

lexicale suivante.

Entcour est une variable globale désignant l'entité lexicale courante.

exemple:

L'analyse de la chaîne

id1+id2\*id3

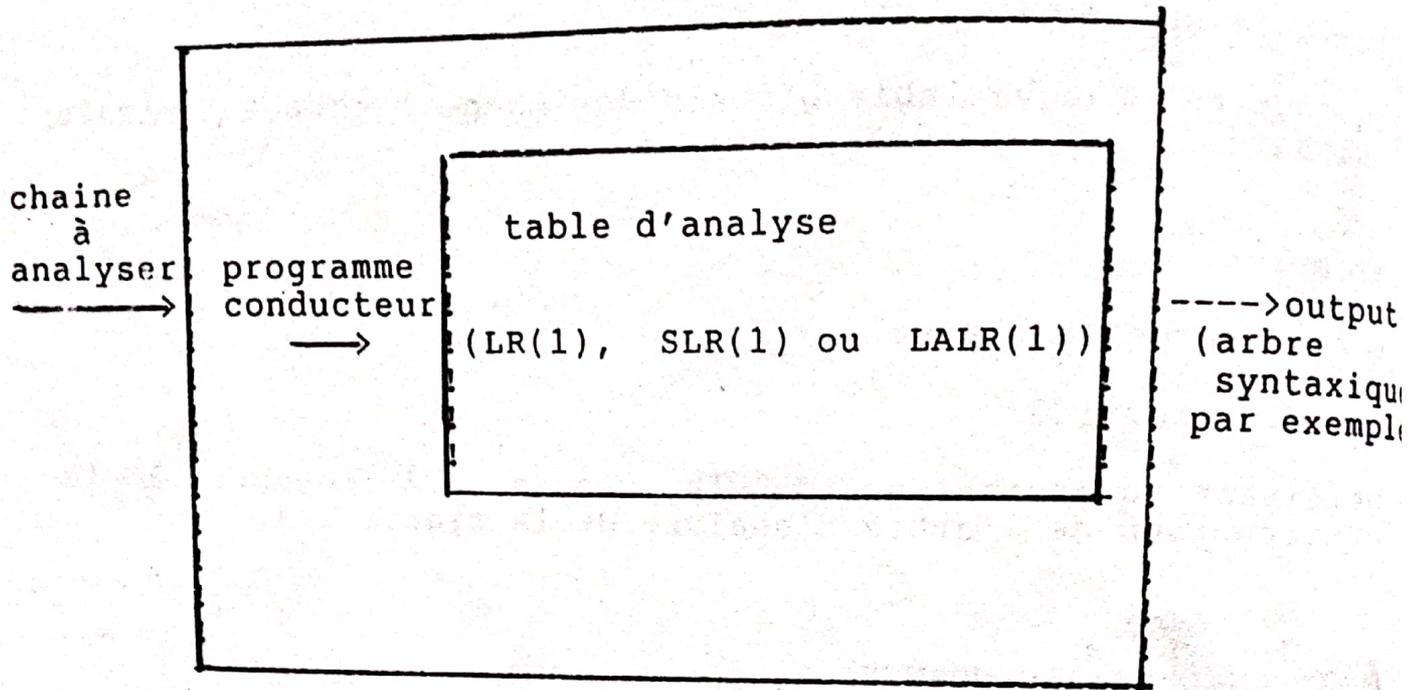
utilisant la grammaire G' de la figure 3.2 conduit à la construction de l'arbre d'analyse de la figure 3.3.

### 3.3 ANALYSE ASCENDANTE.

Nous étudions ici trois méthodes d'analyse ascendante qui sont la SLR(1), la LR(1) et la LALR(1). A travers cette étude, nous verrons les modes d'application de chacune de ces méthodes.

Toutes les trois utilisent un programme qui simule un automate représenté par une table. Ce programme appelé conducteur (ou introducteur) est pratiquement identique dans les trois cas. Ce qui différencie les trois analyses, c'est le contenu de la table d'analyse qui particularise le type d'analyse.

