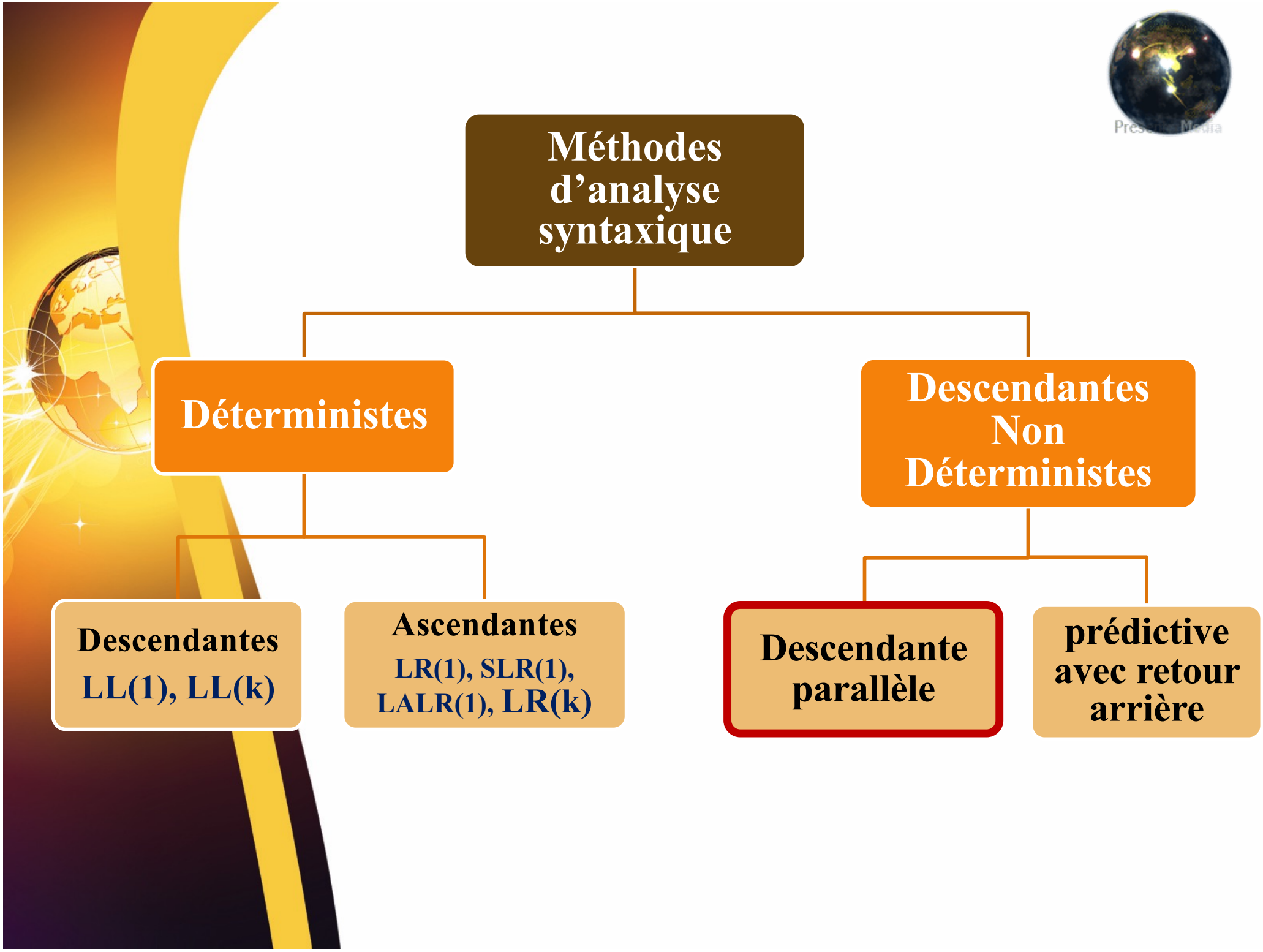
Les méthodes d'analyse syntaxique descendante non déterministe


Méthodes d'analyse syntaxique
Déterministes
Descendantes Non Déterministes
Descendantes
LL(1), LL(k)
Ascendantes
LR(1), SLR(1), LALR(1), LR(k)
Descendante parallèle
prédictive avec retour arrière

