

# CMAT Final Presentation

Language Guru: Michael Berkowitz (meb2235)

Project Manager: Frank Cabada (fc2452)

System Architect: Marissa Ojeda (mgo2111)

Tester: Daniel Rojas (dhr2119)

# Introduction

# What is CMAT?

## LLVM

CMAT compiles to LLVM IR. LLVM is flexible and works across multiple platforms.

## C-Like

Our syntax is inspired by C but some features are inspired by MATLAB.

## Vectors

Vectors are single dimensional arrays that hold integers or floats.

## Exceptions

We have error messages that output when a user uses incorrect semantic statements.

## Matrices

Matrices are 2-dimensional arrays. We allow many operations on matrices such as transpose, matrix multiplication, and scalar multiplication.

## Standard library

We allow extra math, vector, and matrix operations.

# CMAT Timeline



Over 3 months of meetings

5,973 lines of code and tests

273 Git Commits

# How did they do it?

- Communication
- 3-4 Weekly Meetings
- TA Advisor Meetings
- Dividing up tasks individually or in pairs
- Committing working code and branching from master
- Ubuntu: same image, same versions



CMAT



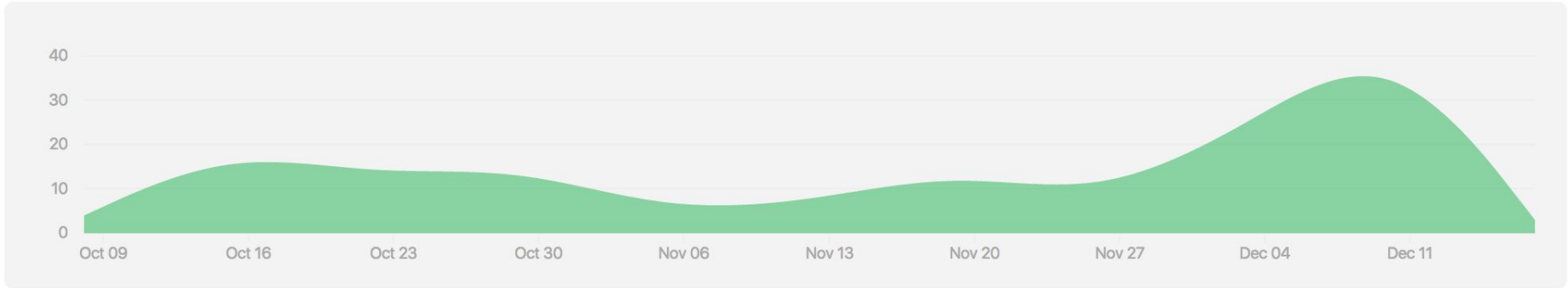
# Git History



Oct 9, 2016 – Dec 18, 2016

Contributions: **Commits** ▾

Contributions to master, excluding merge commits



Let's start programming in  MAT



## Comments

```
/* This is a single-line comment */  
/*This is a long multiple-line  
comment */
```

## Primitives

int, float, bool, void, String, matrix,  
vector

## Control Flow

If, else if, else, while, for, return

## Arithmetic Operators

+ - \* / = ++ --

## Conditional Operators

== != < <= > >=

## Logical Operators

\*\* || !

## Vector

```
| 1 |  
| 1 | 2 | 3 | 4 |  
| 1.0 | 2.0 | 3.5 |  
v:len
```

## Matrix

```
[ 1.0, 2.0; 3.0, 4.0; 5.0, 6.0 ]  
[ 1, 2, 3; 4, 5, 6; 7, 8, 9 ]  
m:rows ; m:cols ; m:tr
```

# Examples of



```
int main() {  
    float f; float g; float h;  
    f = 5.5; g = 6.0;  
    h = g+f; print_float(h);  
    h = g-f; print_float(h);  
    h = g*f; print_float(h);  
    h = g/f; print_float(h);  
    return 0;  
}
```

```
int main() {  
    int i; int j;  
    i=1;  
    while(i < 4) {  
        for(j = 0; j < 3; ++j) {  
            if(i == 1) { print_string("i=1"); }  
            else if(i == 2) { print_string("i=2"); }  
            else { print_string("i=3"); }  
        }  
        i = i+1;  
    }  
    return 0;  
}
```

