CmpE230 Fall '19
Homework 2

Selman Berk Özkurt, Emre Hoşer

2017402195, 2016400366

December 2, 2019

1 Introduction

In this project, we will implement an interpreter for *matlang* programming language through Python. Our source code takes a *.mat* file as an argument and translate it to Python syntax in a Python program, then execute the program and display the output on the terminal.

2 Code

2.1 Internal Layer

In this section, we implement classes and methods that is used for execution of *matlang* language and creating abstraction for the language. There are two classes; *matrix* and *vector*(that inherit the matrix class). *matrix* class include the methods for multiplication, addition, subtraction and indexing.

__init__ → constructor for matrix

__str__ → print overloading method that returns the matrix in the string form that we want to print

set → method for setting matrix elements that takes a python list as a input and replaces list elements in the two dimensional list respectively

__add__ → overloading for addition method that check matrices size whether they are in proper for addition and return the matrices that is result of addition.

__mul__ → overloading for multiplication method that takes two parameter one for matrix and other for scalar, vector or matrix, then check the parameters type and multiply them according to types. If it is matrix-matrix or matrix- vector multiplication, check parameters size whether they are in proper for multiplication. In addition, scalar(inner) product of two vector is specifically determined and asserted to return a scalar to avoid future confusions.

__rmul__ → overloading for right multiplication that is used to ensure multiplication of the matrix and scalar is commutative.

__getitem__ → overloading for indexing that takes input that can be either integer or tuple.

Other methods for this section;

tr → the method that matrix, vector or scalar as input , then check the type of it and transpose it according to type.

vecdot → a helper function for __mul__ method that multiply rows of first matrix and cols second matrix one by one return the result.

myrange → the method for translating inclusion convention of *matlang* language into that of python

sqrt → function that return squareroot of parameter.

printsep → function print '-' 10 times.

countcolon → helper function that count colons to determine whether for loop is nested or not.

myexit → the function used to print error message and terminate the process

checkset → check whether the input list size is proper for matrix size or not.

checkadd→ check whether the matrices sizes is proper or not for addition.

checkmul → check whether the matrices sizes is proper or not for multiplication.

choose → function that check which parameters return according to choose function as mentioned in project description .

2.2 Translation Layer

In this section, we modify the input statement of *matlang* language into our own implementation of python program, so that the python interpreter can execute the commands. The main methods of this section is *statement* method that determines how any individual line should be process. We cannot do internally process of for loop, assignment and definition, so we implement methods for them the other pass into python directly. The *_code* variable keeps the all program that is executed in Python.

Methods ;

statement → first check line is empty or not , then check line include any loop keyword if there is, switch the *_for* true read the statement as working like a while process until *_for* is false (if there is '}') then call the *loop()* function. In addition, it doesn't take white spaces into account.

matchassignment, matchdefiniton →helper functions for statement that check line is either assignment or definition by using some keyword.

assignment → split the line into 2 part (left and right of '=' operator) left one is name as variable and right part is removed from curly braces, then set as list in Python adding into the *_code* variable. If there is scalar assignment, it is gone directly into python code.

definition → split the line 2 part. Right part is defined type of the variable. If it is scalar the python program take in it same id and declare it. If it is vector and matrix, turn it into python declaration syntax and fetch the dimension either integer or tuple.

2.2.1 Loop Processing

After the block is properly formed, the loop procedure is called by the statement method. This procedure divides the block into header and executable statement and by checking the header determines the depth of the loop before calling the corresponding loop processing procedure.

Single or nested loop processing functions begin by fetching the loop variable and their ranges, then generate the equivalent python internal headers, also making use of myrange function. Later, they call statements for executable lines of the loop block with proper indentation for python syntax.

2.3 External Layer

In this section, the program opens the input file with the name specified as command line argument, then read the contents of the input program into a string. Later ,a loop calls the statement for every line. At last, *exec* method execute the python code that is translated from *matlang.*