Le schéma suivant explique le fonctionnement de ces analyses.



programme conducteur.

Il utilise une pile dont le contenu spécifie l'état de l'analyse et un vecteur contenant le reste de la chaîne à analyser. Au départ, la configuration de l'analyse est la suivante:

(s0	,	a1a2.....am#)
-----		-----
etat		chaîne à analyser
initial		(vecteur)
(pile)		

Au cours de l'analyse, la configuration varie et devient :

(s0 x1 x2.....xp-1 sp , ai ai+1.....am )

Pour passer à la configuration suivante, l'analyseur consulte la table d'analyse. La table fait apparaître deux parties. Une partie ACTION et une partie GOTO.

Les types d'action que le simulateur peut rencontrer dans la table sont:

- Décaler ou empiler j (noté dj)
- Réduire j (noté rj)
- Accepter (noté acc)
- Erreur: une erreur est indiquée par une case vide

Les lignes de la table sont indicées par les états de l'automate construits à partir de la grammaire et les colonnes par les symboles de la grammaire et le symbole #.

La case (sp,ai) indique l'action à entreprendre:

Si l'action est:

1-dj, l'analyseur doit empiler l'élément ai ensuite j qui est le numéro de l'état suivant par la transition ai j=goto(sp,ai). (pour ne pas trop encombrer la table, la fonction goto pour les terminaux apparaît dans l'action de décalage.)

2-rj, j indique un numéro de règle de la grammaire. A ce moment la partie droite de cette production apparaît au sommet de la pile. Il faut substituer à cette partie droite, la partie gauche de la règle. On dit qu'on a procédé à une réduction.

A la suite de cette opération, l'analyseur empile le numéro de l'état de transition trouvé dans la deuxième partie de la table c'est à dire

k = goto (spr,A)

où A est le nonterminal apparaissant au sommet de la pile et spr se trouvant juste avant A dans la pile.

3-acc, l'analyseur est arrivé à une situation de succès.

4-une case vide, l'analyseur a abouti à un échec, il passe le contrôle à la routine faisant le diagnostic des erreurs pour la signalisation de cette erreur.

## ANALYSE SYNTAXIQUE

## 3.3.1 Analyse SLR(1).(simple LR)

Dans cette analyse, nous construisons un automate d'états finis déterministe spécifiant les différents états d'avancement de l'analyse.

Un état de l'automate contient un ensemble d'éléments décrivant ce que l'analyse anticipe de rencontrer à partir de l'instant où l'analyse transite vers cet état. Un élément de cet ensemble est appelé item LR(0). Une transition de l'automate par un symbole de grammaire fournit l'état dans lequel serait l'analyse si elle arrive à reconnaître ultérieurement ce symbole.

Une fois l'automate élaboré, nous procédons à la construction de la table. Ce qui caractérise cette analyse des deux autres est que cette analyse tient compte du contexte uniquement lorsqu'elle suspecte un conflit d'actions.

## 3.3.1.1 Item LR(0). -

Soit  $A \rightarrow BCD$  une production de la grammaire du langage. Les différents items LR(0) de cette règle sont:

$A \rightarrow \cdot BCD$

$A \rightarrow B \cdot CD$

$A \rightarrow BC \cdot D$

et  $A \rightarrow BCD \cdot$ .

A travers cet exemple, nous remarquons qu'un item LR(0) est une règle de production dont la partie droite fait figurer un point.

si  $A \rightarrow \cdot$ , le seul item LR(0) de cette règle est :  $A \rightarrow \cdot$ .

## 3.3.1.2 Détermination Des Ensembles Des Items LR(0). -

pour ce faire, nous avons besoin de définir deux notions qui sont la fermeture ou la fermeture d'un item LR(0) et la fonction GOTO.

closure d'un item LR(0).

