MPL: Matrix Processing Language

David Rincon-Cruz(Language Guru),
Chi Zhang(Tester),
Jiangfeng Wang(Team Leader),
Wode Ni(System Architect)
{dr2884, cz2440, jw3107, wn2155}@columbia.edu

May 11, 2017

Contents

1 Introduction

1.1 Motivation and Background

Matrices lie at the intersection of linear algebra and computer vision, however, this can make introductory graphical projects inaccessible without higher level training in mathematics. Many aspects of image manipulation create high overhead or complicated procedures to program. We proposed MPL, a **Matrix Processing Language** that makes it easy to encapsulate functionality on matrices and images.

1.2 Language Description

The goal of MPL was to make system of programmable operations for matrix manipulations in order to conceptually facilitate image processing. Much of the language is built around an abstraction of the convolutions. At a high level, MPL is an environment of imposing functions on each entry of a matrix, that can view its neighbors to dictate its change.

This allows a new range of behaviors not easily programmable otherwise. Context-insensitive operations such as color inversion are basic and lack much functionality. By abstracting images as matrices of values, we can thus individually manage each pixel cell to alter its own value depending on its local context. Examples such as edge detection work on this intuition by detecting a pixel's similarity to it's neighbors.

1.3 style

MPL functions are built around the concept of the *Moore neighborhood*. We will refer a matrix entry's neighbors with reference to this, specifying them by cardinal directions.

2 Tutorial

2.1 Simple Data Types and Syntax

MPL uses a C-like syntax with curly braces indicating scope, parentheses for expressions and semicolons ending statements. There are 5 main types in our language:

- int
- float
- boolean
- string (literals only)
- Mat

The primitives are: `int`, `float`, `boolean`, `string`. `Mat` is a derived type, meaning that in order to use it, you must specify the uniform type of its contents and its dimensions:

```
Mat<int>[512][512] imageMatrix;
  int a;
  float b;
  bool isTrue;
  string str;
  a = 1;
  b = 2.5;
  isTrue = true;
  str = "helloworld";
```

2.2 Simple Program Structure Example

An MPL program has a simple structure. It's a series a function declarations, followed by a main that controls the main flow of the program.

Function declarations include a type, but no arguments. Functions are **applied** to a matrix of a static type, and thus, functions implicitly have access to 9 arguments whenever they are applied. These are accessed with the encodings:

- # **C** The value of the current entry
- # **N** The value of the entry North of the current entry
- # **NE** The value of the entry Northeast of the current entry
- # **E** The value of the entry East of the current entry
- # **SE** The value of the entry Southeast of the current entry
- # **S** The value of the entry South of the current entry
- # **SW** The value of the entry Southwest of the current entry
- # **W** The value of the entry West of the current entry
- # **NW** The value of the entry Northwest of the current entry

2.2.1 Example:Blur

Suppose you want to blur an image. Naturally you'd want every pixel to modify itself to become an average of itself an it's neighbors. Here is a function `blur` defined to do a Gaussian Blur:

```
int blur() {
    int sum;
    sum = #NW + 2*#N + #NE + 2*#W + 2*#E + #SW + 2*#S + #SE;
    sum = #C * 4 + sum;
    return sum / 16;
}
```

We can then, given an matrix of pixel values, apply blur to it and obtain a new matrix with the values of each pixel averaged.

```
    entry @ imageMatrix;
```

With a full main function, we can thus read in an image, blur it, and write out another image.

```
int main() {
    Mat<int>[512][512] imageMatrix;
    pgmread("input.pgm", imageMatrix);
    entry @ imageMatrix;
    pgmwrite("output.pgm", imageMatrix);
}
```

2.3 Compiling and Running a MPL program

To compile and run a .mpl file, simply feed the file name as input to the "run.sh" under src folder as follows:

```
    ./run.sh samplefile.mpl
```

2.4 User Input

All user-defined functions take no arguments because of the implicit access they'll have on a matrix entry and its neighborhood.

However, MPL provides built-in functions that allow the programmer to specify file paths of input and output, or functions to output to the console. These are the only functions allowed to take in arguments and return void.

```
void print(int i)
void print(float f)
void print(boolean b)
```

`print()` can take 3 types of arguments and will output to the console the value of the integer, float, or boolean.

```
void prints(string s)
```

`prints()` takes a string literal and output a message. This is useful for logging purposes.

```
void printm(mat m)
void print_board(mat m)
```

`printm()` takes an initialized matrix or ints or floats and prints to the console (in matrix format) the values inside. `print_board()` prints the matrix with bits representing zero and nonzero values.

```
void matread(string s, mat m)
void pgmread(string s, mat m)
void ppmread(string s, mat m)
```

These read functions, take a filepath in a string literal and initialize the values into a matrix variable.

```
void matread(string s, mat m)
void pgmread(string s, mat m)
void ppmread(string s, mat m)
```

These read functions, take a filepath in a string literal and initialize the values into a matrix variable.

```
void matwrite(string s, mat m)
void pgmwrite(string s, mat m)
void ppmwrite(string s, mat m)
```

These write functions, take a filepath in a string literal and print the values of the matrix into the file.

2.5 Usage

Our motivations were to allow entry programmers access to cool image projects, so go explore! Although our focus is image manipulation, there's other things you can try too! As you'll see in our *Conway's Game of Life* simulator, it's pretty good at two-dimensional simulations of physical grids - you could even try building a heat-diffusion model. Possibilities are as endless as your imagination.

3 Language Reference Manual

3.1 Lexical Elements

3.1.1 Identifiers

Identifiers are tokens used for naming varaibles and functions. They are case sensitive and should start with a letter and can follow with letters, digits or underscores. Below describes the definition of identifiers:

$$\begin{aligned} identifier &:= \texttt{(letter)(letter | digit | underscore)*} \\ digit &:= \texttt{'0'-'9'} \\ letter &:= \texttt{'A'-'Z''a'-'z'} \end{aligned}$$

3.1.2 Keywords

Keywords are case sensitive and are reserved for different uses in the language, so they cannot be used as identifiers. Below lists all the keywords in MPL:

int	float	boolean
true	false	print
#N	#S	#W
#E	#NW	#SW
#NE	#SE	#C
if	else	elseif
OR	NOT	AND
return	neg	Mat
void	matread	matwrite
printm	prints	while
print_board		

3.1.3 Literals

Literals are constant string, numeric, or boolean values, such as "helloworld", 100, or false. Each literal has a specific type it belongs to and cannot be casted to other types. Assign a literal to another type that it does not belong to will cause an error.

Integer Literals Integer literals are whole numbers represented represented by a sequence of 0-9 digits. An integer can be either positive or negative. To represent negative integers, a keyword "neg" is used.
Examples: 123; neg 321

Boolean Literals Boolean literals have values either true or false. "true" and "false" are both reserved as keywords.
Examples: true; false

Operators Operators are used for arithmetic operations such as addition, subtraction, multiplication and division. Operators can be applied on integers, float numbers and matrices.
We also add a few operators for easier manipulation of matrices calculation. `@` operator is used when applying

a function to a matrix. `@` takes a single function and applies it to every entry of a value matrix.
Examples: `+`; `-`; `*`; `/`;`@`

Delimiters We use white space to separate different tokens in the code.

Parentheses and Braces Parentheses and braces are used to better format the structure of code and limit the scope of variables. Local variables can only be accessed within the scope of code which is identified in the pair of curly braces.

Commas and Semicolons Commas are used to separate function arguments. Semicolons are used to terminate a sequence of code.

Comments Comment is denoted by `//`, such as `// COMMENT`
We also have block comment, which comments out section in between delimiters.

```
/* Comment
Yet another line of comment*/
```

3.2 Data Types

MPL uses strict typing. All variable types should be known at compile time and typecasting is not allowed.

3.2.1 Primitive Data Types

int 32 bit signed integer ranging from -2147483638 to 2147483647.

float 8-byte double-precision floating point numbers.

boolean 1-byte boolean type, either true or false.

string All text values will be of this type.

3.2.2 Non-primitive Data Types

Mat Matrices are the high level equivalent of their math counterparts and will be singly typed as one of 3: integer matrices, float matrices, and function matrices. Function matrices can be applied to integer and float matrices, and standard matrix operations apply (operators defined subsequently).

Declaring a matrix You can declare a matrix by telling its type and dimensions(row and column) explicitly. For example, a 2*2 integer matrix can be declared as:

```
Mat<int> [2][2] A;
```

Initializing a matrix To initialize a matrix that is previously declared, indicating the values at each entry with curly braces, and separate each row with a semicolon. All the entries in a matrix must be the same type, either int or float. Also, the type and dimension of entries must match the declaration. For example, a 2*2 integer matrix can be initialized as:

```
A = 1, 2; 3, 4;
```

Accessing matrix entry Matrix elements can be accessed by providing the row and column location within brackets next to the identifier of the matrix. Using the above example, to get the first row second entry (integer 2) in the matrix, we can do:

```
A[0][1]
```

3.3 Expressions and operators

3.3.1 Expressions

Expressions in MPL are made of operations between matrix and function. They are made up of one or more operands and operators. Like all other mathematics language, innermost expressions will be evaluated first. Otherwise the expressions with higher order will be evaluated before expressions with lower order. If expressions have the same order, the expressions will be evaluated from left to right.

3.3.2 Non-primitive Data Types

The tables below presents the language operators including assignment operators, mathematical operators, logical operators, comparison operators, logical operators. There are also descriptions and order:

operator	description	order
+	plus	1
-	minus	1
*	multiply	2
/	divide	2
=	assignment	0
<	Less than	0
>	More than	0
>=	Less than or equals	0
<=	More than or equals	0
==	equals	0
@	apply	3

3.4 Statements and Functions

3.4.1 Statement

The if Statement The if statement is used to execute a statement if a specified condition is met. If the specified condition is not met, the statement is skipped over. The general form of an if statement is as follows:

```
if(condition) {
  statement1;
  statement2;
  statement3;
    // ...
}
else {
  statement1;
  statement2;
  statement3;
    // ...
}
```

The while Statement The while statement is used to execute a block of code continuously in a loop until the specified condition is not maintained. If the condition is not met upon initially reaching the while loop, the code is never executed. The general structure of a while loop is as follows:

```
while(condition){
  statement1;
  statement2;
  statement3;
    // ...
}
```

3.4.2 Functions

User Defined Function Definitions User Defined Functions in MPL is recognized as an operation on entries. It is treated more like type of data types. The user defined function will operate on and only on the entries. So the way it built is a little different from the tradition flows: a type key word which would be the type of the entry operated, an initial word of "func" but no return type, a function identifier and a paramter. However, in the func, it asks to return some data, which will be the resulted value of the entry operated. An example is shown below:

```
int fblur (){
  int temp = #C + 1;
  return temp ;
}
```

User Defined Function Calling A user defined function can be used directly on a matrix, which will operate every entry in the matirx:

```
int fblur (){
  int temp = #C + 1;
  return temp ;
}
int main(){
  Mat<int> [2][2] C;
  fblur @ C;
  printm(Result);
// D will be {2 2;
//            2 2;}
}
```

System Function The system function are some functions that are included in the language, which is the built-in functions and can be called in the main program. These are some useful and practical functions like "print", "prints", "printm", "print_board","matread", "matwrite". Usually they will not operate matrix entries. Details in 6.2 built-in functions.

