Analyse Syntaxique descendante

Analyse descendante

- Principe : l'analyse descendante part de l'axiome et tente de reconstituer l'arbre de dérivation par un parcours gauche-droite préfixé
  - elle essaye de dériver l'axiome par une suite de dérivations gauches
- Exemple 1 :

$$S \rightarrow aSbT \mid cT \mid d$$

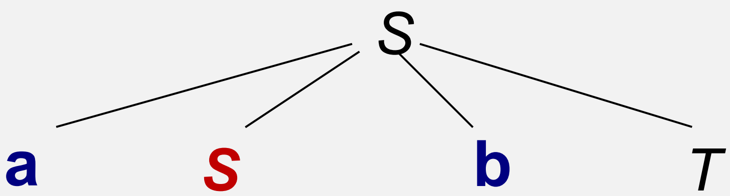
$$T \rightarrow aT \mid bS \mid c$$

avec le mot ***accbbadbc***

  - On part avec l'arbre contenant le seul sommet ***S***
  - La lecture de la première lettre du mot *(a)* nous permet d'avancer la construction

Analyse descendante
Exemple 1

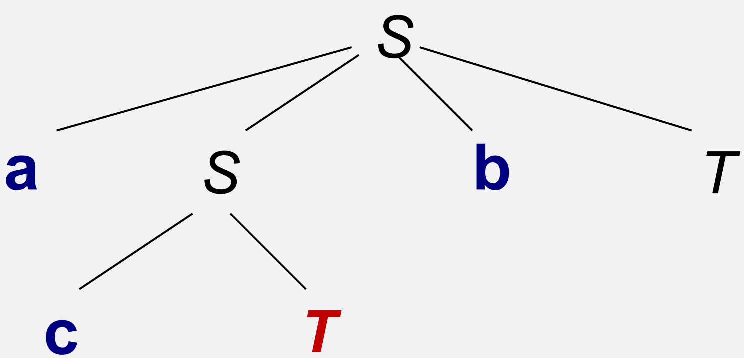
S* → *aSbT* | *cT* | *d
T* → *aT* | *bS* | *c
Avec le mot : *accbbadbc*



