

Government College of Engineering, Jalgaon

(An Autonomous Institute of Govt. of Maharashtra)

Name of Student:.....

PRN:.....

Course Teacher: Sharayu Bonde

Date of Performance:.....

Date of Completion:.....

Experiment No. B: 01

Title: Write a C program to implement operator precedence parsing

Theory:

A grammar that is generated to define the mathematical operators is called operator grammar with some restrictions on grammar. An operator precedence grammar is a context-free grammar that has the property that no production has either an empty right-hand side (null productions) or two adjacent non-terminals in its right-hand side

Operator precedence grammar is kinds of shift reduce parsing method. It is applied to a small class of operator grammars.

1. A grammar is said to be operator precedence grammar if it has two properties:
2. No R.H.S. of any production has $a \in$.
3. No two non-terminals are adjacent.
4. Operator precedence can only established between the terminals of the grammar. It ignores the non-terminal.

There are the three operator precedence relations:

- $a > b$ means that terminal "a" has the higher precedence than terminal "b".
- $a < b$ means that terminal "a" has the lower precedence than terminal "b".
- $a \doteq b$ means that the terminal "a" and "b" both have same precedence.

Precedence table:

	+	*	()	id	\$
+	$>$	$<$	$<$	$>$	$<$	$>$
*	$>$	$>$	$<$	$>$	$<$	$>$
($<$	$<$	$<$	\doteq	$<$	X
)	$>$	$>$	X	$>$	X	$>$
id	$>$	$>$	X	$>$	X	$>$
\$	$<$	$<$	$<$	X	$<$	X

Parsing Action

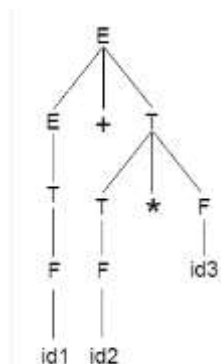
- Both end of the given input string, add the \$ symbol.
- Now scan the input string from left right until the $>$ is encountered.
- Scan towards left over all the equal precedence until the first left most $<$ is encountered.
- Everything between left most $<$ and right most $>$ is a handle.
- \$ on \$ means parsing is successful.

Example

Grammar:

$E \rightarrow E+T/T$
 $T \rightarrow T*F/F$
 $F \rightarrow id$
Given string:
 $w = id + id * id$

Let us consider a parse tree for it as follows:



On the basis of above tree, we can design following operator precedence table:

	E	T	F	id	+	*	\$
E	X	X	X	X	=	X	>
T	X	X	X	X	>	=	>
F	X	X	X	X	>	>	>
id	X	X	X	X	>	>	>
+	X	=	<	<	X	X	X
*	X	X	=	<	X	X	X
\$	<	<	<	<	X	X	X

Now let us process the string with the help of the above precedence table:

 $\$ < id1 > + id2 * id3 \$$
 $\$ < F > + id2 * id3 \$$
 $\$ < T > + id2 * id3 \$$
 $\$ < E = < id2 > * id3 \$$
 $\$ < E = < F > * id3 \$$
 $\$ < E = < T = * < id3 > \$$
 $\$ < E = < T = * = F > \$$
 $\$ < E = + = T > \$$
 $\$ < E = + = T > \$$
 $\$ < E > \$$

Accept.

Sharayu Nivrutti Bonde
Course Teacher