3.5 Context Free Grammar

The $|$ and $\circ$ symbols are CFG syntax, not part of the language.

$$\begin{aligned}
program &\rightarrow functionDefs \circ matrixCode\\
functionDefs &\rightarrow \epsilon | fDecls \circ functionDefs\\
fType &\rightarrow int | float\\
fDecl &\rightarrow fType \circ func \circ fId \circ \{gStatements\}\\
gStatements &\rightarrow \epsilon | gStatement; gStatements\\
fStatement &\rightarrow gExpr;\\
&\quad |return\\
&\quad |if(gExpr)\{fStatements\}else\{fStatements\}\\
&\quad |fvDecl\\
fvDecl &\rightarrow fType \circ id = fExpr;\\
fExpr &\rightarrow (fExpr) | fExpr + fTerm | fExpr - fTerm | fTerm\\
fTerm &\rightarrow fTerm * (fExpr)\\
&\quad |fTerm * number\\
&\quad |fTerm/(fExpr)\\
&\quad |fTerm/number\\
imgDecl &\rightarrow Imgid = Img(String);\\
&\quad |Imgid = Img(matId, matId, matId);\\
MatrixCode &\rightarrow genStatements\\
genStatements &\rightarrow \epsilon | matStatement | imgDecl | gExpr;\\
&\quad |if(gExpr)\{genStatments\}else\{genStatements\}\\
&\quad |while(gExpr)\{genStatements\} | return;\\
gvDecl &\rightarrow gType \circ id = gExpr;\\
gType &\rightarrow int | float | boolean\\
gExpr &\rightarrow (gExpr) | gExpr + gTerm | gExpr - gTerm | gTerm\\
gTerm &\rightarrow gTerm * (gExpr)\\
&\quad |gTerm * number\\
&\quad |gTerm/(gExpr)\\
&\quad |gTerm/number\\
matStatement &\rightarrow matDecl | fMatDecl | matExpr;\\
matDecl &\rightarrow Mat < type > id = matExpr; |\\
&\quad Mat < type > id = [matRows];\\
matRows &\rightarrow [numbersList] | [numbersList]; matRows;\\
numbersList &\rightarrow number | number; numbersList\\
fMatDecl &\rightarrow fMatid = fMatExpr;\\
matExpr &\rightarrow (matExpr)\\
&\quad |matExpr + matExpr\\
&\quad |matExpr - matExpr\\
&\quad |matTerm\\
matTerm &\rightarrow matTerm * (matTerm)\\
&\quad |matTerm * matFuncted\\
&\quad |matTerm/(matExpr)\\
&\quad |matTerm/matFuncted\\
&\quad |matTerm * .(matTerm)\\
&\quad |matTerm * .matFuncted\\
&\quad |matTerm/.(matExpr)\\
&\quad |matTerm/.matFuncted\\
matFuncted &\rightarrow id
\end{aligned}$$

4 Project Plan

4.1 Process used for planning, specification, development and testing

Our team met twice a week to discuss our work done, problems we had and future plan. For most of the problems, we sat together to solve it. If it's still not fixed, we met with our TA Julie for further guidance. In the last month of the semester, we met four times a week to do pair programming together. It's very efficient and each of the teammate benefited a lot from each other. We used facebook messenger group to coordinate weekly meetings and any necessary topic that we thought would be worth discussing about. We also used Github as a version control for code and working documents.

Right after we submitted our project proposal, we had an initial project timeline for each component of the project. Since all components of the project were co-dependent, we decided to have several branches to make sure the work done on different sides do not affect others' work. Once everyone finished their part, we merged code to the general develop branch and resolved any conflicts during the merge. When we had any problem that a single person could not solve on his own, we met as a team to work through the problem and solved it. The project timeline kept changing due to the unpredictable code issues we had, but planning ahead gave us pretty clear tasks to complete all the time. We strived to achive the goal and overall the teamwork went well.

We first worked on scanner, parser and ast to ensure the accuracy and tried not to change it unless really necessary. We spent a long time coming up a good working version of our scanner, parser and ast, thus we had a clear idea of how our language looks like and it saved us a lot of time in later development. The team was then divided into two part, each sub-team worked on codegen or semantic checker. Everyone in the team wrote some test cases to thoroughly test their code before commit and it helped a lot when we merged the code. The full test suite successfully passed on May 5th.

4.2 Style Guide

The following outlines our style guide for Ocaml, version control, Bash and C.

4.2.1 Ocaml

- Comments included if logic is at all confusing
- "if" "then" should be in the same line, "else" align with "if" in the next line
- Newlines between function definitions
- Helper function are written to increase code reusability

4.2.2 Version Control Github

- Multiple branches for sub-teams work and merge
- Master branch only contains the most recent complete working version for a milestone
- Commit with meaningful commit messages to allow for easy tracking of code or roll back if necessary
- Create new branches to incorporate new features

4.2.3 Bash

- Separate actions into discrete statements when possible
- One line per statement
- One space between each token

4.2.4 C

We followed standard C coding conventions to generate C code.

4.3 Project Timeline

Date(actual)	Date(projected)	Description
Jan 22	Jan 22	Finalize team members
Jan 25	Jan 25	Discuss project ideas and group norms
Feb 1	Feb 1	Project idea brainstorm
Feb 3	Feb 3	Language features and syntax discussion
Feb 8	Feb 7	Project proposal
Feb 22	Feb 22	Produce LRM
Mar 1	Feb 28	Finalize CFG
Mar 5	Mar 1	First commit, creation of project directory
Mar 15	Mar 7	Develop preliminary scanner, parser
Mar 25	Mar 17	Hello World
-	May 1	Produce semantic checker and codegen
Apr 23	May 1	Compelete semantic checker
May 3	May 1	Complete Sast
May 3	May 1	Complete Codegen
May 5	May 1	Merge code
May 5	May 1	Produce full test suite
May 5	May 1	Build demo game of life
May 5	May 8	Final presentation
May 10	May 10	Final report

4.4 Roles and Responsiblities

We divided roles with Jiangfeng as Project Manager, Nimo as System Architect, David as Language Guru, Chi as Tester. Throughout the course of the project we had different tasks depending on differing needs at that moment. Below is a specified deliverables contributed to by each team member in our group.

Jiangfeng Wang
Project Manager
- Project planning
- Scanner, Parser, Ast, Semantic Checker
- Sample test cases
- Final report
- Final presentation

Nimo Ni
System Architect
- Scanner, Codegen, files linking
- Sample test cases and scripts
- Maintenance of the codebase
- Demo game
- Final report
- Final presentation

David Rincon-Cruz
Language Guru
- Design CFG and language syntax
- Parser, Ast, Semantic Checker
- Sample test cases
- Final report
- Final presentation

Chi Zhang
Tester
- Code gen
- Scanner Test
- Success and fail test suites
- Final Report
- Final presentation

4.5 Software development environment used

4.6 Project log

- Languages: Ocaml, C(library)
- Programming Editor: Sublime, vim
- Version Control: Git, Github, Travis CI
- Documentation: Overleaf

4.7 Project log

4.7.1 Events Log

- Jan 25: Met as a group for the first time
- Feb 1: Met as a to discuss project ideas
- Feb 3 and Feb 5: Met as a group to brainstorm and finalize project architecture
- Feb 8: Develop project proposal
- Mar 1: David finalizes CFG
- Mar 5: Nimo creates of project directory
- Mar 8: Team met with TA Julie to discuss LRM
- Mar 15:

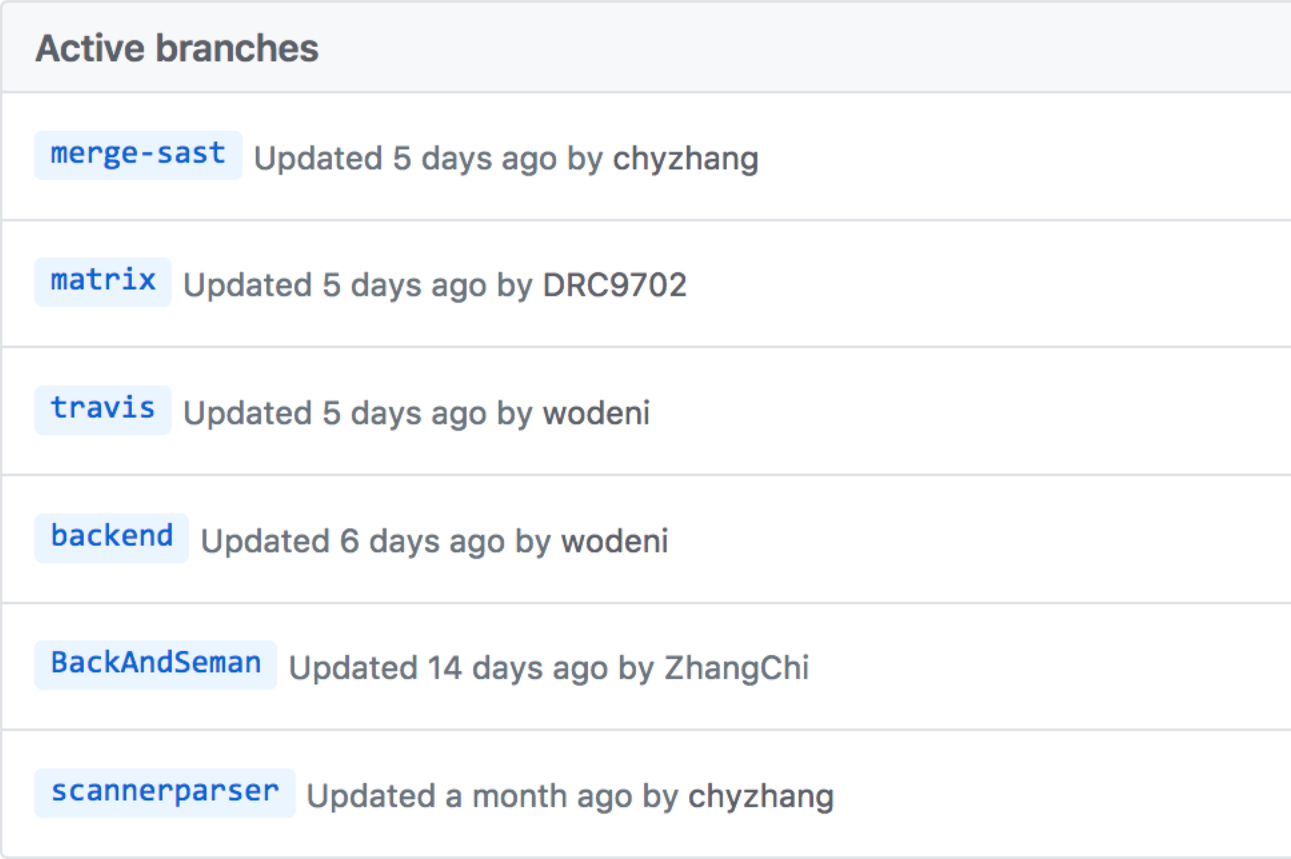
  writes basic code generation, Jiangfeng and David writes basic scanner and parser, Chi writes basic test cases
- Mar 26: Finalizes scanner and parser; everyone agrees on syntax
- Mar 28: Code merge and Hello World compiles
- Apr 1: Met as a team to discuss future project plans, come up with project timeline and preliminary deadlines for team

- Apr 2: Jiangfeng and David starts to work on semantic checker; Nimo and Chi continue to work on Code gen
- Apr 16: Met as a team to come up with final demo idea: game of life
- Apr 30: Jiangfeng and David work on Sast
- May 1: Nimo and Chi work on demo game of life design
- May 5: Complete full test suite, code merge and prepare for final presentation
- May 5: LRM revision
- May 1-7: Demo build, final report, bug fixes
- May 8: Final presentation
- May 10: Finish final report

4.7.2 Git Commit History

The team's Git handles:
Jiangfeng: janewanggg
Nimo: wodeni
Chi: chyzhang
David: DRC9702
The full commit history is included in Appendix.

4.7.3 Active Branches



Active branches

merge-sast Updated 5 days ago by chyzhang

matrix Updated 5 days ago by DRC9702

travis Updated 5 days ago by wodeni

backend Updated 6 days ago by wodeni

BackAndSeman Updated 14 days ago by ZhangChi

scannerparser Updated a month ago by chyzhang

5 Architectural Design

5.1 Overview

The architecture of the MPL compiler consists of six major components: Scanner Parser AST/SAST Code Generation Semantic Checker C I/O library

5.2 Scanner

Contributions from Nimo, Jiangfeng and David
The scanner is responsible for reading in the mpl file and decomposing the full text of the source code into a series of prespecified tokens. Irrelevant characters such as the white spaces and commented sections are omitted in this phase. MPL also process string literals in the Scanner, properly replacing the escaped sequences by the actual characters: for example, from `\n` to the actual line feed character.

5.3 Parser

Contributions from Nimo, Jiangfeng and David
The Parser contains the Context-free grammar of the language. Using the tokens created by Scanner, the Parser walks through the stream of tokens and produces an instance of the Abstract Syntax Tree(AST). Normally, whenever there is a syntax error in the source program, it will be caught by the Parser. It does, however, accepts a superset of the MPL language, because both the Semantic Checker and Code Generator will check for errors, too.

