FIRST AND FOLLOW

The construction of a predictive parser is aided by two functions associated with a grammar G. The functions **FIRST** and **FOLLOW** allow us to fill in the entries of a predictive parsing table for G.

Ex. 1:-

A. Give the predictive parsing table for the following grammar:-

E→E+T/T

T→T*F/F

F→(E)/id

B. Show the moves of the parser for the input (id+id) * id

Solution:-

A. Elimination of left recursion:-

E→TE´
E´ →+T E´/ε

T→FT´

T´ →*F T´/ ε

F→(E)/id

Finding FIRST and FOLLOW:-

FIRST(E)= FIRST(T) = FIRST(F) = { (, id }

FIRST(E´) = { + , ε}

FIRST(T´) = { * , ε}

FOLLOW(E) = {) , $ }

FOLLOW(E´) = FOLLOW(E) = {) , $ }

FOLLOW(T) = FIRST(E´) = { + , ε} + FOLLOW(E´)

= { + ,) ,$ }

FOLLOW(T´) = FOLLOW(T) = { + ,) ,$ }

FOLLOW(F) = FIRST(T´) = { * , ε} + FOLLOW(T´) = { *, + ,) ,$ }

So the Predictive Parsing table is,

T / NT	id	+	*	(	)	$
E	E→TE´			E→TE´		
E´		E´→+TE´			E´→ε	E´→ε
T	T→FT´			T→FT´		
T´		T´→ ε	T´→*FT´		T´→ε	T´→ε
F	F→id			F´→(E)		

B. Moves made by predictive parser on Input id+id*id

STACK	INPUT	OUTPUT
$E	id + id * id$	E→TE´
$E´T	id + id * id$	T→FT´
$E´T´F	id + id * id$	F→id
$E´T´(id	id) + id * id$	Remove id
$E´T´	+id * id$	T´→ε
$E´	+id * id$	E´→+TE´
$E´T(+	+)id * id$	Remove +
$E´T	id * id$	T→FT´
$E´T´F	id * id$	F→id
$E´T´(id	id) * id$	Remove id
$E´T´	* id$	T´→*FT´
$E´T´F(*	*)id$	Remove *
$E´T´F	id$	F→id
$E´T´(id	id)$	Remove id
$E´T´	$	T´→ε
$E´	$	E´→ ε
$	$	

Ex.No:2:- Give the Predictive parsing table for the following Grammar,

S→iEtSS´/a

S´→eS/ε

E→b

Solution:-

Elimination of Left Recursion

The above grammar has no left Recursion. So we move to First and Follow.

First(S) = {i, a}

First(S´) = {e, ε}

First(E) = { b}

Follow(S) = First(S´) = {e, ε} + Follow(S´)

= { e, $}

Follow (E) = {t, $}

Follow (S´) = Follow(S) = {e, $}

So the Predictive Parsing table is,

NT \ T	a	B	e	i	t	$
S	S→a			S→iEtSS´		
S´			S´→eS S´→ε			S´→ε
E		E→b				

Ex.No:3:- Give the Predictive parsing table for the following Grammar,

S→CC

C→cC/d

Solution:-

First(S) = First(C) = {c,d}

Follow(S) = { $ }

Follow(C) = First (C) = { c, d, $}

So the predictive parsing table is

NT \ T	c	d	$
S	S→CC	S→CC	
C	C→cC	C→d	

Ex.No:3:- Give the Predictive parsing table for the following Grammar,

S→iCtSS´/ a
S´→eS/ε
C→b

Solution:-

FIRST(S) = { i, a}

FIRST(S´) = { e, ε}

FIRST(C) = { b}

FOLLOW(S) = FIRST(S´) = {e, ε} + FOLLOW(S´)

= { e, $}

FOLLOW(S´) = FOLLOW(S) = {e, $}

FOLLOW(C) = {t, $}

So the predictive parsing table is

T / NT	a	B	e	i	t	$
S	S→a			S→iCtSS´		
S´			S´→eS S´→ε			S´→ε
C		C→b				

References

1. J. Tremblay, P.G. Sorenson,"The Theory and Practice of Compiler Writing “, McGRAW-HILL,1985.
2. W.M. Waite, L.R. Carter,"An Introduction to Compiler Construction",Harper Collins,New york,1993
3. A.W. Appel,"Modern Compiler Implementation in, CambridgeUniversity Press,1998
4. Internet Papers
5. Aho, R. Sethi, J.D. Ullman," Compilers- Principles, Techniques and Tools"Addison-Weseley, 2007