La closure d'un item  $[A \rightarrow \alpha \cdot \beta]$  est un ensemble d'items LR(0) déterminé comme suit:

- 1- L'item LR(0)  $A \rightarrow \alpha \cdot \beta$  appartient à la closure
- 2- Si  $[X \rightarrow \alpha \cdot \beta]$  est un item LR(0) de la closure, si  $\beta = B\alpha$  où B est un nonterminal et si  $B \rightarrow \lambda$  est une règle de la grammaire alors l'item LR(0)  $[B \rightarrow \cdot \lambda]$  doit être ajouté à cette closure.
- 3- Répéter le pas 2 jusqu'à ce qu'aucun nouvel item LR(0) ne soit créé et ajouté à l'ensemble.

fonction GOTO.

$GOTO(I_i, X)$  existe si dans l'ensemble  $I_i$ , il existe un item LR(0) ayant le format  $[A \rightarrow \alpha \cdot X \beta]$ . Dans ce cas

$$GOTO(I_i, X) = \text{closure}([A \rightarrow \alpha \cdot X \beta]).$$

exemple:

soit  $G_1$  la grammaire

$$E \rightarrow E + T \quad (1)$$

$$/ T \quad (2)$$

$$T \rightarrow T * F \quad (3)$$

$$/ F \quad (4)$$

$$F \rightarrow (E) \quad (5)$$

$$/ id \quad (6)$$

En augmentant la grammaire, nous obtenons la grammaire

suivante:

$$E' \rightarrow E$$

$$E \rightarrow E + T$$

$$/ T$$

$$T \rightarrow T * F$$

$$/ F$$

$$F \rightarrow (E)$$

$$/ id$$

L'état initial de l'automate d'analyse sera la closure de l'item LR(0)  $[E' \rightarrow \cdot E]$ . Les autres états se déduiront à partir de ce dernier à l'aide de la fonction GOTO. Les ensembles des item LR(0) sont les suivants:

I<sub>0</sub> :  $[E' \rightarrow \cdot E]$   
 $[E \rightarrow \cdot E + T]$   
 $[E \rightarrow \cdot T]$   
 $[T \rightarrow \cdot T * F]$   
 $[T \rightarrow \cdot F]$   
 $[F \rightarrow \cdot (E)]$   
 $[F \rightarrow \cdot id]$

I<sub>1</sub>: GOTO(I<sub>0</sub>, E)  
 $[E' \rightarrow E \cdot]$   
 $[E \rightarrow E \cdot + T]$

I<sub>2</sub>: GOTO(I<sub>0</sub>, T)  
 $[E \rightarrow T \cdot]$   
 $[T \rightarrow T \cdot * F]$

I<sub>3</sub>: GOTO(I<sub>0</sub>, F)  
 $[T \rightarrow F \cdot]$

I<sub>4</sub>: GOTO(I<sub>0</sub>, ( )  
 $[F \rightarrow ( \cdot E)]$   
 $[E \rightarrow \cdot E + T]$

```
[E-->.T]
[T-->.T*F]
[T-->.F]
[F-->.(E)]
[F-->.id]
```

```
I5: GOTO (I0, id )
    [F-->id.]
```

```
I6: GOTO(I1,+)
    [E-->E+.T]
    [T-->.T*F]
    [T-->.F]
    [F-->.(E)]
    [F-->.id]
```

```
I7: GOTO (I2,*)
    [T-->T*.F]
    [F-->.(E)]
    [F-->.id]
```

```
I8: GOTO(I4,E)
    [F-->(E.)]
    [E-->E.+T]
```

```
I9: GOTO(I6,T)
    [E-->E+T.]
    [T-->T.*F]
```

```
GOTO(I4,T) = I2
GOTO(I4,F) = I3
GOTO(I4,()) = I4
GOTO(I4,id) = I5
GOTO(I6,F) = I3
GOTO(I6,()) = I4
GOTO (I6,id) = I5
```

```
I10: GOTO (I7,F)
    [T-->T*F.]
```