5.4 Semantic Checker

Contributions from Jiangfeng and David
The Semantic Checker handles errors that are hard to detect by the Parser, and infer some extra information about the AST node. It then decorates the original AST by building a Semantically-checked Abstract Syntax Tree(SAST) based on the raw AST. Typical tasks of the semantic checker include verifying the type and dimensionalities of matrices, adding the type of identifiers to the SAST and etc.

5.5 Code Generator

Contributions from Nimo and Chi
The code generator receives a syntax tree as input from the semantic checker. Then by calling llvm building functions and c functions, we can generate a `.ll` file, which contains LLVM Intermidiate Representation (LLVM IR) Code. The `.ll` file is a platform independent representation of the actual assembly code. A command line tool, `llc`, can then be used to generate platform-specific assembly code.
One other task of the Code generator is to define entry points for the C function that MPL uses as built-in functions, mostly for I/O purposes.

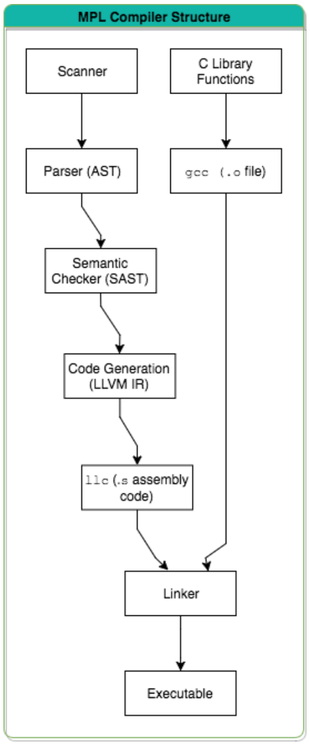


Figure 1: A diagram showing the flow of MPL. The source code is loaded to the scanner and the compiler generates an executable at the end

5.6 Linkage with the C IO Library

Contributions from Nimo, Chi, and David
Some of the built-in functions in MPL are implemented in C and

Layer	**Lead Developer**
Unit Tests	Chi
Scanner	Jane, David, Nimo
Parser	Jane, David, Nimo
Semantic	Jane, David
Codegen	Nimo, Chy
C Library	Nimo, David, Chy

Figure 2: Lead Developer of Each Layer.

compiled to a `.o` file. In the compilation process, `gcc` linker is used to connect the `.s` file and the `.o` files and produce a single executable file.

6 Testing

6.1 Representative Language Programs

6.1.1 GCD

GCD code

```
int gcd(){
  if(#C>#W) {
    return #C - #W;
  }
  else{
    return #C;
  }
}

int main() {
  int h;
  Mat<int> [1][2] m;
  m = [50, 40];
  while (m[0][0] !=  m[0][1]){
    gcd @ m;
  }
  h = m[0][0];
  print(h);
}
//print 10
```

GCD is porgrammed by using a 1-D matrix. The two integers that are tested are the entry of the matrix. By keep modifying these two entries with Euclid's algorithm, we can get the gcd of the two integers.

Game of Life

```
int evolve() {
    int i;
    int sum;
    i = 0;

    /* compute the number of neighbors alive */
    sum = #NW + #N + #NE + #W + #E + #SW + #S + #SE;

    if(#C == 1)
        if(sum == 2 || sum == 3) return 1;
        else return 0;
    else
        if(sum == 3) return 1;
        else return 0;
}

int main() {
    Mat<int>[220][900] board;
    matread("ship.bin", board);

    while(true) {
        print_board(board, 100);
        evolve @ board;
    }
}
//print a ship flying from right to left
```

This is a program running for Conway's Game of Life meaning that its evolution is determined by its initial state, requiring no further input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves, or, for advanced "players", by creating patterns with particular properties. The printboard function and matread function are implemented with C.

6.2 Testing Suite and Justification

The test for our language can be separated in two parts: Scanner Test and Program Test. The Scanner Test is built to check if Scanner works well and recognize everything we want. It helps a lot in early stage of the development. The Program Test are used to ensure that MPL works well under right language grammar. It is composed of success and failure test.We named The success tests are named with the prefix ẗest_änd fail tests are named with f̈ail_.̈ The .out files and .err file are the golden reference that are used to compare the result. All of these can be found in the test/ directory.

6.2.1 Scanner

To run scanner test, go into test/Scanner/scripts and run ./build.sh and then run ./test.sh. The success running will have output from ScannerTest.ml and compare with the .out file.

The sucess test for Scanner should be something that can be recongnized in our MPL, while the fail test will tell there shouldn't be such String appear. The test includes:

Sucess			
_arithmetic.out	_arithmetic.test	_assignment.out	_assignment.test
_base_scanner.out	_base_scanner.test	_comment.out	_comment.test
_conditionals.out	_conditionals.test	_control_flow.out	_control_flow.test
_delimiters.out	_delimiters.test	_function.out	_function.test
_identifier.out	_identifier.test	_literal.out	_literal.test
_main_function.out	_main_function.test	_matrix.out	_matrix.test
_misc.out	_misc.test	_types.out	_types.test
_mixed_arithmetic.out	_mixed_arithmetic.test		
fail:			
_illegal_carrot.out	_illegal_carrot.test	_illegal_dollar.out	_illegal_dollar.test
_illegal_percent.out	_illegal_percent.test	_illegal_period.out	_illegal_period.test
_illegal_pound.out	_illegal_pound.test	_illegal_tilde.out	_illegal_tilde.test

6.3 Program Tests

The Program tests are used to help check functionality of standard language aspects including built-in functions, while, if and variables. For MPL 's program is similar to microC. The test cases for the aspects of language are closed to microC's test. The testall.sh is modified to run our language effectively. To run test, go to MPL/src and "make", then go to MPL and run ./testall.sh. For MPL focus on the matrix, printm and apply /@ are the most important test cases.

6.3.1 Printm Test

```
//test-printm.mpl
int main(){
    Mat<int>[2][2] m;
    m = [1, 2; 3, 4];
    printm(m);
}
//golden reference for test-printm.mpl
[
1, 2;
3, 4;
]
```

The printm will print the matrix. It is important for when we run our matrix-flow program, it will be all about matrix. We test most of functions with this printm function.

6.3.2 Apply Test

```
//test-apply.mpl
int entryf() {
    return 1;
}

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    entryf @ m;
    printm(m);
    return 0;
}
//golden reference for test-apply.mpl
[
1, 1, 1;
1, 1, 1;
1, 1, 1;
]
//test-apply2.mpl
int entryf() {
    return #N;
}

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    entryf @ m;
    printm(m);
    return 0;
}
//golden reference for test-apply2.mpl
[
7, 8, 9;
1, 2, 3;
4, 5, 6;
]
```

Apply is the special way in MPL to modify matrix. The test on apply is very sucessful. In the early stage of apply, the matrix's entry was changed once the entry is passed. The neighbor of it (#S,#E,#SE) will read the modified the value, which should not happen. The origin apply can pass the first test of apply: test-apply.mpl. However it cannot pass the second apply test, test-apply2.mpl.

6.3.3 Other Success and Fail test

The other success and fail tests are listed below. Both of them guarantee the program can run under the right grammar and algorithm and give errors when the rules are violated.

Sucess			
test-all1.mpl	test-all1.out	test-all2.mpl	test-all2.out
test-apply.mpl	test-apply.out	test-apply2.mpl	test-apply2.out
test-func.mpl	test-func.out	test-if1.mpl	test-if1.out
test-if2.mpl	test-if2.out	test-if3.mpl	test-if3.out
test-if4.mpl	test-if4.out	test-if5.mpl	test-if5.out
test-local1.mpl	test-local1.out	test-local2.mpl	test-local2.out
test-mat.mpl	test-mat.out	test-matall.mpl	test-matall.out
test-matread.mpl	test-matread.out	test-matwrite.mpl	test-matwrite.out
est-ops1.mpl	test-ops1.out	test-print-board.mpl	test-print-board.out
test-print.mpl	test-print.out	test-printm.mpl	test-printm.out
test-prints.mpl	test-prints.out	test-var1.mpl	test-var1.out
test-while1.mpl	test-while1.out	test-while2.mpl	test-while2.out
test-imgread.mpl	test-imgread.out		
fail:			
fail-assign1.err	fail-assign1.mpl	fail-assign2.err	fail-assign2.mpl
fail-expr1.err	fail-expr1.mpl	fail-func1.err	fail-func1.mpl
fail-func4.err	fail-func4.mpl	fail-func5.err	ail-func5.mpl
fail-func6.err	fail-func6.mpl	fail-func7.err	fail-func7.mpl
fail-func9.err	fail-func9.mpl	fail-global1.err	fail-global1.mpl
fail-if1.err	fail-if1.mpl	fail-if2.err	fail-if2.mpl
fail-if3.err	fail-if3.mpl	fail-nomain.err	fail-nomain.mpl
fail-return1.err	fail-return1.mpl	fail-return2.err	fail-return2.mpl
fail-while1.err	fail-while1.mpl	fail-while2.err	fail-while2.mpl

7 Lesson Learned

7.1 Jiangfeng Wang

Though I was quite confused with Ocaml syntax at first, the project helped me feel very comfortable with programming in Ocaml. The class examples, microC and projects from previous semester are great sources to learn about Ocaml. When doing team project, it is important to stay in touch with the whole teams and to hold weekly meetings to inform others which step you are at and what problems you have encountered. Our group held a few coding Hackathons at the end of the semester and it went very well.

Last but not the least, I think that testing is the most important part of the project. We did not have the full test suite set up early and caused some issues when we tried to merge the code. Once language syntax is set up, test suite should be built up so that we can make sure every aspect of the language is considered. If the test fails, it's a great indication of the issues of the language. When we coded, we normally thought from the aspect of a coder, but testing gave us a different aspect of users, and thus can find different issues and forces programmers to think of the language in more details.

7.2 Chi Zhang

The most important thing I have learned is always try to learn code well. For this class, previous project can give student great help. But that doesn't mean people can copy and paste. The logic in Ocaml can be complex. Any structure and logic difference can break the flow easily. It is more important to read other's code and learn and feel why people write code like that. With great understanding of code, we improved our efficiency in the later stage of work. Also, specially for codegen, understand .ll file can be very important. It can be extremely difficult for people to debug facing codegen code. However, reading the .ll and seeing what you generate and checking what you should generate, then going back to codegen is always the short path.

7.3 Wode “Nimo” Ni

Starting early is definitely the ultimate solution to everything, but most of the time, it is not going to happen. At the early stage, it is truly difficult for anyone to take his/her own time to perform the setups and learn about things needed to start the project. I think people should start frequent meetings (more than 3 times a week) as soon as possible. Nobody will take the responsibility unless everyone is watching. There is a learning curve at the beginning, and people should overcome it as a group, not lazy individuals. It is also vital for all the team members to become familiar with all the tools, for example, `git` for version control, as soon as possible,

This project did provide me with a solid foundation of functional programming, and more importantly, a new way of thinking as a maker of the language, not the user. This one more level of abstraction changed the way I look at programming languages, and definitely furthered my understanding of them.

I also learned about the importance of writing clean, well-documented, and possibly elegant code. When the code base started to expand, it became increasingly hard to understand others’ work, and sometimes even my own work. At the end, it was very rewarding to see our Game of Life simulation program working with only twenty lines of code or so. Beautiful stuff.

7.4 David Rincon-Cruz

I think one of the most important lessons in development I learned was how to explore an unknown search space with a team and to become comfortable with Ocaml and higher level concepts by experimenting.

Pair programming and task division was crucial to the success of my team as it allowed active code-reviewing, shared ownership, and sanity checks. Developing individually wasn’t as effective as catch-up time was always necessary.

Another important practice to our development was the emphasis of segmenting the code into readable sections. Halfway through the development, the semantic side of the code underwent complete refactoring to create a framework that was easy to expand and allowed testing of individual components. This also made it easy to develop code simultaneously and allowed expansion of functionality to be divided into explicit features.