a ccbbadbc

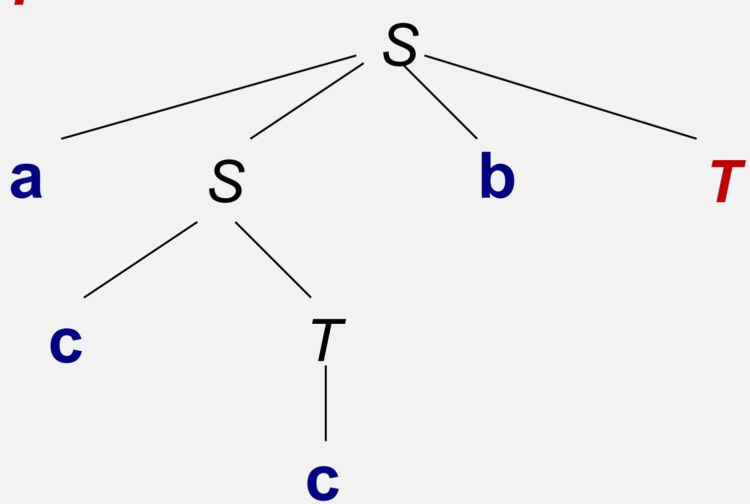
S* ⇒ *aSbT

- La lecture de la deuxième lettre du mot (***c***) nous amène à



ac cbbadbc

S* ⇒ *aSbT* ⇒ *acTbT



acc bbadbc

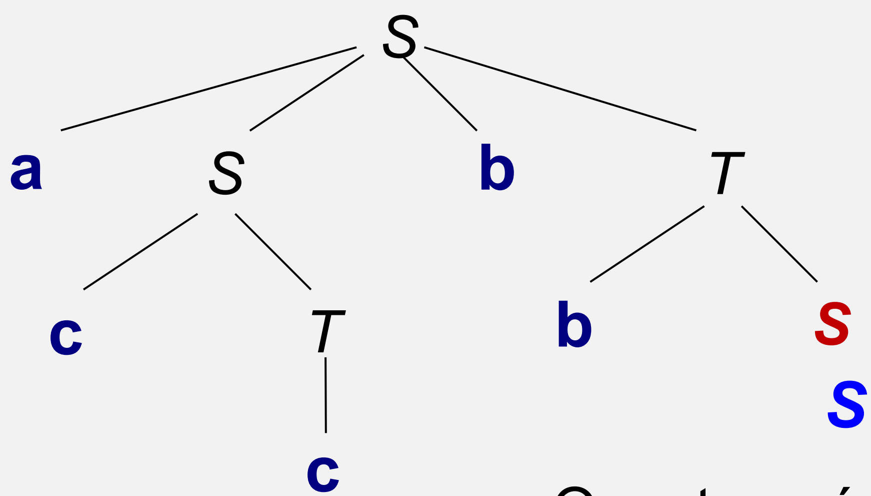
S* ⇒ *aSbT* ⇒ *accbT

Analyse descendante

Exemple 1

$$S \rightarrow aSbT \mid cT \mid d$$
$$T \rightarrow aT \mid bS \mid c$$

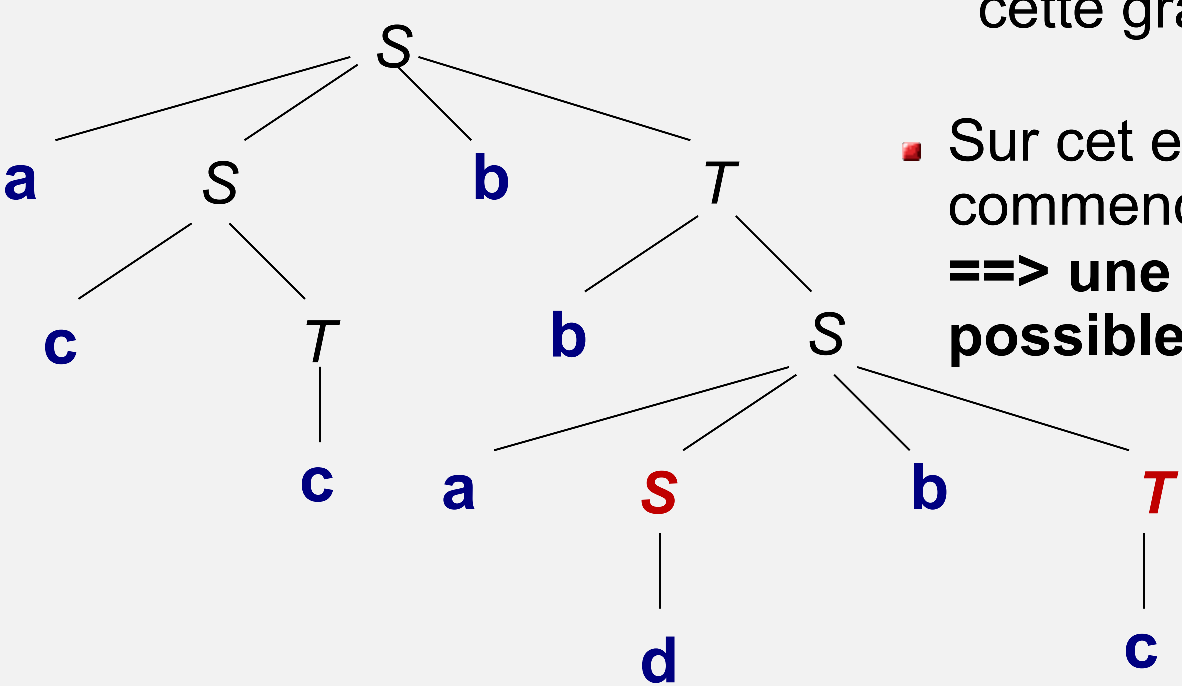
Avec le mot : ***accbbadbc***



accbb adbc

$$S \Rightarrow aSbT \Rightarrow accbbS$$

- On a trouvé un arbre de dérivation donc ce mot appartient au langage engendré par cette grammaire

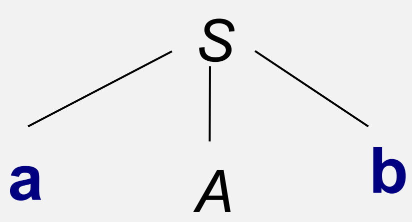


- Sur cet exemple chaque règle commence par **un terminal différent ==> une seule règle de production est possible**

Analyse descendante
Exemple 2

$S \rightarrow aAb$
$A \rightarrow cd \mid c$
Avec le mot : ***acb***

- La lecture de la première lettre du mot *(a)* nous amène à :



a cb

- Une seule production possible : $S \Rightarrow aAb$

- En lisant le *c*, on ne sait pas s'il faut prendre la règle $A \rightarrow cd$ ou $A \rightarrow c$
  - Pour le savoir il faut lire aussi la lettre suivante *(b)*
  - *Ou alors essayer les différentes possibilités*
    - *on essaye ,* $A \rightarrow cd$ *on aboutit à un échec,*
    - *on retourne en arrière et on essaye la deuxième règle* $A \rightarrow c$

Analyse descendante
Exemple 3

- Exemple 3 : grammaire des expressions arithmétiques :

