LR(1) parser, also known as *Shift-Reduce* parser.

$$\begin{aligned} E &\longrightarrow E+T \mid E-T \mid T \\ T &\longrightarrow T*F \mid T\,/\,F \mid F \\ F &\longrightarrow (E) \mid n \end{aligned}$$

let n be any positive integer.

LR(1) parsing table: ($ is the end of the expression)

	n	$+$	$-$	$*$	$/$	(	)	$	E	T	F
0	shift 5					shift 4			1	2	3
1		shift 6	shift 6					accept			
2		$E \to T$	$E \to T$	shift 7	shift 7		$E \to T$	$E \to T$			
3		$T \to F$	$T \to F$	$T \to F$	$T \to F$		$T \to F$	$T \to F$			
4	shift 5					shift 4			8	2	3
5		$F \to \mathrm{n}$	$F \to \mathrm{n}$	$F \to \mathrm{n}$	$F \to \mathrm{n}$		$F \to \mathrm{n}$	$F \to \mathrm{n}$			
6	shift 5					shift 4				9	3
7	shift 5					shift 4					10
8		shift 6	shift 6				shift 11				
9		$E \to E+T$ $E \to E-T$	$E \to E+T$ $E \to E-T$	shift 7	shift 7		$E \to E+T$ $E \to E-T$	$E \to E+T$ $E \to E-T$			
10		$T \to T*F$ $T \to T\,/\,F$	$T \to T*F$ $T \to T\,/\,F$	$T \to T*F$ $T \to T\,/\,F$	$T \to T*F$ $T \to T\,/\,F$		$T \to T*F$ $T \to T\,/\,F$	$T \to T*F$ $T \to T\,/\,F$			
11		$F \to (E)$	$F \to (E)$	$F \to (E)$	$F \to (E)$		$F \to (E)$	$F \to (E)$			

Note: If the entry contains two reduction rules, which one should be used will be determined by the terminal symbol popped from the stack.