SER 502 Team - 18

MOWA Programming Language

Vijay Ram Giduturi - 1223292304
Srikruthi Vedantham - 1223040832
Nikhil Alapati - 1222336349
Phani Teja Inaganti - 1223458470
Krishna Chandra Sen Dadi - 1222678613

Overview

- Features of language
- Tools used
- Tools installation
- Flow of language
- Grammar of the language
- Future Scope & Limitations

Features of language

- MOWA is an imperative language.
- It supports arithmetic, unary, ternary, relational & logical operations. It accepts datatypes like integer, strings and boolean.
- Conditional statements and loops (along with for-range) can also be performed in this language.
- “show” is used as a print function in this language
- And few extra functionalities likes code comment, end of line (@) etc. can be used.

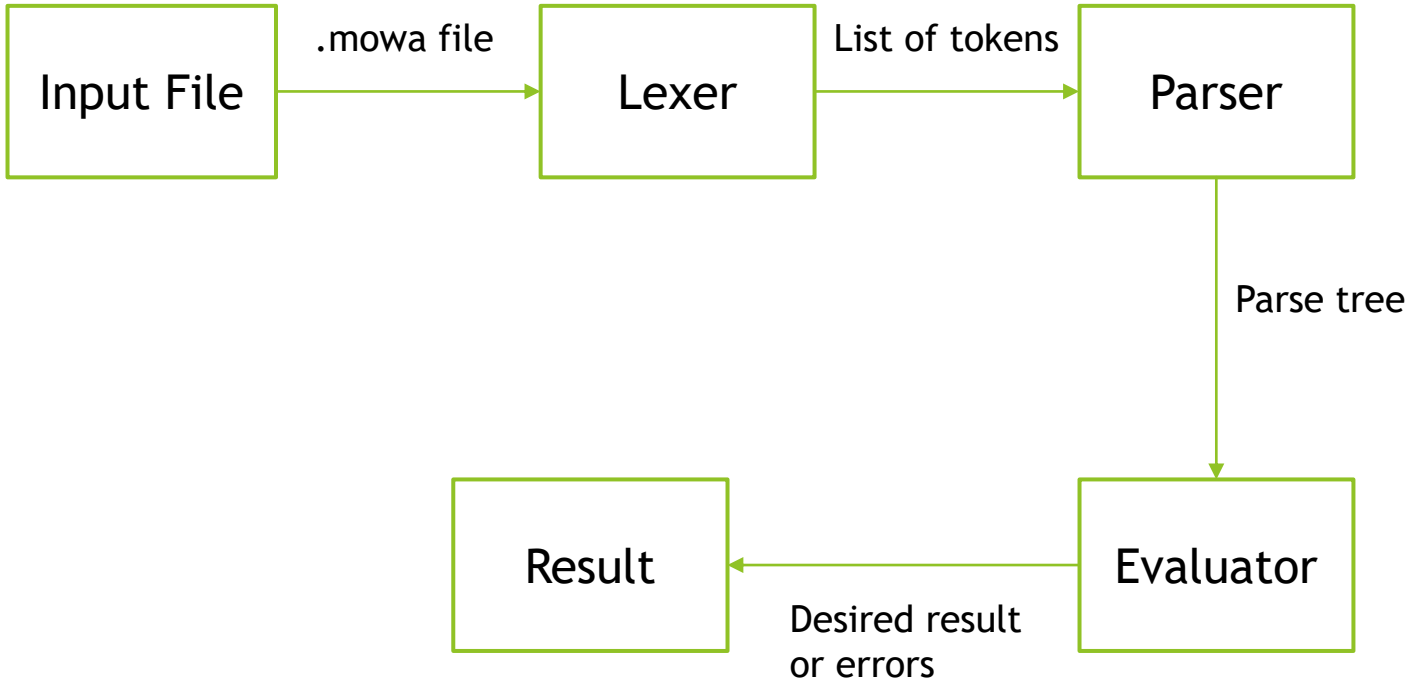
Tools used

- **Python** is used to build lexer.
- **PySwip** library is used to interface python with prolog.
- Parser & Semantic Analysis (Evaluator) are designed with the help of SWI **Prolog**.
- For lexical analysis, **Sly** library of python is used.

Tools installation – For MacOS

- Install python3 on your machine and set environment variable called PATH.
  - brew install python3
- Install SWI-Prolog of version 8.2.10.
  - brew install swi-prolog
- Use python command to install pip, then use pip command to install the necessary libraries like PySwip, Sly.
  - pip3 install sly
  - pip3 install pyswip

Flow of language



Grammar of language

Source code:

- It is the program written by user and given as input to lexer for analysis. It has .mowa as the extension of file.

Lexer:

- To perform lexical analysis, we will use SLY library of python. This is the phase where program is taken as input & generates a list of tokens.
- It returns errors if found any and ignores blank or new lines, spaces and comments.

Grammar of language

Parser:

- Parser will take list of tokens from lexer as input and will generate a parse tree. It will check for syntax and produces a tree structure.

Evaluator:

- This will check the semantics of the grammar, will evaluate to give an output. If it encounters an error, then those will be displayed.

Output:

- After semantic analysis, the output for that program is generated. Output can be desired result or errors that are occurred.

Grammar of language

Data Types:

- Integers – num
- Strings – ntr
- Boolean – bool

Operators:

- Arithmetic - *,/,+,- (represented by →)
- Unary - ++,--
- Relational - <,>,<=,>=,==,~=
- Logical - ^(and), |(or), ~(not)

Grammar of language

Looping Statements:

- For loop
- While loop
- For in range

Printing Statements:

- show()

Comments:

- Single line comment - #

Future Scope & Limitations

- Multiline comment - ## …. ##
- Extra Operators and data structures Implementations.