$$E \rightarrow TE' \mid G$$

$$G \rightarrow -E$$

$$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid nb$$

Avec le mot **nb*nb +nb/nb**

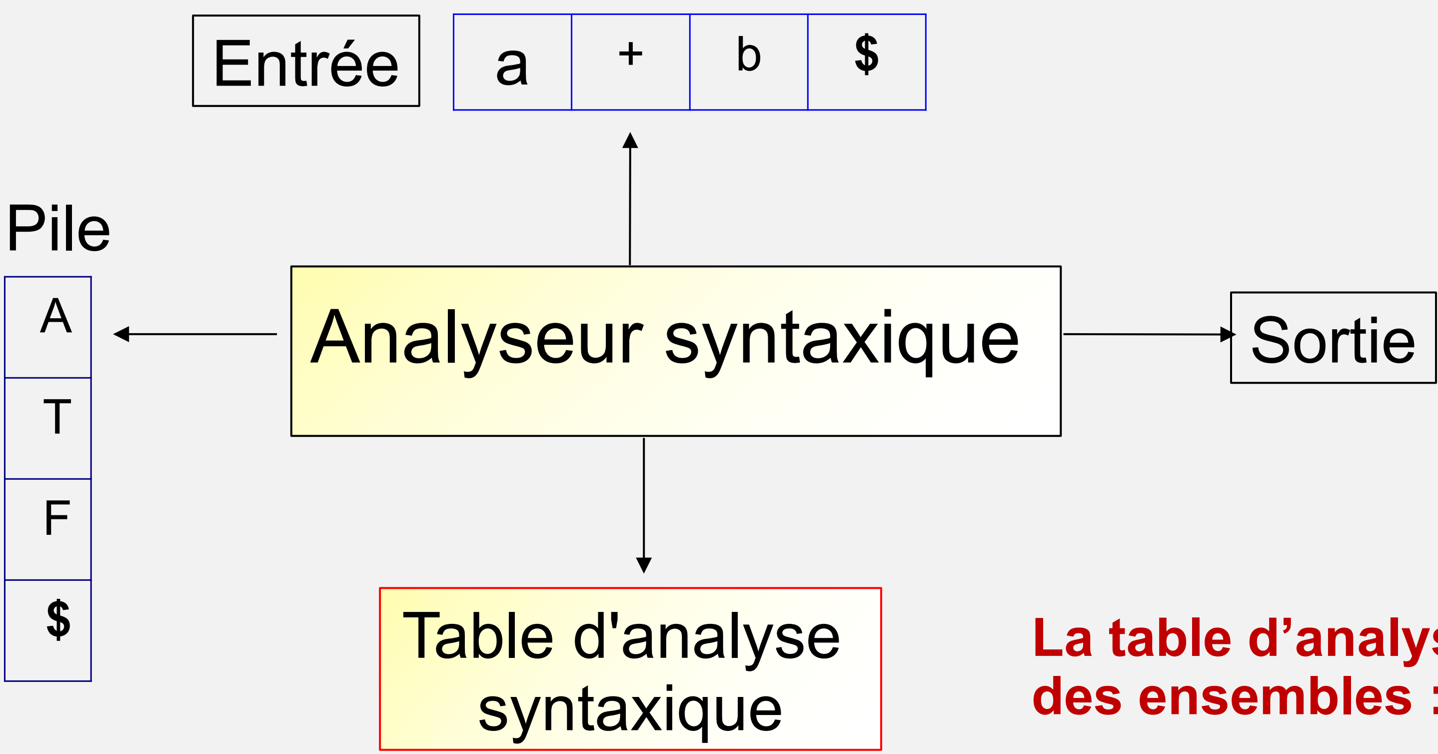
- Quelle règle de production choisir ?? !!

- Utiliser **une table d'analyse** :
  - Elle indique **quelle production utiliser** quand je lis **tel caractère** et que j'en suis à **dériver tel non terminal**
  - Pour construire une table d'analyse, on a besoin des ensembles **PREMIER** et **SUIVANT**

Analyse descendante

Analyse prédictive sans récursivité

- L’analyseur syntaxique prédictif comporte :
  - **un tampon d'entrée**: initialisé avec la chaîne d’entrée suivie du symbole $
  - **une pile**: initialisée avec le symbole de départ par-dessus le symbole $
  - **une table d'analyse (notée *M*)** *:* où ***M* [A, +]** indique : quelle production utiliser si le symbole (non terminal) sur le dessus de la pile est ***A*** et que le prochain symbole en entrée est **+**



La sortie de l’analyseur consiste en la suite de productions utilisées

La table d’analyse est définie à partir des ensembles : PREMIER et SUIVANT

Table d'analyse LL(1)

Exemple de la table d'analyse

Table d'analyse de la grammaire des expressions arithmétiques :

$E \rightarrow TE'$
$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$
$T \rightarrow FT'$
$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$
$F \rightarrow (E) \mid nb$

La table d'analyse obtenue est :

Symbole d'entrée

	nb	**+**	**-**	*****	**/**	**(**	**)**	**$**
E	$E \rightarrow TE'$					$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$	$E \rightarrow -TE'$				$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$					$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$	$T' \rightarrow /FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow nb$					$F \rightarrow (E)$		

La table d'analyse est définie à partir des ensembles : PREMIER et SUIVANT

Table d'analyse LL(1)
Calcul de PREMIER

Soit α un non terminal

- On définit l'ensemble PREMIER(α) comme étant l'ensemble des **terminaux** qui débutent les chaînes générées à partir de (α) plus ε si α peut générer ε

Formellement: $PREMIER(\alpha) = \{\mathbf{a} \mid \alpha \Rightarrow^* \mathbf{a}\omega\} \cup \{\varepsilon \mid \alpha \Rightarrow^* \varepsilon\}$

- *Exemple :* $S \rightarrow Ba$
  $B \rightarrow cP \mid bP \mid P \mid \varepsilon$
  $P \rightarrow dS$

$S \Rightarrow^* a$ donc $a \in PREMIER(S)$;

$S \Rightarrow^* \mathbf{cPa}$ donc $c \in PREMIER(S)$

$S \Rightarrow^* \mathbf{bPa}$ donc $b \in PREMIER(S)$;