Ultimately, this project helped me develop an appreciation for the work a compiler does from a theoretical point of view. Independent of code generation, it answers as many questions it can without becoming undecidable. "Useless" compiler messages are ultimately the result of limitations on static analysis, not a shortcoming on the development. Just returning the faulty line number is a feat I can truly appreciate the scope of.

8 Acknowledgements

We would like to express our special thanks of gratitude to our professor Edwards as well as our TA Julie who taking time to meet with us, helped our team get on the right track and gave us the opportunity to do this wonderful project, and we came to know plenty of new things.

9 Appendix

9.1 Source Code

This section lists all source files of the MPL Compiler.

- mpl.ml
- ast.ml
- codegen.ml
- scanner.mll
- parser.mly
- sast.ml
- utils.c

```
(*
 * File: mpl.mll
 * Date: 2017-03-11
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni    <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang         <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 *
 * Top-level of the mpl compiler: scan & parse the input,
 * check the resulting AST, generate LLVM IR, and dump the module
 *)

type action = Ast | LLVM_IR | Compile

let _ =
  let action = if Array.length Sys.argv > 1 then
    List.assoc Sys.argv.(1) [ ("-a", Ast);  (* Print the AST only *)
                              ("-l", LLVM_IR);  (* Generate LLVM, don't check *)
                              ("-c", Compile) ] (* Generate, check LLVM IR *)
  else Compile in
  let lexbuf = Lexing.from_channel stdin in
  let ast    = Parser.program Scanner.token lexbuf in
  let sast   = Semant.check ast
  in
  (match action with
    (*  Ast -> print_string (Ast.string_of_program ast) | *)
    LLVM_IR -> print_string (Llvm.string_of_llmodule (Codegen.translate sast))
  | Compile -> let m = Codegen.translate sast in
    (* Llvm_analysis.assert_valid_module m;  *)
    print_string (Llvm.string_of_llmodule m))
}
```

```
(*
 * File: scanner.mll
 * Date: 2017-03-11
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni    <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang         <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 *)

{ open Parser

exception LexError of string

(* string parsing from OCaml compiler code :-) *)
let string_buff = Buffer.create 256
let reset_string_buffer () = Buffer.clear string_buff
let store_string_char c = Buffer.add_char string_buff c
let store_string_snip str = Buffer.add_string string_buff str
let get_stored_string () = Buffer.contents string_buff

let char_for_backslash = function
    'n' -> '\n'
  | 't' -> '\t'
  | 'b' -> '\b'
  | 'r' -> '\r'
  | c    -> c

let decimal_code c d u =
  100 * (Char.code c - 48) + 10 * (Char.code d - 48) + (Char.code u - 48)

let char_for_hexadecimal_code d u =
  let d1 = Char.code d in
  let val1 = if d1 >= 97 then d1 - 87
             else if d1 >= 65 then d1 - 55
             else d1 - 48
  in
  let d2 = Char.code u in
  let val2 = if d2 >= 97 then d2 - 87
             else if d2 >= 65 then d2 - 55
             else d2 - 48
  in
  Char.chr (val1 * 16 + val2)

let lex_warning lexbuf msg =
  let p = Lexing.lexeme_start_p lexbuf in
  Printf.eprintf "MPL warning:\nFile \"%s\", line %d, character %d: %s.\n"
    p.Lexing.pos_fname p.Lexing.pos_lnum
    (p.Lexing.pos_cnum - p.Lexing.pos_bol + 1) msg;
  flush stderr

let incr_loc lexbuf delta =
  let pos = lexbuf.Lexing.lex_curr_p in
  lexbuf.Lexing.lex_curr_p <- { pos with
    Lexing.pos_lnum = pos.Lexing.pos_lnum + 1;
    Lexing.pos_bol = pos.Lexing.pos_cnum - delta;
  }
;;
}

let newline = '\n' | "\r\n"
let whitespace = [' ' '\t']
let consecutive_strings = ['"'] whitespace* ['"']
let backslash_escapes = ['\\' '"' '\'' 'n' 't' 'b' 'r']
let digit      = ['0'-'9']
let integer    = digit+
let exp = (’e’ | ’E’) (’+’ | ’-’)? digit+
let float_re = ’.’ digit+ exp? | digit+ (’.’ digit* exp? | exp)

rule token = parse
  newline            { Lexing.new_line lexbuf; token lexbuf }
| whitespace         { token lexbuf }
| "/*"               { comment 0 lexbuf }           (* Comments *)
| ’"’              { reset_string_buffer ();    (* String literals *)
      parse_string lexbuf;
      (*handle_lexical_error string lexbuf;*)
      STRLIT(get_stored_string ()) }
(* Punctuation *)
| ’(’      { LPAREN }
| ’)’      { RPAREN }
| ’{’      { LBRACE }
| ’}’      { RBRACE }
| ’[’      { LBRACKET }
| ’]’      { RBRACKET }
| ’;’      { SEMI }
| ’,’      { COMMA }

(* Operators *)
| ’+’      { PLUS }
| ’-’      { MINUS }
| ’*’      { TIMES }
| ’/’      { DIVIDE }
| ’=’      { ASSIGN }
| "=="     { EQ }
| "!="     { NEQ }
| ’<’      { LT }
| "<="     { LEQ }
| ">"      { GT }
| ">="     { GEQ }
| "&&"    { AND }
| "||"     { OR }
| "!"    { NOT }
(* | "neg"     { NEG } *)
| "@"      { APPLY }
| ".@"     { MATAPP }
| "^"      { TRANS }
| ".*"     { EMULT }
| "./"     { EDIV }

(* Keywords *)
| "if"     { IF }
| "else"   { ELSE }
| "elseif" { ELSEIF }
(* | "for"    { FOR } *)
| "while"  { WHILE }
| "return" { RETURN }
| "int"    { INT }
| "float"  { FLOAT }
| "boolean"{ BOOL }
| "string" { STRING }
| "void"   { VOID }
| "true"   { TRUE }
| "false"  { FALSE }
| "func"   { FUNC }
| "null"   { NULL }
| "new"    { NEW }
| "\#C"     { CENTER }
| "\#N"     { NORTH }
| "\#S"     { SOUTH }
| "\#W"     { WEST }
| "\#E"     { EAST }
| "\#NW"    { NWEST }
| "\#NE"    { NEAST }
| "\#SW"    { SWEST }
| "\#SE"    { SEAST }


(* Built-in Types *)
| "Img"    { IMG }
| "Mat"    { MAT }
| "fMat"   { FMAT }

(* Integer literals, identifiers, and others *)
| integer as lxm { INTLIT(int_of_string lxm) }
| float_re as lxm { FLOATLIT(float_of_string lxm) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']* as lxm { ID(lxm) }
| eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped char)) }

(*
and comment = parse
  "*/" { token lexbuf }
| _    { comment lexbuf }
*)

(* Block comment parsing: nested comments are allowed *)
and comment level = parse
    "*/"  { if level = 0 then token lexbuf
            else comment (level-1) lexbuf }
  | newline { Lexing.new_line lexbuf; comment level lexbuf }
  | "/*"    { comment (level+1) lexbuf }
  | eof     { raise (LexError("unterminated comment!")) }
  | _       { comment level lexbuf }

(* fancy string parsing from OCaml compiler code :-) *)
and parse_string = parse
    consecutive_strings  { parse_string lexbuf }
  | '"'     { () }
  | newline { Lexing.new_line lexbuf; parse_string lexbuf }
  | '\\' ("\010" | "\013" | "\013\010") ([' ' '\009'] * as spaces)
    { incr_loc lexbuf (String.length spaces);
      parse_string lexbuf }
  | '\\' (backslash_escapes as c)
    { store_string_char(char_for_backslash c);
      parse_string lexbuf }
  | '\\' 'x' (['0'-'9' 'a'-'f' 'A'-'F'] as d) (['0'-'9' 'a'-'f' 'A'-'F'] as u)
    { store_string_char (char_for_hexadecimal_code d u) ;
      parse_string lexbuf }
  | '\\' (['0'-'9'] as c) (['0'-'9'] as d) (['0'-'9']  as u)
    { let v = decimal_code c d u in
      if v > 255 then
       lex_warning lexbuf
        (Printf.sprintf
          "illegal backslash escape in string: `\\%c%c%c'" c d u) ;
      store_string_char (Char.chr v);
      parse_string lexbuf }
  | '\\' (_ as c)
    { lex_warning lexbuf
        (Printf.sprintf "illegal backslash escape in string: `\\%c'" c) ;
      store_string_char '\\' ;
      store_string_char c ;
      parse_string lexbuf }
  | '\010'
    { store_string_char '\010';
      incr_loc lexbuf 0;
      parse_string lexbuf }
  | eof { raise(LexError("unterminated string")) }
  | _ as c
    { store_string_char c;
      parse_string lexbuf }
```

```

(*
 * File: ast.ml
 * Date: 2017-03-11
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni    <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang         <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 *)

type op = Add | Sub | Mult | Div | Equal | Neq | Less | Leq | Greater | Geq |
          And | Or | Apply | Matapp | Emult | Ediv

type uop = Neg | Not

type typ = Int | Bool | Float | Void | String (*Sorry Nimo*)
                  | Mat of typ * int * int
                  | FMat of typ * int * int
                  | Img

(*type num = IntLit of int | FloatLit of float*)

type bind = typ * string
type var_dec = typ * string

type expr =
    IntLit of int
  | FloatLit of float
  | BoolLit of bool
  | MatrixLit of expr list list
  | FMatrixLit of string list list
  | StrLit of string
  | Id of string
  | Binop of expr * op * expr
  | Unop of uop * expr
  | Assign of string * expr
  | Call of string * expr list
  | Noexpr
  | Null
  | MatrixAccess of string * expr * expr (*changed string*int*int *)

type stmt =
    Block of stmt list
  | Expr of expr
  | Return of expr
  | If of expr * stmt * stmt
(*  | For of expr * expr * expr * stmt *)
  | While of expr * stmt

type func_decl = {
    typ      : typ;
    fname    : string;
    formals  : bind list;
    locals   : bind list;
    body     : stmt list;
  }

type program = func_decl list

(* Pretty-printing functions *)
let string_of_op = function
    Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Equal -> "=="
  | Neq -> "!="
  | Less -> "<"
  | Leq -> "<="
  | Greater -> ">"
  | Geq -> ">="
  | And -> "&&"
  | Or -> "||"

let string_of_uop = function
    Neg -> "-"
  | Not -> "!"

let rec string_of_expr = function
    IntLit(l) -> string_of_int l
  | FloatLit(l) -> string_of_float l
  | BoolLit(true) -> "true"
  | BoolLit(false) -> "false"
  | Id(s) -> s
  | Binop(e1, o, e2) ->
      string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_expr e2
  | Unop(o, e) -> string_of_uop o ^ string_of_expr e
  | Assign(v, e) -> v ^ " = " ^ string_of_expr e
  | Call(f, el) ->
      f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^ ")"
  | Noexpr -> ""

let rec string_of_stmt = function
    Block(stmts) ->
      "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
  | Expr(expr) -> string_of_expr expr ^ ";\n";
  | Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
  | If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^ string_of_stmt s
  | If(e, s1, s2) ->  "if (" ^ string_of_expr e ^ ")\n" ^
      string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
  | While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt s

let rec string_of_typ = function
    Int -> "int"
  | Float -> "float"
  | Bool -> "bool"
  | Void -> "void"
  | Mat(t, i1, j1) -> "Mat <"^ string_of_typ t ^ "> ("^ string_of_int i1 ^ ", " ^ string_of_int j1 ^ ")"
  | FMat(t, i1, j1) -> "FMat <"^ string_of_typ t ^ "> ("^ string_of_int i1 ^ ", " ^ string_of_int j1 ^ ")"

let string_of_vdecl (t, id) = string_of_typ t ^ " " ^ id ^ ";\n"

let string_of_fdecl fdecl =
  string_of_typ fdecl.typ ^ " " ^
  fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.locals) ^
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "}\n"

let string_of_program (vars, funcs) =
  String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^
  String.concat "\n" (List.map string_of_fdecl funcs)
```

```
(*
 * File: sast.mly
 * Date: 2017-04-11
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni    <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang         <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 *)

open Ast

type sexpr =
    SIntLit of int
  | SFloatLit of float
  | SBoolLit of bool
  | SStrLit of string
  | SMatrixLit of sexpr list list * typ
  | SFMatrixLit of string list list * typ
  | SId of string * typ
  | SBinop of sexpr * op * sexpr * typ
  | SUnop of uop * sexpr * typ
  | SAssign of string * sexpr * typ
  | SCall of string * sexpr list * typ
  | SNull of typ
  | SMatrixAccess of string * sexpr * sexpr * typ

let get_expr_type_info sexpr = match sexpr with
   |SIntLit _ ->                Int
   |SFloatLit _ ->              Float
   |SBoolLit _ ->               Bool
   |SMatrixLit (_,x) ->         x
   |SFMatrixLit(_,x) ->         x
   |SId (_,x) ->                x
   |SStrLit _ ->                String
   |SBinop (_,_,_,x) ->         x
   |SUnop (_,_,x) ->            x
   |SAssign (_,_,x) ->          x