## ANALYSE SYNTAXIQUE

```
I11:GOTO (I8,))
      [F-->(E).]
```

```
GOTO(I7,()) = I4
GOTO(I7,id) = I5
GOTO(I8,+) = I6
GOTO(I9,*) = I7
```

Ce que nous venons de faire ici c'est calculer toutes les fonctions GOTO(Ii,x) possibles et de nommer de façon unique les ensembles d'items LR(0) déterminés. Nous pouvons schématiser ces ensembles d'items LR(0) (états de l'automate) et la fonction GOTO (arcs de transition) à l'aide d'un automate d'états finis déterministe. La figure 3.4 montre cet automate

### 3.3.1.3 Construction De La Table D'analyse SLR(1). -

La table d'analyse SLR(1) est composée comme mentionné auparavant de deux parties qui sont action et GOTO. Pour déterminer les actions et les goto, nous devons consulter les ensembles d'items LR(0) item par item et appliquer les règles suivantes :

1- Si dans Ii, il existe un item LR(0)  $[A \rightarrow \alpha a \beta]$  où a est un terminal et si  $GOTO(Ii, a) = Ij$  alors mettre dans la case (i,a) de la table l'action dj.

2- Si dans Ii, il existe l'item LR(0)  $[S' \rightarrow S.]$  alors mettre dans la case (i,#) de la table l'action accepter.

3- Si dans Ii, il existe un item LR(0)  $[a \rightarrow \alpha.]$  alors pour tout symbole appartenant à SUIVANT(A) mettre dans la case (i,a) de la table, l'action rj où j est le numéro de la règle  $A \rightarrow \alpha$  de la grammaire. (Nous numérotions les règles de la grammaire afin de ne pas trainer les règles dans la table)

4- Si dans Ii, il existe un item LR(0)  $[A \rightarrow \alpha X \beta]$  où X est un nonterminal et si  $GOTO(Ii, X) = Ij$  alors mettre dans la case (i,X) de la table le numéro j.

5- Toutes les cases vides qui restent après le parcours de tous les items LR(0) indiquent l'existence d'une erreur.

La table d'analyse SLR(1) de la grammaire  $G'$  est montrée sur la figure 3.5.