$S \Rightarrow^* \mathbf{dSa}$ donc $d \in PREMIER(S)$

Il n'y a pas de dérivation $S \Rightarrow^* \varepsilon$

Donc $PREMIER(S)=\{a,b,c,d\}$

$B \Rightarrow^* \mathbf{dS}$ donc $d \in PREMIER(B)$

$aB \Rightarrow^* \mathbf{a\alpha}$ donc $PREMIER(aB)=\{a\}$

$BSb \Rightarrow^* \mathbf{ab}$ donc $a \in PREMIER(BS)$

Table d'analyse LL(1)
Calcul de PREMIER

On peut calculer les ensembles *PREMIER à* l'aide des **règles** suivantes :

- *PREMIER* $(\varepsilon) = \{\varepsilon\}$
- *PREMIER* $(a\alpha) = \{a\}$ */*a est un terminal*/*
- *Si* $X \rightarrow \alpha$ *est une production, PREMIER* $(\alpha) \subseteq$ *PREMIER(X)*
- *Si* $\varepsilon \notin$ *PREMIER(X)*

  alors *PREMIER*$(X\alpha)$ = *PREMIER(X)*

  sinon *PREMIER*$(X\alpha) = \{\text{PREMIER}(X) / \{\varepsilon\}\} \cup$ *PREMIER*(α)

Table d'analyse LL(1)
Calcul de PREMIER (exemples)

Exemple 1 : grammaire des expressions arithmétiques :

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$

$F \rightarrow (E) \mid nb$

- *PREMIER(E)=PREMIER(T)* **= {(, nb}** ; $\varepsilon \notin$ *PREMIER(T)*
- *PREMIER(E')=PREMIER(+TE')* $\cup$ *PREMIER(-TE')* $\cup$ *PREMIER(*ε*)=* **{+, -, ε}**
- *PREMIER(T)=PREMIER(F)* **= {(, nb}**
- *PREMIER(T')=* **{ *, /, ε}**
- *PREMIER(F) =* **{(, nb}**

Table d'analyse LL(1)
Calcul de PREMIER (exemples)

Exemple 2 :

$S \rightarrow ABCe$

$A \rightarrow aA \mid \varepsilon$

$B \rightarrow cB \mid bB \mid \varepsilon$

$C \rightarrow de \mid da \mid dA$

- $PREMIER(S)=\{a,b,c,d\}$
- $PREMIER(A)=\{a,\varepsilon\}$
- $PREMIER(B)=\{b,c,\varepsilon\}$
- $PREMIER(C)=\{d\}$

Exemple 3 :

$S \rightarrow c \mid ABS$

$A \rightarrow B \mid a$

$B \rightarrow b \mid \varepsilon$

- $PREMIER(S)=\{a,b,c\}$
- $PREMIER(A)=\{a,b,\varepsilon\}$
- $PREMIER(B)=\{b,\varepsilon\}$

Table d'analyse LL(1)
Calcul de SUIVANT

- *SUIVANT(A)* est l'ensemble de tous les terminaux qui peuvent apparaître immédiatement à droite de *A*

  Formellement : $SUIVANT(A) = \{a \mid S \Rightarrow^* \alpha A a \beta\}$..

- Exemple : $S \rightarrow Sc \mid \mathbf{B}a$

  $B \rightarrow B\mathbf{P}a \mid b\mathbf{P}b \mid \mathbf{P} \mid \varepsilon$

  $\mathbf{P} \rightarrow dS$

$\{a, b, c, d\} \subseteq SUIVANT(S)$ car il y a les dérivations :

- $S \Rightarrow Sc$
- $S \Rightarrow^* dSa \; (S \Rightarrow Ba \Rightarrow Pa \Rightarrow dSa)$
- $S \Rightarrow^* bdSba$
- $S \Rightarrow^* dSdSaa : (S \Rightarrow Ba \Rightarrow BPaa \Rightarrow PPaa \Rightarrow dSdSaa)$

Table d'analyse LL(1)
Calcul de SUIVANT (algorithme)

On peut calculer les ensembles *SUIVANT* de tous les non terminaux d'une grammaire à l'aide des **règles** suivantes :

I. Si ***S*** est le symbole de départ (l'axiome), alors ***\$*** $\in$ ***SUIVANT* (*S*)**

II. Si ***A*** → α ***B*** β est une production où ***B* est un non terminal** alors ajouter PREMIER (β) – $\{\varepsilon\}$ à *SUIVANT*(*B*) (***sauf*** ε)

III. Si ***A*** → α ***B*** est une production ou si ***A*** → α ***B***β est une production et que $\varepsilon \in$ ***PREMIER*(β)**, alors **ajouter *SUIVANT(A)* à *SUIVANT*(*B*)**

Table d'analyse LL(1)
Calcul de SUIVANT (Exemple 1)

Exemple 1 : grammaire des expressions arithmétiques :

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$

$F \rightarrow (E) \mid nb$

Règle I : On commence par ajouter *{$} à SUIVANT(E)*

Table d'analyse LL(1)

Calcul de SUIVANT (Exemple 1)

$E \rightarrow TE'$
$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$
$T \rightarrow FT'$
$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$
$F \rightarrow (E) \mid nb$

Exemple 1 : grammaire des expressions arithmétiques :

