DEUG MIAS2. Institut Galilée
Option : Langages formels et compilation. 1999–2000
TD : Feuille 4.

Des exercices sont énoncés au verso de cette feuille.

La grammaire G_1.

Alphabets : $Vt = \{+, *, (,), \mathbf{id}\}$ et $Vn = \{E, T, F\}$. Axiome : E.

Productions et une table d'analyse LR pour G_1.

1 : $E \longrightarrow E + T$
2 : $E \longrightarrow T$
3 : $T \longrightarrow T * F$
4 : $T \longrightarrow F$
5 : $F \longrightarrow (E)$
6 : $F \longrightarrow \mathbf{id}$

	\$	$+$	$*$	(	)	**id**	E	T	F
0				4		5	1	2	3
1	Acc	6							
2	$r2$	$r2$	7		$r2$				
3	$r4$	$r4$	$r4$		$r4$				
4				4		5	8	2	3
5	$r6$	$r6$	$r6$		$r6$				
6				4		5		9	3
7				4		5			10
8		6			11				
9	$r1$	$r1$	7		$r1$				
10	$r3$	$r3$	$r3$		$r3$				
11	$r5$	$r5$	$r5$		$r5$				

Exemple d'analyse dans G_1, avec la table précédente : $\mathbf{id} + \mathbf{id} * \mathbf{id}$

Pile	Pile d'états	Entrée	Actions
ε	0	$\mathbf{id} + \mathbf{id} * \mathbf{id}$ \$	(5 : lecture de **id**)
id	0 5	$+ \mathbf{id} * \mathbf{id}$ \$	$r6 : F \longrightarrow \mathbf{id}$
F	0 3	$+ \mathbf{id} * \mathbf{id}$ \$	$r4 : T \longrightarrow F$
T	0 2	$+ \mathbf{id} * \mathbf{id}$ \$	$r2 : E \longrightarrow T$
E	0 1	$+ \mathbf{id} * \mathbf{id}$ \$	(6 : lecture de $+$)
$E+$	0 1 6	$\mathbf{id} * \mathbf{id}$ \$	(5 : lecture de **id**)
$E + \mathbf{id}$	0 1 6 5	$* \mathbf{id}$ \$	$r6 : F \longrightarrow \mathbf{id}$
$E + F$	0 1 6 3	$* \mathbf{id}$ \$	$r4 : T \longrightarrow F$
$E + T$	0 1 6 9	$* \mathbf{id}$ \$	(7 : lecture de $*$)
$E + T*$	0 1 6 9 7	**id** \$	(5 : lecture de **id**)
$E + T * \mathbf{id}$	0 1 6 9 7 5	\$	$r6 : F \longrightarrow \mathbf{id}$
$E + T * F$	0 1 6 9 7 10	\$	$r3 : T \longrightarrow T * F$
$E + T$	0 1 6 9	\$	$r1 : E \longrightarrow E + T$
E	0 1	\$	*Acc*

La grammaire G_2.

Alphabets : $Vt = \{\texttt{=}, *, \mathbf{id}\}$ et $Vn = \{S, G, D\}$. Axiome : S.

Productions et une table d'analyse LR pour G_2.

1 : $S \longrightarrow G\texttt{=}D$
2 : $S \longrightarrow D$
3 : $G \longrightarrow *D$
4 : $G \longrightarrow \mathbf{id}$
5 : $D \longrightarrow G$

	\$	=	$*$	**id**	S	G	D
0			4	5	1	2	3
1	*Acc*						
2	$r5$	6					
3	$r2$						
4			4	5		8	7
5	$r4$	$r4$					
6			4	5		8	9
7	$r3$	$r3$					
8	$r5$	$r5$					
9	$r1$	$r1$					

Comment lire une table d'analyse.

- la première colonne contient les numéros des états d'un automate,
- la première ligne contient le symbole \$ et tous les symboles de la grammaire : ces derniers constituent le ¡¡vocabulaire¿¿ de l'automate,
- dans la table elle–même, les numéros isolés définissent les transitions d'un automate fini :
  - pour les symboles terminaux, une transition correspond à une lecture (d'où l'expression `shift` (décalage) utilisée par Yacc dans ses tables),
  - pour tout non terminal X, les transitions indiquées doivent s'effectuer après toute ¡¡réduction¿¿ sur une règle relative à X (Yacc utilise l'expression `goto`),
- l'entrée de cet automate est numérotée 0 (les sorties ne jouent aucun rle ici et ne sont donc pas mentionnées),
- les autres informations portées dans la table sont les ¡¡actions¿¿ suivantes :
  - *Acc* : accepter (fin d'analyse d'une phrase ¡¡correcte¿¿),
  - ri : réduire par la règle numéro i,
  - les cases vides correspondent à des erreurs de syntaxe (les `error` de Yacc).

Exercices à faire sur chacune de ces grammaires.

- Tracer le diagramme de transition de l'automate défini par la table.
- Comment faire des diagnostics d'erreur ? reporter ces diagnostics sur la table d'analyse.
- Faire l'analyse de phrases écrites dans le vocabulaire (terminal) de la grammaire considérée.
- Proposer des actions sémantiques et les exécuter sur les analyses précédentes.
- Ecrire un programme Yacc capable de produire un analyseur syntaxique exécutant les actions sémantiques.