The construction of both top-down and bottom-up parsers is aided by two functions $FIRST()$ and $FOLLOW()$**.**

Define $FIRST(\alpha)$, where α is any string of grammar symbols, to be the set of **terminals** that begin strings derived from α. If $\alpha \overset{*}{\Rightarrow} \in$, then $\in$ is also in $FIRST(\alpha)$**.**

To compute $FIRST(X)$

1. If X is a terminal, then $FIRST(X) = \{X\}$
2. If X is a nonterminal and $X \to Y_1 Y_2 \cdots Y_k$ is a production, then place a in $FIRST(X)$ if for some i, a is in $FIRST(Y_i)$, and $\in$ is all of $FIRST(Y_1), \cdots, FIRST(Y_{i-1})$; that is: $Y_1 \cdots Y_{i-1} \overset{*}{\Rightarrow} \in$. If $\in$ is in $FIRST(Y_j)$ for all $j = 1,2, \cdots, k$, then add $\in$ to $FIRST(X)$. For example, everything in $FIRST(Y_1)$ is surely in $FIRST(X)$. If Y_1 does not derive $\in$, then we add nothing more to $FIRST(X)$but if $Y_1 \Rightarrow \in$, then we add $FIRST(Y_2)$, and so on.
3. If $X \to \in$ is a production, then add $\in$ to $FIRST(X)$.

To compute $FOLLOW(A)$

1. Place \$ in $FOLLOW(S)$ where S is the start symbol, and \$ is the input right endmarker.
2. If there is a production $A \to \alpha B \beta$, then everything in $FIRST(\beta)$ except $\in$ is in $FOLLOW(B)$.
3. If there is a production $A \to \alpha B$, or a production $A \to \alpha B \beta$, where $FIRST(\beta)$ contains $\in$, then everything in $FOLLOW(A)$ is in $FOLLOW(B)$.

Consider the grammar.

	left side		*right side*
1	*E*	→	*E* **+** *T*
2	*E*	→	*T*
3	*T*	→	*T* ***** *F*
4	*T*	→	*F*
5	*F*	→	**(** *E* **)**
6	*F*	→	**id**

Table 1. Set of productions expressions

Step	Action	Justification
1	$FIRST(E) = FIRST(T)$	FIRST, rule 2
2	$FIRST(T) = FIRST(F)$	FIRST, rule 2
3	$FIRST(F) = \{$**(**,**id**$\}$	FIRST, rule 1 applied to productions 5 and 6

Step	Action	Justification
1	$FOLLOW(E) = \$$	FOLLOW, rule 1
2	$FOLLOW(E) = FOLLOW(E) \cup$ {**+**,**)**} ={\$,**+**,**)**}	FOLLOW, rule 2, productions 1 and 5
3	$FOLLOW(T) = FOLLOW(E) =$ {\$,**+**,**)**}	FOLLOW, rule 2, production 2
4	$FOLLOW(T) = FOLLOW(T) \cup$ {*****} = {\$,*****,**+**,**)**}	FOLLOW, rule 2, production 3
5	$FOLLOW(F) = FOLLOW(T) =$ {**\$**,*****,**+**,**)**}	FOLLOW, rule 3, production 4

	$FIRST()$	$FOLLOW()$
E	{**(**,**id**}	{**$**,**+**,**)**}
F	{**(**,**id**}	{**$**,**+**,*,**)**}
T	{**(**,**id**}	{**$**,**+**,*,**)**}