Name : Javeria Rizwan

Stu ID : F20212660 98

Section : Compiler construction - V4

Home Work 02 : Syntax Analyzer

Recursive descent Parser : To implement a syntax Analyzer we uses 'recursive descent parser'. Its a kind of Top-down parser i.e, it builds the parse tree from top to bottom.

- We can implement this parser by carefully writing a grammer means eliminating left recursion and left factoring from it.
- No left recursion in given cfg.

Left factoring

Program → void main () { StmtSet }

StmtSet → Stmt S

S → StmtSet | ϵ

Stmt → DecStmt | AssignStmt

DecStmt → Type idlist ;

Type → int | float

idlist → idL

L → , idlist | ϵ

AssignStmt → id = EXP ;

EXP → OPR OP OPR

OP → + | *

OPR → id | num

- Recursive descent parsing uses a set of mutually recursive functions to process the input based on grammer.

Parse tree:

Input string :

```
void main (){
    int id, id;
    id = id+num;
}
```