# Vector Matrix Manipulation

```
#include <stdlib.cmat>

int main() {
    int i; float f;

    matrix int [2,3] mi;
    matrix int [3,2] mj;
    matrix int [2,2] mt;

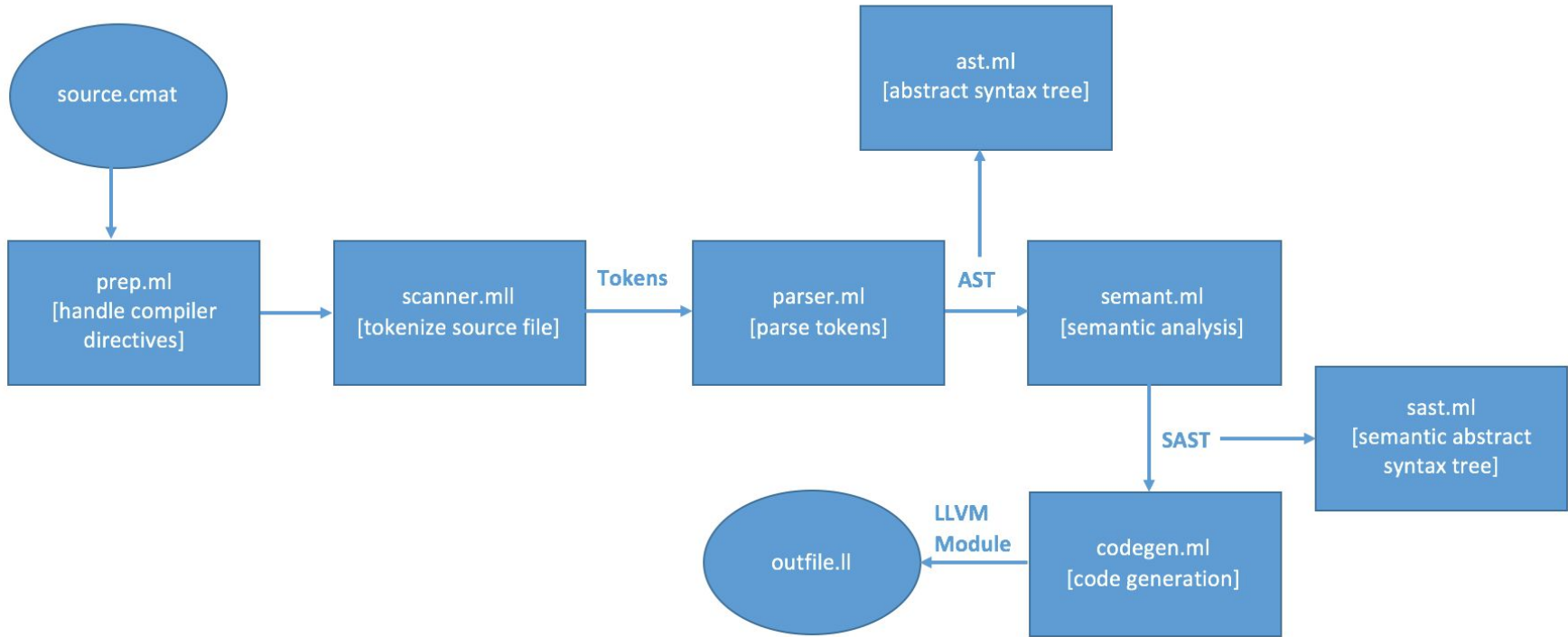
    vector int [2] vi;
    vector float [2] vf;

    i=2; f=5.5;
    vi = |1|2|;
    vf = |1.1|2.2| ;
    mi = [ 1,2,3;
          4,5,6 ];
    mj = [ 1,2;
          3,4;
          5,6 ];

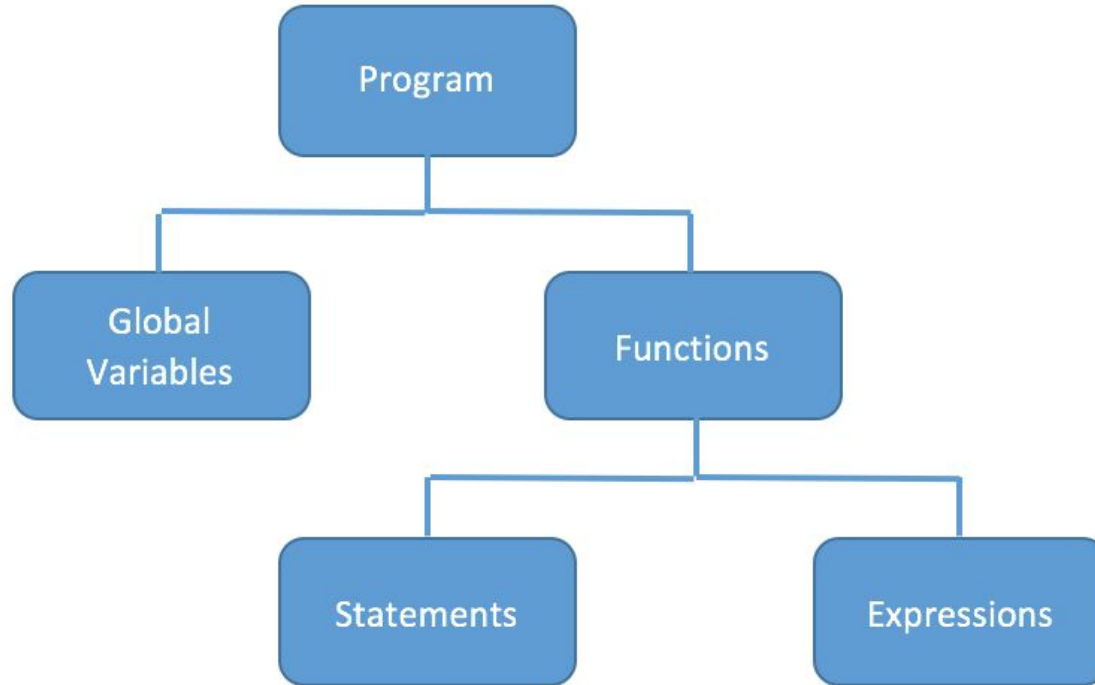
    mi = i*mi;
    mi = mi+mi;
    mi = mi - mi;
    mt = mi*mj;

    vf = f*vf;
    vi = vi+vi;
    vf = vf-vf;
    vi = mt*vi;
    return 0;
}
```