P(E)= {(, nb}
P(E')= {+, -, ε}
P(T)= {(, nb}
P(T')= { *, /, ε}
P(F)= {(, nb}

Règle II : Appliquons cette règle, symbole par symbole :

Symbole	Contraintes identifiées
$E \rightarrow \underline{T}E'$	PREMIER{E'}-{ε} ⊆ SUIVANT(T) ⇒ {+, -}⊆ *SUIVANT(T)*
$E' \rightarrow +\underline{T}E'$	PREMIER{E'}-{ε} ⊆ SUIVANT(T) ⇒ {+, -}⊆ *SUIVANT(T)*
….	
$F \rightarrow (\mathbf{E})$	*PREMIER{)}-{ε} ⊆ SUIVANT(E) ⇒ {)}⊆ SUIVANT(E)*

Règle III : Appliquons cette règle, symbole par symbole :

Symbole	justification	Contraintes
$E \rightarrow \underline{T}E'$ $E \rightarrow T\underline{E'}$	T est suivi de ε (situé à la fin)	SUIVANT{E} ⊆ SUIVANT(T) *SUIVANT{E} ⊆ SUIVANT(E')*
$E' \rightarrow +\underline{T}E'$ $E' \rightarrow +T\underline{E'}$ $E' \rightarrow -\underline{T}E'$ ...	suivi de ε situé à la fin suivi de ε	SUIVANT{E'} ⊆ SUIVANT(T) *SUIVANT{E'} ⊆ SUIVANT(E')* *SUIVANT{E'} ⊆ SUIVANT(T)* …
$F \rightarrow (\mathbf{E})$	–	–

Table d'analyse LL(1)
Calcul de SUIVANT (Exemple)

grammaire des expressions arithmétiques :

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$

$F \rightarrow (E) \mid nb$

On trouve enfin la solution suivante :

- *SUIVANT(E)={$,) }*
- *SUIVANT(E')={$,) }*
- *SUIVANT(T)={+, -, $,) }*
- *SUIVANT(T')= { +, -, $,) }*
- *SUIVANT(F) = {*, /, +, -, $,) }*

Table d'analyse LL(1)

Calcul de PREMIER et de SUIVANT

- Exemple 2 :
  - $S \rightarrow aSb \mid cd \mid Ae$
  - $A \rightarrow aAdB \mid \varepsilon$
    $B \rightarrow bb$

- *PREMIER(S)={a, c, e}*
- *PREMIER(A)={a,ε}*
- *PREMIER(B)={b}*

- *SUIVANT(S)={\$, b}*
- *SUIVANT(A)={e, d}*
- *SUIVANT(B)={e, d}*

I. Si *S* est le symbole de départ (l'axiome), alors $\$ \in$ *SUIVANT* (*S*)
II. Si $A \rightarrow \alpha B \beta$ est une production où **B est un non terminal** alors ajouter PREMIER (β) − {ε} à *SUIVANT*(*B*) (***sauf*** ε)
III. Si $A \rightarrow \alpha B$ est une production ou si $A \rightarrow \alpha B\beta$ est une production et que $\varepsilon \in$ ***PREMIER***(β), alors **ajouter *SUIVANT(A)* à *SUIVANT*(*B*)**

Table d'analyse LL(1)
Construction de la table d'analyse (algorithme)

- Une table d'analyse est un tableau à deux dimensions qui indique quelle production utiliser pour chaque non terminal *A* et chaque terminal *a* (tel que *a* est le prochain symbole en entrée)

Entrée : Une grammaire *G*

Sortie: Une table d'analyse *M*

Méthode :

1. Pour chaque production $A \rightarrow \alpha$, faire :

   (a) Pour tout ***a*** $\in$ ***PREMIER(***α**)** (a $\neq \varepsilon$), rajouter ***A*** $\rightarrow \alpha$ *dans la case* ***M*[*A*, *a*];**

   (b) Si $\varepsilon \in$***PREMIER(***α**)**, ajouter ***A***$\rightarrow \alpha$ **à** ***M*[*A*, *b*]** pour chaque

   b$\in$***SUIVANT*(*A*)** où *b* est un terminal ou $.

2. Chaque case de ***M[A,a] vide*** doit être initialisée à un signalement **d'erreur**

Table d'analyse LL(1)

Construction de la table d'analyse (exemple 1)

Table d'analyse de la grammaire des expressions arithmétiques :

$E \rightarrow TE'$
$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$
$T \rightarrow FT'$
$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$
$F \rightarrow (E) \mid nb$

1. Pour chaque production $A \rightarrow \alpha$, faire :
   - (a) Pour tout ***a*** $\in$ ***PREMIER(α)*** (a ≠ ε), rajouter ***A*** $\rightarrow$*α dans la case* ***M[A, a]***;
   - (b) Si ε ∈***PREMIER(α)***, ajouter ***A***→*α* **à *M*[*A, b*]** pour chaque ***b***∈***SUIVANT(A)*** où *b* est un terminal ou $.
2. Chaque case de ***M[A,a] vide*** doit être initialisée à un signalement **d'erreur**

- *PREMIER(E)={(, nb}*
- *PREMIER(E')={+, -, ε}*
- *PREMIER(T)={(, nb}*
- *PREMIER(T')= { *, /, ε}*
- *PREMIER(F) = {(, nb}*

