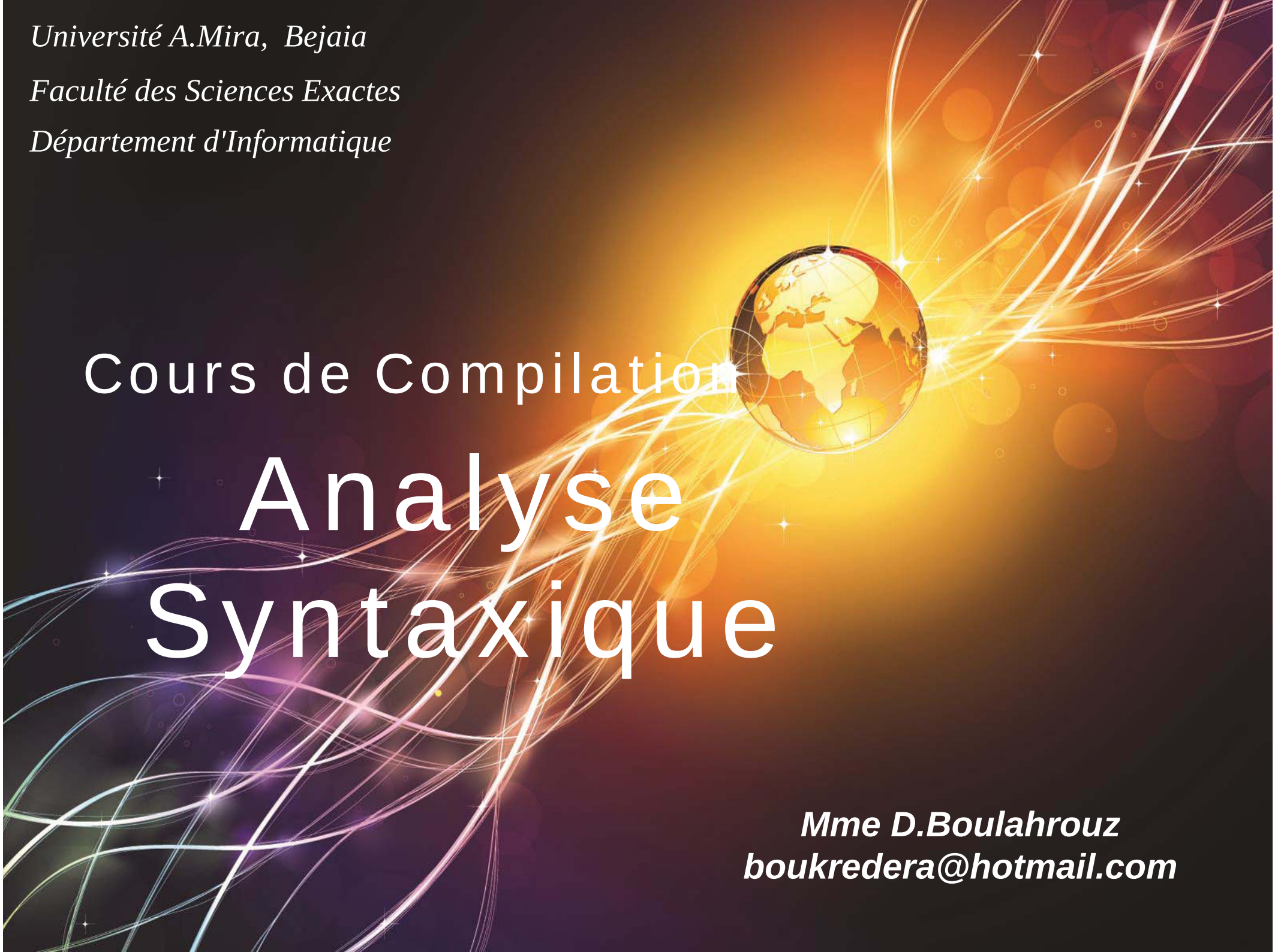
Université A.Mira, Bejaia
Faculté des Sciences Exactes
Département d'Informatique



