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
In the `main` function, we declare the variable `a` and assign it the value of `657 + 64`. Then, within a `while` loop that runs indefinitely, we print the value of `a` and call the `printm` function with an argument that computes the inverse of the sum of the matrices `m(42) + m(42)`. The `m` function is defined as a user-defined function that takes an integer `ans` as its argument and returns a matrix of integers. The matrix returned by `m` is dependent on the value of `ans`. The `printStr` function is used to print a string message.
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

<table>
<thead>
<tr>
<th>Zero</th>
<th>One</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Matrix</td>
<td>● Matrix</td>
<td>● Matrix</td>
</tr>
<tr>
<td>● Vector</td>
<td>● Some Matrix Operations</td>
<td>● Many Matrix Operations</td>
</tr>
<tr>
<td>● Matrix/Vector Operations</td>
<td>● Variable declaration as statement</td>
<td>● Variable declaration as statement</td>
</tr>
<tr>
<td>● String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● String concatenation: “a”+”b”</td>
<td></td>
<td>● Fancy automated test-suite</td>
</tr>
<tr>
<td>● Fancy ruby-like string interpolation: “#{num}”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Structs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Function that return Multiple Values</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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...
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.
Matrix Structure

Scalar

Vector

Matrix

\[
\begin{pmatrix}
1 & 1 & 4 & 5 & 1 & 1 \\
0 & 4 & 6 & 8 & 9 & 7 \\
\end{pmatrix}
\]
Syntax for Matrix

The operators and some snippets
func main() int{
    matrix a = [[1,2,3]];  
    matrix b = [[2,3,4]];  
    printd(a dot b);  
    return 0;   
}

Result: 20
func main() int {
    matrix a = [[1,2],[4,5]];
    printm(a * a);
    return 0;
}

Result:
(
[9.00 12.00 ]
[24.00 33.00 ]
)
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)
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?
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](https://slidescarnival.com)
- Photographs by [Unsplash](https://unsplash.com)
Extra Screenshots

From the code to the test-suite
Test summary:
Passed: 74/74
Failed: 0/74
Skipped: 1

```
py testall.py --dir future_improvements

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