- *SUIVANT(E)={$,) }*
- *SUIVANT(E')={$,) }*
- *SUIVANT(T)={+, -, $,) }*
- *SUIVANT(T')= { +, -, $,) }*
- *SUIVANT(F) = {*, /, +, -, $,) }*

Production	Étape	Information	Table d'analyse
$E \rightarrow TE'$	1(a) 1(b)	*PREMIER(TE')={(, nb}* *ε∉PREMIER(TE') : rien à faire*	*Ajouter E → TE' dans la case* M[E,(] Ajouter *E → TE' dans la case* M[E,nb]
$E' \rightarrow +TE'$	1(a)	*PREMIER(+TE')={+}*	*Ajouter E' →+TE' dans la case* M[E',+]
$E' \rightarrow -TE'$	1(a)	*PREMIER(-TE')={-}*	*Ajouter E' →+TE' dans la case* M[E',-]
$E' \rightarrow \varepsilon$	1(a) 1(b)	*Rien à faire* *SUIVANT(E')={$,)}*	*Ajouter E' → ε* dans la case M[E',)] ajouter *E' → ε* dans la case M[E',$]
...			

Table d'analyse LL(1)

Construction de la table d'analyse (exemple 1)

Table d'analyse de la grammaire des expressions arithmétiques :

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid -TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid /FT' \mid \varepsilon$

$F \rightarrow (E) \mid nb$

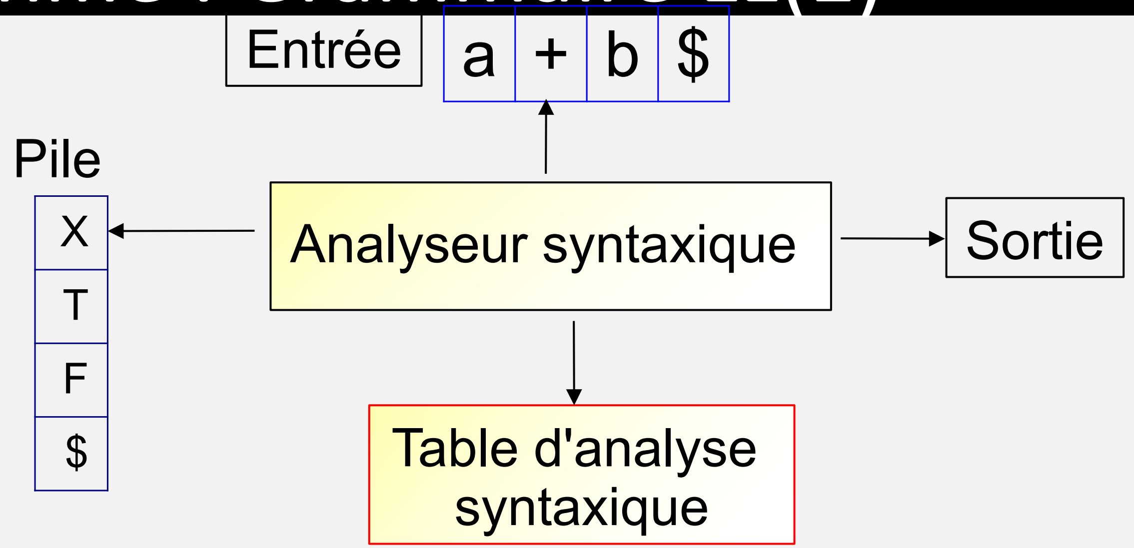
La table d'analyse obtenue est :

Symbole d'entrée

	nb	**+**	**-**	*****	***/***	***(***	***)***	***$***
E	$E \rightarrow TE'$					$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$	$E \rightarrow -TE'$				$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$					$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$	$T' \rightarrow /FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow nb$					$F \rightarrow (E)$		

Analyse descendante prédictive

Algorithme : Grammaire LL(1)



- Dans une configuration où ***X*** est le sommet de la pile et ***a*** est le prochain symbole dans l'entrée, l’analyseur effectue une action parmi les suivantes :
  - Si ***X* = *a* = *$*,** l'analyseur arrête ses opérations et annonce une analyse réussie
  - Si ***X* = *a* ≠ *$***, l'analyseur dépile ***X*** et fait avancer le pointeur de l'entrée
  - Si ***X*** est un non terminal, le programme de l'analyseur consulte la table d'analyse en position ***M[X, a]*. La case consultée fournit soit une production à utiliser, soit une indication d'erreur**
    - Si, par exemple, *M[X, a]* = ***{X → UVW}***, alors l'analyseur **dépile *X*** et empile ***W*, *V* et *U*, dans cet ordre**
    - En cas d'erreur, l'analyseur s'arrête et signale une erreur

