Compiler Design Course Project Presentation: End-Sem

Compiler for Subset of R langauge

Group Members

Sr. No	Name	Roll No	PRN
1	Sarthak Pithe	9	12210166
2	Hrishikesh Potnis	10	12211239
3	Soham Nimale	53	12210227
4	Madhur Vaidya	70	12211223

Division: CS-TY-D

Batch: 3

Group: TY-83

Project Guide:

Prof. Dr. Sheetal Phatangre

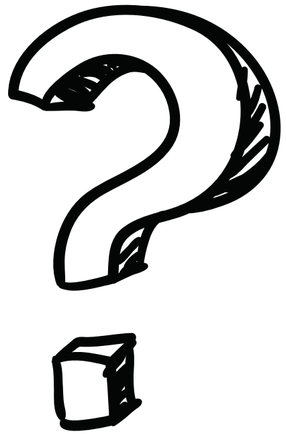
Department of Computer Engineering
Vishwakarma Institute of Technology, Pune

Introduction

- R Language
- Interpreted language
- Interpreters
- Compilers vs Interpreters
- Why a compiler

Problem Statement

To create a compiler for subset of R language



Literature Review

Name	Author	Summary
A Byte Code Compiler for R	Luke Tierney	A basic byte code compiler for R language
A fast abstract syntax tree interpreter for R	Tomas Kalibera, Petr Maj, Floreal Morandat, Jan VitekAuthors Info & Claims	Built R interpreter on top of a Java virtual machine

Methodology

Methodology of the project can be divided into 3 phases:

- Lexical Analysis
- Syntax Analysis
- Semantic Analysis

Lexical Analysis

- Identifying tokens using the lexer
- For example:

Syntax Analysis

- Checking for syntax related errors
- Done using grammar rules in bison
- The input file is parsed and grammar rules are verified
- For example:

var <- 10

var <<- 10

Semantic Analysis

- Type checking in arithmetic expressions
- Checking number of parameters in function calls
- Checking existence of used variables
- For example:

var <- 10 + 10

var <- "Hello" + 10

Outcome

```
------------ ------ ------ -------------------------
Name         Type   Scope   Return Type/Value
------------ ------ ------ -------------------------
m            3      1      Value: 12
my_array1    5      0      Array of size 6: [1.000000, 78.000000, 34.000000, 90.000000, 23.0000
my_array2    5      0      Array of size 4: [12.000000, 34.000000, 67.000000, 12.000000]
my_array3    5      0      Array of size 10: [1.000000, 2.000000, 3.000000, 4.000000, 5.000000,
s            3      1      Value: 10
x            1      0      Value: 1.000000
z            2      0      Value: "$hello@ this is symbol *"
float        1      0      Value: 3.140000
array        1      0      Value: 0.000000
sum_func     6      0      3
  Parameters (2):
    Param Name   Param Type
    ------------ ------------
    m            -1
    s            -1
my_array     5      0      Array of size 5: [1.000000, 2.000000, 3.000000, 4.000000, 5.000000]
result       3      1      Value: -1414878808
a            1      0      Value: 2.144533
b            3      0      Value: 2
c            2      0      Value: C
------------ ------ ------ -------------------------
Symbol Table Ends here!!
```

Symbol Table

Demonstration

Conclusion

With our compiler, the following features can be enjoyed:

- Helps in reducing the execution time
- Includes Error checking (Syntax + Type checking)
- Displaying the symbol table with easy to understand visuals
- Scope resolution for variable re-declaration errors

References

[1]https://en.wikipedia.org/wiki/R_(programming_language)

[2]https://www.geeksforgeeks.org/r-programming-language-introduction/

[3]https://stackoverflow.com/questions/1677021/is-r-an-interpreted-or-compiled-programming-language

[4]https://medium.com/@akshay.sharma81296/does-r-need-a-compiler-b4b0f4a118fd

Thank You!