   |SCall (_,_,x) ->            x
   |SNull x ->                  x
   |SMatrixAccess (_,_,_,x) ->  x


type sstmt =
    SBlock of sstmt list
  | SExpr of sexpr
  | SIf of sexpr * sstmt * sstmt
  (*| SFor of sexpr * sexpr * sexpr * sstmt*)
  | SWhile of sexpr * sstmt
  | SReturn of sexpr

type sfunc_decl = {
  styp          : typ;
  sfname      : string;
  sformals    : bind list;
  slocals     : bind list;
  sbody       : sstmt list;
}

type sprogram = sfunc_decl list
```

```ocaml

(*
 * File: sement.ml
 * Date: 2017-03-28
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni    <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang         <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 *)

open Ast
open Sast
(*module StringMap = Map.Make(String)*)
module FuncMap = Map.Make(String)

(* Semantic checking of a program. Returns void if successful,
   throws an exception if something is wrong.

   Check each global variable, then check each function *)
let requireIntegers tlist str =
    let _ = List.map(
          fun t ->  match t with
            IntLit(_) -> true
          | _ -> raise (Failure(str))
    ) tlist in
    true

let requireFloats tlist str =
    let _ = List.map(
        fun t ->  match t with
            FloatLit(_) -> true
          | _ -> raise (Failure(str))
    ) tlist in
    true

let requireBools tlist str =
    let _ = List.map(
        fun t ->  match t with
            BoolLit(_) -> true
          | _ -> raise (Failure(str))
    ) tlist in
    true

let requireAllMatrices tlist str =
    let _ = List.map(
        fun t -> match t with
            Mat(typ1, i1, j1) -> true
          | _ -> raise (Failure(str))
    ) tlist in
    true

let checkAllMatrixLiterals d2list str =
    let i = List.length d2list in
    let j = List.length (List.hd d2list) in
    let t = List.hd (List.hd d2list) in
        match t with
            IntLit(_) -> List.map (fun lst -> requireIntegers lst str) d2list; Mat(Int, i, j)
          | FloatLit(_) -> List.map (fun lst -> requireFloats lst str) d2list; Mat(Float, i, j)
          | BoolLit(_) -> List.map (fun lst -> requireBools lst str) d2list; Mat(Bool, i, j)
          | _ -> raise (Failure("Matrix literals must be of the same type"))

let rec checkUnique lst =
  if (List.length lst)==1 then true else ((List.hd lst) ==(List.nth lst 1) && (checkUnique(List.tl lst)))

let checkMatrixDimensions d2list str =
    if ((checkUnique (List.map List.length d2list))==true) then true else raise(Failure(str))

let getArithBinopType t1 t2 op =
  match(t1, t2) with
  (Int, Int) -> Int
  | (Float, Float) -> Float
  | (Int, Mat(typ, i, j)) -> (match op with
                                Mult -> Mat(typ, i, j)
                                | _ -> raise(Failure("Only valid operation between int and matrix is multiplication.")))
  | (Float, Mat(typ, i, j)) -> (match op with
                                Mult -> Mat(typ, i, j)
                                | _ -> raise(Failure("Only valid operation between float and matrix is multiplication.")))
  | (Mat(typ, i, j),Int) -> (match op with
                                Mult -> Mat(typ, i, j)
                                | _ -> raise(Failure("Only valid operation between int and matrix is multiplication.")))
  | (Mat(typ, i, j),Float) -> (match op with
                                Mult -> Mat(typ, i, j)
                                | _ -> raise(Failure("Only valid operation between float and matrix is multiplication.")))
  | (Mat(typ1, i1, j1), Mat(typ2, i2, j2)) ->
    (match op with
      Add | Sub -> if typ1=typ2 && i1=i2 && j1=j2 then Mat(typ1, i1, j1)
            else raise(Failure("Matrices must be of same type and dimensions for +/-"))
      | Mult -> if typ1=typ2 && j1=i2 then Mat(typ1, i1, j2)
            else raise(Failure("M1(a,b) and M2(c,d) must have b=c for *"))
      | Emult|Ediv -> if typ1=typ2 && i1=i2 && j1=j2 then Mat(typ1, i1, j2)
            else raise(Failure("M1(a,b) and M2(c,d) must have matching dimensions .* and ./"))
      | _ -> raise(Failure("No matrices division")))
  | _ -> raise(Failure("Invalid type for arithmetic operand"))

(* let getLogicalBinopType t1 t2 op = function *)
let getLogicalBinopType t1 t2 op =
  match (t1, t2) with
  (Int, Int) -> Bool
  | (Float, Float) -> Bool
  | (Bool, Bool) -> Bool
  | _ -> raise(Failure("Invalid type for logical operand"))

let getEqualityBinopType t1 t2 op =
  match (t1, t2) with
    (Int, Int) -> Bool
  | (Float, Float) -> Bool
  | _ -> raise(Failure("Invalid type for logical operand"))


(*fd is where you feed function_decls*)
let checkFunction fd s = try FuncMap.find s fd
       with Not_found -> raise (Failure ("unrecognized function " ^ s))

let checkApply t1 t2 op fd =
    let func = checkFunction fd t1 in
         let t21 = Sast.get_expr_type_info t2 in
        match t21 with
        Mat(typ,_,_) -> if func.typ==typ then t21 else raise(Failure("Function and Matrix Type don't match for apply"))
        | _ -> raise(Failure("T2 must be a matrix type"))

(*Checks that a list of strings are functions with a specific type*)
let requireFunctionsWithType tlist fd typ =
    let _ = List.map (fun func -> if func.typ==typ then () else raise(Failure(""^func.fname^" has wrong type: "^(string_of_typ typ)))) (List.map (fun s -> checkFunction fd s) tlist)
    in true

(*Checks that a Fmatrix of functions is homogeneous*)
let checkFMatFunctions d2list fd =
    let i = List.length d2list in
    let j = List.length (List.hd d2list) in
    let func = checkFunction fd (List.hd (List.hd d2list)) in
    let typ = func.typ in
    let _ = List.map (fun lst -> requireFunctionsWithType lst fd typ) d2list in FMat(typ,i,j)


let checkBinop op e1 e2 fd=
  let t1 = Sast.get_expr_type_info e1 and t2 = Sast.get_expr_type_info e2 in
  match op with
  Add | Mult | Sub | Div -> getArithBinopType t1 t2 op
  | Equal | Neq -> getEqualityBinopType t1 t2 op
  | And | Or -> getLogicalBinopType t1 t2 op
  | Less | Leq | Greater | Geq -> getEqualityBinopType t1 t2 op
  | _ -> raise(Failure("Invalid operand in getBinopType"))


let checkMatIndex m i j = match(m,i,j) with
  (Mat(typ,row,col), IntLit(i1), IntLit(j1)) -> if (i1<0)||(i1>=row)
                                    then raise(Failure("Out of bounds access - row:"^(string_of_int row)^" i:"^(string_of_expr i)))
                                    else if (j1<0)||(j1>=col)
                                        then raise(Failure("Out of bounds access - col:"^(string_of_int col)^" j:"^(string_of_expr j)))
                                        else typ
  |(Mat(typ,row,col), Id(_), Id(_)) -> typ
  |(Mat(typ,row,col), IntLit(i1), Id(_)) -> if (i1<0)||(i1>=row)
                                    then raise(Failure("Out of bounds access - row:"^(string_of_int row)^" i:"^(string_of_expr i)))
                                    else typ
  |(Mat(typ,row,col), Id(_), IntLit(j1)) -> if (j1<0)||(j1>=col)
                                            then raise(Failure("Out of bounds access - col:"^(string_of_int col)^" j:"^(string_of_expr j)))
                                            else typ
  | _-> raise(Failure("Invalid arguments in accessing matrix"))

let numlitToSlit n = match n with
    IntLit(n) -> SIntLit(n)
  | FloatLit(n) -> SFloatLit(n)
let getLitType n = match n with
    IntLit(_) -> Int
  | FloatLit(_) -> Float
  | _ -> raise(Failure("Must be int or float type"))

let getString s = match s with
    SId(n, _) -> n
  | _ -> raise(Failure("Cannot find symbol"))

let mlitToSmlit n =
    let t = getLitType(List.hd (List.hd n)) in
    let slit = (List.map (fun nl -> (List.map numlitToSlit nl)) n) in
    let r = List.length n and c = List.length (List.hd n) in
    SMatrixLit(slit, Mat(t,r,c))
(*
let get_expr_type_info sexpr = match sexpr with
    |SIntLit _ ->                Int
    |SFloatLit _ ->              Float
    |SBoolLit _ ->               Bool
    |SMatrixLit (_,x) ->         x
    |SId (_,x) ->               x
    |SStrLit _ ->               String
    |SBinop (_,_,_,x) ->        x
    |SUnop (_,_,x) ->           x
    |SAssign (_,_,x) ->         x
    |SCall (_,_,x) ->           x
    |SNull ->                   Void
    |SMatrixAccess (_,_,_,x) -> x

let exprToSexpr expr = match expr with
    | IntLit(n)  ->             SIntLit(IntLit(n))
    | FloatLit(n) ->            SFloatLit(FloatLit(n))
    | BoolLit(n) ->             SBoolLit(BoolLit(n))
    | MatrixLit(n) ->           checkMatrixLit n
    | Id(n) ->                  SId(n, getIdType n)
    | StrLit(n) ->              StrLit(n)
    | Binop(e1, op, e2) ->      checkBinop e1 op e2
    | Unop(op, e) ->            checkUnop op e
    | Assign(s, e) ->           checkAssign s e
    | Call(s, e) ->             SCall(s, (List.map exprToSexpr e), )
    | Null ->                   SNull
    | MatrixAccess(s, e1, e2) -> checkMatrixAccess s e1 e2
*)
let check (functions) =

  (* Raise an exception if the given list has a duplicate *)
  let report_duplicate exceptf list =
    let rec helper = function
  n1 :: n2 :: _ when n1 = n2 -> raise (Failure (exceptf n1))
      | _ :: t -> helper t
      | [] -> ()
    in helper (List.sort compare list)
  in

  (* Raise an exception if a given binding is to a void type *)
  let check_not_void exceptf = function
      (Void, n) -> raise (Failure (exceptf n))
    | _ -> ()
  in

  (* Raise an exception of the given rvalue type cannot be assigned to
     the given lvalue type *)
  let check_assign lvaluet rvaluet err =
  match (lvaluet,rvaluet) with
      Mat(t1, i1, j1), Mat(t2, i2, j2) -> if t1=t2 && i1=i2 && j1=j2 then lvaluet else raise err
     | _-> if lvaluet == rvaluet then lvaluet else raise err
  in

  (**** Checking Global Variables ****)
(*
  List.iter (check_not_void (fun n -> "illegal void global " ^ n)) globals;

  report_duplicate (fun n -> "duplicate global " ^ n) (List.map snd globals);
*)
  (**** Checking Functions ****)
  if List.mem "print" (List.map (fun fd -> fd.fname) functions)
  then raise (Failure ("function print may not be defined")) else ();

  report_duplicate (fun n -> "duplicate function " ^ n)
    (List.map (fun fd -> fd.fname) functions);

  (* Function declaration for a named function *)
  let built_in_decls =  FuncMap.add "print"
     { typ = Void; fname = "print"; formals = [];
       locals = []; body = [] }
     (FuncMap.add "printb" { typ = Void; fname = "printb"; formals = [(Bool, "x")];
       locals = []; body = [] }
     (FuncMap.add "prints" { typ = Void; fname = "prints"; formals = [(String, "x")];
       locals = []; body = [] }
     (FuncMap.add "printm" { typ = Void; fname = "printm"; formals = [];
       locals = []; body = [] }
     (FuncMap.add "matread"{ typ = Void; fname = "matread"; formals = [];
        locals = []; body = []}
     (FuncMap.add "pgmread"{ typ = Void; fname = "pgmread"; formals = [];
        locals = []; body = []}
     (FuncMap.add "ppmread"{ typ = Void; fname = "pgmread"; formals = [];
        locals = []; body = []}
     (FuncMap.add "matwrite"{ typ = Void; fname = "matwrite"; formals = [];
        locals = []; body = []}
     (FuncMap.add "pgmwrite"{ typ = Void; fname = "pgmwrite"; formals = [];
        locals = []; body = []}
     (FuncMap.add "ppmwrite"{ typ = Void; fname = "pgmwrite"; formals = [];
        locals = []; body = []}
     (FuncMap.singleton "print_board"{ typ = Void; fname = "print_board"; formals = [];
        locals = []; body = []}
     ))))))))))
   in

  let function_decls = List.fold_left (fun m fd -> FuncMap.add fd.fname fd m)
                         built_in_decls functions
  in

  let function_decl s = try FuncMap.find s function_decls
       with Not_found -> raise (Failure ("unrecognized function " ^ s))
  in

  let _ = function_decl "main" in (* Ensure "main" is defined *)

let check_function func =
    let module StringMap = Map.Make(String) in
(*
    List.iter (check_not_void (fun n -> "illegal void formal " ^ n ^
      " in " ^ func.fname)) func.formals;

    report_duplicate (fun n -> "duplicate formal " ^ n ^ " in " ^ func.fname)
      (List.map snd func.formals);
*)
