Analyseur syntaxique

Sommaire (Analyse ascendante)

- Principe
- Table analyse SLR
- Algorithme de construction des Fermetures d'un ensemble d'items I
- Exemple
- Algorithme de construction des collections des items d'une grammaire
- Construction de la table d'analyse SLR
- Principe d'analyse
- Exercice

Analyse ascendante

Principe

construire un arbre de dérivation du bas (les feuilles, ie les unités lexicales) vers le haut (la racine, ie l'axiome de départ).

Le modèle général utilisé est le modèle par **décallages-réductions.** C'est à dire que l'on ne s'autorise que deux opérations :

- **décallage (shift)** : décaller d'une lettre le pointeur sur le mot en entrée
- **réduction (reduce)** : réduire une chaîne (suite consécutive de terminaux et non terminaux à gauche du pointeur sur le mot en entrée et finissant sur ce pointeur) par un non-terminal en utilisant une des règles de production

Exemple

$S \rightarrow aSbS|c$ avec le mot $u = aaacbaacbcbcbcbacbc$

$\boxed{a}aacbaacbcbcbcbacbc$ on ne peut rien **réduire**, donc on **décalle**

$a\boxed{a}acbaacbcbcbcbacbc$ on ne peut rien réduire, donc on décalle

$aa\boxed{a}cbaacbcbcbcbacbc$ on ne peut rien réduire, donc on décalle

$aaa\boxed{c}baacbcbcbcbacbc$ ah ! On peut réduire par $S \to c$

$aaa\boxed{S}baacbcbcbcbacbc$ on ne peut rien réduire, donc on décalle

...

$aaaSbaa\boxed{c}bcbcbcbacbc$ on peut utiliser $S \to c$

$aaaSbaa\boxed{S}bcbcbcbacbc$ on ne peut rien réduire, donc on décalle

$aaaSbaaS\boxed{b}cbcbcbacbc$ on ne peut rien réduire, donc on décalle

$aaaSbaaSb\boxed{c}bcbcbacbc$ On peut utiliser $S \to c$

$aaaSbaaSb\boxed{S}bcbcbacbc$ On peut utiliser $S \to aSbS$

$aaaSba\boxed{S}bcbcbacbc$ décallage

$aaaSbaS\boxed{b}cbcbacbc$ décallage

$aaaSbaSb\boxed{c}bcbacbc$ réduction par $S \to c$

$aaaSbaSb\boxed{S}bcbacbc$ réduction par $S \to aSbS$

$aaaSb\boxed{S}bcbacbc$ réduction par $S \to aSbS$

$aa\boxed{S}bcbacbc$ décallage

$aaS\boxed{b}cbacbc$ décallage

$aaSb\boxed{c}bacbc$ réduction par $S \to c$

$aaSb\boxed{S}bacbc$ réduction par $S \to aSbS$

$a\boxed{S}bacbc$ décallage

$aS\boxed{b}acbc$ décallage

$aSb\boxed{a}cbc$ décallage

$aSba\boxed{c}bc$ réduction par $S \to c$

$aSba\boxed{S}bc$ décallage

$aSbaS\boxed{b}c$ décallage

$aSbaSb\boxed{c}$ réduction par $S \to c$

$aSbaSb\boxed{S}$ réduction par $S \to aSbS$

$aSb\boxed{S}$ réduction par $S \to aSbS$

$\boxed{S}$ **terminé !!!!** On a gagné, le mot est bien dans le langage.