Cours de Compilation

Analyse Syntaxique

Mme D.Boulahrouz
boukredera@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* $A \in N$, $\alpha \in (T \cup N)$

4. Méthodes d'analyse syntaxique

Analyseurs Syntaxiques

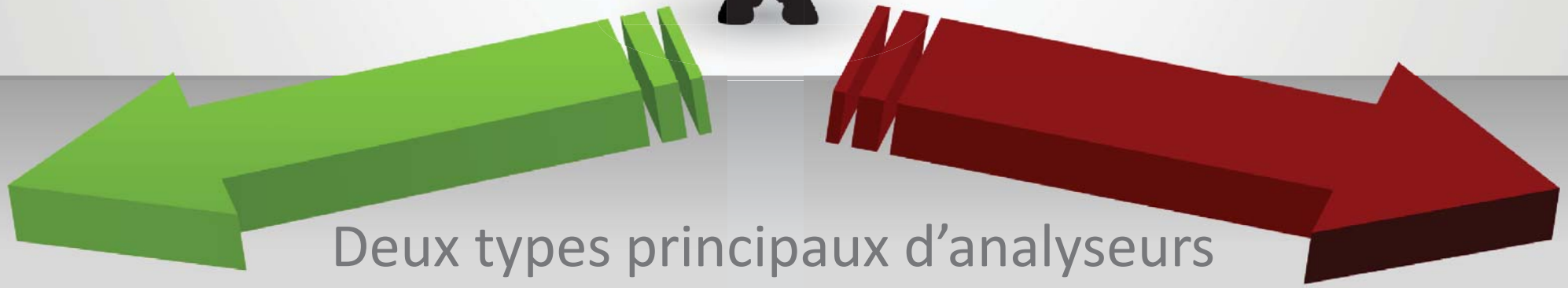
On distingue 02 types généraux d'analyseurs syntaxiques: Descendants et Ascendants

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)**

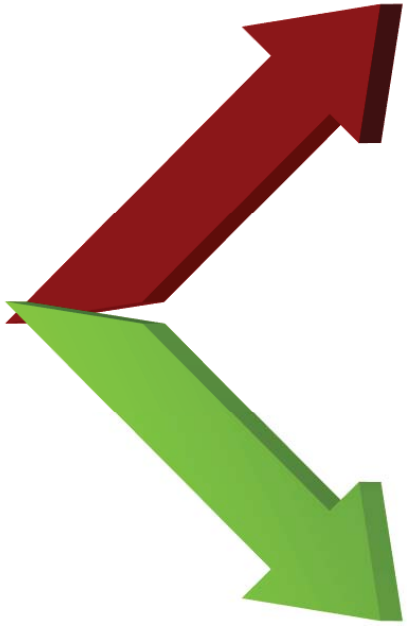
Analyse Ascendante

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



Deux types principaux d'analyseurs syntaxique

Analyseurs Syntaxiques



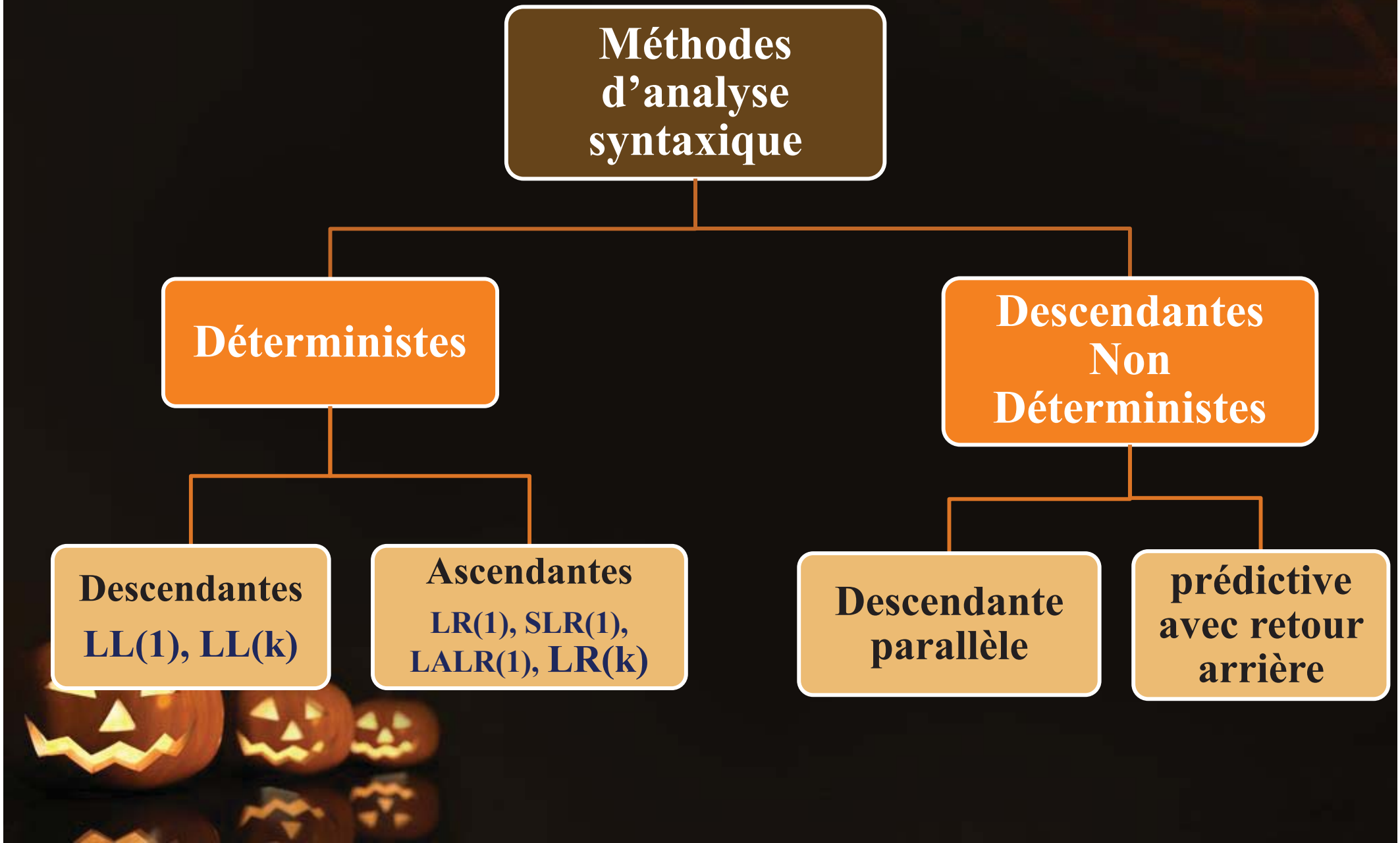
Parmi ces 2 familles d'analyseurs, on distingue les analyseurs déterministes et non déterministes

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 ambigüe,*** *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



Analyseurs Syntaxiques Descendants

Conditions d'analyse syntaxique descendante

Afin de réaliser l'analyse syntaxique ***descendante****, il faut que la grammaire soit* ***de type 2****,* ***factorisée****, et* ***non récursive à gauche****. La dernière condition permet de limiter les retours arrière et les boucles.*

Récursivité gauche

Une grammaire ayant au moins une production de la forme $A \Rightarrow A\alpha$ *est appelée grammaire* ***récursive gauche***.

Les ***analyses descendantes*** *ne fonctionnent pas avec des grammaires récursives gauches.*

La récursivité gauche *peut être éliminée en réécrivant la grammaire.*

Récursivité gauche Directe ou indirecte

- *Une grammaire est dite **récursive à gauche**, de façon **directe**, si elle s'écrit de la manière suivante :*

$$A \rightarrow A\alpha / \beta \quad \text{Avec : } A \in N \text{ et } \alpha, \beta \in (T \cup N)^*$$

- *Et elle est **récursive gauche indirectement** si elle s'écrit comme suit :*

$$A \rightarrow B\alpha_1 / \beta_1$$
$$B \rightarrow A\alpha_2 / \beta_2 \quad \text{Avec : } A \in N \text{ et } \alpha_1, \beta_1, \alpha_2, \beta_2 \in (T \cup N)^*$$

Elimination de la récursivité gauche directe et indirecte

5.1 Elimination de la récursivité gauche

*Cas de **récursivité gauche directe** :*

Soit la grammaire: $A \longrightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots. \mid A\alpha_m \mid \beta_1 \mid \ldots\ldots. \mid \beta_n$

Pour supprimer la récursivité gauche, nous réécrivons cette grammaire sous la forme:

$$A \longrightarrow \beta_1 A' \mid \beta_2 A' \mid \ldots\ldots. \mid \beta_n A'$$
$$A' \longrightarrow \alpha_1 A' \mid \alpha_2 A' \mid \ldots\ldots. \mid \alpha_m A' \mid \varepsilon$$

*avec les **βi** ne commençant pas par **A**, et les **αi** ne commençant pas par **A'** et ne produisent pas la chaîne vide.*

Exemple

soit la grammaire récursive à gauche:

$$E \longrightarrow E + T \mid T$$
$$T \longrightarrow T * F \mid F$$
$$F \longrightarrow id \mid (E)$$

$$\alpha 1 = + T,\ \beta = T$$
$$\alpha 1 = * F,\ \beta = F$$

La grammaire, non récursive à gauche, équivalente est:

$$E \longrightarrow T\,E'$$
$$E' \longrightarrow + T\,E' \mid \varepsilon$$
$$T \longrightarrow F\,T'$$
$$T' \longrightarrow * F\,T' \mid \varepsilon$$
$$F \longrightarrow id \mid (E)$$

5.1 Elimination de la récursivité gauche

*Cas de **récursivité gauche indirecte** :*

*Pour la transformation d'une grammaire **récursive gauche indirecte** , on utilise «**la substitution**» pour faire apparaitre la récursivité directe qu'on éliminera comme vu précédemment:*

G1:
- S → Aa | b
- A → cA| Sd |c

S ⇒ Aa ⇒ **S**da

G2:
- S → Ba | baA
- A → ccA| Sb
- B → Ad | ε

S ⇒ Ba ⇒ Ada ⇒ **S**bda

G1 et G2 sont récursives à gauche indirectement

Exemple1 récursivité gauche indirecte

soit la grammaire récursive à gauche:

G1:
- S → Aa | b
- A → cA| Sd |c

1. *Faire apparaitre la récursivité gauche directe en remplaçant A dans S:*

- S → cAa | **S**da | ca | b
- A → cA | Sd | c

2. *Eliminer la récursivité gauche directe dans S:*

- S → cAaS' | caS' | bS'
- S' → daS' | ε
- A → cA | Sd | c

La grammaire, non récursive à gauche, équivalente à G1.

Factorisation d'une grammaire

C'est une *des* ***conditions*** *permettant de faire une analyse syntaxique* ***descendante déterministe***

5.2 Factorisation gauche

*Si **les parties droites** de plusieurs productions commencent par le **même préfixe**, lors de l'analyse il n'est pas évident de choisir la 'bonne' production.*

Soit les S-productions

$$S \longrightarrow A\,\beta1 \mid A\,\beta2 \mid \ldots\ldots \mid A\,\beta n$$

avec A le préfixe commun.

En factorisant à gauche, on a:

$$S \longrightarrow A\,A'$$

$$A' \longrightarrow \beta1 \mid \beta2 \mid \ldots\ldots \mid \beta n$$

5. Rappels sur les grammaires

Exemple

$$G:\begin{cases} S \rightarrow aS/bA \\ A \rightarrow aA/ab/c \end{cases}$$

G n'est pas factorisée.

$$G':\begin{cases} S \rightarrow aS/bA \\ A \rightarrow aB/c \\ B \rightarrow A/b \end{cases}$$

G' grammaire factorisée équivalente à G