Compiling in **CMAT**



# AST Representation



# Standard Library

## Math functions

mod, powi, powf, sin, cos

## Vector/Matrix functions

dot product functions, rotation matrix generation functions

## Standard IO functions

vector and matrix printing functions

Testing CMAT

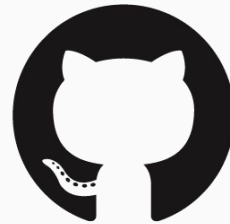


# Testing Infrastructure

- Combo of Travis CI and Github
- Set up Travis; runs on every git push
- Alerted by email only on status change
  - working branch → broken
  - broken branch → fixed
- Sorted tests by
  - Component
    - Fail
    - Pass
- NOT test-driven development
  - Too mercurial of a project



## Travis CI














## GitHub

# frankcabada / plt build passing



































































Current **Branches** Build History Pull Requests

More options 

## Default Branch

 master  130 builds	 #358 passed  about 11 hours ago	 1c8f656  frankcabada					
--	---	--	---	---	---	---	---

## Active Branches

 mike-new  101 builds	 #359 passed  12 minutes ago	 1c8f656  frankcabada					
 fcabada-stdlib  4 builds	 #357 passed  about 12 hours ago	 1c8f656  frankcabada					
 marissa_gc  8 builds	 #356 passed  about 12 hours ago	 123ae96  Marissa Ojeda					
 fcabada-global-dupe-bug  1 builds	 #339 passed  about 18 hours ago	 cacd8b0  frankcabada					
 fcabada-free-bugfix  1 builds	 #338 passed  about 18 hours ago	 f47356c  frankcabada					
 fcabada-tests  3 builds	 #325 passed  a day ago	 9ccd83c  frankcabada					

Demo [MAT