

$$6.25 + 5,5 + 3.75 + 3 = 18.5$$

NOM: NIGET
PRÉNOM: Tom

Compilation

mercredi 20 octobre 2021

Question 1:

Pourquoi utilise t'on des techniques d'analyse différentes pour les analyseurs lexicaux et syntaxiques?

Car dans une phrase les lexèmes sont des éléments atomiques, indépendants, alors que les syntagmes sont par essence récursifs. L'analyse lexicale produit une liste de jetons, alors que l'analyse syntaxique produit un arbre. Ce sont deux structures radicalement différentes qui nécessitent des méthodes d'analyse différentes.

et qui relèvent des ~~deux domaines~~ ~~de~~ ~~langages~~ ~~de~~ ~~langages~~ ≠ (rationnel vs algébrique)

Question 2:

Soit la grammaire G_1 suivante:

$G \rightarrow G E \text{ ' ; ' } \mid \epsilon$

$E \rightarrow \text{id} \mid \text{id} [L] \mid \text{id} = E \mid \text{int}$

$L \rightarrow L , E \mid E$

(r1, r2)
(r3, r4, r5, r6)
(r7, r8)

où id, int, ' ; ', '['] ', ', ' et '=' dénotent des terminaux.

Calculer les premiers et les suivants de G, E et L.

PREM(G) = id, int, ε ✓ SUIV(G) = id, int, \$ ✓
PREM(E) = id, int ✓ SUIV(E) = '[', ']', '=', 'int', 'ε' ✓
PREM(L) = id, int ✓ SUIV(L) = '[', ']' ✓

Question 3:

Construire la table d'analyse LL de cette grammaire (remplissez la table avec les numéros de règles).

	id	int	;	[]	,	=	\$
G	r1 + r2	r1 + r2						r2
E	r3, r4, r5	r6						
L	r8, r7	r8, r7						

Pourquoi cette grammaire n'est pas LL(1)?

Il y a un conflit pour (E, id) : il faudrait factoriser à gauche.

ou

Également, il y a une récursivité à gauche sur G qui cause aussi des conflits.

Question 4:

Construire la table LR(0) de la grammaire G_1 (il y a ici plus de lignes que nécessaire):

état	id	int	;	[]	,	=	\$	G	E	L
0	s3	s4	r2	r2	r2	r2	r2		1		
1	s3	s4						accept		5	OK
2	r2	r2	r2	r2	r2	r2	r2	r2			
3	r3	r3	r3	r3	r3	r3	r3	r3			
4	r6	r6	r6	r6	r6	r6	r6	r6			
5			s8								
6	s3	s4								10	9
7	s3	s4								11	OK
8	r1	r1	r1	r1	r1	r1	r1	r1			
9					s12						
10	r8	r8	r8	r8	r8	r8	r8	r8			
11	r5	r5	r5	r5	r5	r5	r5	r5			
12	r4	r4	r4	r4	r4	r4	r4	r4			
13	s3	s4								14	OK
14	r7	r7	r7	r7	r7	r7	r7	r7			
15											
16											
17											
18											
19											

5.5

Qu'en concluez vous?

pas grave -
 qu'il y a une récursion droite et donc qu'il devrait y avoir un conflit
 shift-reduce, mais que j'ai fait une erreur quelque part
 → en fait j'ai creuvé c'était à l'état 3. → PAS LR(0)

Question 5:

Construire la table SLR(1) de la grammaire G_1 :

- Mettez ici seulement les lignes différentes de la table précédente;
- N'oubliez pas d'indiquer les numéros d'états.

Là encore, il y a plus de lignes que nécessaire.

état	id	int	;	[]	,	=	\$	G	E	L
3	r3	r3	r3	s6	r3	r3	s7				

Qu'en concluez vous?

La grammaire G_1 est SLR(1).

OK

Question 6:

On réécrit la grammaire G_1 pour essayer de la rendre LL(1). Pour cela, on lui enlève les récursivités gauches et on la factorise. On en profite aussi pour y rajouter les appels de fonctions. On arrive à la grammaire G_2 suivante:

$G \rightarrow E \text{ ; } G \mid \epsilon$	(r1, r2)
$E \rightarrow \text{id } E_p \mid \text{int}$	(r3, r4)
$E_p \rightarrow ' (' P ') ' \mid ' [' L '] ' \mid ' = ' E \mid \epsilon$	(r5, r6, r7, r8)
$P \rightarrow L \mid \epsilon$	(r9, r10)
$L \rightarrow E L_p$	(r11)
$L_p \rightarrow ' , ' E L_p \mid \epsilon$	(r12, r13)

Calculer les premiers et les suivants de G , E , E_p , P , L et L_p .

PREM(G) = id, int, ε

SUIV(G) = \$

PREM(E) = id, int

SUIV(E) = ' (' , ' [' , ' = ' , ε

PREM(E_p) = ' (' , ' [' , ' = ' , ε

SUIV(E_p) = ' (' , ' [' , ' = ' , ε

PREM(P) = id, int + ε

SUIV(P) = ε

PREM(L) = id, int

SUIV(L) = ' (' , ' [' , ' = ' , ε

PREM(L_p) = ε

SUIV(L_p) = ' (' , ' [' , ' = ' , ε

Question 7:

Construire la table d'analyse LL de cette grammaire.

	id	int	;	[]	()	,	=	\$
G	r1	r1								r2
E	r3	r4								
Ep			r8	r6	r8	r5	r8	r8	r7	
P	r9	r3					r10			
L	r11	r11								
Lp					r13		r13	r12		

Est-ce que, finalement, la grammaire G_2 est LL(1)?

Oui, il n'y a aucun conflit.