6. Analyse descendante non déterministe

6.1 Analyse descendante parallèle

- *C'est une **analyse descendante** dans laquelle il y a construction d'un ensemble d'arbre de dérivation **en même temps**.*
- *Les **arbres** qui ne peuvent pas correspondre à la chaine d'entrée sont **abandonnés.***
- ***A chaque dérivation,** un non-terminal en même temps par **tous** ses **MDP** .*

6. Analyse descendante non déterministe

Exemple (analyse parallèle)

Soit la grammaire G suivante:

$$\begin{cases} <Bloc> \rightarrow début <LD> ; <LI> fin \\ <LD> \rightarrow d \mid d ; <LD> \\ <LI> \rightarrow i \mid i ; <LI> \end{cases}$$

Analyser la chaine : ***début d ; d ; i ; i fin#***

Pas	Cible	Chaine	Action
01	<Bloc>	Début d;d;i;i fin#	Remplacer <Bloc> par ses MDP
02	Début <LD>;<LI> fin	Début d;d;i;i fin#	Égalité : avancer .
03	<LD>;<LI> fin	d;d;i;i fin#	Remplacer <LD> par ses MDP
04	d;<LI> fin d;<LD> ; <LI> fin	d;d;i;i fin# d;d;i;i fin#	Égalité : avancer Égalité : avancer
05	;<LI> fin ;<LD> ; <LI> fin	;d;i;i fin# ;d;i;i fin#	Égalité : avancer Égalité : avancer
06	<LI> fin <LD> ; <LI> fin	d;i;i fin# d;i;i fin#	Remplacer <LI> par ses MDP Remplacer <LD> par ses MDP
07	i fin i ; <LI> fin	d;i;i fin# d;i;i fin#	Blocage Blocage
	d;<LI> fin d;<LD> ; <LI> fin	d;i;i fin# d;i;i fin#	Égalité : avancer Égalité : avancer
08	;<LI> fin ;<LD> ; <LI> fin	;i;i fin# ;i;i fin#	Égalité : avancer Égalité : avancer
09	<LI> fin <LD> ; <LI> fin	i;i fin# i;i fin#	Remplacer <LI> par ses MDP Remplacer <LD> par ses MDP

Pas	Cible	Chaine	Action
10	i fin i ; <LI> fin	i;i fin# i;i fin#	Égalité : avancer Égalité : avancer
	d;<LI> fin d;<LD> ; <LI> fin	i;i fin# i;i fin#	Blocage Blocage
11	fin ; <LI> fin	;i fin# ;i fin#	Blocage égalité : avancer
12	<LI> fin	i fin #	Remplacer <LI>par ses MDP
13	i fin i ; <LI> fin	i fin # i fin #	Égalité : avancer Égalité : avancer
14	fin ; <LI> fin	fin # fin #	Égalité : avancer Blocage
15	Vide	#	**Chaine ∈ langage**

6. Analyse descendante non déterministe

Algorithme
(analyse parallèle)

```
Début
    Tant que Cible ≠vide
          Faire
             Si tête.cible = non-terminal
                      Alors Remplacer non-terminal par ses MDP
                      Sinon Si tête.cible =TC
                                  Alors Avancer dans la chaine et la cible
                                  Sinon Blocage
                            Fsi;
             Fsi;
          Fait;
    Si (cible = vide) et (TC = #)
             Alors chaine ∈ langage
             Sinon chaine ∉  langage
    Fsi;
Fin.
```

6. Analyse descendante non déterministe

Algorithme
(analyse parallèle)

Début

Tant que Cible ≠vide

Remarque :

1. *Cette technique n'est **pas efficace** car elle nécessite l'examen de plusieurs arbres en parallèle, ce qui est **couteux en temps et en espace mémoire**.*

*2. Pour appliquer cette méthode, il faut que la grammaire soit **non récursive gauche**.*

et la cible

Fsi;

Fsi;

Fait;

