

# UE 4TIN603U – Compilation – Licence 3 – 2018-2019

## TD4 - Grammaires non LR – Premiers calculs sémantiques

### 1. Analyse LR

Soit la grammaire augmentée suivante :

- 1)  $E' \rightarrow E$
- 2)  $E \rightarrow E + B$
- 3)  $E \rightarrow E * B$
- 4)  $E \rightarrow B$
- 5)  $B \rightarrow \text{nb}$

La construction de la table d'analyse par la méthode SLR donne ceci :

État	Action					
	*	+	nb	#	E	B
0			S3		G1	G2
1	S5	S4		R1 & OK		
2	R4	R4		R4		
3	R5	R5		R5		
4			S3			G6
5			S3			G7
6	R2	R2		R2		
7	R3	R3		R3		

FIGURE 1 – Table d'analyse SLR

Où :

- $S_i$  signifie décalage (*shift*) et aller à l'état  $i$
- $G_i$  signifie aller à l'état  $i$
- $R_i$  signifie réduire (*reduce*) par la règle  $i$

- (a) Construire les ensembles d'items qui ont permis la construction de la table 1
- (b) Analyser le mot  $\text{nb} + \text{nb} * \text{nb} \#$
- (c) Construire l'arbre d'analyse étant donné l'analyse obtenue
- (d) Que doit-on conclure de la grammaire en examinant cet arbre ?

## 2. Grammaire étendue des expressions

Soit les expressions arithmétiques faites d'opérateurs habituels ( $+$ ,  $\times$ ,  $-$ ,  $/$ ), les expressions de comparaison ( $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $=$ ,  $\neq$ ) et les expressions logiques ( $\vee$ ,  $\wedge$ ,  $\neg$ ).

- (a) Écrire une grammaire n'utilisant que trois symboles `exprArithm`, `exprComp` et `exprLog` pour rassembler toutes ces expressions en y ajoutant les constantes et les notations parenthésées.

Compiler l'analyseur syntaxique et observer les informations données par `Beaver`. Les corrections apportées automatiquement sont-elles satisfaisantes ?

- (b) Réécrire la même grammaire en tenant compte des propriétés suivantes de l'algèbre :

- Les multiplications sont prioritaires sur les additions.  $(a + b \times c = a + (b \times c))$
- Les conjonctions sont prioritaires sur les disjonctions.  $(a \vee b \wedge c = a \vee (b \wedge c))$
- Les opérateurs unaires sont prioritaires sur les opérateurs binaires  $(a + -b = a + (-b))$ .
- Tous les opérateurs binaires sont associatifs à gauche.  $(a + b + c = (a + b) + c)$
- Les opérateurs unaires sont associatifs à droite.  $(- - -a = -(-(-a)))$
- Les opérandes finales des opérations logiques sont des expressions de comparaison ou des constantes logiques.
- Les opérandes finales des opérations de comparaison sont des expressions arithmétiques ou des nombres.
- Les opérandes finales des opérations arithmétiques sont des nombres

- (c) Écrire une classe qui étend `beaver.Symbol` pour chacune des trois expressions.
- (d) Y implémenter les méthodes permettant les calculs sur ces expressions.
- (e) Écrire un analyseur syntaxique qui prend n'importe quelle expression (logique, de comparaison, arithmétique) et affiche le résultat en sortie.