Matcat

Mariam Khmaladze, Davit Barblishvili, James Ryan, Andreas Cheng
System Architect Language Guru Tester Manager

Agenda

- Matcat Overview
- Language Key Features
- Compiler Architecture
- Key Implementation Details
- Demo
- Wrap Up
- Q&A

Matcat Overview

Intro and Evolution

Goals

- Convenient matrix **manipulation**
- Convenient matrix **operations**
- **Rich** matrix related built-in functions
- **Polymorphic operators** that work for primitive types and matrices

Language Properties

- Imperative
- **Statically** scoped
- **Strongly** Typed
- **Matrix** supports
  - Special data type
- C-like syntax

Matcat in One Slide

Declaring function

Our matrix data type 😎

Formal argument

```
func m(int ans) matrix {
        return [[ans, 0],[0, ans]];
}
```

Return statement

Polymorphic Operators 😎

Declaring variables as a statement

```
func main() int {
        printStr("I can only show you the door.");
        int a = 657 + 64;
        while (true) {
                print(a);
                printm(inv( m(42) + m(42) ));
        }
}
```

Built-in functions

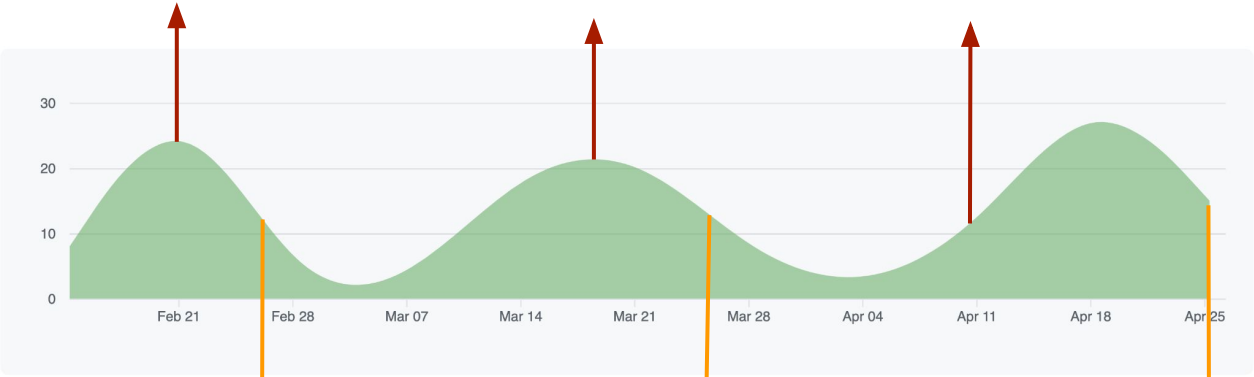
Calling user-defined function

Timeline

Initializing Repo: lots of renaming

Create ast, sast, codegen based on MicroC

Start working on: `matrixLibrary.c`, adding `Vdecl`



LRM/Parser

Hello World

Presentation

Language Evolution - The iterations

Zero

- Matrix
- Vector
- Matrix/Vector Operations
- String
- String concatenation: “a”+”b”
- Fancy ruby-like string interpolation: “#{num}”
- Structs
- Function that return Multiple Values

One

- Matrix
- Some Matrix Operations
- Variable declaration as statement

Current

- Matrix
- **Many** Matrix Operations
- Variable declaration as statement
- Fancy automated test-suite

Available Built-In Functions

- **inv(matrix mt) : :** inverse of matrix
- **isInv(matrix mt) : :** checks if the matrix is invertible
- **det(matrix mt) : :** determinant of a matrix
- **check_symmetry(matrix mt) : :** checks if the matrix is symmetric
- **rotate90(matrix mt) : :** rotates the matrix in the clockwise direction
- **transpose(matrix mt) : :** transpose of a matrix