    List.iter (check_not_void (fun n -> "illegal void local " ^ n ^
      " in " ^ func.fname)) func.locals;

    report_duplicate (fun n -> "duplicate local " ^ n ^ " in " ^ func.fname)
      (List.map snd func.locals);

    let neighbor_names = [ "\#NW"; "\#N"; "\#NE"; "\#W"; "\#C"; "\#E"; "\#SW"; "\#S"; "\#SE" ]
    in

    (* Type of each variable (global, formal, or local *)
    let symbols = List.fold_left (fun m s -> StringMap.add s func.typ m) (List.fold_left (fun m (t, n) -> StringMap.add n t m)
  StringMap.empty (func.locals )) neighbor_names
    in

    let initSyms = ref( List.fold_left (fun m (t, n) -> StringMap.add n false m) StringMap.empty (func.locals))
    in

    let type_of_identifier s =
      try StringMap.find s symbols
      with Not_found -> raise (Failure ("undeclared identifier " ^ s))
    in

    let updateInitSyms var =
        initSyms := StringMap.add var true !initSyms
    in
    let checkInitSyms n =
        let i = StringMap.find n !initSyms in
        if i == false then raise(Failure("The variable " ^ n ^ " has not been initialized"))
        else true
     in

    (* Return the type of an expression or throw an exception *)
    let rec expr = function
       IntLit(n) -> SIntLit(n)
      | FloatLit(n)  -> SFloatLit(n)
      | BoolLit(n) -> SBoolLit(n)
      | StrLit(n) -> SStrLit(n)
      | Id(s) -> SId(s, type_of_identifier s)
      | MatrixLit s ->  ((checkMatrixDimensions s "Malformed matrix"; checkAllMatrixLiterals s "All matrix literals must be of the same type")); (mlitToSmlit s)
      | MatrixAccess(m,i,j) -> let t = checkMatIndex (type_of_identifier m) i j in
            SMatrixAccess(m, numlitToSlit i, numlitToSlit j, t)
      | Binop(e1, op, e2) as e ->
             (match (e1,op) with
             (Id s,Apply) -> SBinop(SId(s,get_expr_type_info (expr e2)), op, (expr e2), checkApply s (expr e2) op function_decls)
            | _ -> let t1 = expr e1 and t2 = expr e2 in
          SBinop(t1, op, t2, checkBinop op t1 t2 function_decls))
      | Unop(op, e) as ex -> SUnop(op, (expr e), (let t1 = expr e in
                                                  let t = Sast.get_expr_type_info t1 in
   (match op with
     Neg when t = Int -> Int
   | Neg when t = Float -> Float
   | Not when t = Bool -> Bool
         | _ -> raise (Failure ("illegal unary operator"))) ))
      (*| Noexpr -> Void*)
      | Assign(var, e) as ex -> SAssign(var, (expr e), (let lt = type_of_identifier var
                                and rt1 = (expr e) in
                                let  rt = (Sast.get_expr_type_info rt1) in
                                let  ret = check_assign lt rt (Failure ("illegal assignment " ^ string_of_typ lt ^
             " = " ^ string_of_typ rt )) in
                                let _ = updateInitSyms var in
                                ret ))
      | Call("printm", actuals) -> let a = (List.map expr actuals) in
                        let s = getString (List.hd a) in
                        SCall("printm", a, (if (List.length actuals != 1)
                                           then raise(Failure("Too many arguments to printm"))
                                           else let t1 = expr (List.hd actuals) in
                                           let _ = checkInitSyms s in
                                           let t = Sast.get_expr_type_info t1 in
                                           (match t with
                                           |Mat(Int,_,_) -> Void
                                           |Mat(Float,_,_) -> Void
                                           |_ -> raise(Failure("Wrong argument type. [Not mat<int> or mat<float>]")))
                                    )) (*match scall*)
      | Call("print", actuals) -> let a = (List.map expr actuals) in
                                        if (List.length actuals != 1)
                                           then raise(Failure("Too many arguments to print"))
                                           else (
                                           let l = List.hd a in
                                           (match l with
                                           SIntLit(_) -> SCall("print", a, Int)
                                           | SFloatLit(_) ->  SCall("print", a, Float)
                                           | SBoolLit(_) -> SCall("print",a,Bool)
                                           | SBinop(_,_,_,x) -> SCall("print", a, x)
                                           | _ -> SCall("print", a,
                                           (let t1 = expr (List.hd actuals) in
                                           let s = getString (List.hd a) in
                                           let _ = checkInitSyms s in
                                           let t = Sast.get_expr_type_info t1 in
                                           (match t with
                                           |Int -> Int
                                           |Float -> Float
                                           |Bool -> Bool
                                           |_ -> raise(Failure("Wrong argument type. [int or float or bool]")))
                                    )))) (*match scall*)
      | Call("prints", actuals) -> let a = (List.map expr actuals) in

                                        if (List.length actuals != 1)
                                           then raise(Failure("Too many arguments to prints"))
                                           else(
                                              let str = List.hd actuals in
                                              (match str with
                                              StrLit(_)->SCall("prints", a, String)
                                            | _ -> (SCall("prints", a,
                                                (let t1 = expr (List.hd actuals) in
                                                let s = getString (List.hd a) in
                                                let _ = checkInitSyms s in
                                                let t = Sast.get_expr_type_info t1 in
                                                    (match t with
                                                    |String -> String
                                                    |_ -> raise(Failure("Wrong argument type. [String]")))
                                    ))))) (*match scall*)
      | Call("matread", actuals) -> let a = (List.map expr actuals) in
              SCall("matread", a,    (if (List.length actuals != 2)
                                    then raise(Failure("matread only accepts 2 arguments"))
                                    else ( let a1 = List.hd actuals and a2 = expr (List.nth actuals 1) in
                                    let t2 = Sast.get_expr_type_info a2 in
                                    let var = getString a2 in
                                        match (a1,t2) with
                                        StrLit(_),Mat(_,_,_) -> updateInitSyms var; Int
                                        | _ -> raise(Failure("matread takes string literal and matrix type"))
                                     ) ))
     | Call("matwrite", actuals) -> let a = (List.map expr actuals) in
              SCall("matwrite", a,     (if (List.length actuals != 2)
                                    then raise(Failure("matwrite only accepts 2 arguments"))
                                    else ( let a1 = List.hd actuals and a2 = expr (List.nth actuals 1) in
                                    let t2 = Sast.get_expr_type_info a2 in
                                    let var = getString a2 in
                                        match (a1,t2) with
                                        StrLit(_),Mat(_,_,_) -> updateInitSyms var; Void
                                        | _ -> raise(Failure("matwrite takes string literal and matrix type"))
                                     ) ))
      | Call("pgmread", actuals) -> let a = (List.map expr actuals) in
              SCall("pgmread", a,    (if (List.length actuals != 2)
                                    then raise(Failure("pgmread only accepts 2 arguments"))
                                    else ( let a1 = List.hd actuals and a2 = expr (List.nth actuals 1) in
                                    let t2 = Sast.get_expr_type_info a2 in
                                    let var = getString a2 in
                                        match (a1,t2) with
                                        StrLit(_),Mat(_,_,_) -> updateInitSyms var; Int
                                        | _ -> raise(Failure("pgmread takes string literal and matrix type"))
                                     ) ))
      | Call("ppmread", actuals) -> let a = (List.map expr actuals) in
              SCall("ppmread", a,    (if (List.length actuals != 4)
                                    then raise(Failure("ppmread only accepts 4 arguments"))
                                    else ( let a1 = List.hd actuals and a2 = expr (List.nth actuals 1)
                                    and a3 = expr(List.nth actuals 2) and a4 = expr(List.nth actuals 3)
                                    in
                                    let t2 = Sast.get_expr_type_info a2 and t3 = Sast.get_expr_type_info a3
                                    and t4 = Sast.get_expr_type_info a4
                                    in
                                    let var2 = getString a2 and var3 = getString a3 and var4 = getString a4 in
                                        match (a1,t2,t3,t4) with
                                        StrLit(_),Mat(_,x2,y2),Mat(_,x3,y3),Mat(_,x4,y4) ->
                                            if (x2!=x3)||(x3!=x4)||(y2!=y3)||(y3!=y4) then raise(Failure("All matrices must have the same dimensions")) else (); updateInitSyms var2;
                                        updateInitSyms var3; updateInitSyms var4; Int
                                        | _ -> raise(Failure("ppmread takes string literal and 3 matrix types"))
                                     ) ))
     | Call("pgmwrite", actuals) -> let a = (List.map expr actuals) in
              SCall("pgmwrite", a,    (if (List.length actuals != 2)
                                    then raise(Failure("pgmwrite only accepts 2 arguments"))
                                    else ( let a1 = List.hd actuals and a2 = expr (List.nth actuals 1) in
                                    let t2 = Sast.get_expr_type_info a2 in
                                    let var = getString a2 in
                                        match (a1,t2) with
                                        StrLit(_),Mat(_,_,_) -> updateInitSyms var; Void
                                        | _ -> raise(Failure("pgmwrite takes string literal and matrix type"))
                                     ) ))
      | Call("ppmwrite", actuals) -> let a = (List.map expr actuals) in
              SCall("ppmwrite", a,    (if (List.length actuals != 4)
                                    then raise(Failure("ppmwrite only accepts 4 arguments"))
                                    else ( let a1 = List.hd actuals and a2 = expr (List.nth actuals 1)
                                    and a3 = expr(List.nth actuals 2) and a4 = expr(List.nth actuals 3)
                                    in
                                    let t2 = Sast.get_expr_type_info a2 and t3 = Sast.get_expr_type_info a3
                                    and t4 = Sast.get_expr_type_info a4
                                    in
                                    let var2 = getString a2 and var3 = getString a3 and var4 = getString a4 in
                                        match (a1,t2,t3,t4) with
                                        StrLit(_),Mat(_,x2,y2),Mat(_,x3,y3),Mat(_,x4,y4) ->
                                            if (x2!=x3)||(x3!=x4)||(y2!=y3)||(y3!=y4) then raise(Failure("All matrices must have the same dimensions")) else (); updateInitSyms var2;
                                        updateInitSyms var3; updateInitSyms var4; Int
                                        | _ -> raise(Failure("ppmwrite takes string literal and 3 matrix types"))
                                     ) ))
     | Call("print_board", actuals) -> let a = (List.map expr actuals) in
              SCall("print_board", a,    (if (List.length actuals != 2)
                                    then raise(Failure("Print_board only accepts 2 arguments"))
                                    else ( let a1 = expr(List.hd actuals) and a2 = expr (List.nth actuals 1) in
                                    let t1 = Sast.get_expr_type_info a1 in
                                    let t2 = Sast.get_expr_type_info a2 in
                                    let var = getString a1 in
                                    match (t1, t2) with
                                        (Mat(_,_,_), Int) -> updateInitSyms var; Void
                                        | _ -> raise(Failure("Print_board takes string literal and matrix type"))
```

```
                                   ) ))

    | Call(fname, actuals) as call -> let a = (List.map expr actuals) in
            SCall(fname, a, (let fd = function_decl fname in
       if List.length actuals != List.length fd.formals then
         raise (Failure ("expecting " ^ string_of_int
           (List.length fd.formals) ^ " arguments in " ^ string_of_expr call))
       else
         List.iter2 (fun (ft, _) e -> let et1 = expr e in
            let et = Sast.get_expr_type_info et1 in
            ignore (check_assign ft et
              (Failure ("illegal actual argument found " ^ string_of_typ et ^
              " expected " ^ string_of_typ ft ^ " in " ^ string_of_expr e))))
           fd.formals actuals;
         fd.typ))
  in

  let check_bool_expr e = if (Sast.get_expr_type_info (expr e)) != Bool
  then raise (Failure ("expected Boolean expression"))
   else (expr e) in

  (* Verify a statement or throw an exception *)
  let rec stmt = function
Block sl ->(* SBlock(let rec check_block = function
         [Return _ as s] -> stmt s
       | Return _ :: _ -> raise (Failure "nothing may follow a return")
       | Block sl :: ss -> check_block (sl @ ss)
       | s :: ss -> stmt s ; check_block ss
       | [] -> (expr ())
      in check_block sl)*)
      SBlock(convertStmtToSStmt sl)
    | Expr e -> SExpr (expr e)
    | Return e -> SReturn(let t = expr e in
                          let t1 = Sast.get_expr_type_info t in
                          if t1 = func.typ then expr e else
       raise (Failure ("return gives invalid type")))

    | If(p, b1, b2) -> SIf((check_bool_expr p),(stmt b1), (stmt b2))
    (*| For(e1, e2, e3, st) -> ignore (expr e1); check_bool_expr e2;
                             ignore (expr e3); stmt st*)
    | While(p, s) -> SWhile((check_bool_expr p), stmt s)

and convertStmtToSStmt sl = List.map stmt sl in
 let convertFdeclToSFdecl function_decls fdecl =
  {
      sfname = fdecl.fname;
      styp   = fdecl.typ;
      sformals = fdecl.formals;
      slocals = fdecl.locals;
      sbody = (convertStmtToSStmt fdecl.body);
  }
 in
  let sast = convertFdeclToSFdecl function_decls func
  in

  (ignore(stmt (Block func.body)));sast

  in

  (List.map check_function functions)
  (*
   List.map (fun x -> if ((List.length x.locals)==0) then raise(Failure("No Locals:"^x.fname))
   else check_function x) functions
  *)
```

```
(*
 * File: codegen.ml
 * Date: 2017-03-13
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni    <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang         <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 *)
open Sast
open Semant
open Exceptions
module L = Llvm
module A = Ast
module S = Sast

module StringMap = Map.Make(String)
module MatrixMap = Map.Make(String)

let translate (functions) =
  let context    = L.global_context () in
  let the_module = L.create_module context "MPL"

  (* ------------------------------------------------- *
   |              llvm tyoes declarations               |
   * ------------------------------------------------- *)
  and i32_t      = L.i32_type  context
  and i8_t       = L.i8_type   context
  and i1_t       = L.i1_type   context
  and void_t     = L.void_type context
  and float_t    = L.double_type context
  and array_t    = L.array_type
  and pointer_t  = L.pointer_type

  in
(*
  let matrix_int_t m n   = array_t (array_t i32_t m) n
  and matrix_float_t m n = array_t (array_t float_t m) n
  and matrix_t t m n   = array_t (array_t t m) n
in
*)

  let rec func_ptr_t typ =
      let arr   = Array.make 9 (ltype_of_typ typ) in
      let ftype = L.function_type (ltype_of_typ typ) arr in
      pointer_t ftype
  and

  (* Find out the llvm type by Ast type *)
  ltype_of_typ = function
      A.Int   -> i32_t
    | A.Float -> float_t
    | A.Bool  -> i1_t
    | A.Void  -> void_t
    | A.Mat(typ, rows, cols) ->
        (match typ with
                A.Int     -> array_t (array_t i32_t cols) rows
                | A.Float -> array_t (array_t float_t cols) rows
                | _ -> raise(Exceptions.UnsupportedMatrixType)        )
    | A.FMat(typ, rows, cols) -> array_t (array_t   (func_ptr_t typ) cols) rows
    | _ -> raise(Exceptions.UnsupportedType)
  in


  (* ------------------------------------------------- *
   |               Built-in Functions                  |
   * ------------------------------------------------- *)
  let printf_t             = L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
  let printf_func          = L.declare_function "printf" printf_t the_module in

  let printm_int_t         = L.function_type i32_t [| L.pointer_type i8_t; i32_t; i32_t |] in
  let printm_int_func      = L.declare_function "printm_int" printm_int_t the_module in
  let printm_float_t       = L.function_type i32_t [| L.pointer_type i8_t; float_t; float_t |] in
  let printm_float_func    = L.declare_function "printm_float" printm_float_t the_module in

  let matrw_int_t          = L.function_type i32_t [| L.pointer_type i8_t; L.pointer_type i8_t; i32_t; i32_t |] in
  let matrw_float_t        = L.function_type i32_t [| L.pointer_type i8_t; L.pointer_type i8_t; i32_t; i32_t |] in
  let matread_int_func     = L.declare_function "matread_int" matrw_int_t the_module in
  let matread_float_func   = L.declare_function "matread_float" matrw_float_t the_module in
  let matwrite_int_func    = L.declare_function "matwrite_int" matrw_int_t the_module in
  let matwrite_float_func  = L.declare_function "matwrite_float" matrw_float_t the_module in

  let memcpy_t             = L.function_type i32_t [| L.pointer_type i8_t; L.pointer_type i8_t; i32_t|] in
  let memcpy_func          = L.declare_function "memcpy" memcpy_t the_module in

  let print_board_t        = L.function_type i32_t [| L.pointer_type i8_t; i32_t; i32_t; i32_t |] in
  let print_board_func     = L.declare_function "print_board" print_board_t the_module in

  let pgmread_func         = L.declare_function "pgmread" matrw_int_t the_module in
  let pgmwrite_func        = L.declare_function "pgmwrite" matrw_int_t the_module in
  let ppmrw_int_t          = L.function_type i32_t [| L.pointer_type i8_t; L.pointer_type i8_t;
                             L.pointer_type i8_t; L.pointer_type i8_t; i32_t; i32_t |] in
  let ppmread_func         = L.declare_function "ppmread" ppmrw_int_t the_module in
  let ppmwrite_func        = L.declare_function "ppmwrite" ppmrw_int_t the_module in

  (* Define each function so we can call it *)
  let function_decls =
    let function_decl m fdecl =
      let name  = fdecl.S.sfname in
      let lltyp = (ltype_of_typ fdecl.S.styp) in
      let arr   = Array.make 9 lltyp in
      let ftype =
          if(name = "main") then
              L.function_type lltyp [| lltyp |]
          else
              L.function_type lltyp arr
          in
      StringMap.add name (L.define_function name ftype the_module, fdecl) m in
    List.fold_left function_decl StringMap.empty functions in


  (* Fill in the body of the given function *)
  let build_function_body fdecl =
    let (the_function, _) = StringMap.find fdecl.S.sfname function_decls in
    let builder           = L.builder_at_end context (L.entry_block the_function) in
    (* format strings for printf *)
    let int_format_str    = L.build_global_stringptr "%d\n" "fmti" builder in
    let string_format_str = L.build_global_stringptr "%s\n" "fmts" builder in

    let get_mat_dimensions t = match t with
        A.Mat(typ, rows, cols) -> (typ, rows, cols)
        | _                    -> raise ( UnsupportedMatrixType )
    in

    (* An array of string representation of the 9 neighbhors *)
    let typ = fdecl.S.styp in
    let neighbor_names = [ "\#NW"; "\#N"; "\#NE"; "\#W"; "\#C"; "\#E"; "\#SW"; "\#S"; "\#SE"
```

```
         ] in
    let neighbor_list = List.map (fun x -> (typ, x)) neighbor_names
    in

    (* Construct the function's "locals": formal arguments and locally
       declared variables.  Allocate each on the stack, initialize their
       value, if appropriate, and remember their values in the "locals" map *)
    let local_vars =
        let add_formal m (t, n) p = L.set_value_name n p;
            let local = L.build_alloca (ltype_of_typ t) ("sharp" ^ n) builder in
            ignore (L.build_store p local builder);
            StringMap.add n local m in

  let add_local (m,mat_m) (t, n) =
    let local_var = L.build_alloca (ltype_of_typ t) n builder in
    (match t with
      A.Mat(typ, row, cols) ->
                let dim = get_mat_dimensions t in
                ((StringMap.add n local_var m), (StringMap.add n dim mat_m))
      | _ -> ((StringMap.add n local_var m), mat_m))
    in

    let formals =
        if(fdecl.S.sfname <> "main") then
            (List.fold_left2 add_formal StringMap.empty neighbor_list
            (Array.to_list (L.params the_function)))
        else StringMap.empty in

    (* Add the local variables to a new map *)
    List.fold_left add_local (formals, MatrixMap.empty) fdecl.S.slocals in

  (* ------------------------------------------------- *
   |               Helper Functions                    |
   * ------------------------------------------------- *)
    (* Return the value for a variable or formal argument *)
    let getSlocal (a, _) = a in
    let getMlocal (_, b) = b in
    let lookup n = try StringMap.find n (getSlocal local_vars)
        with Not_found -> raise(Exceptions.LocalNotFound("unknown variable name: "^n))
    in

    let find_matrix_type matrix =
      match (List.hd (List.hd matrix)) with
        S.SIntLit _   -> ltype_of_typ (A.Int)
      | S.SFloatLit _ -> ltype_of_typ (A.Float)
      | S.SBoolLit _  -> ltype_of_typ (A.Bool)
      | S.SId (s, t)  -> let func_type = L.type_of (fst (StringMap.find s function_decls))
                        in pointer_t func_type
      | _             -> raise (UnsupportedMatrixType) in

    let idx n m          = [| L.const_int i32_t n; L.const_int i32_t m |] in
    let idx_gep      n m = [| L.const_int i32_t 0; L.const_int i32_t n; L.const_int i32_t m |] in

    let lookupM = function
        S.SId(s, t) -> MatrixMap.find s (getMlocal local_vars)
        | _ -> raise(Exceptions.UnsupportedMatrixType)
    in

    let lookup_matrixid = function
        S.SId(s, t) -> lookup s
        | _ -> raise(Exceptions.UnsupportedMatrixType)
    in

    let get_string_by_id = function
        | S.SId(s, t) -> s
        | _ -> raise(Exceptions.UnsupportedMatrixType)  (* TODO *)
    in
    let build_matrix ast_typ l expr builder =
        let typ            = ltype_of_typ ast_typ in
        let i32_list       = List.map (List.map (expr builder)) l in
        let list_of_arrs  = List.map Array.of_list i32_list in
        let arrs_of_arrs  = Array.of_list (List.map (L.const_array typ) list_of_arrs) in
        L.const_array (array_t typ (List.length (List.hd l))) arrs_of_arrs
    in

    let build_matrix_access i j s rows cols builder assign =
        let ptr = L.build_gep (lookup s) [| L.const_int i32_t 0; i; j|] s builder in
        if assign then ptr
        else L.build_load ptr s builder
    in