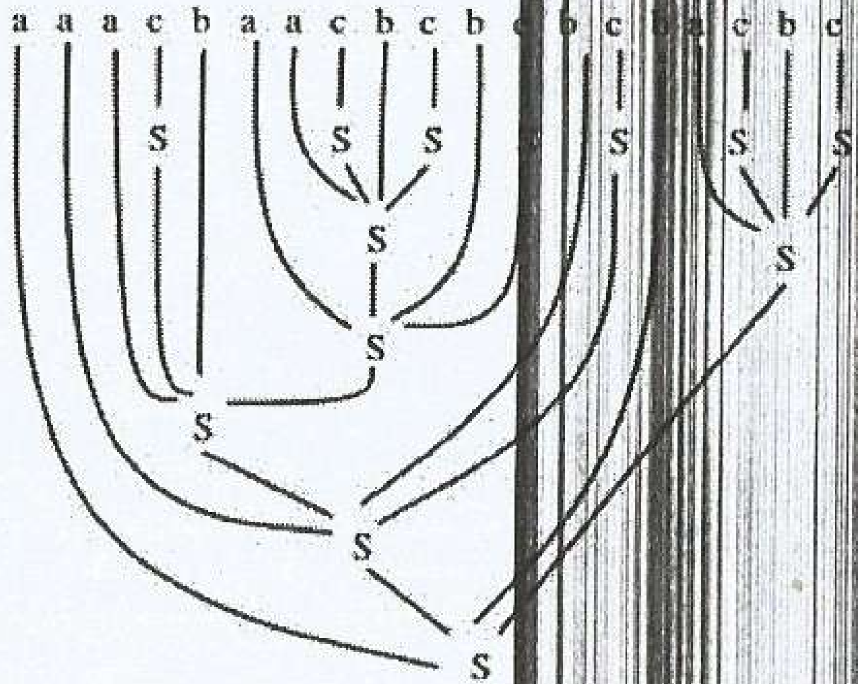


Table analyse SLR

(on l'appelle comme ça parce que, de même que la méthode descendante vue précédemment ne permettait d'analyser que les grammaires LL(1), cette méthode va permettre d'analyser les grammaires dites LR).
Cette table va nous dire ce qu'il faut faire quand on lit une lettre a et qu'on est dans un état i

- soit on **décalle**. Dans ce cas, on empile la lettre lue et on va dans un autre état j. Ce qui sera noté dj
- soit on **réduit** par la règle de production numéro p, c'est à dire qu'on remplace la chaîne en sommet de pile (qui correspond à la partie droite de la règle numéro p) par le non-terminal de la partie gauche de la règle de production, et on va dans l'état j qui dépend du non-terminal en question. On note ça rp
- soit on **accepte** le mot. Ce qui sera noté ACC
- soit c'est une **erreur**. Case vide

Construction de la table d'analyse : utilise aussi les ensembles SUIVANT (et donc PREMIER), plus ce qu'on appelle des fermetures de 0-items. Un 0-item (ou plus simplement *item*) est une production de la grammaire avec un "." quelque part dans la partie droite. Par exemple (sur la gram ETF) : $E \to E.+T$ ou encore $T \to F.$ ou encore $F \to .(E)$

Algorithme de construction des Fermetures d'un ensemble d'items I

1- Mettre chaque item de I dans Fermeture(I)
2- Pour chaque item i de Fermeture(I) de la forme $A \to \alpha.B\beta$
    pour chaque production $B \to \gamma$
        rajouter (s'il n'y est pas déja) l'item $B \to .\gamma$ dans Fermeture(I)
    finpour
  finpour
3- Recommencer 2 jusqu'à ce qu'on n'ajoute rien de nouveau

Algorithme de construction des Fermetures d'un ensemble d'items

1- Mettre chaque item de I dans Fermeture(I)
2- Pour chaque item i de Fermeture(I) de la forme $A \rightarrow \alpha.B\beta$
    pour chaque production $B \rightarrow \gamma$
        rajouter (s'il n'y est pas déja) l'item $B \rightarrow .\gamma$ dans Fermeture(I)
    finpour
finpour
3- Recommencer 2 jusqu'à ce qu'on n'ajoute rien de nouveau

Exemple

soit la grammaire ETF (des expressions arithmétiques)

$$\begin{cases} (1) \quad E \rightarrow E + T & (3) \quad T \rightarrow T * F & (5) \quad F \rightarrow (E) \\ (2) \quad E \rightarrow T & (4) \quad T \rightarrow F & (6) \quad F \rightarrow \text{nb} \end{cases}$$

soit l'ensemble d'items $\{T \rightarrow T * .F \ , \ E \rightarrow E. + T\}$.
La fermeture de cet ensemble d'items est :

$$\{T \rightarrow T * .F \ , \ E \rightarrow E. + T \ , \ F \rightarrow .\text{nb} \ , \ F \rightarrow .(E)\}$$

Analyse ascendante

Transition par X d'un ensemble d'items I

$\Delta(I, X) =$ Fermeture(tous les items $A \to \alpha X.\beta$) où $A \to \alpha.X\beta \in I$

Exemple

Soit l'ensemble d'items $I = \{T \to T * .F\ ,\ E \to E. + T\ ,\ F \to .\text{nb}\ ,\ F \to .(E)\}$, on aura
$\Delta(I, F) = \{T \to T * F.\}$
$\Delta(I, +) = \{E \to E + .T\ ,\ T \to .T * F\ ,\ T \to .F\ ,\ F \to .\text{nb}\ ,\ F \to .(E)\}$
etc.