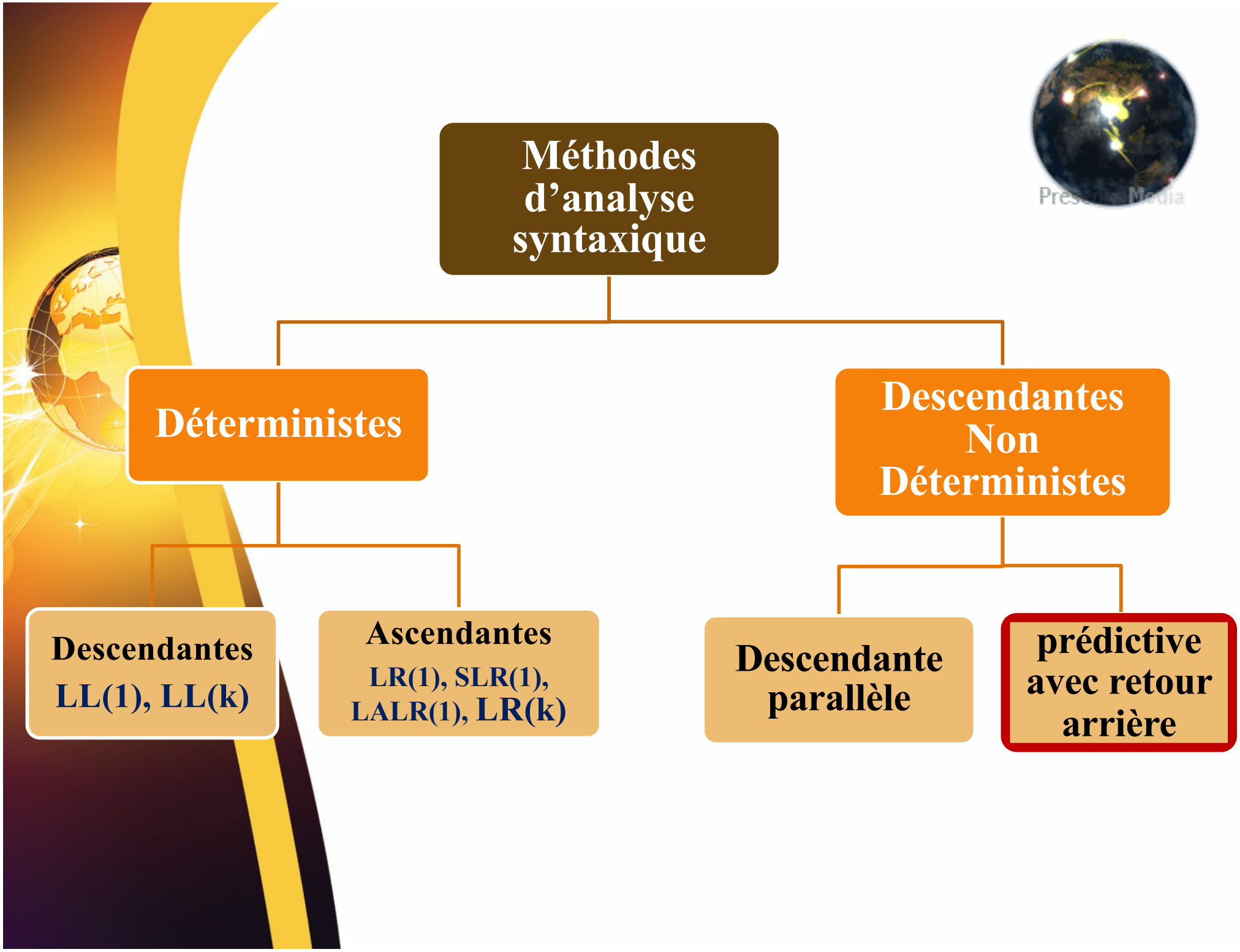
Si (cible = vide) et (TC = #)

Alors chaine ∈ langage

Sinon chaine ∉ langage

Fsi;

Fin.


Méthodes d'analyse syntaxique
Déterministes
Descendantes
LL(1), LL(k)
Ascendantes
LR(1), SLR(1), LALR(1), LR(k)
Descendantes Non Déterministes
Descendante parallèle
prédictive avec retour arrière

6. Analyse descendante non déterministe

6.2 Analyse prédictive avec retour arrière

*L'**analyseur** cherche partant de **l'axiome** à atteindre la chaine en effectuant une série de dérivation.*

***Si** l'analyseur parvient à construire un tel arbre, on conclut que la chaine appartient au langage , **sinon** l'analyseur effectue des retours arrières pour essayer une autre possibilité de dérivation.*

6. Analyse descendante non déterministe

Exemple (prédictive avec retour arrière)

Soit la grammaire G suivante:

$$G:\begin{cases} S \rightarrow ca \mid b \mid cSA \\ A \rightarrow c \mid b \end{cases}$$

