19CSE401 – Compiler Design

RDP construction

Name: Ravella Abhinav

Roll No.: CB.EN.U4CSE19453

RDP:

```
package assignment;

import org.antlr.v4.runtime.Token;

publicclass RDP {
	Lexspeclexer;
	Token token;

	RDP(Lexspeclexer)
	{
		this.lexer = lexer;
	}

	publicbooleanGoal() {
		System.out.println("VISIT Goal()");
		token = lexer.nextToken();
		if(Expr()) {
			if(token.getType() == -1) {
				returntrue;
			} else {
				returnFAIL();
			}
		} else {
			returnFAIL();
		}
	}
```

```
publicbooleanFAIL() {
        System.out.println("FAILED");
        returnfalse;
}

publicbooleanExpr() {
        System.out.println("VISIT Expr()");
        if(Term()) {
                returnEPrime();
        } else {
                returnFAIL();
        }
}

publicbooleanTerm() {
        System.out.println("VISIT Term()");
        if(Factor()) {
                returnTPrime();
        } else {
                returnFAIL();
        }
}

publicbooleanEPrime() {
        System.out.println("VISIT EPrime()");
        if(token.getType() == 8) {
                token = lexer.nextToken();
                if(Term()) {
                        returnEPrime();
                } else {
                        returnFAIL();
                }
        } elseif(token.getType() == 11 || token.getType() == -1) {
                returntrue;
                } else {
                        returnFAIL();
                }
        }

        publicbooleanTPrime() {
                System.out.println("VISIT TPrime()");
                if(token.getType() == 9) {
                        token = lexer.nextToken();
                        if(Factor()) {
                                returnTPrime();
                        } else {
                                returnFAIL();
                        }
                } elseif(token.getType() == 8 || token.getType() == 11 ||
token.getType() == -1) {
                        returntrue;
                } else {
                        returnFAIL();
                }
        }
        publicbooleanFactor() {
                System.out.println("VISIT Factor()");
                if(token.getType() == 10) {
                        token = lexer.nextToken();
                        if(!Expr()) {
                                returnFAIL();
                        }
                        if(token.getType() != 11) {
                                returnFAIL();
                        }
                        token = lexer.nextToken();
                        returntrue;
                } elseif(token.getType() == 6 || token.getType() == 7){
                        token = lexer.nextToken();
                        returntrue;
```

```
            }else {
                   returnFAIL();
            }
       }

}
```

Lexspec.g4:

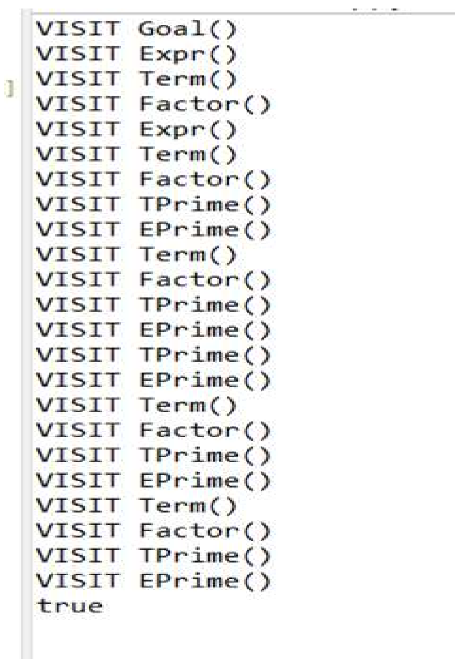
```
lexergrammar Lexspec;

Expr :TermEPrime;
EPrime :PLUSORMINUSTermEPrime
       | EOF;
Term :FactorTPrime;
TPrime :MULTORDIVFactorTPrime
       | EOF;
Factor :OBExprCB
       | NUM
       | NAME;
NUM : [0-9]+;
NAME: [a-zA-Z]+;
PLUSORMINUS :'+'
                    | '-';
MULTORDIV: '*'
             | '/';
OB: '(';
CB: ')';
WS : [ \n\t\r]+ ->skip;
```

Mymain.java

```
package assignment;

import java.io.IOException;

import org.antlr.v4.runtime.*;

public class mymain {


            @SuppressWarnings("deprecation")

            public static void main(String[] args) throws IOException

            {

                        // TODO Auto-generated method stub

                        try

        {

                                    CharStream input = new ANTLRFileStream("C:\\Users\\Sujit\\eclipse-
workspace2\\assignment\\src\\input");

Lexspeclexer = new Lexspec(input);

           RDP rdp = new RDP(lexer);

System.out.println(rdp.Goal());

        }

catch(Throwable t)

        {

System.out.println("Exception: "+t);

t.printStackTrace();

        }

            }


}
```

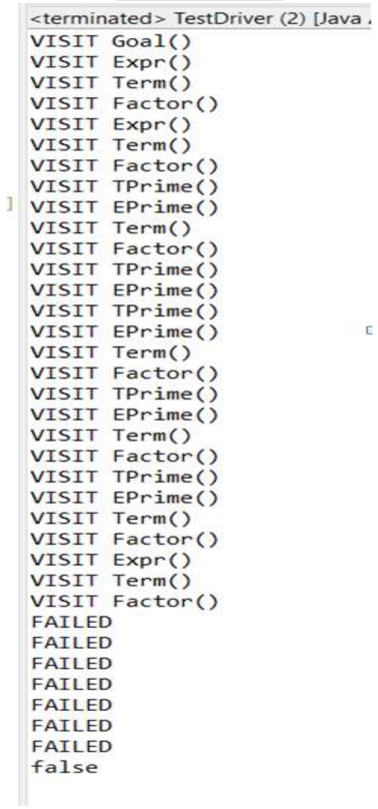
Input/Output:

1. (5*3)-9+4



```
VISIT Goal()
VISIT Expr()
VISIT Term()
VISIT Factor()
VISIT Expr()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
true
```

2. **Input: (5*3)-9+4-(**



```
<terminated> TestDriver (2) [Java
VISIT Goal()
VISIT Expr()
VISIT Term()
VISIT Factor()
VISIT Expr()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT TPrime()
VISIT EPrime()
VISIT Term()
VISIT Factor()
VISIT Expr()
VISIT Term()
VISIT Factor()
FAILED
FAILED
FAILED
FAILED
FAILED
FAILED
FAILED
false
```