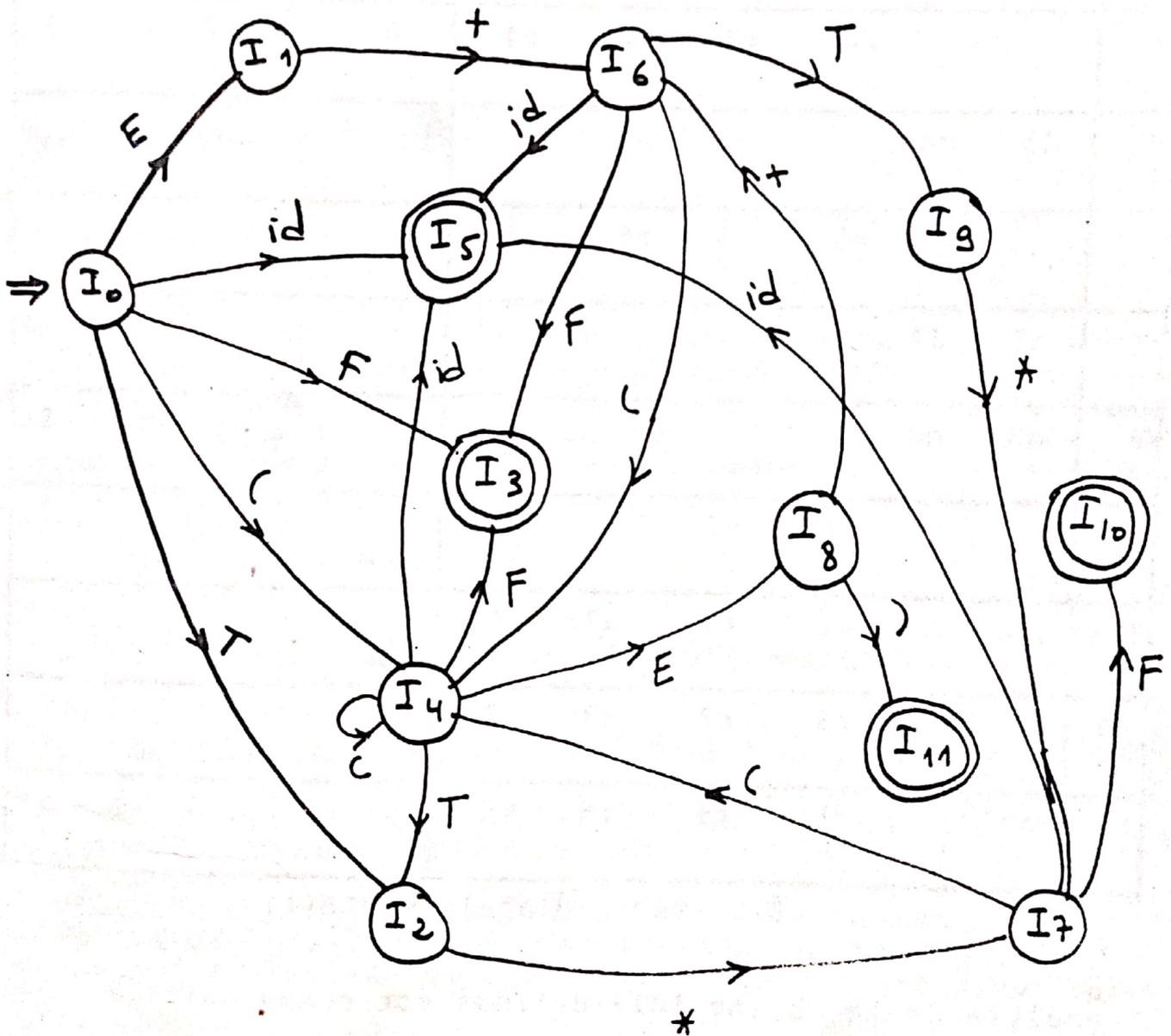


figure 3.4: Automate d'analyse SLR(1).

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

Figure 3.5: Table d'analyse SLR(1).

L'analyse de la chaîne  $id1+id2*id3\#$  est comme suit:

0

 $id1+id2*id3\#$

0id15	+id2*id3	#
0F3	+id2*id3	#
0T2	+id2*id3	#
0E1	+id2*id3	#
0E1+6	id2*id3	#
0E1+6id25	*id3	#
0E1+6F3	*id3	#
0E1+6T9	*id3	#
0E1+6T9*7	id3	#
0E1+6T9*7id35		#
0E1+6T9*7F10		#
0E1+6T9		#
0E1		#
acc		

### 3.3.2 Analyse LR(1) (left To Right).

Le fonctionnement de l'analyse LR(1) est similaire à celui de l'analyse SLR(1).

La nuance qui existe entre les deux approches réside dans le fait que l'analyse LR(1) considère le contexte ou la première entité lexicale de la chaîne à analyser pour la construction des ensembles des items et donc de la table LR(1). de ce fait, cette analyse est plus efficace et accepte une plus grande classe de grammaires. (l'analyse LR(k) effectue une prélecture des k premiers tokens de la chaîne d'entrée).

#### 3.3.2.1 Construction Des Ensembles D'items LR(1). -

Un item LR(1) a la forme  $[A \rightarrow \alpha \cdot \beta, a]$  où  $A \rightarrow \alpha \beta$  est une règle de la grammaire et 'a' un terminal. a est appelé lookahead ou entité de prélecture de l'item LR(1), '1' indiquant la longueur du lookahead.

La notion de closure n'est plus la même dans ce cas puisque l'item LR(1) possède une information en plus qui est le lookahead. Il revient donc de définir la closure d'un item LR(1).

definition: