CLASSMATE
Date
Page

ASSIGNMENT-9

TITLE: RECURSIVE DESCENT PARSER

PROBLEM STATEMENT: STUDY RDP

THEORY:

- RECURSIVE DESCENT PARSER:
  - A recursive descent parser is a top-down parser.
  - It is called so because it builds a parse tree from the top (the start symbol) down, from left to right, using an input sequence as a target as it is scanned from left to right.
  - The actual tree is not constructed is implicit as a sequence of function calls.

- WORKING OF RDP:
  - The parser uses a recursive function to each grammar rule (i.e., corresponding to each non-terminal symbol in the language).
  - For simplicity, the name of the non-terminal can be used as name of the function the body of each recursive function mirrors the right side of the method corresponding rule.
  - In order for this method to work, one might must be able to decide which function to call based on next input symbol.

For example,
given grammar

$S \rightarrow fL$

$L \rightarrow +S \mid \varepsilon$

CLASSMATE
Date
Page

T → UM
M → *T
U → (S) | V
V → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Implementation:

```
S() {
    T();
    L();
}
T() {
    V();
    M();
}
U() {
  if ( input [i] == '(' )
    { i++ ;
      S()
      if (input [i] == ')')
        i++ ;
      else
        error = 1
    }
}
M() {
  if ( input [i] == '*' ) {
      i++ ;
      T()
    }
}
```

```
L(){
  if (input[i] == '+') {
        i++;
        S()
  }
}
```

Sample input - output

4+5
accepted

4+4*
rejected

4+(4*4)
accepted

- CONCLUSION:
In this assignment, we implemented RDP for given grammar.