Report

CS-3512 Programming Languages

Programming Project – 01

Table of Contents

1. Introduction
   1. Problem
      1. Constrains
   2. Input Format
   3. Output Format
2. Structure
   1. LexicalAnalysar
      1. Scanner.java
      2. Regex.java
      3. Token.java
      4. NameOfToken.java
   2. Parser
      1. Parser.java
      2. ExeptionHandle.java
   3. AST
      1. AST.java
      2. Node.java
      3. NodeType.java
      4. ExeptionHandleStandardize.java
   4. CSE Machine
      1. BetaClass.java
      2. CSEMachine.java
      3. DeltaClass.java
      4. Environment.java
      5. EtaClass.java
      6. Error.java
      7. BackupNode.java
      8. TupleClass.java

# **Introduction**

## **1.1. Problem**

* Implementing a lexical analyzer and a parser for the RPAL language.
* Output of the parser should be the Abstract Syntax Tree (AST) for the given input program.
* Implementing an algorithm to convert the Abstract Syntax Tree (AST) in to Standardize Tree (ST) and implement CSE machine.
* Program should be able to read an input file which contains a RPAL program.
* Output of the program should match the output of “rpal.exe“ for the relevant program.

### **1.1.1.Constrains**

* Should not use ‘lex’. ‘yacc’ or any such tool.
* Must use C/C++ or Java for this project.

## **1.2.Input Requirements**

Read an input file which contains a RPAL program.

Program can be executed using following instructions.

**Input Format**

Eg: Here is one input file

let Sum(A) = Psum (A,Order A )

where rec Psum (T,N) = N eq 0 -> 0

| Psum(T,N-1)+T N

in Print ( Sum (1,2,3,4,5) )

## **1.3.Output Requirements**

Output of the program should match the output of “rpal.exe“ for the relevant program.

**Output Format**

Output of the above example program is: 15

# **Structure**

## **LexicalAnalysar**

### **2.1.1. Scanner.java**

### **2.1.2.LexicalRegexPatterns.java**

### **2.1.3.Token.java**

### **2.1.4.TokenType.java**

## **2.2.Parser**

### **2.2.1.ParseExeption.java**

### **2.2.2.Parser.java**

## **2.3.AST**

### **2.3.1.AST.java**

### **2.3.2.ASTNode.java**

### **2.3.3.ASTNodeType.java**

### **2.3.4.StandedizeExeption.java**

## **2.4.CSE Machine**

### **2.4.1.Beta.java**

### **2.4.2.CSEMachine.java**

### **2.4.3.Delta.java**

### **2.4.4.Environment.java**

### **2.4.5.Eta.java**

### **2.4.6.EvaluationError.java**

### **2.4.7.NodeCopier.java**

### **2.4.8.Tuple.java**