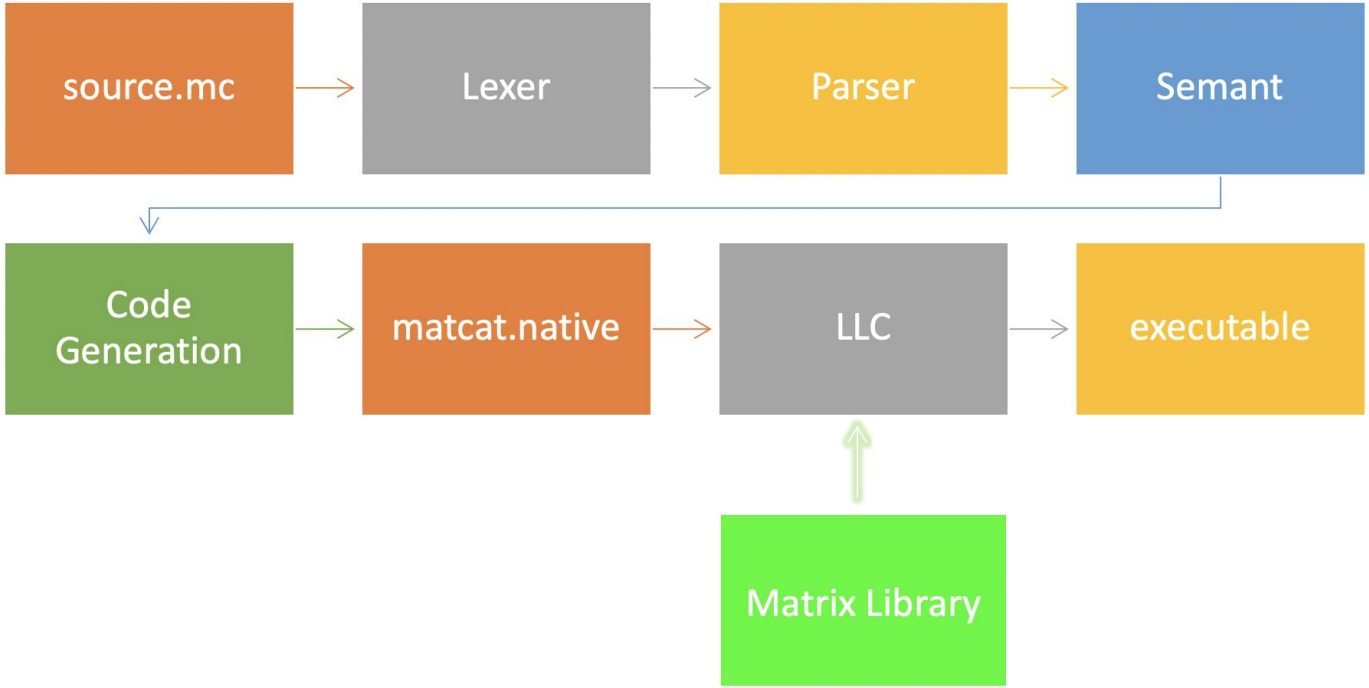
Language Key Features

- Linear Algebra
- Matrix Operations
  - +, −, *, /, ^
  - [i , :] , [: , j] , [: , :] , [i , j]
  - dot product

Compiler Architecture

The Structure...

Compiler Architecture



Matrix Overview

- A new non-orthodox data type in Matcat;
- It does not require the user to define dimensions;
- numbers of columns and rows are accounted internally;
- Defined, implemented, and maintained in C using two-dimensional array.

	0	1	2
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

Column Index.
Accessible → [:,j]

Matrix Diagonal.
Accessible → [:,:]

Row Index.
Accessible → [i,:]

Matrix Structure

$$1$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 4 & 5 & 1 & 1 \\ 0 & 4 & 6 & 8 & 9 & 7 \end{bmatrix}$$

Scalar

Vector

Matrix

Syntax for Matrix

The operators and some snippets

Snippets: Dot Product

```
func main() int{
    matrix a = [[1,2,3]];
    matrix b = [[2,3,4]];
    printd(a dot b);
    return 0;
}
```

Result: 20

Snippets: Matrix Multiplication

```
func main() int {
    matrix a = [[1,2][4,5]];
    printm(a * a);
    return 0;
}
```

