

Analyse ascendante

- ▶ A l'inverse des méthodes descendantes qui sont basées sur une dérivation de l'axiome les méthodes d'analyse par décalage-réduction sont des méthodes ascendantes qui partent de la chaîne (des feuilles) pour remonter à l'axiome.
- ▶ A chaque étape la partie droite d'une production présente dans la chaîne est remplacée par sa partie gauche.
- ▶ Si on fait les bons choix (et si la chaîne appartient au langage) on remonte ainsi à l'axiome.

Analyse ascendante : exemple

Considérons la grammaire:

$$\begin{aligned} S &\rightarrow aABe \\ A &\rightarrow Abc|b \\ B &\rightarrow d \end{aligned}$$

Et $w = abbcde$, le terminal le plus à gauche, b peut être remplacé par A pour obtenir $aAbcde$. $aAbcde$ peut être réécrit en $aAde$, puis en $aABe$ et finalement en S . Cela correspond à la séquence de dérivations droite suivante :

$$S \xRightarrow[d]{\Rightarrow} aABe \xRightarrow[d]{\Rightarrow} aAde \xRightarrow[d]{\Rightarrow} aAbcde \xRightarrow[d]{\Rightarrow} abbcde$$

$\Rightarrow_d$ signifie qu'on applique des dérivations droites à partir de S pour obtenir w .

Notez également que l'utilisation de la production $A \rightarrow b$ sur $aAbcde$ aurait donné $aAAcde$ ce qui ne permettait pas de revenir à S .

Manche

Si, il existe $A \rightarrow \beta$ tel que :

$$S \xRightarrow[d]{*} \alpha A w \xRightarrow[d]{} \alpha \beta w, \quad w \in \Sigma^*$$

Le couple $(A \rightarrow \beta, |\alpha| + 1)$ est appelé un manche de $\alpha \beta w$. Notez que :

- La chaîne w qui suit β est composée uniquement de terminaux.
- Un manche est identifié à la fois par sa production et par la position de son membre droit dans la chaîne.
- Un manche n'existe qu'à partir d'une dérivation de l'axiome. Donc, si on a une phrase γ identifier β dans γ tel que $\gamma = \alpha \beta w$, $w \in \Sigma^*$ ne suffit pas.

Il faut également que $S \xRightarrow{*} \gamma$.

Intuitivement, un manche est un mot que l'on va saisir pour remonter une dérivation.

Manche : Exemple

Considérons :

$$\begin{aligned} S &\rightarrow aABe \\ A &\rightarrow Abc|b \\ B &\rightarrow d \end{aligned}$$

et :

$$\underset{d}{S} \Rightarrow \underset{d}{aABe} \Rightarrow \underset{d}{aAde} \Rightarrow \underset{d}{aAbcde} \Rightarrow abbcde$$

- ▶ $(A \rightarrow b, 2)$ est un manche de $abbcde$ ($\alpha = a, \beta = b, w = bcde$)
- ▶ $(A \rightarrow Abc, 2)$ est un manche de $aAbcde$
- ▶ $(B \rightarrow d, 3)$ est un manche de $aAde$
- ▶ $(S \rightarrow aABe, 1)$ est un manche de $aABe$

Elagage du manche

Partant d'un mot composé de terminaux $w = \gamma_n$, on trouve un manche $(A_n \rightarrow \beta_n, i_n)$ et on remplace dans γ_n , β_n en position i_n par A_n pour obtenir γ_{n-1} , on cherche à nouveau le manche de γ_{n-1} pour obtenir γ_{n-2} et ainsi de suite jusqu'à $\gamma_0 = S$. On a donc :

$$S = \gamma_0 \underset{d}{\Rightarrow} \gamma_1 \underset{d}{\Rightarrow} \gamma_2 \dots \underset{d}{\Rightarrow} \gamma_{n-1} \underset{d}{\Rightarrow} \gamma_n = w$$