La sortie de l’analyseur : la suite de productions utilisées

Analyse descendante prédictive
Exemple 1

- Exemple : grammaire des expressions arithmétiques :
  - $E \to TE'$
  - $E' \to +TE' \mid -TE' \mid \varepsilon$
  - $T \to FT'$
  - $T' \to *FT' \mid /FT' \mid \varepsilon$
  - $F \to (E) \mid nb$

	nb	**+**	**-**	*****	**/**	**(**	**)**	**$**
E	$E \to TE'$					$E \to TE'$		
E'		$E \to +TE'$	$E \to -TE'$				$E' \to \varepsilon$	$E' \to \varepsilon$
T	$T \to FT'$					$T \to FT'$		
T'		$T' \to \varepsilon$	$T' \to \varepsilon$	$T' \to *FT'$	$T' \to /FT'$		$T' \to \varepsilon$	$T' \to \varepsilon$
F	$F \to nb$					$F \to (E)$		

Analyse prédictive de la chaîne ***nb + nb * nb***

Analyse descendante prédictive

Exemple 1

Analyse prédictive de la chaîne *nb + nb * nb*

	nb	**+**	**-**	*****	***/***	***(***	***)***	***$***
E	$E \rightarrow TE'$					$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$	$E \rightarrow -TE'$				$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$					$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$	$T' \rightarrow /FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow nb$					F → (E)		

PILE	ENTREE	SORTIE
E$	*nb+nb*nb$*	$E \rightarrow TE'$
TE'$	nb+nb*nb$	$T \rightarrow FT'$
FT'E'$	nb+nb*nb$	F → *nb*
nbT'E'$	nb+nb*nb$	Consommer le terminal nb
T'E'$	+nb*nb$	$T' \rightarrow \varepsilon$
E'$	+nb*nb$	$E' \rightarrow +TE'$
+*TE'$*	+nb*nb$	*Consommer le terminal +*
TE'$	nb*nb$	$T \rightarrow FT'$
FT'E'$	nb*nb$	$F \rightarrow nb$
nbT'E'$	nb*nb$	*Consommer le terminal nb*
T'E'$	*nb$	**...**
$	$	***Succès***

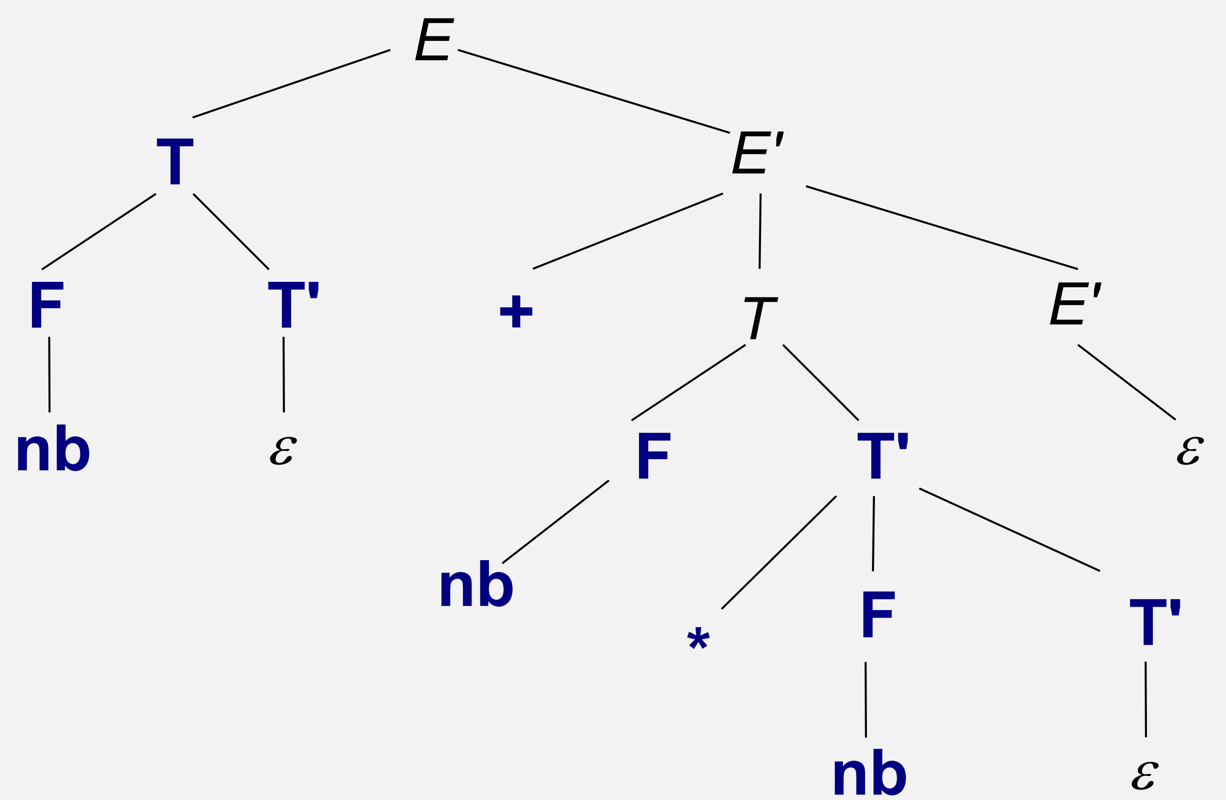
Analyse descendante prédictive

Exemple 1

Analyse prédictive de la chaîne *nb* + *nb* $*$ *nb*

- On oubtient l'arbre syntaxique suivant :

