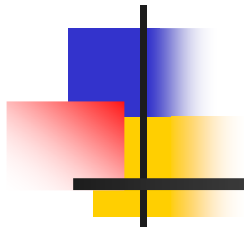


ANALYSE SYNTAXIQUE ASCENDANTE



*ESI – École nationale Supérieure en
Informatique*



Analyseur syntaxique ascendant

- Etude des analyseurs syntaxiques ascendants déterministes.
- Analyse par précédence et analyse LR.
- L'analyse syntaxique ascendante est également appelée analyse par décalage/réduction.



Analyseur syntaxique ascendant

- Le principe général de fonctionnement des analyseurs syntaxiques ascendants est de lire les entités du texte source et par une série de réductions (remplacer des MDP par des MGP) remonter à l'axiome de la grammaire du langage.
 - MDP : Membre Droit d'une règle de Production
 - MGP : Membre Gauche d'une règle de Production



Analyse par Précédence Simple

- Relations de priorité entre tous les symboles de la grammaire.
- Définir des relations.



Analyse par Précédence Simple

- **Définitions des ensembles PREMIER et DERNIER :**

$\text{PREMIER}(A) = \{X \mid A \Rightarrow^+ X\alpha, X \in (N \cup T), \alpha \in (N \cup T)^*\}$

$\text{DERNIER}(A) = \{X \mid A \Rightarrow^+ \alpha X, X \in (N \cup T), \alpha \in (N \cup T)^*\}$



Analyse par Précédence Simple

Relations entre 2 symboles $S1$, $S2 \in (NUT)^2$:

- **$S1 = S2$**

- Si \exists 1 production $A \rightarrow \dots S1S2 \dots$

- **$S1 < S2$**

- Si \exists une production $A \rightarrow \dots S1X \dots$ et $S2 \in \text{PREMIER}(X)$

- **$S1 > S2$**

- Si \exists une production $A \rightarrow \dots XY \dots$ et $S1 \in \text{DERNIER}(x)$ et $S2 \in \text{Début}(Y)$; $S2 \in T$

Analyse par Précédence Simple

Relations entre 2 symboles $S1$, $S2 \in (NUT)^2$

<p>Diagram illustrating the relation $S1 = S2$. The root node S has two children: $\dots X \dots$ and $\dots S1 S2 \dots$.</p>	<p>Diagram illustrating the relation $S1 < S2$. The root node S has two children: $\dots S1 X \dots$ and $\dots S2 \dots$.</p>	<p>ou</p> <p>Diagram illustrating the relation $S1 > S2$. The root node S has two children: $\dots X Y \dots$ and $\dots S1 S2 \dots$.</p>
$S1 = S2$	$S1 < S2$	$S1 > S2$



Analyse par Précédence Simple

- **Grammaire de précédence simple :**
 - i. Il existe au plus une relation de précédence simple entre symboles de la grammaire ;
 - ii. Deux règles de production ne peuvent pas avoir un même membre droit :
 - iii. Les productions vides ne sont pas permises.



Analyse par Précédence Simple

- **Théorème :**

- Si une grammaire G est de précédence simple alors G est non ambiguë.



Analyse par Précédence Simple

- **Algorithme d'analyse de $\omega\#$:**
 - Ajout de la production $Z \rightarrow \#S\#$
 - La pile contient initialement $\#$;
 - SP désigne le symbole en sommet de pile
 - tc désigne le terme courant de la chaîne à analyser ;
 - Initialement tc pointe sur le début de la chaîne ω ;
 - ts désigne le terme suivant tc dans la chaîne à analyser ;



Analyse par Précédence Simple

■ **Algorithme d'analyse (suite):**

Si $(SP=tc)$ ou $(SP < tc)$

Alors empiler (relation); empiler (tc); $tc=ts$;

Sinon Si $(SP > tc)$

Alors Tant que $(SP \neq <)$ **Faire** Dépiler; **FinTq** ;

Dépiler ; // reconnaissance $X \rightarrow MDP$

Si $(X = \text{Axiome})$ et $(SP = \#)$ et $(tc = \#)$

Alors "Syntaxiquement correcte" ; **FinSi** ;

Empiler (Relation(SP, X)) ; Empiler (X);

Sinon "Erreur Syntaxique" ; **FinSi** ;

FinSi ; Reprendre l'analyse ;

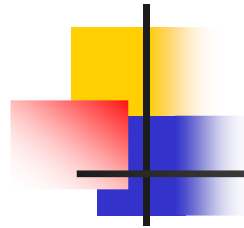


Analyse par Précédence Simple

- **Exemple d'analyse :**
- La grammaire G engendrant des expressions avec parenthèses balancées.

