \rightarrow -(E) \rightarrow -(E + E) \rightarrow -(id + id)$$

*Où  $\rightarrow$  dénote une "dérivation directe".*



# 2. Grammaires non contextuelles

(à contexte libre -type2-)

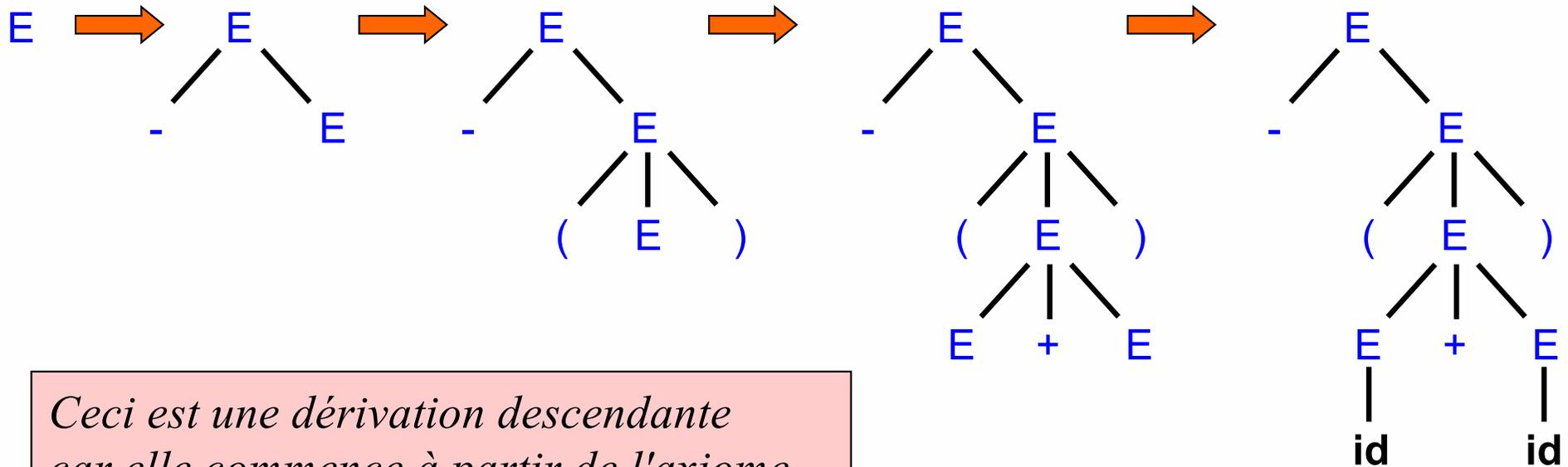


## Exemple2

$E \rightarrow E + E \mid E * E \mid (E) \mid -E \mid id$

*Examinons cette dérivation:*

$E \rightarrow -E \rightarrow -(E) \rightarrow -(E + E) \rightarrow -(id + id)$



*Ceci est une dérivation descendante car elle commence à partir de l'axiome*

*arbre de dérivation*



# 2. Grammaires non contextuelles

(à contexte libre -type2-)