Result:

```
(
[9.00 12.00 ]
[24.00 33.00 ]
)
```

Snippets: Scalar * Matrix

```
func main() int {
    matrix a;
    a = [[1,0,0],[0,1,0],[0,0,1]];
    printm(4.2 * a);
    return 0;
}
```

Result:

```
(
[4.20 0.00 0.00 ]
[0.00 4.20 0.00 ]
[0.00 0.00 4.20 ]
)
```

Snippets: Matrix Inverse

```
func main() int {
    matrix a;
    a = [[1,0],[0,1]];
    printm(inv(a));
    return 0;
}
```

Result:

```
(
[1.00 -0.00 ]
[-0.00 1.00 ]
)
```

Demo

Sample programs

Wrap up

Future work, challenges and lessons learnt

Future Work

- Accept any types in the matrix
- Make it works like a interpreter
- Better semantic checking
- Allowing library import
  - `#include<cmath>`
- More built-in data types
  - `String, List, Tuple`
- `Struct/Class`
- Integrate the automated testsuite on GitHub

Challenges

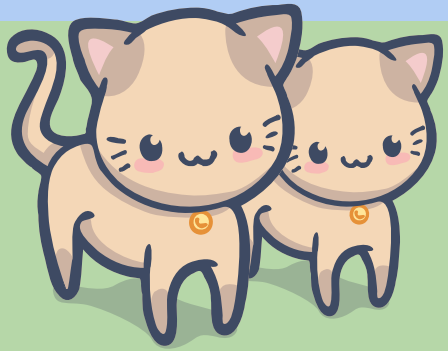
- Learning Ocaml
- Learning LLVM
- Learning Git
- Timezone 😿
- A bit lost after the Hello World
- “Personnel changes”
  - Teammate and TA are changed

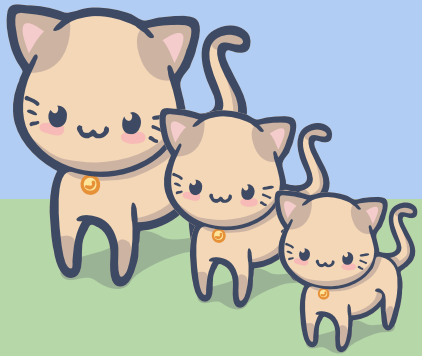
Lessons Learnt

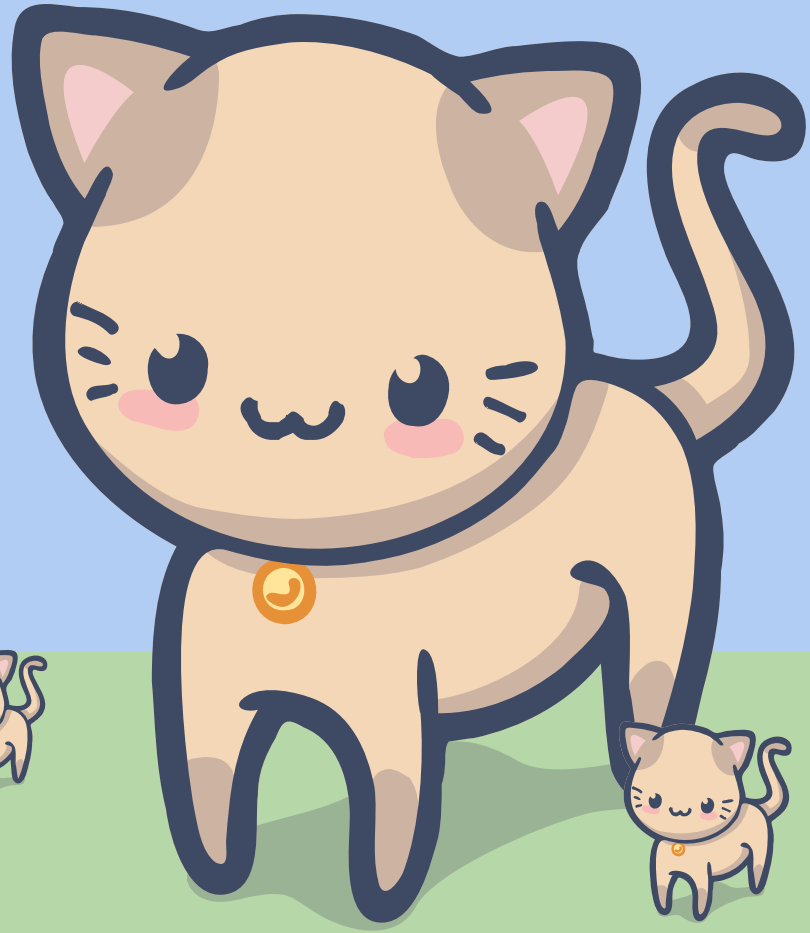
- Functional Programming
- Compiler
- Collaboration
- Linear Algebra

