|  |
| --- |
| CS 440 |
| Programming Languages and Translators |
|  |
| MCL: White Paper |
| 01/17/2017 |

|  |
| --- |
| Yuyang Luo;  Chen Can  yluo41@hawk.iit.edu; zdthm007@gmail.com |

**MCL:**

**Matrix Calculating Language**

*Introduction*

MCL is designed to be a simple and intuitive language for performing mathematical calculation on matrixes. A large number of language that wide use in this days like C, C++, Java, Python do not provide base data type of Matrix and operators, they need user use libraries or define the data type and given the operators by themselves. Although the MATLAB Language provide the data type and operators of matrixes, but it a really large environment that need user spend a while to set up, will take lots of space and some function user may not use at all. MCL will provide most operators for matrixes’ calculation as a lightweight language.

*type*

MCL language will be an imperative language, most of the rules will be same as the rules of C language.

*Simple*

MCL will be an intuitive language, it provides most of the operators and functionality for matrixes’ calculation like MATLAB and also provide the most common control flow constructs, such as for loop and if conditionals. MCL will be easy to use, have the syntax that easy to understand, it should be readable and understandable to anyone who have programming experience and familiar with array manipulation functionality.

*Robust*

Program written in MCL are reliable by the simplicity of the language design. There are only three types of variables in MCL language: integers, real numbers and matrixes of real numbers. This allow avoid some error in matrixes calculation. Additional there is extensive size checking in most of the operation, for example, if use attempts to multiply to matrixes that has different numbers of row and column, an error message will be issued. And in MCL language, any calculation that will produce complex number will also issue error message.

*Architecture*

MCL will be an interpreted language. The interpreter will be written in Java, which makes MCL program could be run with any platform that running JVM. (or written in C++, and different platform need different interpreter.)

*Interpreted*

MCL will have a number of operation-intensive manipulations implemented as built in function and operators that needed to be interpreted. And the interpreter also allowed user input program use command line.

*Data types*

There are 2 data types in MCL: integers and matrixes. Integers are elements in matrixes and provide basic calculation, and all data saved in matrixes have to be integer. When user try to declare a matrix, the number of row and column for this matrix have to be given as integer.

*Error handling*

The interpreter will catch most of the lexical and syntactic errors, and will check the row and column of matrixes before calculation, and catches the error of the matrixes’ form and complex number product.

*Example of Syntax*

Here is a sample program in MCL:

matrix a[3,4]={1,2,3,4,1,2,3,4,1,2,3,4};

matrix b[4,3]={1,2,3};

matrix c = a\*b;

matrix c[0,0]=-1;

print c;

In this sample, we first initialize a matrix a of 3 row and 4 columns and a matrix b of 4 row 3 columns, then product the matrix c of a multiply b, then add 1 to each element in c and print it.

*Conclusion*

MCL will be a simple, efficient, light language for matrix calculation. Using intuitive syntax and allow users use little programming to finish matrixes calculation.