1. Enhance the above grammar by and find new grammar call it G3

a. Left factor rules 5&6. Call the new term OI

I => '-' OI

OI => ZD

OI => O DS

b. Left factor rules 7&8. Call the new term OD

DS => D OD

OD=> DS

OD=> ƛ

c. G3 After update

S → N $

N → I F

I → ZD

I → O Ds

I → '-' OI

OI → ZD

OI → O DS

DS → D OD

OD → DS

OD → λ

D → ZD

D → O

ZD → '0'

O → ['1'-'9']

F → . Ds

F → λ

2 . Build the nullable, first and follow table for G3

	Nullable	first	Follow
S	F	'-' '0' ['1'-'9']	None
N	F	'-' '0' ['1'-'9']	$
I	F	'-' '0' ['1'-'9']	• $
OI	F	'0' ['1'-'9']	. $
DS	F	'0' ['1'-'9']	. $
OD	F	'0' ['1'-'9']	. $ '0' ['1'-'9']
D	F	'0' ['1'-'9']	. $ '0' ['1'-'9']
ZD	F	'0'	. $ '0' ['1'-'9']

3 . Build the LL(1) parse table for G3

	'-'	'0'	['1'-'9']	$	.
S	S → N $	S → N $	S → N $		
N	N → I F	N → I F	N → I F		
I	I→ '-' OI	I→ ZD	I→ O DS		
OI		OI→ ZD	OI→ O DS		
DS		DS →D OD	DS →D OD		
OD		OD→DS	OD→DS	OD→ λ	OD→ λ
D		D→ZD	D→O		
ZD		ZD→'0'			
O			0→['1'-'9']		
F				F→ λ	F→. DS

4: Write the JLex regular expression specification file to recognize the following tokens.

JLex Regular Expression	Token Type
"." [1-9] 0 "-" {WS} .	DOT NUM ZERO MINUS SPACE ERROR

```
import java.io.*;
enum TokenType{NUM , ZNUM , MINUS , DOT , EOF ,
ERROR}
class Token{
TokenType type;
String value;
public Token (TokenType type , String value){
this.type = type;
this.value = value;
}
}
%%
%class Lexer
%public
%function getNextToken
%type Token
%char
%eofval{
return new Token(TokenType.EOF,"");
%eofval}
%{
 public static void main(String[]args) throws
IOException{
 String number = "-23.45";
 Lexer l = new Lexer(new StringReader(number));
 Token token;
 while((token = l.getNextToken()).type !=
TokenType.EOF)
 System.out.println(token.type + "\t" +
token.value);
 }
%}
%line
%char
%%
<YYINITIAL> [1-9] {return new Token (TokenType.NUM
, yytext());}
<YYINITIAL> [0] {return new Token (TokenType.ZNUM ,
yytext());}
<YYINITIAL> "-" {return new Token (TokenType.MINUS
, yytext());}
<YYINITIAL> "." {return new Token (TokenType.DOT ,
yytext());}
<YYINITIAL> [\r\t\n\f\ ]* {}
<YYINITIAL> . {System.out.println("ERROR");}
```

5: Write a top down parser code for G3

```
import java.io.*;
public class NumberParser {
private Token currentToken = null;
public Lexer lexer;
public NumberParser(Lexer lexer) {
this.lexer = lexer;
}
public void eat(TokenType type) {
if (currentToken.type == type) {
try {
currentToken = lexer.getNextToken();
}
catch (Exception ex) {
Error("unexpected token " + type);
}
}
else {
Error("unexpected token " + type);
}
}
public void Error(String msg) {
System.out.println(msg);
System.exit(0);
}

public Token getNextToken() {
try {
if(currentToken == null)
currentToken = lexer.getNextToken();
return currentToken;
}
catch (Exception ex) {
Error(ex.getMessage());
return null;
}
}
public void S() {
Token nt = getNextToken();
```

```
switch(nt.type){
case MINUS :
N();
eat(TokenType.EOF);
break;
case ZNUM :
N();
eat(TokenType.EOF);
break;
case NUM :
N();
eat(TokenType.EOF);
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
```

```
public void N() {
Token nt = getNextToken();
switch(nt.type){
case MINUS :
I();
F();
break;
case ZNUM :
N();
F();
break;
case NUM :
N();
F();
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void I() {
Token nt = getNextToken();
switch(nt.type){
case MINUS :
eat(TokenType.MINUS);
OI();
break;
case ZNUM :
```

```
ZD();
break;
case NUM :
O();
DS();
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void OI() {
Token nt = getNextToken();
switch(nt.type){
case ZNUM :
ZD();
break;
case NUM :
O();
DS();
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void DS() {
Token nt = getNextToken();
switch(nt.type){
case ZNUM :
D();
OD();
break;
case NUM :
D();
OD();
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void OD() {
Token nt = getNextToken();
switch(nt.type){
case ZNUM :
DS();
break;
```

```
case NUM :
DS();
break;
case EOF :
break;
case DOT :
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void D() {
Token nt = getNextToken();
switch(nt.type){
case ZNUM :
ZD();
break;
case NUM :
O();
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void ZD() {
Token nt = getNextToken();
switch(nt.type){
case ZNUM :
eat(TokenType.NUM);
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public void O() {
Token nt = getNextToken();
switch(nt.type){
case NUM :
eat(TokenType.NUM);
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
```

```
public void F() {
Token nt = getNextToken();
switch(nt.type){
case EOF :
break;
case DOT :
eat(TokenType.DOT);
DS();
break;
default :
Error("unexpected token " + nt.type +
"\t");
}
}
public static void main(String[] args) {
String msg = "12345";
StringReader s = new StringReader(msg);
Lexer lexer = new Lexer(s);
NumberParser parser = new
NumberParser(lexer);
parser.S();
System.out.println("Success");
}
}
```

Amr Emad Abdel Hady Amer

Section : 5