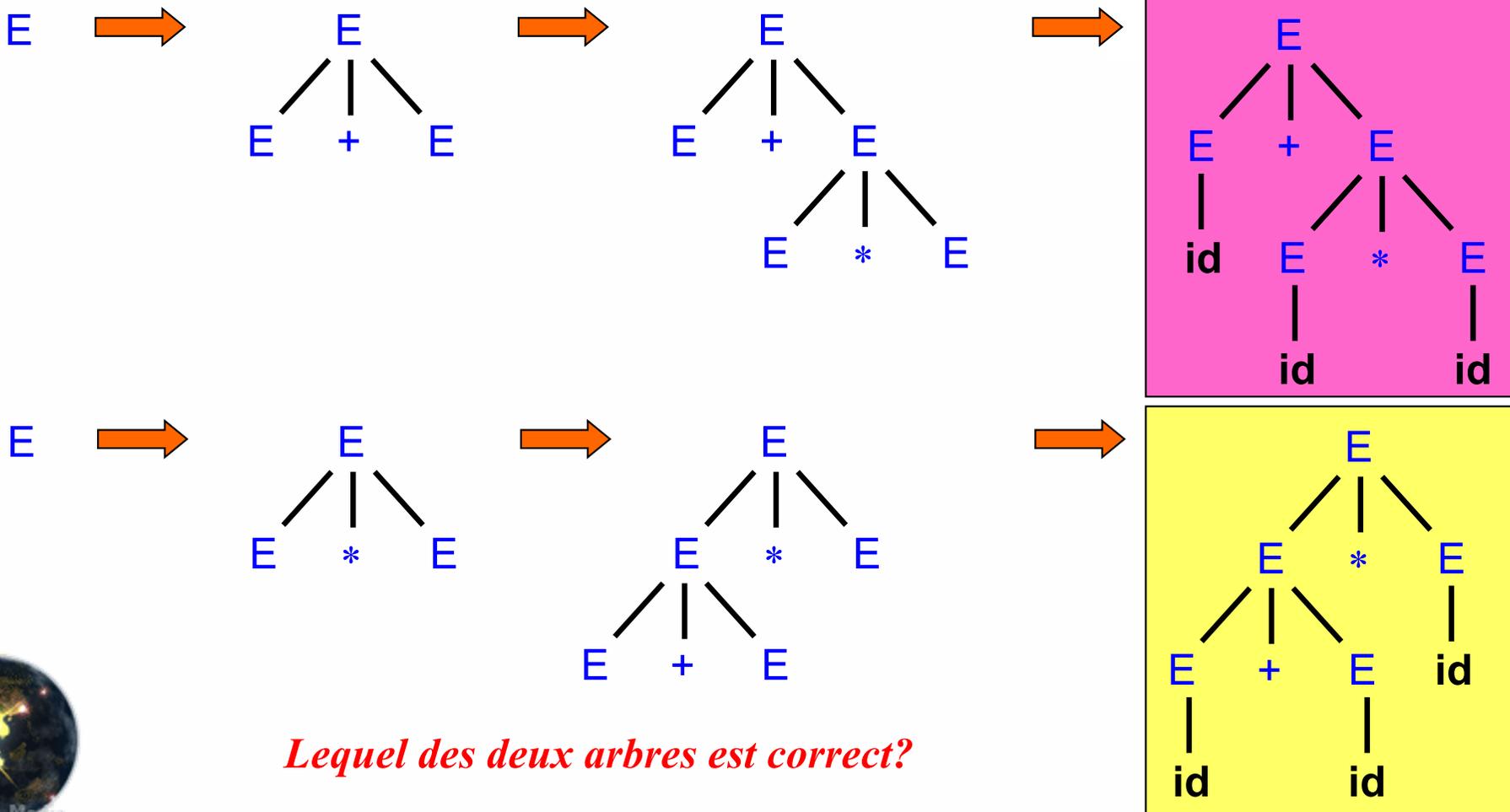
## Exemple 3

$E \rightarrow E + E \mid E * E \mid (E) \mid -E \mid id$



Trouver une dérivation pour l'expression:

**id + id \* id**



*Lequel des deux arbres est correct?*



# 2. Grammaires non contextuelles

(à contexte libre -type2-)



## Exemple3

$E \rightarrow E + E \mid E * E \mid (E) \mid - E \mid id$

Trouver une dérivation pour l'expression:

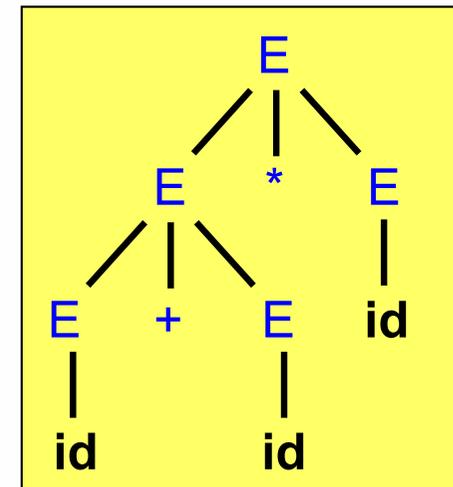
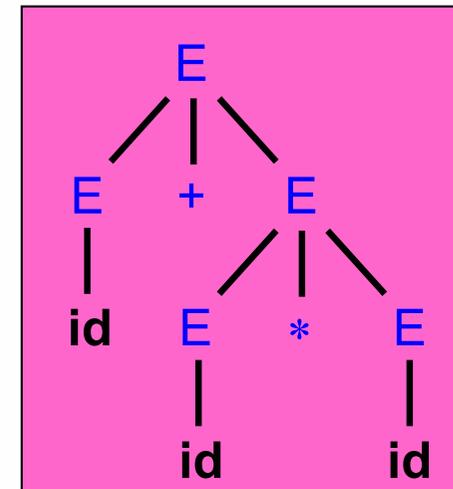
**id + id \* id**

*D'après la grammaire, les deux arbres sont corrects.*



PresenterMedia

Une grammaire produisant plus d'un arbre de dérivation pour une phrase d'entrée est appelée grammaire **ambiguë**.



PresenterMedia

## 2. Grammaires non contextuelles

(à contexte libre -type2-)



### Notion de Réduction

*C'est **l'inverse de la dérivation** , elle consiste à remplacer, une fois que l'on reconnaît un MDP par la partie gauche (MGP), ainsi jusqu'à arriver à l'axiome .*

➤ *La **réduction** est une description de la **construction ascendante** de l'arbre syntaxique. Elle est utilisée par les **analyseurs ascendants***



# 4. Méthodes d'analyse syntaxique



Analyseurs  
Syntaxiques



*On distingue 02 types  
généraux d'analyseurs  
syntaxiques: **Descendants**  
et **Ascendants***



## Analyse Descendante

méthodes **descendantes**  
qui construisent des  
arbres d'analyse en  
partant de la **racine**  
(axiome) vers les  
feuilles (pgm source) →  
**par dérivation (Top-Down)**



## Analyse Ascendante

méthodes **ascendantes**  
qui construisent des  
arbres d'analyse en  
partant **des feuilles** vers  
la **racine** → **par**  
**réduction (Bottom-up)**



Deux types principaux d'analyseurs  
syntaxique

# 4. Méthodes d'analyse syntaxique

 Analyseurs  
Syntaxiques



*Parmi ces 2 familles  
d'analyseurs, on distingue  
les analyseurs **déterministes**  
et **non déterministes***



# 4. Méthodes d'analyse syntaxique



## Analyseurs Syntaxiques

- **Analyseurs Déterministes:** *si une seule règle de production est applicable dans chaque étape de configuration de l'analyseur.*
  - *En se basant uniquement sur le premier token, l'analyseur sait déterminer quelle règle utiliser pour la dérivation ( LL, LR, ...)*
- **Analyseurs non Déterministes:** *Si la grammaire est ambiguë, l'analyse déterministe est totalement impossible. Dans ce cas, il faut des méthodes qui explorent les différentes possibilités.*
  - *Les analyseurs basés sur les méthodes prédictives avec retour arrière et les méthodes descendantes parallèles sont des exemples de ce type d'analyseurs*



# Méthodes d'analyse syntaxique

Méthodes  
d'analyse  
syntaxique

Déterministes

Descendantes  
LL(1), LL(k)

Ascendantes  
LR(1), SLR(1),  
LALR(1), LR(k)

Descendantes  
Non  
Déterministes

Descendante  
parallèle

prédicative  
avec retour  
arrière

