
PA3 - ANTLR F1VAE

Programming Languages (SWE3006-41)
Fall, 2023

Instructor :

Sungjae Hwang

- jason.sungjae.hwang@gmail.com
- <https://softsec-lab.github.io>

TA :

- Kyeonghwan Min

Introduction

- Deadline : 2023/12/06 (Delay Submission 2023/12/08, 25% deduction per day.)
- Implement Multiple Support F1VAE with ANTLR.
- Submit source codes and report.
 - You will not get any points if your source code does not compiles well.
 - Submit "PA3_StudentID.zip" through icampus.
 - The zip file should contains :
 - > AstCall.java, AstNodes.java, BuildAstVisitor.java, Evaluate.java, Expr.g4, program.java, Report
- Please leave the questions in the google sheet.
https://docs.google.com/spreadsheets/d/19T2K5LmaounOnL-LGaF-nzdu1G_jJXY5nRjSMQr3hVc/edit#gid=323035301
* Avoid using email or the iCampus message for inquires.

- Extend your PA1's ANTLR Project to support Multiple Support F1VAE.
 - Explanation of Multiple Support F1VAE is in lecture Week 10.

```
prog      ::= decl_list expr | expr
decl_list ::= decl decl_list | decl
decl      ::= def var var_list = expr endef | def var = expr endef
var_list  ::= var var_list | var
expr      ::= let var = expr in expr
              | var()
              | var(expr_list)
              | expr + expr
              | expr - expr
              | (expr)
              | number
              | ~ (expr)
              | var
expr_list ::= expr, expr_list | expr
```



```
daily@DESKTOP-AJKPT60:~/2023_Fall_PL/antlr-f1vae$ java program
def f = 3 endef def g x y = x + y endef let x = f() in g(1, x);
DECL
    f
    3.0
DECL
    g
    x
    y
    ADD
    x
    y
LETIN
    x
    Call
    f
    Call
    g
    1.0
    x
0.0
0.0
4.0
```

- You have to implement **Semantic Error Messages** too.
 - Your program's identifier is defined by Lexical Scope manner.

```
def f x = x + z z free identifier endef (*  $\Lambda = [f \mapsto (x, x + z)]$  *)  
let y = 3 in (*  $\sigma = [y \mapsto 3]$  *)  
let z = 4 in (*  $\sigma = [y \mapsto 3, z \mapsto 4]$  *)  
f(z - y) (*  $\Lambda, \emptyset \vdash 1 + z$ : free id error *)
```

```
daily@DESKTOP-AJKPT60:~/2023_Fall_PL/antlr-f1vae$ java program  
def f x = x + z endef let y = 3 in let z = 4 in f(z - y);  
Error: Free identifier z detected.
```

- So, your program should print error messages like picture above.

- There are three types of Semantic Error Messages.
- Free Identifier
 - "Error: Free Identifier {free identifier's name} detected."
- Undefined Function
 - "Error: Undefined function {undefined function's name} detected."
- Argument Mismatch
 - "Error: The number of arguments of {function's name} mismatched, Required: {the number of function's argument}, Actual: {the number of passed argument}"

- Free Identifier

```
daily@DESKTOP-AJKPT60:~/2023_Fall_PL/antlr-f1vae$ java program  
x + 1;  
Error: Free identifier x detected.
```

```
daily@DESKTOP-AJKPT60:~/2023_Fall_PL/antlr-f1vae$ java program  
def x = x + 1 endef x();  
Error: Free identifier x detected.
```

- Undefined Function

```
daily@DESKTOP-AJKPT60:~/2023_Fall_PL/antlr-f1vae$ java program  
def x x y = x + 1 endef y();  
Error: Undefined function y detected.
```

- Argument Mismatch

```
daily@DESKTOP-AJKPT60:~/2023_Fall_PL/antlr-f1vae$ java program  
def x x y = x + 1 endef x();  
Error: The number of arguments of x mismatched, Required: 2, Actual: 0
```

```

17  * Given that all visit_* methods are called in a top-down fashion,
18  * we can be sure that the order in which we add declared variables in the `vars` is
19  * identical to how they are declared in the input program.
20  */
21  private List<String> vars; // stores all the variables declared in the program so far
22  private List<String> semanticErrors; // 1. duplicate declaration 2. reference to undeclared
23  // Note that semantic errors are different from syntax errors.
24
25  @Override
26  public Expression visitDeclaration(DeclarationContext ctx) {
27      Token idToken = ctx.ID().getSymbol(); // equivalent to: ctx.getChild(0).getSymbol()
28      int line = idToken.getLine();
29      int column = idToken.getCharPositionInLine() + 1;
30
31      String id = ctx.getChild(0).getText();
32      if (vars.contains(id)) {
33          semanticErrors.add("Error: variable" + id + " already declared ");
34      }
35  }
36
37  @Override
38  public Expression visitMultiplication(MultiplicationContext ctx) {

```

Description	Resource	Path	Location	Type
Errors (1 item)				
This method must return a result of type Expression	AntlrToExpres...	/AntlrExampleVisitor...	line 26	Java Problem

https://www.youtube.com/watch?v=zo_oIHzKLqw&list=PL5dxAmCmjv_4FGYtGzcVBeoS-BobRTJLq&index=4

https://www.youtube.com/watch?v=zo_oIHzKLqw&list=PL5dxAmCmjv_4FGYtGzcVBeoS-BobRTJLq&index=5

- Restriction of some strings
 - Function Name : alphabet and number -> [a-zA-Z0-9]+
 - Parameter : alphabet and underbar(_) -> [a-zA-Z_]+
 - Variable : alphabet, underbar(_), dash(-) -> [a-zA-Z_ -]+
- Test cases
 - All test cases are based on F1VAE syntax. So, there're no syntax error cases.
 - If you are struggling with ambiguous case, feel free to leave question in Q&A sheet. Your question will helpful for other students.

- Test cases (90pt) + Report (10pt)
- Test cases of this assignment is identical to PA2's test cases.
- Your output should not display any redundant message
 - Only print AST and evaluation or Semantic Error Message.
 - Otherwise, it is considered wrong output
- Report
 - Concisely explain your extended grammar and code.
 - And explain your experience comparing implementation on OCaml and ANTLR.
 - You have to submit PDF file format.
 - Report file's name should be {Your_Student_ID}.pdf

Cheating

- All assignments should be written individually
- **Cheating will make you fail this course**
 - Copying, retyping, outsourcing, submitting copies of others and etc.
 - We will actively check for plagiarism and AI-generated code.
 - We have an automated system that computes the similarity between the submitted materials.
 - Disciplinary actions will follow.

Reference

- The Definitive ANTLR4 Reference - Terence Parr
- <http://antlr.org> > Dev Tools > Resources
 - Documentation
 - <https://github.com/antlr/antlr4/blob/master/doc/index.md>
 - Runtime API (look into “Java Runtime” for ANTLR4 APIs)
 - <http://www.antlr.org/api/>
- Java util package
 - ■ <https://www.tutorialspoint.com/java/util/index.htm>