$S \rightarrow A)$

$A \rightarrow (\mid Aa \mid AS$



Analyse par Précédence Simple

- Table des relations :

	S	A	()	a	#
S			>	>	>	
A	=	<	<	=	=	
(>	>	>	
)			>	>	>	>
a			>	>	>	
#		<	<			

Analyse par Précédence Simple

Exemple d'analyse

Pile (sommet à droite)	Relation	Restant de la chaîne à analyser	Action
#	<	(a(a))#	Décaler
# < (>	a(a))#	Réduction : $A \rightarrow ($
# < A	=	a(a))#	Décaler
# < A = a	>	(a))#	Réduction : $A \rightarrow Aa$
# < A	<	(a))#	Décaler
# < A < (>	a))#	Réduction : $A \rightarrow ($
# < A < A	=	a))#	Décaler
# < A < A = a	>))#	Réduction : $A \rightarrow Aa$
# < A < A	=))#	Décaler
# < A < A =)	>)#	Réduction : $S \rightarrow A)$
# < A = S	>)#	Réduction : $A \rightarrow AS$
# < A	=)#	Décaler
# < A =)	>	#	Réduction : $S \rightarrow A)$
# S		#	Chaîne acceptée



Analyse par Précédence Simple

- **Remarque :**

- Si S est l'axiome d'une grammaire, les relations $(\# = S)$ et $(S = \#)$ sont prohibées et ne seront jamais portées sur la table des relations.
- Inclure ces relations risque de refuser artificiellement des grammaires de l'analyse de précédence simple.



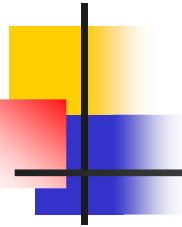
Analyse par Précédence Simple

- **Limites de l'analyse par précédence simple :**
 - La méthode d'analyse par précédence simple est facile à implémenter mais la classe des grammaires de précédence simple reste "**restreinte**".
 - En effet, la grammaire des expressions arithmétique qui est incontournable (utilisée par les compilateurs, tableurs, ...) n'est pas de précédence simple.

Analyse par Précédence Simple

- Table des relations pour la grammaire des expressions arithmétiques :

	E	T	F	+	*	()	i	#
E				=			=		
T				>	=		>		>
F				>	>		>		>
+		= <	<			<		<	
*			=			<		<	
(= <	<	<			<		<	
)				>	>		>		>
i				>	>		>		>
#	<	<	<			<		<	



Analyse par Précédence Simple

- Extension de l'analyse par Analyse de Précédence Faible.
- Méthode plus générale : Analyse LR



Analyse LR

- **Signification de LR (k) :**

L : Left to right parsing, constructing a

R : Rightmost derivation in reverse, using

k : k tokens to take an analysis decision.



Analyse LR

- L'analyse LR(k) est la méthode par décalage/réduction sans rebroussement la plus générale connue ;
- La classe des grammaires qui peuvent être analysées par la méthode LR est un sur-ensemble strict de la classe des grammaires qui peuvent être analysées par les méthodes prédictives (grammaires LL).
- K est généralement égal à 1.



Analyse LR

- Les méthodes LR sont les méthodes les plus utilisées par les générateurs de compilateurs ; Yacc par exemple utilise la méthode LALR(1) pour générer l'analyseur syntaxique



Analyse LR

- **Analyses LR :**

SLR : Simple LR

LALR : Look Ahead LR

LR



Analyse SLR

Méthode constructive de la table d'analyse

- **Définition**

- Un item LR(0) d'une grammaire G est une production avec un point repérant une position du membre droit de cette production.

- **Exemples :**

- La production $A \rightarrow XYZ$ fournit quatre items que sont :
 - $[A \rightarrow .XYZ]$
 - $[A \rightarrow X.YZ]$
 - $[A \rightarrow XY.Z]$
 - $[A \rightarrow XYZ.]$
- La production $A \rightarrow \varepsilon$ fournit l'item : $[A \rightarrow .]$



Analyse SLR

Méthode constructive de la table d'analyse

- **Interprétation :**

- Le point dans un item sépare le membre droit de production en partie gauche (avant le point) et droite(après le point) ;
- cet item "signifie" qu'on a reconnu une chaîne dérivée de la partie gauche et on attend de reconnaître une chaîne dérivée de la partie droite.

- **Remarque :**

- Un item peut être codé par deux entiers : le numéro de production et la position du point dans le membre droit de cette production



Analyse SLR

Méthode constructive de la table d'analyse

- **Fermeture d'un ensemble d'items :**

- L'idée centrale d'un analyseur SLR est de construire à partir de la grammaire un automate d'états finis AFD qui reconnaît les contextes gauches d'une grammaire.
- Les items sont regroupés en ensembles qui constitueront les états de l'AFD
- La grammaire initiale sera augmentée de la règle $S' \rightarrow S$ (où S est l'axiome de la grammaire) pour obtenir l'état initial de l'AFD.



Analyse SLR

Méthode constructive de la table d'analyse

- **Fermeture (I):**
- Soit I un ensemble d'items ;
- Placer chaque item de I dans Fermeture(I) ;
- **Si** $[A \rightarrow \alpha.B\beta]$ est dans Fermeture(I) et $B \rightarrow \gamma$ est une production
- **Alors** ajouter l'item $[B \rightarrow .\gamma]$ à Fermeture(I) s'il n'y est pas
- **FinSi**
- Appliquer cette règle jusqu'à ce qu'aucun item ne puisse être ajouté à Fermeture(I).



Analyse SLR

Méthode constructive de la table d'analyse

- **Exemple :**
- Considérer la grammaire G dont les productions sont :
 - $E \rightarrow E + T \mid T$
 - $T \rightarrow T * F \mid F$
 - $F \rightarrow (E) \mid i$
- La grammaire est augmentée de la règle $E' \rightarrow E$;
- Fermeture ($[E' \rightarrow .E]$) = $\{ [E' \rightarrow .E], [E \rightarrow .E+T], [E \rightarrow T], [T \rightarrow .T*F], [T \rightarrow .F], [F \rightarrow .(E)], [F \rightarrow .i] \}$



Analyse SLR

Méthode constructive de la table d'analyse

- **Opération transition (ou GOTO)**
- L'opération GOTO (I, X), où I est un état et X un symbole de la grammaire, est définie comme suit :
- $\text{GOTO}(I, X) = \text{Fermeture}([A \rightarrow \alpha X \beta])$;
- tel que : $[A \rightarrow \alpha X \beta] \in I$.



Analyse SLR

Méthode constructive de la table d'analyse

- Collection canonique C , d'ensembles d'items LR(0)
- **Procédure** Items (G') ;
 - $C = \{\text{Fermeture}([S' \rightarrow S])\}$;
 - **Répéter**
 - **Pour** (**chaque** ensemble d'items I de C et **chaque** symbole X)
 - **tel que** $\text{GOTO}(I, X)$ est non vide et non encore dans C
 - **Faire** ajouter $\text{GOTO}(I, X)$ à C ;
 - **Fait** ;
 - **Jusqu'à** ce qu'aucun nouvel ensemble ne puisse être ajouté à C



Analyse SLR

Méthode constructive de la table d'analyse

- **Exemple :**
- Construire la collection canonique pour la grammaire augmentée G' dont les productions sont données :
- $E' \rightarrow E$
- $E \rightarrow E + T \mid T$
- $T \rightarrow T * F \mid F$
- $F \rightarrow (E) \mid i$



Analyse SLR

Méthode constructive de la table d'analyse

- $I_0 = \{[E' \rightarrow .E], [E \rightarrow .E+T], [E \rightarrow .T], [T \rightarrow .T*F], [T \rightarrow .F], [F \rightarrow .(E)], [F \rightarrow .i]\}$
- $I_1 = \text{GOTO}(I_0, E) = \{[E' \rightarrow E.], [E \rightarrow E.+T]\}$
- $I_2 = \text{GOTO}(I_0, T) = \{[E \rightarrow T.], [E \rightarrow T.*F]\}$
- $I_3 = \text{GOTO}(I_0, F) = \{[T \rightarrow F.]\}$
- $I_4 = \text{GOTO}(I_0, () = [F \rightarrow (.E)], [E \rightarrow .E+T], [E \rightarrow .T], [T \rightarrow .T*F], [T \rightarrow .F], [F \rightarrow .(E)], [F \rightarrow .i]\}$
- $I_5 = \text{GOTO}(I_0, i) = \{[F \rightarrow i.]\}$
- $I_6 = \text{GOTO}(I_1, +) = \{[E \rightarrow E+.T], [T \rightarrow .T*F], [T \rightarrow .F], [F \rightarrow .(E)], [F \rightarrow .i]\}$
- $I_7 = \text{GOTO}(I_2, *) = \{[T \rightarrow T*.F], [F \rightarrow .(E)], [F \rightarrow .i]\}$
- $I_8 = \text{GOTO}(I_4, E) = \{[E \rightarrow E.+T], [F \rightarrow (E.)]\}$
- $\text{GOTO}(I_4, T) = I_2$
- $\text{GOTO}(I_4, F) = I_3$
- $\text{GOTO}(I_4, () = I_4$
- $\text{GOTO}(I_4, i) = I_5$



Analyse SLR

Méthode constructive de la table d'analyse

- $I_9 = \text{GOTO}(I_6, T) = \{[E \rightarrow E+T.], [T \rightarrow T.*F]\}$
- $\text{GOTO}(I_6, F) = I_3$
- $\text{GOTO}(I_6, () = I_4$
- $\text{GOTO}(I_6, i) = I_5$
- $I_{10} = \text{GOTO}(I_7, F) = \{[T \rightarrow T*F.]\}$
- $\text{GOTO}(I_7, () = I_4$
- $\text{GOTO}(I_7, i) = I_5$
- $I_{11} = \text{GOTO}(I_8,)) = \{[F \rightarrow (E).]\}$
- $\text{GOTO}(I_8, +) = I_6$
- $\text{GOTO}(I_9, *) = I_7$



Analyse SLR

Méthode constructive de la table d'analyse

- **Algorithme de construction de la table SLR(1)**
- i) Construire $C = \{I_0, I_1, \dots, I_n\}$ la collection canonique d'items LR(0) de G' ;
- ii) L'état i est construit à partir de I_i ;
- iii) Les actions d'analyse pour l'état i sont obtenues par :
 - Si** $[A \rightarrow \alpha.a\beta]$ est dans I_i , $a \in T$ et $GOTO(I_i, a) = I_j$
Alors ACTION[i, a] = "Décaler j " ; **FinSi** ;
 - Si** $[A \rightarrow \alpha.]$ est dans I_i
Alors ACTION[i, b] = "Réduire par $A \rightarrow \alpha$ " pour chaque $b \in SUIV(A)$; **FinSi**
 - Si** $[S' \rightarrow S.]$ est dans I_i **Alors** ACTION[$i, \#$] = "Accepter« ; **FinSi** ;
- iv) Transition SUCCESSEUR pour l'état i
 - Si** $GOTO(I_i, A) = I_j$; $A \in N$
Alors SUCCESSEUR [i, A] = j ; **FinSi** ;
- iv) Les entrées non définies par les règles iii) et iv) sont mises à "Erreur"



Analyse SLR

Méthode constructive de la table d'analyse

- **Remarques :**
- Si la table d'analyse ainsi construite est mono-définie alors la grammaire est SLR(1). On pourra alors faire une analyse SLR(1).
- L'état initial de l'analyseur est celui construit à partir de l'ensemble d'items contenant l'item $[S' \rightarrow .S]$.



Analyse SLR

Méthode constructive de la table d'analyse

- **Algorithme d'analyse :**
- La pile contient au départ l'état initial de l'automate S_0 ;
- Le pointeur de chaîne P_s pointe sur le premier symbole de la chaîne à analyser ;
- Soit S l'état en sommet de pile ;
- Soit a le symbole pointé par P_s ;



Analyse SLR

Méthode constructive de la table d'analyse

- **Si** ACTION[S,a] = "Décaler T"
- **Alors** empiler(a) ; empiler(T) ;
- avancer Ps sur le prochain symbole ;
- **Sinon** **Si** ACTION[S,a] = "Réduire par $A \rightarrow \alpha$ "
- **Alors** dépiler $2*|\alpha|$ symboles ;
- Soit U le nouvel état en sommet de pile ;
- empiler (A); empiler(SUCCESSEUR[U,A];
- **Sinon** **Si** ACTION[S,a] = "Accepter"
- **Alors** retourner (Réussite());
- **Sinon** retourner (Echec()); **FinSi** ;
- **FinSi** ;
- **FinSi** ;



Analyse SLR

Méthode constructive de la table d'analyse

- **Exemple :**
- Construire la table d'analyse SLR(1) pour la grammaire augmentée G' ($E' \rightarrow E$) dont les productions numérotées sont données ci après :
- **(1)** $E \rightarrow E + T$
- **(2)** $E \rightarrow T$
- **(3)** $T \rightarrow T * F$
- **(4)** $T \rightarrow F$
- **(5)** $F \rightarrow (E)$
- **(6)** $F \rightarrow i$



Analyse SLR

Méthode constructive de la table d'analyse

- Calcul des ensembles SUIVANT :


	SUIVANT
E	+) #
T	+) * #
F	+) * #

Analyse SLR

Méthode constructive de la table d'analyse

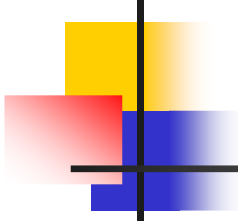
La collection canonique étant déjà calculée, on obtient la table d'analyse SLR(1) suivante :

	+	*	()	i	#	E	T	F
0			D 4		D 5		1	2	3
1	D 6					Accepter			
2	R (2)	D 7		R (2)		R (2)			
3	R (4)	R (4)		R (4)		R (4)			
4			D 4		D 5		8	2	3
5	R (6)	R (6)		R (6)		R (6)			
6			D 4		D 5			9	3
7			D 4		D 5				10
8	D 6			D 11					
9	R (1)	D 7		R (1)		R (1)			
10	R (3)	R (3)		R (3)		R (3)			
11	R (5)	R (5)		R (5)		R (5)			

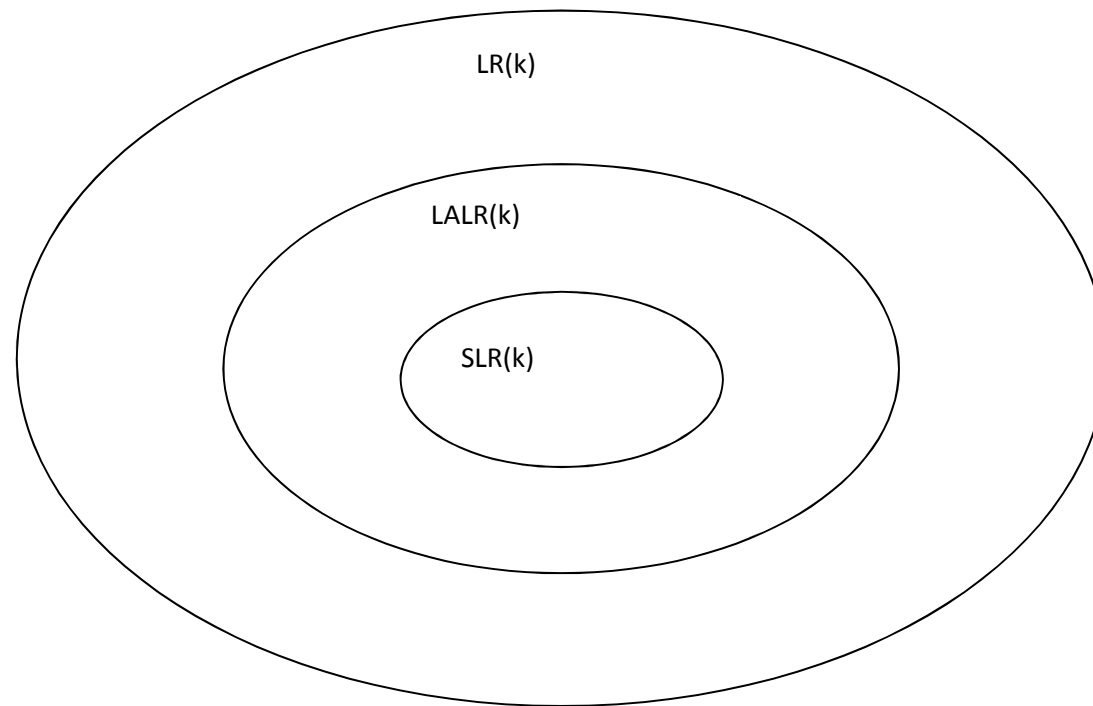



Analyse LR(1) et LALR(1)

- Méthodes analogues à SLR(1) mais beaucoup plus générales.
- **Comparaison des trois méthodes.**



Analyse LR(1) et LALR(1)





Analyse LR(1) et LALR(1)

- Méthodes analogues à SLR(1) mais beaucoup plus générales.
- **Comparaison des trois méthodes.**