Algorithme de construction des collections des items d'une grammaire

0- Rajouter l'axiome S' avec la production : $S' \to S$
1- $I_0 \leftarrow$ Fermeture($\{S' \to .S\}$)
   Mettre I_0 dans Collection
2- Pour chaque $I \in$ Collection
      Pour chaque X tq $\Delta(I, X)$ est non vide
            ajouter $\Delta(I, X)$ dans Collection
      finpour
   finpour
3- Recommencer 2 jusqu'à ce qu'on n'ajoute rien de nouveau

Analyse ascendante

Exemple

$I_0 = \{S \to .E, E \to .E+T, E \to .T, T \to .T*F, T \to .F, F \to .nb, F \to .(E)\}$
$\Delta(I_0, E) = \{S' \to E., E \to E.+T\} = I_1$ (terminal pour la règle $S' \to E$)
$\Delta(I_0, T) = \{E \to T., T \to T.*F\} = I_2$ (terminal pour la règle 2)
$\Delta(I_0, F) = \{T \to F.\} = I_3$ (terminal pour la règle 4)
$\Delta(I_0, () = \{F \to (.E), E \to .E+T, E \to .T, T \to .T*F, T \to .F, F \to .nb, F \to .(E)\} = I_4$
$\Delta(I_0, nb) = \{F \to nb.\} = I_5$ (terminal pour la règle 6)
$\Delta(I_1, +) = \{E \to E+.T, T \to .T*F, T \to .F, F \to .nb, F \to .(E)\} = I_6$
$\Delta(I_2, *) = \{T \to T*.F, F \to .nb, F \to .(E)\} = I_7$
$\Delta(I_4, E) = \{F \to (E.), E \to E.+T\} = I_8$
$\Delta(I_4, T) = \{E \to T., T \to T.*F\} = I_2$ déja vu, ouf
$\Delta(I_4, F) = \{T \to F.\} = I_3$
$\Delta(I_4, nb) = \{F \to nb.\} = I_5$
$\Delta(I_4, () = \{F \to (.E), E \to .E+T, E \to .T, T \to .T*F, T \to .F, F \to .nb, F \to .(E)\} = I_4$
$\Delta(I_6, T) = \{E \to E+T., T \to T.*F\} = I_9$ (terminal pour règle 1)
$\Delta(I_6, F) = I_3$
$\Delta(I_6, nb) = I_5$
$\Delta(I_6, () = I_4$
$\Delta(I_7, F) = \{T \to T*F.\} = I_{10}$ (terminal pour règle 3)
$\Delta(I_7, nb) = I_5$
$\Delta(I_7, () = I_4$
$\Delta(I_8,)) = \{F \to (E).\} = I_{11}$ (terminal pour règle 5)
$\Delta(I_8, +) = \{E \to E+.T, T \to .T*F, T \to .F, F \to .nb, F \to .(E)\} = I_6$
$\Delta(I_9, *) = I_7$
OUFFFFFF ...!

0- Rajouter l'axiome S' avec la production : $S' \to S$
1- $I_0 \leftarrow$ Fermeture$(\{S' \to .S\})$
  Metire I_0 dans Collection
2- Pour chaque $I \in$ Collection
    Pour chaque X tq $\Delta(I, X)$ est non vide
      ajouter $\Delta(I, X)$ dans Collection
    finpour
  finpour
3- Recommencer 2 jusqu'à ce qu'on n'ajoute rien de nouveau

