Interpreter Construction

The project basically deals with the construction of interpreter for the python statements. The code is designed for the specific statements in the python language

statements s ::= x = e | print e | s s

expressions e ::= x | n | e + e | e - e | e \* e | e / e

variables x

integers n

I have used ast package for the generation of Abstract Syntax tree skipping the steps for the construction of lexical analyzer and the parse tree. The purpose of the project is to check the semantic correctness of the statements in an efficient manner, so AST is used for this purpose. The entire project is implemented in Python language.

The statements are read sequentially from a file and AST is generated for these statements. Only assign and print statements are to be evaluated in this grammar. The nodes of the abstract syntax tree are parsed to evaluate the statements. For the evaluation of the statements, we use symbol table to store the variables so that it can be used for evaluation of the expression on the right side of the assignment operator or the print. Symbol table also helps us in checking the semantic correctness of the code.

Whenever a variable is encountered in the right side of the Assign statement or print statement, we retrieve the variable value from the symbol table. If the variable is not present, then error is thrown. This ensures semantic correctness of the code. If the new variable is present on the left side of assign operator, then it is stored in symbol table for the future use. This is repeated till the we reach end of file.