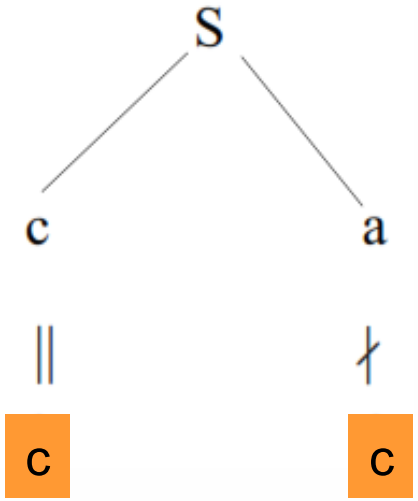
Analyser la chaine : ***ccab***

6. Analyse descendante non déterministe

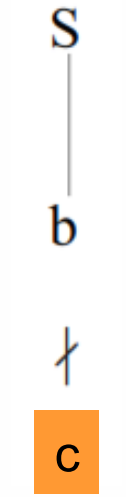
Exemple (prédictive avec retour arrière)

$$G:\begin{cases} S \rightarrow ca \mid b \mid cSA \\ A \rightarrow c \mid b \end{cases}$$

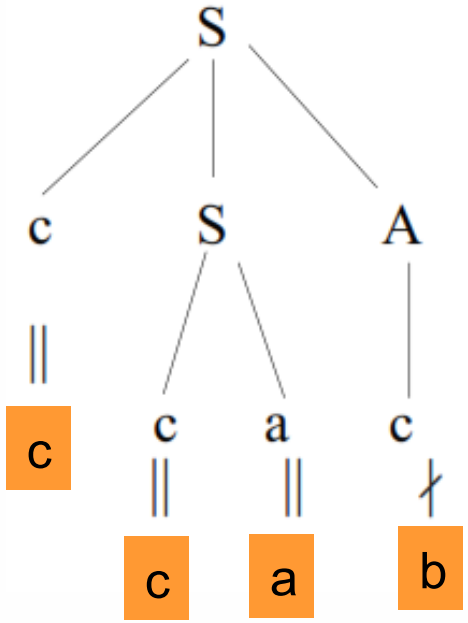
ccab



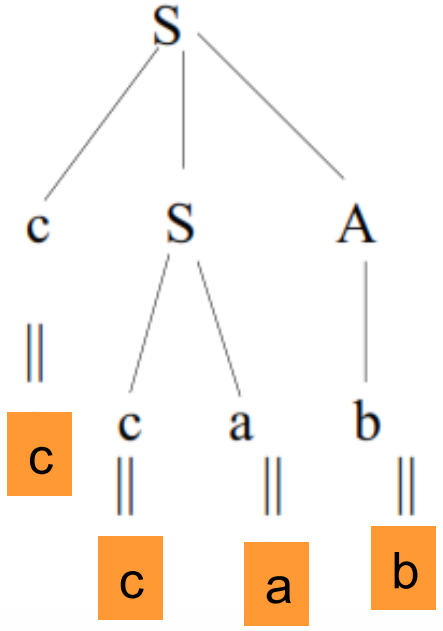
ECHEC: *effectuer un* ***retour arrière***



ECHEC: *effectuer un* ***retour arrière***



ECHEC: *effectuer un* ***retour arrière***



La chaine **ccab** appartient au langage

6. Analyse descendante non déterministe

Algorithme
(prédictive avec retour arrière)

```
Début
    Pour un non-terminal donné:  essayer les alternatives dans l'ordre
        Si TC = terminal du MDP
            Alors Avancer
            Sinon Pour le même terminal : passer à l'alternative suivante
                Si aucune alternative ne marche
                    Alors Revenir au sous arbre précédent
                Fsi;
        Fsi;
        Si aucune alternative ne marche et le non-terminal de retour est l'axiome
            Alors chaine n'appartient pas au langage
        Fsi;
Fin.
```

6. Analyse descendante non déterministe

Algorithme
(prédictive avec retour arrière)

Début

Pour un non-terminal donné: essayer les alternatives dans l'ordre

S

uivante

F

S est l'axiome

Alors chaîne n'appartient pas au langage

Fsi;

Fin.