Construction de la table d'analyse SLR

1- Construire la collection d'items $\{I_o,\ldots,I_n\}$

2- l'état i est contruit à partir de I_i :

a) pour chaque $\Delta(I_i,a)=I_j$: mettre **decaller j** dans la case $M[i,a]$

b) pour chaque $\Delta(I_i,A)=I_j$: mettre **aller en j** dans la case $M[i,A]$

c) pour chaque $A\rightarrow\alpha.$ (sauf $A=S'$) contenu dans I_i :

mettre **reduire** $A\rightarrow\alpha$ dans **chaque** case $M[i,a]$ où $a\in$SUIVANT(A)

d) si $S'\rightarrow S.\in I_i$: mettre **accepter** dans la case $M[i,\$]$

la table d'analyse LR de cette grammaire est

état	nb	+	*	(	)	$	E	T	F
0	d5			d4			1	2	3
1						ACC			
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									

	PREMIER	SUIVANT
E	nb (	$ +)
T	nb (	$ + *)
F	nb (	$ + *)

(1) $E\rightarrow E+T$

(2) $E\rightarrow T$

(3) $T\rightarrow T*F$

(4) $T\rightarrow F$

(5) $F\rightarrow (E)$

(6) $F\rightarrow$ nb

Construction de la table d'analyse SLR

1- Construire la collection d'items $\{I_o, \ldots, I_n\}$

2- l'état i est contruit à partir de I_i :

a) pour chaque $\Delta(I_i, a) = I_j$: mettre `decaller j` dans la case $M[i, a]$

b) pour chaque $\Delta(I_i, A) = I_j$: mettre `aller en j` dans la case $M[i, A]$

c) pour chaque $A \rightarrow \alpha.$ (sauf $A = S'$) contenu dans I_i :

mettre `reduire` $A \rightarrow \alpha$ dans **chaque** case $M[i, a]$ où $a \in$ SUIVANT(A)

d) si $S' \rightarrow S. \in I_i$: mettre `accepter` dans la case $M[i, \$]$

la table d'analyse LR de cette grammaire est

état	nb	+	*	(	)	$	E	T	F
0	d5			d4			1	2	3
1		d6				ACC			
2		r2	d7		r2	r2			
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			

	PREMIER	SUIVANT
E	nb (	$ +)
T	nb (	$ + *)
F	nb (	$ + *)

(1) $E \rightarrow E + T$
(2) $E \rightarrow T$
(3) $T \rightarrow T * F$
(4) $T \rightarrow F$
(5) $F \rightarrow (E)$
(6) $F \rightarrow nb$

Analyse ascendante

Principe d'analyse

Analyseur syntaxique : On part de l'état 0 et on empile et dépile non seulement les symboles (comme lors de l'analyseur LL) mais aussi les **états successifs.**

Exemple : l'analyse du mot $m = 3 + *4\$$

pile	entrée	action
$\$\boxed{0}$	$3 + *4\$$	d5
$\$\boxed{0}3\boxed{5}$	$+ *4\$$	r6 : $F \to$ nb

pile	entrée	action
$\$\boxed{0}$	$3 + *4\$$	d5
$\$\boxed{0}3\boxed{5}$	$+ *4\$$	r6 : $F \to$ nb
$\$\boxed{0}F$	$+ *4\$$	je suis en 0 avec F : je vais en 3
$\$\boxed{0}F\boxed{3}$	$+ *4\$$	r4 : $T \to F$
$\$\boxed{0}T$	$+ *4\$$	je suis en 0 avec T : je vais en 2
$\$\boxed{0}T\boxed{2}$	$+ *4\$$	r2 : $E \to T$
$\$\boxed{0}E$	$+ *4\$$	je suis en 0 avec E : je vais en 1
$\$\boxed{0}E\boxed{1}$	$+ *4\$$	d6
$\$\boxed{0}E\boxed{1}+\boxed{6}$	$*4\$$	ERREUR!! Ce mot n'appartient