Thanks!

Any questions?







Credits

Special thanks to Xijiao Li, our TA,
(λ Stephen . λ A . Edwards),
and those wonderful past projects:

- PixelPlusPlus
- Shoo
- Coral

Also, special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

Extra Screenshots

From the code to the test-suite

```
test-matrixMult.mc...                PASSED
test-scalarDivDouble.mc...           PASSED
test-addFloats.mc...                 PASSED
test-matrixAccess2D.mc...            PASSED
test-scaleMatrixDouble.mc...         PASSED
test-for2.mc...                      PASSED
test-ifelse1.mc...                   PASSED
test-matrixAccess1D.mc...            PASSED
test-addMatrix.mc...                 PASSED
test-matrixMult1.mc...               PASSED
fail-assign2.mc...                   PASSED. Failed at semantic checking.
fail-funcInLoop.mc...                PASSED. Failed at semantic checking.
fail-multipleTypes.mc...             PASSED. Failed at semantic checking.
fail-for4.mc...                      PASSED. Failed at semantic checking.
fail-ifelse2.mc...                   PASSED. Failed at semantic checking.
fail-for5.mc...                      PASSED. Failed at semantic checking.
fail-ifelse3.mc...                   PASSED. Failed at semantic checking.
fail-assign3.mc...                   PASSED. Failed at semantic checking.
fail-emptyMatrixDeclare.mc...        PASSED. Failed at semantic checking.
fail-for1.mc...                      PASSED. Failed at semantic checking.
fail-funcInFunc.mc...                PASSED. Failed at semantic checking.
fail-while1.mc...                    PASSED. Failed at semantic checking.
fail-binop-floats-ints.mc...         PASSED. Failed at semantic checking.
fail-call-nonfunc.mc...              PASSED. Failed at semantic checking.
fail-nestedMatrix.mc...              PASSED. Failed at semantic checking.
fail-illegalVarName.mc...            PASSED. Failed at semantic checking.
fail-helloworld1.mc...               PASSED. Failed at semantic checking.
fail-returnTypeMismatch.mc...        PASSED. Failed at semantic checking.
fail-topLevelReturn.mc...            PASSED. Failed at semantic checking.
fail-for2.mc...                      PASSED. Failed at semantic checking.
fail-assign1.mc...                   PASSED. Failed at semantic checking.
fail-for3.mc...                      PASSED. Failed at semantic checking.
fail-ifWithoutExpr.mc...             PASSED. Failed at semantic checking.
fail-ifelse1.mc...                   PASSED. Failed at semantic checking.

Test summary:
Passed: 74/74
Failed: 0/74
Skipped: 1
```

```
❯ py testall.py --dir future_improvements
test-transposeMatrix.mc...          FAILED. Output does not match.
        Expected: b'(\n[1.00 4.00 ]\n[2.00 5.00 ]\n[3.00 6.00 ]\n)'
        Actual:   b'(\n[0.00 0.00 ]\n[2.00 5.00 ]\n[3.00 6.00 ]\n)'
test-assignMatrixTypes.mc...        FAILED
fail-noReturn.mc...                 FAILED, it should not compile.
fail-badMatrixIndex.mc...           FAILED, it should not compile.

Test summary:
Passed: 0/4
Failed: 4/4
```