

TP (Grammaires et langages) : Syntactic analysis and evaluation of an arithmetic expression

One aim to make you write a bottom-up parser From the ambiguous grammar G1 below. For this purpose, we have previously constructed the analysis table LALR(1),

G1:

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

and the sets of following symbols for each non-terminal appearing in the left part of each rule

successeurs (E') = { \$ }

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

State	Terminals						Non terminals
	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	

Conflicts that remain present in this analysis table will be resolved as follows:

- Left associativity for + makes it possible to remove the d4/r2 conflict from state 7 in favor of r2.
- Left associativity for * allows the d5/r3 conflict to be lifted from state 8 in favor of r3.
- The priority of * compared to + favors d5 in the d5/r2 conflict of state 7 and r3 in the d4/r3 conflict of state 8.

1. Implement the LALR automaton using the State design pattern and the data structures described in the course handout.
2. Construct and Interpret the expression: this will involve evaluating the expression within the analysis program, and displaying the calculated value.
3. Your program will be tested by the teaching team.

Annexe

