

First set:

	Expr	Expr'	Term	Term'	Factor
First	(, id, num	+, -, ε	(, id, num	*, /, ε	(, id, num

Follow set:

Expr	Expr'	Term	Term'	Factor
eof,)	eof,)	+, -, eof,)	+, -, eof,)	*, /, +, -, eof,)

Algorithm:

1. For each production $A \rightarrow \alpha$ of the grammar, perform steps 2, 3 and 4.
2. For each terminal a in $\text{First}(a)$, add the production $A \rightarrow \alpha$ to $M[A, a]$
3. If ϵ is in $\text{First}(\alpha)$, add $A \rightarrow \epsilon$ to $M[A, b]$ for each terminal b in $\text{Follow}(A)$.
4. If ϵ is in $\text{First}(\alpha)$ and **eof** is in $\text{Follow}(A)$, add $A \rightarrow \alpha$ to $M[A, \text{eof}]$.
5. Make each undefined entry be an **error**.

Grammar:

Expr → **Term Expr'**
Expr' → **+ Term Expr' | - Term Expr' | ε**
Term → **Factor Term'**
Term' → *** Factor Term' | / Factor Term' | ε**
Factor → **(Expr) | num | id**

Parse table:

	eof	+	-	*	/	()	id	num
Expr						$E \rightarrow TE'$		$E \rightarrow TE'$	$E \rightarrow TE'$
Expr'	$E' \rightarrow \epsilon$	$E' \rightarrow +TE'$	$E' \rightarrow -TE'$				$E' \rightarrow \epsilon$		
Term						$T \rightarrow FT'$		$T \rightarrow FT'$	$T \rightarrow FT'$
Term'	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$	$T' \rightarrow /FT'$		$T' \rightarrow \epsilon$		
Factor						$F \rightarrow (E)$		$F \rightarrow id$	$F \rightarrow num$