Elagage du manche : Exemple

Soit la grammaire

$$E \rightarrow E + E | E * E | (E)$$

$$E \rightarrow id$$

et $w = id1 + id2 * id3$. On obtient la séquence de réductions suivante :

Proto phrase droite	Manche	Production
$id1 + id2 * id3$	$id1$	$E \rightarrow id$
$E + id2 * id3$	$id2$	$E \rightarrow id$
$E + E * id3$	$id3$	$E \rightarrow id$
$E + E * E$	$E * E$	$E \rightarrow E * E$
$E + E$	$E + E$	$E \rightarrow E + E$
E		

Notons que nous avons 2 manches possibles à la ligne 3 ($E \rightarrow E + E$ et $E \rightarrow id$) que l'on résout en utilisant la priorité de la multiplication.

Analyse par décalage-réduction avec une pile

L'analyseur est constitué d'une pile et d'un tampon d'entrée. A chaque étape, l'extrémité droite du manche (si il existe) se trouve en sommet de pile. L'analyseur ajoute un \$ en fin de la chaîne à reconnaître et en début de pile puis effectue les actions suivantes :

1. Si on ne trouve pas de manche dans la pile, on effectue une action *décaler*: On déplace le prochain symbole du tampon d'entrée dans la pile.
2. Si il existe un manche ($A \rightarrow \beta$) en sommet de pile, on effectue une action *réduire* : la séquence de symboles définissant β est remplacée par A dans la pile.
3. Si la pile contient $\$S$, S l'axiome et le tampon d'entrée $\$$ l'analyseur effectue une action *accepter* : La chaîne d'entrée est reconnue.
4. Si aucun manche ne peut être trouvé par une action décaler, l'analyseur effectue une action *erreur*.

Analyse par décalage-réduction : Exemple

Pile	Entrée	Action
(1) \$	$id_1 + id_2 * id_3 \$$	décaler
(2) \$ id_1	$+id_2 * id_3 \$$	réduire par $E \rightarrow Id$
(3) \$ E	$+id_2 * id_3 \$$	décaler
(4) \$ $E+$	$id_2 * id_3 \$$	décaler
(5) \$ $E+id_2$	$*id_3 \$$	réduire par $E \rightarrow Id$
(6) \$ $E+E$	$*id_3 \$$	décaler
(7) \$ $E+E*$	$id_3 \$$	décaler
(8) \$ $E+E*id_3$	\$	réduire par $E \rightarrow Id$
(9) \$ $E+E*E$	\$	réduire par $E \rightarrow E * E$
(10) \$ $E+E$	\$	réduire par $E \rightarrow E + E$
(11) \$ E	\$	accepter

Notez le conflit décaler/réduire ligne 6 résolu en utilisant la priorité des opérateurs. Il existe en fait 2 types de conflit: décaler/réduire et réduire/réduire quand il existe des non terminaux A_1, A_2 tels que $A_1 \rightarrow \beta$ et $A_2 \rightarrow \beta$.

Préfixes viables

- ▶ Un préfixe viable est un préfixe d'une proto-phrased ne s'étendant pas au-delà de la fin du manche le plus à droite de la proto-phrased.
- ▶ Par exemple: $E + E$ est un préfixe viable de $E + E * id_3$
- ▶ Dans l'algorithme précédent, on peut donc décaler des symboles du tampon d'entrée tant que le contenu de la pile (lu de la base au sommet) constitue un préfixe viable de la proto-phrased constitué de la pile et du tampon d'entrée.

Analyseurs LR

LR(k)

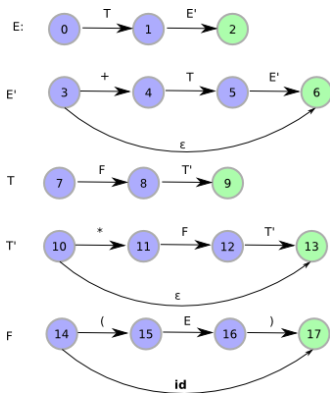
Left to right scanning of the input constructing a
Right most derivation in reverse using k symbols.

- ▶ Tous les langages que vous avez vu, peuvent être reconnus par un analyseur LR.
- ▶ La méthode d'analyse LR est la méthode par décalage réduction sans rebroussement la plus connue. Elle peut cependant être implantée aussi efficacement que d'autres méthodes utilisant le même principe.
- ▶ L'ensemble des grammaires reconnaissables par un analyseur LR est un sous-ensemble strict des grammaires pouvant être reconnues par un analyseur prédictif.
- ▶ Les analyseurs LR peuvent détecter très tôt les erreurs de syntaxe.
- ▶ Les tables d'analyse LR sont trop grandes pour être rédigées 'à la main'. Heureusement de très bons outils (Yacc, Bison) existent.

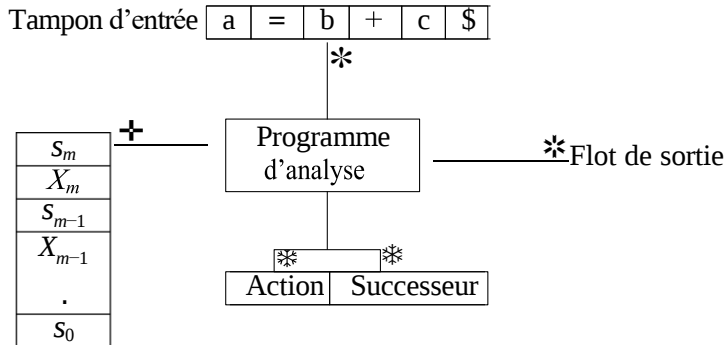
Analyse LR : Idée de base

$$\begin{aligned}E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \epsilon \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid \epsilon \\ F &\rightarrow (E) \mid \text{id}\end{aligned}$$

- Les algorithmes d'analyse précédents prennent une décision sur la vue de T en sommet de pile et la lecture d'un symbole dans le tampon d'entrée.
- Voit-on T depuis l'état 0 ou depuis l'état 4 ?
- Les méthodes d'analyse précédentes (dont celle des langages LL(1)) manquent de 'mémoire'.



Analyse LR : schéma



Analyse LR : schéma

- ▶ La pile contient une séquence $s_0 X_1 \dots X_m s_m$ alternant des états s_i et des symboles X_i représentant les transitions entre ces états.
- ▶ L'état s_m résume donc l'état de la pile.
- ▶ Si a est le symbole courant du tampon d'entrée et s_m le sommet de la pile $\text{Action}[s_m, a]$ correspond à une des actions suivantes :
 - ▶ décaler s , où s est un état,
 - ▶ réduire par une production $A \rightarrow \beta$
 - ▶ accepter
 - ▶ erreur
- ▶ Successeur[s, X] (s un état, X un non terminal) renvoie un état.

Analyse LR : Algorithme

Une configuration de l'analyseur est donnée par le couple (pile, tampon d'entrée) : $(s_0X_1 \dots X_ms_m, a_i \dots, a_n\$)$ représentant la proto-phrased : $X_1X_2 \dots X_{m-1}X_ma_i \dots, a_n$.

1. Si $\text{Action}[s_m, a_i] = \text{décaler } s$. On décale a_i et on empile s pour passer dans la configuration : $(s_0X_1 \dots X_ms_ma_i s, a_{i+1} \dots, a_n\$)$
2. Si $\text{Action}[s_m, a_i] = \text{réduire par } A \rightarrow \beta = X_{m-r+1} \dots X_m$. On passe dans la configuration : $(s_0X_1 \dots s_{m-r}A s, a_i \dots, a_n\$)$
avec $s = \text{Successeur}[s_{m-r}, A]$. Le tampon d'entrée reste inchangé.
3. Si $\text{Action}[s_m, a_i] = \text{accepter}$: renvoyer succès.
4. Si $\text{Action}[s_m, a_i] = \text{erreur}$, appeler une routine de récupération d'erreur.

Analyse LR : Algorithme

fonction analyseLR (w\$: chaîne) : Booleen

Déclaration a : symbole courant du tampon, s : Sommet de pile

début

répéter

si Action[s, a] = décaler s' alors

empiler a puis s'

avancer sur le prochain symbole du tampon d'entrée

sinon

si Action[s, a] = réduire par $A \rightarrow \beta$ alors

dépiler $2|\beta|$ symboles

$s' \leftarrow$ nouveau sommet de pile

empiler A puis Successeur[s, A]

finsi

si Action[s, a] = Erreur alors Erreur()

finsi

jusqu'à ce que Action[s, a] = accepter

fin

Analyse LR : Exemple

Toutes les méthodes d'analyse LR sont basées sur l'algorithme précédent. La différence intervient sur les méthodes d'initialisation des tables Action et Successeur.

Considérons la grammaire :

- (1) $E \rightarrow E + T$
- (2) $E \rightarrow T$
- (3) $T \rightarrow T * F$
- (4) $T \rightarrow F$
- (5) $F \rightarrow (E)$
- (6) $F \rightarrow \text{id}$

avec les conventions suivantes :

- di : décaler et empiler i
- rj : réduire par la production j
- acc : accepter
- entrée vide : erreur

Analyse LR : Exemple

On obtient les tables Action/ successeur suivantes (nous verrons plus loin comment les calculer).

Etat	Action						Successeur		
	id	+	*	(	)	\$	E	T	F
0	d5			d4			1	2	3
1		d6				acc			
2		r2	d7		r4	r4			
3		r4	r4		r4	r4			
4	d5			d4			8	2	3
5		r6	r6		r6	r6			
6	d5			d4				9	3
7	d5			d4					10
8		d6			d11				
9		r1	d7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

Analyse syntaxique ascendante

Page 18 sur 83

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d7

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$
(9) 0 E 1	+id\$	d 6

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$
(9) 0 E 1	+id\$	d 6
(10) 0 E 1 + 6	id\$	d 5

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$
(9) 0 E 1	+id\$	d 6
(10) 0 E 1 + 6	id\$	d 5
(11) 0 E 1 + 6 id 5	\$	r par $F \rightarrow id$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$
(9) 0 E 1	+id\$	d 6
(10) 0 E 1 + 6	id\$	d 5
(11) 0 E 1 + 6 id 5	\$	r par $F \rightarrow id$
(12) 0 E 1 + 6 F 3	\$	r par $T \rightarrow F$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$
(9) 0 E 1	+id\$	d 6
(10) 0 E 1 + 6	id\$	d 5
(11) 0 E 1 + 6 id 5	\$	r par $F \rightarrow id$
(12) 0 E 1 + 6 F 3	\$	r par $T \rightarrow F$
(13) 0 E 1 + 6 T 9	\$	r $E \rightarrow E + T$

Analyse LR : Exemple

Pile	Entrée	Action
(1) 0	id*id+id\$	d 5
(2) 0 id 5	*id+id\$	r par $F \rightarrow id$
(3) 0 F 3	*id+id\$	r par $T \rightarrow F$
(4) 0 T 2	*id+id\$	d 7
(5) 0 T 2 * 7	id+id\$	d 5
(6) 0 T 2 * 7 id 5	+id\$	r par $F \rightarrow id$
(7) 0 T 2 * 7 F 10	+id\$	r $T \rightarrow T * F$
(8) 0 T 2	+id\$	r $E \rightarrow T$
(9) 0 E 1	+id\$	d 6
(10) 0 E 1 + 6	id\$	d 5
(11) 0 E 1 + 6 id 5	\$	r par $F \rightarrow id$
(12) 0 E 1 + 6 F 3	\$	r par $T \rightarrow F$
(13) 0 E 1 + 6 T 9	\$	r $E \rightarrow E + T$
(14) 0 E 1	\$	accepter