Example:

Consider the Grammar

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \epsilon$

$F \rightarrow (E) \mid id$

construct a predictive parsing table.

Solution:

$FIRST(E) = \{(, id\}$

$FIRST(E') = \{+, \epsilon\}$

$FIRST(T) = \{(, id\}$

$FIRST(T') = \{*, \epsilon\}$

$FIRST(F) = \{(, id\}$

$FIRST(+) = \{+\}$

$FIRST(*) = \{*\}$

$FIRST(() = \{(\}$

$FIRST()) = \{)\}$

$FIRST(id) = \{id\}$

* The following algorithm can be used to construct a predictive parsing table.

Algorithm: construction of predictive Parsing table.

Input : A Grammar G

Output : Parsing Table M

Method :

1. For each production $A \rightarrow \alpha$ of the grammar do step 2 and step 3
2. For each terminal a in $FIRST(\alpha)$, add $A \rightarrow \alpha$ to $M[A, a]$

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

If ϵ is in FIRST(α) and \$ is in FOLLOW(A) add $A \to \alpha$ to M[A, \$]

4. Make each undefined entry to be error

* We can apply this algorithm to the grammar,

$E \Rightarrow TE'$
$E' \Rightarrow +TE' | \epsilon$
$T \Rightarrow FT'$
$T' \Rightarrow *FT' | \epsilon$
$F \Rightarrow (E) | id$.

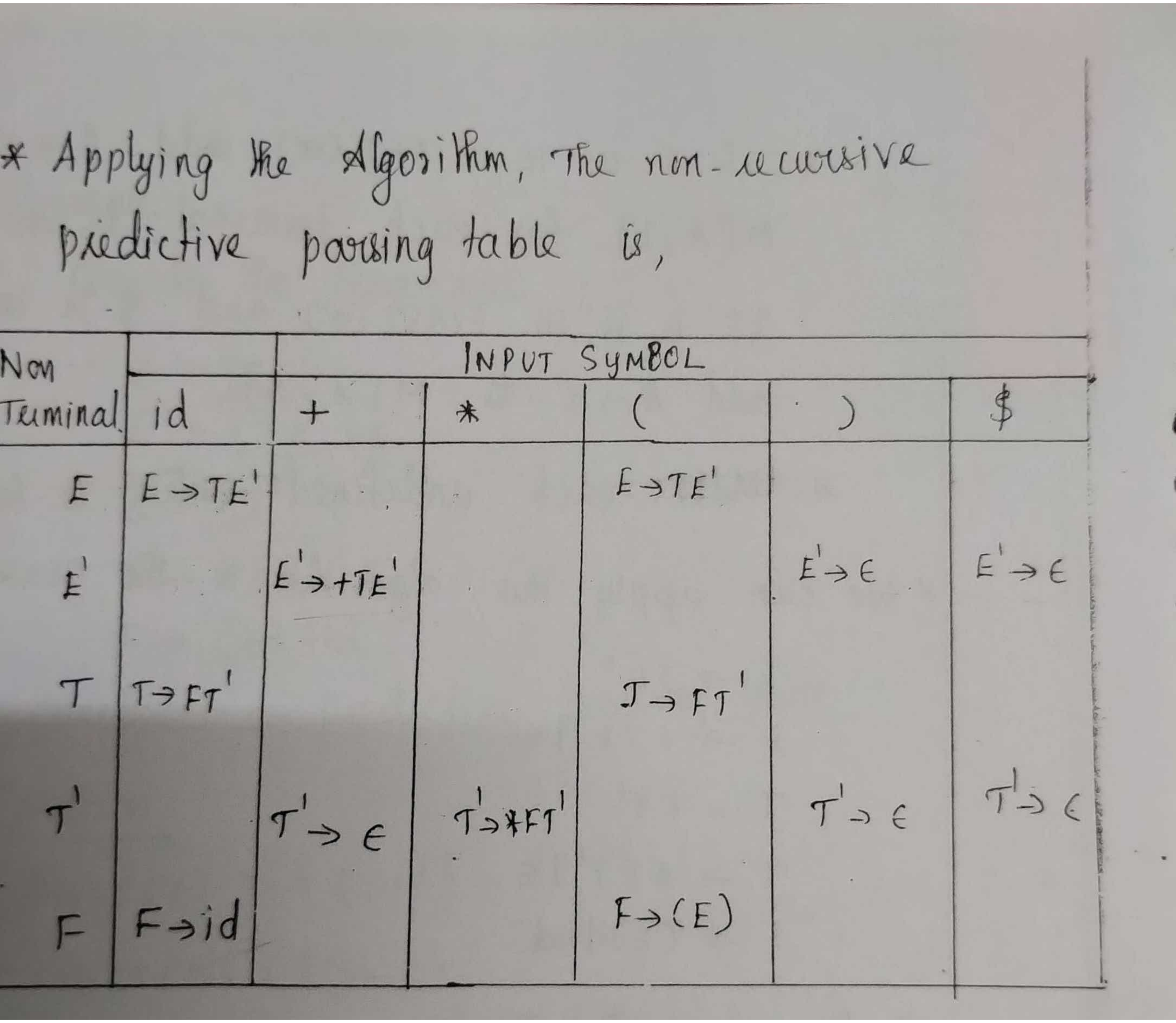
* According to ~~step~~ 2

FIRST(TE') = FIRST(T) = {(, id}

∴ M[T, (] = E → TE'
M[T, id] = E → TE'

* This step 2 is applied to all the productions

FOLLOW(E) = {\$,)}
FOLLOW(E') = {\$,)}
FOLLOW(T) = {), \$, +}
FOLLOW(T') = {), \$, +}
FOLLOW(F) = {+, *,), \$}



* Applying the Algorithm, The non-recursive predictive parsing table is,

Non Terminal	INPUT SYMBOL					
	id	+	*	(	)	$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Example:

Consider the Grammar

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \epsilon$

$F \rightarrow (E) \mid id$.

* In this predictive parsing table, blanks are error entries and non-blanks indicate a production with which to expand the top non-terminal on the stack.

Non Terminal	Input Symbol					
	id	+	*	(	)	$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

* The Parser traces out a left most derivation for this input. i.e) the productions output are those of a left most derivation.

STACK	INPUT	OUTPUT
$\$E$	$id+id*id\$$	
$\$E'T$	$id+id*id\$$	$E \rightarrow TE'$
$\$E'T'F$	$id+id*id\$$	$T \rightarrow FT'$
$\$E'T'id$	$id+id*id\$$	$F \rightarrow id$
$\$E'T'$	$+id*id\$$	
$\$E'$	$+id*id\$$	$T' \rightarrow \epsilon$
$\$E'T+$	$+id*id\$$	$E' \rightarrow +TE'$
$\$E'T$	$id*id\$$	
$\$E'T'F$	$id*id\$$	$T \rightarrow FT'$
$\$E'T'id$	$id*id\$$	$F \rightarrow id$
$\$E'T'$	$*id\$$	
$\$E'T'F*$	$*id\$$	$T' \rightarrow *FT'$
$\$E'T'F$	$id\$$	
$\$E'T'id$	$id\$$	$F \rightarrow id$
$\$E'T'$	$\$$	$T' \rightarrow \epsilon$
$\$E'$	$\$$	$E' \rightarrow \epsilon$