Remarque :

*1. Les **dérivations** se font de **gauche à droite** et de **haut en bas**, les retours arrières se font dans le sens inverse.*

*2. Il faut que la **grammaire** soit **non récursive gauche.***

Chapitre 03

Analyse Syntaxique

7. *Les méthodes d'analyse descendante déterministe :*

*7.1 Analyse par les **procédures récursives** **(descente récursive)***

*7.2 **Grammaire LL(1)***

7.3 L'analyseur LL(1)

*7.4 **Grammaires LL(k)***

Analyse par les procédures récursives

7. Analyse descendante déterministe

7.3 Analyse par les procédures récursives (descente récursive) :

- *Elle consiste à associer à chaque non-terminal une* ***procédure*** *qui traite tous les MDP du non-terminal . L'analyse se fait par* ***appel aux différentes procédures****.*

- *L'analyse est déterministe, soit la chaine à analyser est correcte ou erronée* ***(pas de retour arrière)****. De plus, l'analyse est efficace et puissante.*

7. Analyse descendante déterministe

Exemple (Descente récursive)

$$G:\begin{cases} Z \rightarrow S\# \\ S \rightarrow aB \mid Aa \\ A \rightarrow BA \mid b \\ B \rightarrow c \end{cases}$$

*Augmenter la grammaire par un nouvel axiome **Z** tel que* ***Z→S#***

Procédure Z( )
Début
S( );
Si *Tc = #* ***alors*** *« Chaine Acceptée »*
Sinon *Erreur (« Chaine rejetée »);*
Fsi
Fin

on suppose que Erreur() est une fonction qui affiche un message et termine le programme.

7. Analyse descendante déterministe

Exemple (Descente récursive)

$$G:\begin{cases} Z \rightarrow S\# \\ S \rightarrow aB \mid Aa \\ A \rightarrow BA \mid b \\ B \rightarrow c \end{cases}$$

```
Procédure S( )
  Début
    Si Tc = 'a' alors  tc ← ts;
                       B( );
    Sinon  A( );
                 Si Tc = 'a' alors tc ← ts;
                 Sinon Erreur (« Chaine rejetée »);
                 Fsi
    Fsi
  Fin
```

7. Analyse descendante déterministe

Exemple (Descente récursive)

$$G:\begin{cases} Z \rightarrow S\# \\ S \rightarrow aB / Aa \\ A \rightarrow BA / b \\ B \rightarrow c \end{cases}$$

Procédure A( )

Début

Si Tc = 'b' alors *tc ← ts;*

Sinon ***B( );***

A( );

Fsi

Fin

7. Analyse descendante déterministe

Exemple (Descente récursive)

$$G:\begin{cases} Z \rightarrow S\# \\ S \rightarrow aB \mid Aa \\ A \rightarrow BA \mid b \\ B \rightarrow c \end{cases}$$

```
Procédure B( )
  Début
  |   |  Si Tc = 'c' alors  tc ← ts;
  |   |  Sinon Erreur (« Chaine rejetée »);
  |   |  Fsi
  |  Fin
```

Remarque: si on une règle: **A → BC | C**, dans ce cas la **procedure A( )** sera:

```
Procédure A( )
  Début
  |   |  Si Tc € Deb (BC) alors  B( );
  |   |                          C( );
  |   |  Sinon C( );
  |   |  Fsi
  |  Fin
```

7. Analyse descendante déterministe

Exemple (Descente récursive)

$$G:\begin{cases} Z \rightarrow S\# \\ S \rightarrow aB / Aa \\ A \rightarrow BA / b \\ B \rightarrow c \end{cases}$$

*Analyse de la chaine **cba#***

	Tc	Chaine	Action
Z	c	cba#	Appel S
ZS	c	cba#	Appel A
ZSA	c	cba#	Appel B
ZSAB	c	cba#	Fin de B
ZSA	b	ba#	Appel A
ZSAA	b	ba#	Fin A et Tc := Tc+1
ZSA	a	a#	Fin A et Tc := Tc+1
ZS	a	a#	Fin S et Tc:=Tc+1
Z	#	#	Chaine correcte

7. Analyse descendante déterministe

Remarques (descente récursive) :

1. *Si la grammaire est récursive gauche , il y a nécessité de la **transformer** .*
2. *Si on veut faire une analyse déterministe, il faut **factoriser** la grammaire.*
3. *La méthode présente un inconvénient, car elle est liée à la grammaire.*
4. *Méthode simple à implémenter et efficace.*