

# TP de compilation : Analyse syntaxique et évaluation d'une expression arithmétique

A partir de la grammaire ambiguë G1,

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

Ce tp a pour but de vous faire écrire un analyseur ascendant. Pour cela, nous avons préalablement construit la table d'analyse LALR(1)

successeurs ( $E'$ ) = { \$ }

successeurs ( $E$ ) = { +, \*, ), \$ }

<u>Etat</u>	<u>terminaux</u>							<u>non terminaux</u>
	val	+	*	(	)	\$		E
0	d3			d2				1
1		d4	d5			accepter		
2	d3			d2				6
3		r5	r5		r5	r5		
4	d3			d2				7
5	d3			d2				8
6		d4	d5		d9			
7		d4 r2	d5 r2		r2	r2		
8		d4 r3	d5 r3		r3	r3		
9		r4	r4		r4	r4		

Les conflits qui demeurent présents dans cette table d'analyse seront résolus de la manière suivante :

- L'associativité à gauche pour + permet de lever le conflit d4/r2 de l'état 7 en faveur de r2.
- L'associativité à gauche pour \* permet de lever le conflit d5/r3 de l'état 8 en faveur de r3.
- La priorité de \* par rapport à + privilégie d5 dans le conflit d5/r2 de l'état 7 et r3 dans le conflit d4/r3 de l'état 8.

1. Implémenter l'automate LALR grâce au design pattern State et aux structures de données décrites dans le polycopié de cours.
2. Construire et Interpréter l'expression : il s'agira d'évaluer l'expression à l'intérieur du programme d'analyse, et d'en afficher la valeur calculée.
3. Votre programme sera testé par l'équipe enseignante.

# Annexe