Exemple : l'analyse du mot $m = 3 + 4 * 2\$$

pile	entrée	action
$\$\boxed{0}$	$3 + 4 * 2\$$	d5
$\$\boxed{0}3\boxed{5}$	$+4 * 2\$$	r6 : $F \to$ nb
$\$\boxed{0}F$	$+4 * 2\$$	je suis en 0 avec F : je vais en 3
$\$\boxed{0}F\boxed{3}$	$+4 * 2\$$	r4 : $T \to F$
$\$\boxed{0}T$	$+4 * 2\$$	je suis en 0 avec T : je vais en 2
$\$\boxed{0}T\boxed{2}$	$+4 * 2\$$	r2 : $E \to T$
$\$\boxed{0}E$	$+4 * 2\$$	je suis en 0 avec E : je vais en 1
$\$\boxed{0}E\boxed{1}$	$+4 * 2\$$	d6
$\$\boxed{0}E\boxed{1}+\boxed{6}$	$4 * 2\$$	d5
$\$\boxed{0}E\boxed{1}+\boxed{6}4\boxed{5}$	$*2\$$	r6 : $F \to$ nb
$\$\boxed{0}E\boxed{1}+\boxed{6}F$	$*2\$$	je suis en 6 avec F : je vais en 3
$\$\boxed{0}E\boxed{1}+\boxed{6}F\boxed{3}$	$*2\$$	r4 : $T \to F$
$\$\boxed{0}E\boxed{1}+\boxed{6}T$	$*2\$$	en 6 avec T : je vais en 9
$\$\boxed{0}E\boxed{1}+\boxed{6}T\boxed{9}$	$*2\$$	d7
$\$\boxed{0}E\boxed{1}+\boxed{6}T\boxed{9}*\boxed{7}$	$2\$$	d5
$\$\boxed{0}E\boxed{1}+\boxed{6}T\boxed{9}*\boxed{7}2\boxed{5}$	$\$$	r6 : $F \to$ nb
$\$\boxed{0}E\boxed{1}+\boxed{6}T\boxed{9}*\boxed{7}F$	$\$$	en 7 avec F : je vais en 10
$\$\boxed{0}E\boxed{1}+\boxed{6}T\boxed{9}*\boxed{7}F\boxed{10}$	$\$$	r3 : $T \to T * F$
$\$\boxed{0}E\boxed{1}+\boxed{6}T$	$\$$	en 6 avec T : je vais en 9
$\$\boxed{0}E\boxed{1}+\boxed{6}T\boxed{9}$	$\$$	r1 : $E \to E + T$
$\$\boxed{0}E$	$\$$	en 0 avec E : je vais en 1
$\$\boxed{0}E\boxed{1}$	$\$$	ACCEPTÉ!!!

Analyse ascendante

Principe d'analyse

Analyseur syntaxique : On part de l'état 0 et on empile et dépile non seulement les symboles (comme lors de l'analyseur LL) mais aussi les **états successifs.**

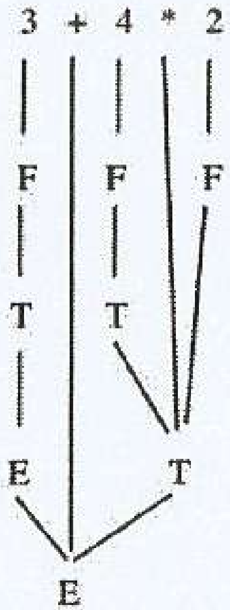
Exemple : l'analyse du mot $m = 3 + *4\$$

pile	entrée	action
$\$\boxed{0}$	$3 + *4\$$	d5
$\$\boxed{0}3\boxed{5}$	$+ * 4\$$	r6 : $F \rightarrow$ nb

état	nb	+	*	(	)	\$	E	T	F
0	d5			d4			1	2	3
1		d6				ACC			
2		r2	d7		r2	r2			
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			

Exemple : l'analyse du mot $m = 3 + 4 * 2\$$

pile	entrée	action
$\$\boxed{0}$	$3 + 4 * 2\$$	d5
$\$\boxed{0}3\boxed{5}$	$+ 4 * 2\$$	r6 : $F \rightarrow$ nb



Exercice

Soit la grammaire G définie par:

P → a R | eps
R → b a R | eps

1. Trouver V_T et V_N
2. Définir grammaire augmentée G' de G.
3. Soit I_0= Fermeture (S'→ .P) où S' est le nouveau axiome
   1. Calculer I_0
   2. Trouver les: Transition (I_0,P) et Transition(I_0,R)??

Soit la table d'analyse (SLR) suivante:

	a	b	$	P	R
0					
1					
2					
3					
4					
5					
6					

4. Compléter la table ci-joint.
5. Analyser les mots aba, abb? Que pouvez vous déduire?
6. Trouver l'arbre de dérivation s'il existe?