FIRST and FOLLOW sets

To compute FIRST(X) for all grammar symbols X, apply the following rules until no more terminals or ϵ can be added to any FIRST set.

1. If X is a terminal, then FIRST$(X) = \{X\}$

2. If X is a nonterminal, and FIRST$(X) = Y_1 Y_2 \dots Y_n$ is a production rule, then

    - everything in FIRST(Y_1) is in FIRST(X)
    - if $Y_1 \rightarrow \epsilon$ contains, then everything in FIRST(Y_2) is also in FIRST(X)
    - repeat for $Y_2 \rightarrow \epsilon$, and so on…

3. If $X \rightarrow \epsilon$, then add ϵ to FIRST(X)

To compute FOLLOW(A) for all nonterminals A, apply the following rules until nothing can be added to any FOLLOW set.

1. Place \$ in FOLLOW(S), where S is the start symbol, and \$ is the input right end-marker (i.e. end of the program code).

2. If there is a production rule $X \rightarrow \alpha A \beta$, then everything in FIRST(β) except ϵ is in FOLLOW(A)

3. If there is a production rule $X \rightarrow \alpha A$, or a production $X \rightarrow \alpha A \beta$ where FIRST(β) contains ϵ, then everything in FOLLOW(X) is in FOLLOW(A)

Given our LL grammar:

$S \rightarrow E$
$E \rightarrow TE'$
$E' \rightarrow +TE' \mid -TE' \mid \epsilon$
$T \rightarrow FT'$
$T' \rightarrow *FT' \mid /FT' \mid \epsilon$
$F \rightarrow (E) \mid \mathbf{id}$

Compute the FIRST sets:

- FIRST$(F) = \{$ (, **id** $\}$
- FIRST$(T') = \{ *, \ /, \ \epsilon \}$
- FIRST$(T) =$ FIRST$(F) = \{$ (, **id** $\}$
- FIRST$(E') = \{ +, \ -, \ \epsilon \}$
- FIRST$(E) =$ FIRST$(T) = \{$ (, **id** $\}$
- FIRST$(S) =$ FIRST$(E) = \{$ (, **id** $\}$

Compute the FOLLOW sets:

- $\text{FOLLOW}(S) = \{\ \$\ \}$
- $\text{FOLLOW}(E) = \{\ )\ \}\ \cup\ \text{FOLLOW}(S)\ = \{\ ),\quad \$\ \}$
- $\text{FOLLOW}(E') =\ \text{FOLLOW}(E) = \{\ ),\quad \$\ \}$
- $\text{FOLLOW}(T) =\ \text{FIRST}(E')\ \cup\ \text{FOLLOW}(E/E') = \{\ +,\quad -,\quad),\quad \$\ \}$
- $\text{FOLLOW}(T') =\ \text{FOLLOW}(T) = \{\ +,\quad -,\quad),\quad \$\ \}$
- $\text{FOLLOW}(F) =\ \text{FIRST}(T')\ \cup\ \text{FOLLOW}(T/T')\ = \{\ *,\quad /,\quad +,\quad -,\quad),\quad \$\ \}$

Generate the predictive parsing table, $M[A, \alpha]$, (for LL grammars) from the FIRST/FOLLOW sets.

For each production rule $A \rightarrow \alpha$ of the grammar, do the following:

1. For each terminal a in FIRST(A), add $A \rightarrow \alpha$ to $M[A, a]$

2. If ϵ is in FIRST(α), then for each terminal b in FOLLOW(A), add $A \rightarrow \alpha$ to $M[A, b]$.

3. Every other entry in the parsing table, $M[A, a]$, implicitly generates an **error**

Non-Terminal	Input Symbol							
	id	+	-	*	/	(	)	$
S	$S \rightarrow E$					$S \rightarrow E$		
E	$E \rightarrow TE'$					$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$	$E' \rightarrow -TE'$				$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$							
T'				$T' \rightarrow *\ FT'$	$T' \rightarrow /FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow \textbf{id}$							