SORTIE
$E \longrightarrow TE'$
$T \longrightarrow FT'$
F $\longrightarrow$ *nb*
Consommer le terminal nb
$T' \longrightarrow \varepsilon$ $E' \longrightarrow +TE'$
Consommer le terminal +
$T \longrightarrow FT'$
$F \longrightarrow nb$
Consommer le terminal nb
... ***Succès***



Analyse descendante prédictive
Algorithme : Grammaire LL(1)

Entrée : Une chaîne ***w*** et la table d'analyse ***M*** associée à une grammaire ***G***

Sortie : Une dérivation à gauche d'abord de ***w*** si $w \in L(G)$ ou erreur sinon

Algorithme :

- initialiser la pile avec l'axiome ***S*** par-dessus ***$*** et le tampon d'entrée avec ***w****$*
- faire pointer ***ip*** sur le premier symbole de l'entrée
- **Répéter**
  - soit ***X*** le symbole du **sommet de la pile** et ***a*** le symbole pointé par ***ip***;
  - **si** *X* est un terminal ou *$* **alors**
    - **si** $X = a$ **alors**
      - dépiler *X* et avancer ***ip***
    - **sinon** *erreur* ()
  - **sinon /*** *X* est un non-terminal */
    - **si** $M[X, a] = X \rightarrow Y_1 \dots Y_k$ **alors début**
      - dépiler *X*
      - empiler $Y_k, \dots, Y_1$, dans l'ordre
      - afficher la production $X \rightarrow Y_1 \dots Y_k$ *(dérivation utilisée)*
    - **fin**
    - **sinon** *erreur* () /* $M[X, a]$ est vide*/
- **jusqu'à ce que** $X = \$$ ou erreur /* la pile est vide */

Analyse descendante prédictive

Exemple 2

Analyser la chaîne (nb + *nb)nb*

	nb	**+**	**-**	*****	**/**	**(**	**)**	**$**
E	$E \rightarrow TE'$					$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$	$E \rightarrow -TE'$				$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$					$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$	$T' \rightarrow /FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow nb$					$F \rightarrow (E)$		

PILE	ENTREE	SORTIE
E$	*(nb+nb)nb$*	$E \rightarrow TE'$
TE'$	*(nb+nb)nb$*	$T \rightarrow FT'$
FT'E'$	*(nb+nb)nb$*	$F \rightarrow (E)$
(E)T'E'$	*(nb+nb)nb$*	*Consommer le terminal (*
...	...	...
E')T'E'$	*)nb$*	$E' \rightarrow \varepsilon$
)T'E'$	*)nb$*	*Consommer le terminal)*
T'E'$	*nb$*	***Erreur***

Le mot (nb+nb)nb n'appartient pas au langagegénéré par cette grammaire

Analyse descendante prédictive
Grammaire LL(1)

- L'algorithme d'analyse descendante prédictive présenté précédement ne peut être appliqué qu'aux grammaires LL(1)
- On appelle grammaire LL(1) une grammaire pour laquelle chaque case de la table d'analyse contient au plus une règle de production
- **LL(1)** signifie
  - **L** : Left to right scanning (parcourt l'entrée de gauche à droite)
  - **L** : Leftmost derivation (dérivation gauches)
  - **1** : un seul symbole de pré-vision est nécessaire
- **Une grammaire ambiguë ou récursive à gauche ou non factorisée à gauche n'est pas LL(1)**

Analyse descendante prédictive
Grammaire LL(1)

- Attention : une grammaire non ambiguë, non récursive à gauche et factorisée à gauche n'est pas toujours LL(1)
- Exemple : la grammaire $S \rightarrow aTb \mid \varepsilon$

$$T \rightarrow cSa \mid d$$

n'est pas LL(1) or elle n'est pas récursive à gauche, elle est factorisée à gauche et elle n'est pas ambiguë !!

Conclusion

- Pour utiliser l'analyse descendante vue ci-dessus il faut vérifier que notre grammaire est bien LL(1)
- Comment **?**
  - Etant donnée une grammaire
    - **La rendre non ambiguë** : il n'y a pas de méthodes
    - Éliminer la récursivité à gauche si nécessaire
    - La factoriser à gauche si nécessaire
    - Construire la table d'analyse
    - Et espérer que ça soit LL(1) (i.e : chaque case de la table d'analyse contient au plus une règle de production)
    - Sinon, il faut concevoir une autre méthode pour l'analyse syntaxique

Exercices

- *Donner la table d'analyse pour les grammaires suivantes :*

$S \rightarrow iEtSS' \mid a$
$S' \rightarrow eS \mid \varepsilon$
$E \rightarrow b$

- Lesquelles de ces grammaires sont LL(1) ?
  - **$S \rightarrow ABBA$**
    - **$A \rightarrow a \mid \varepsilon$**
    - **$B \rightarrow b \mid \varepsilon$**
  - **$S \rightarrow aSe \mid B$**
    - **$B \rightarrow bBe \mid C$**
    - **$C \rightarrow cCe \mid d$**
  - **$S \rightarrow Abc$**
    - **$A \rightarrow a \mid \varepsilon$**
    - **$B \rightarrow b \mid \varepsilon$**
  - **$S \rightarrow Ab$**
    - **$A \rightarrow a \mid B \mid \varepsilon$**
    - **$B \rightarrow b \mid \varepsilon$**