Assignment-9

33119

Title : Recursive Descent Parser (RDP)

Problem Statement : Study of recursive descent parser.

Objective : 1.) To understand basic principles of top-dow parsing.
2.) Study RDP.

Theory :

1.) RDP is a top-down parser, so called because it builds a parse tree from top-down & from left to right.

2.) It uses recursive f^n corresponding to each grammer rule.

3.) We need to decide which f^n to call based on the next i/p. symbol.

Algorithm :

Steps :

1.) Apply left recursion method & remove left recursio grammer if any.

2.) Apply left factoring.

3.) Compute first set

Grammer -
1.) $S \rightarrow TL$
2.) $L \rightarrow +S|e$
3.) $T \rightarrow UM$
4.) $M \rightarrow *T|e$
5.) $U \rightarrow (S)|V$
6.) $V \rightarrow 0|1|\ldots|9$

4.) Compute first set

1.) first (v) = {0, . . . , 9}

2.) first (U) = first ((S)) U first (v) =
{ (} U { (0, ... 9) } = { (0, ... 9}

3.) first (m) = first (*T) U first (e) = {*, e}

4.) first (T) = first (UM) = first (U) = { (0, ... 9) }

5.) first (L) = first (+S) U first (e) = { +, e }

5.) first (S) = first (TL) = first (T)

6.) first (S) = first (TL) = first (T) = { (0, . . . , 9) }

* RDP :

```
Parse_S () {                    // S → TL
        parse_T();
        parse_L();
}
```

```
parse_L() {            // L → +S
   f (lookahead == '+') {
        match ("+");
        parse_S();
   }
    //L
   else . . .
}
```

```
Parse_T () {          // T → UM
    parse_U();
    parse_m();
}
parse_L() {           // L → +S
    if (lookahead == "+") {
        match ("+");
        parse_S();
    }
    else . . .
}
```

```
parse-T() {                    // T → VM
    parse-V();
    parse-M();
}

parse-U() {
    if (lookahead == "(") {            // U → (S)
            match("(");
            parse-S();
            match (")");
    }
    else  parse-v();            // U → V
}

parse-v() {
    if (lookahead == "0") {
          match("0");
    }
    else  if (lookahead == "1") {
            match ("1");
       } else....
}
 else if (lookahead == "g") {
          match ("g");
 } else error ();
}
```

Conclusion : Thus we have studied Recursive Descent Parser (RDP) and implemented it successfully.