    let get_builder bb = L.builder_at_end context bb
    in

    let get_neighbor mat i j xi xj row col b =
        let rem x y = L.build_srem x y "tmp" b in
        let add x y = L.build_add  x y "tmp" b in
        let xi_l    = L.const_int i32_t xi in
        let xj_l    = L.const_int i32_t xj in
        let x = if (xi == -1) then
            rem (add (rem (add i xi_l) row) row) row
            else rem (add i xi_l) row
        in
        let y = if (xj == -1) then
            rem (add (rem (add j xj_l) col) col) col
            else rem (add j xj_l) col
        in
        L.build_load (L.build_gep mat [| L.const_int i32_t 0; x; y |] "build_gep"  b) "
            build_load" b
    in


    (* Build instructions for apply operation, this will translate a single
     * apply to 9 distinct llvm function calls.
     * @fname = string of the function name
     * @mat   = loaded llvalue that is a matrix *)
    let build_apply f_expr mat n b =
      let (typ, rows, cols) = (lookupM mat) in
        let dim = L.const_int i32_t (4 * rows * cols) in
        let mat_str = get_string_by_id mat in
        let f = get_string_by_id f_expr in
        let (fdef, fdecl) = StringMap.find f function_decls in
        let result = f ^ "_result" in

        (* Declare outter counter *)
        let id_ptr      = L.build_in_bounds_gep (lookup mat_str) (idx 0 0) "
            build_in_bounds_gep" b in
        let mat_ptr     = L.build_bitcast id_ptr (pointer_t i8_t) "mat_ptr" b in
        let arr         = Array.make 9 (L.const_int i32_t 0) in
        (* let old_mat     = L.build_alloca (array_t (array_t i32_t cols) rows) "old_mat" b
            in *)
        let old_mat     = L.build_malloc (array_t (array_t i32_t cols) rows) "old_mat" b in
        let old_mat_ptr = L.build_bitcast old_mat (pointer_t i8_t) "old_mat_ptr" b in
        let iptr        = L.build_alloca i32_t "outter_count" b in
        let jptr        = L.build_alloca i32_t "inner_count" b in
        ignore(L.build_store (L.const_int i32_t 0) iptr b);
        ignore(L.build_call memcpy_func [|  old_mat_ptr; mat_ptr; dim  |] "memcpy" b);

        let outter_pred_bb = L.append_block context "outter" the_function in
        ignore (L.build_br outter_pred_bb b);

        let outter_builder = L.builder_at_end context outter_pred_bb in
        let i = L.build_load iptr "outter_countv" outter_builder in
        let outter_bool_val = L.build_icmp L.Icmp.Slt i (L.const_int i32_t rows) "
            outter_bool_val" outter_builder in

        let outter_body_bb = L.append_block context "outter_body" the_function in

        (* Declare inner counter *)
        let outter_body_builder = L.builder_at_end context outter_body_bb in
        ignore(L.build_store (L.const_int i32_t 0) jptr outter_body_builder);

          let inner_pred_bb = L.append_block context "inner" the_function in
          ignore (L.build_br inner_pred_bb outter_body_builder);
          let inner_builder = L.builder_at_end context inner_pred_bb in

          let j = L.build_load jptr "inner_countv" inner_builder in
          let inner_bool_val = L.build_icmp L.Icmp.Slt j (L.const_int i32_t cols) "inner_bool_val" inner_builder in

          let inner_body_bb = L.append_block context "inner_body" the_function in
          let inner_body_builder = L.builder_at_end context inner_body_bb in

          (* The actual code for function application *)
          let entry = L.build_gep (lookup mat_str) [| L.const_int i32_t 0; i; j |] mat_str inner_body_builder in

          (* for all the nine neighbors *)
          let arr = Array.make 9 (L.const_int i32_t 0) in
          for n = -1 to 1 do
              for m = -1 to 1 do
                  let index  = 3 * (m + 1) + (n + 1) in
                  arr.(index) <- get_neighbor old_mat i j m n
                        (L.const_int i32_t rows) (L.const_int i32_t cols) inner_body_builder;
            done
          done;

          let res = L.build_call fdef arr result inner_body_builder in
          ignore(L.build_store res entry inner_body_builder);
          ignore(L.build_store (L.build_add j (L.const_int i32_t 1) "tmp" inner_body_builder) jptr inner_body_builder); (* j++ *)
          ignore(L.build_br inner_pred_bb inner_body_builder);

          let inner_merge_bb = L.append_block context "inner_merge" the_function in
          ignore(L.build_cond_br inner_bool_val inner_body_bb inner_merge_bb inner_builder);
          ignore(L.build_store (L.build_add i (L.const_int i32_t 1) "tmp"
            (get_builder inner_merge_bb)) iptr (get_builder inner_merge_bb)); (* i++ *)
          ignore(L.build_br outter_pred_bb (get_builder inner_merge_bb));


        let outter_merge_bb = L.append_block context "outter_merge" the_function in
        ignore (L.build_cond_br outter_bool_val outter_body_bb outter_merge_bb outter_builder);
        let outter_merge_builder = get_builder outter_merge_bb in
        let ret = L.build_load (L.build_gep (lookup mat_str) [| L.const_int i32_t 0
            ; L.const_int i32_t 0; L.const_int i32_t 0 |] n outter_merge_builder) n outter_merge_builder in
        ignore(L.build_free old_mat outter_merge_builder);
        (ret, outter_merge_builder)
     in

    let find_fptr_by_id typ builder = function
        S.SId (id, t) ->  L.build_bitcast (fst (StringMap.find id function_decls)) typ "func_ptr" builder
        | _ -> raise(Exceptions.UnsupportedMatrixType)
    in

    (* get an i8_t pointer by llvalue of a matrix, useful for C function calls *)
    let get_mptr m b =
        let arr_ptr = L.build_in_bounds_gep m (idx 0 0) "build_in_bounds_gep" b in
         L.build_bitcast arr_ptr (pointer_t i8_t) "mat_ptr" b
    in

  (* -------------------------------------------------- *
   |                Expression Builder                  |
   * -------------------------------------------------- *)
    let rec expr builder expression =  match expression with
        S.SIntLit(i)                  -> L.const_int i32_t i
    | S.SFloatLit(i)                -> L.const_float float_t i
      | S.SBoolLit b                  -> L.const_int i1_t (if b then 1 else 0)
      | S.SStrLit s                   -> L.build_global_stringptr s "str_lit" builder
      | S.SMatrixLit(l, Mat(t, r, c)) -> build_matrix t l expr builder
      (* | S.SNoexpr                  -> L.const_int i32_t 0 *)
      | S.SId (s, t)                  -> L.build_load (lookup s) s builder (* lookup s *)
      | S.SBinop (e1, op, e2, t) ->
          (match op with
            A.Add     -> L.build_add  (expr builder e1) (expr builder e2) "tmp" builder
          | A.Sub     -> L.build_sub  (expr builder e1) (expr builder e2) "tmp" builder
          | A.Mult    -> L.build_mul  (expr builder e1) (expr builder e2) "tmp" builder
          | A.Div     -> L.build_sdiv (expr builder e1) (expr builder e2) "tmp" builder
          | A.And     -> L.build_and  (expr builder e1) (expr builder e2) "tmp" builder
          | A.Or      -> L.build_or   (expr builder e1) (expr builder e2) "tmp" builder
          | A.Equal   -> L.build_icmp L.Icmp.Eq  (expr builder e1) (expr builder e2) "tmp" builder
          | A.Neq     -> L.build_icmp L.Icmp.Ne  (expr builder e1) (expr builder e2) "tmp" builder
          | A.Less    -> L.build_icmp L.Icmp.Slt (expr builder e1) (expr builder e2) "tmp" builder
          | A.Leq     -> L.build_icmp L.Icmp.Sle (expr builder e1) (expr builder e2) "tmp" builder
          | A.Greater -> L.build_icmp L.Icmp.Sgt (expr builder e1) (expr builder e2) "tmp" builder
          | A.Geq     -> L.build_icmp L.Icmp.Sge (expr builder e1) (expr builder e2) "tmp" builder
          | A.Apply   -> fst (build_apply e1 e2 "tmp"  builder)
          | _         -> raise(Exceptions.InvalidUnaryOperation)
          (* TODO: EMult, EDiv, Matapply *)
          )
      | S.SUnop(op, e, t) ->
          let e' = expr builder e in
          (match op with
            A.Neg     -> L.build_neg
          | A.Not     -> L.build_not)
          e' "tmp" builder
      | S.SMatrixAccess (s, e1, e2, t) ->
            let (typ, rows, cols) = MatrixMap.find s (getMlocal local_vars) in
                (build_matrix_access (expr builder e1) (expr builder e2) s rows cols builder false)
      | S.SAssign (s, e, t) -> let e' = expr builder e in
                ignore (L.build_store e' (lookup s) builder); e'
      | S.SCall ("print", [e], t)  -> (match t with
          A.Int  -> L.build_call printf_func [| int_format_str ; (expr builder e) |] "printf" builder
        | A.Bool -> L.build_call printf_func [| int_format_str ; (expr builder e) |] "printf" builder
        | A.Mat(typ, rows, cols) -> let (typ, rows, cols) = (lookupM e) in
            let id = lookup_matrixid e
            in (match typ with
            A.Int -> L.build_call printm_int_func [| (get_mptr id builder); (L.const_int i32_t rows);
                (L.const_int i32_t cols) |] "printm_int" builder
            | A.Float -> L.build_call printm_float_func [| (get_mptr id builder); (L.const_int i32_t rows);
                (L.const_int i32_t cols) |] "printm_float" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType) )
        | _      -> raise(Exceptions.IllegalArgument("from print")))
      | S.SCall ("prints", [e], t) ->
      L.build_call printf_func [| string_format_str ; (expr builder e) |] "printf" builder
      | S.SCall ("print_board", [e1; e2], t) ->
    let (typ, rows, cols) = (lookupM e1) in
        let id = lookup_matrixid e1 in
        (match typ with
            A.Int -> L.build_call print_board_func [| (get_mptr id builder) ; (L.const_int i32_t rows);
                (L.const_int i32_t cols); (expr builder e2) |] "print_board" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType))
      | S.SCall ("printm", [e], t) ->
    let (typ, rows, cols) = (lookupM e) in
        let id = lookup_matrixid e
                in (match typ with
                            A.Int -> L.build_call printm_int_func [| (get_mptr id builder); (L.const_int i32_t rows);
                                (L.const_int i32_t cols) |] "printm_int" builder
                            | A.Float -> L.build_call printm_float_func [| (get_mptr id builder); (L.const_int i32_t rows);
                                (L.const_int i32_t cols) |] "printm_float" builder
                            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | S.SCall ("matwrite", [e1; e2], t) ->
    let (typ, rows, cols) = (lookupM e2) in
        let id = lookup_matrixid e2
        in (match typ with
            A.Int -> L.build_call matwrite_int_func [| (expr builder e1); (get_mptr id builder);
                (L.const_int i32_t rows); (L.const_int i32_t cols) |] "matwrite_int" builder
            | A.Float -> L.build_call matwrite_float_func [| (expr builder e1); (get_mptr id builder);
                (L.const_int i32_t rows); (L.const_int i32_t cols) |] "matwrite_float" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | S.SCall ("matread", [e1; e2], t)  ->
    let (typ, rows, cols) = (lookupM e2) in
        let id = lookup_matrixid e2
        in (match typ with
            A.Int -> L.build_call matread_int_func [| (expr builder e1); (get_mptr id builder);
                (L.const_int i32_t rows); (L.const_int i32_t cols) |] "matread_int" builder
            | A.Float -> L.build_call matread_float_func [| (expr builder e1); (get_mptr id builder);
                (L.const_int i32_t rows); (L.const_int i32_t cols) |] "matread_float" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | S.SCall ("ppmwrite", [e1; e2; e3; e4], t) ->
    let (typ, rows, cols) = (lookupM e2) in
        let mat1 = lookup_matrixid e2 in
        let mat2 = lookup_matrixid e3 in
        let mat3 = lookup_matrixid e4
        in (match typ with
            A.Int -> L.build_call ppmwrite_func [| (expr builder e1); (get_mptr mat1 builder);
                (get_mptr mat2 builder); (get_mptr mat3 builder); (L.const_int i32_t rows);
                (L.const_int i32_t cols) |] "ppmwrite" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | S.SCall ("ppmread", [e1; e2; e3; e4], t) ->
    let (typ, rows, cols) = (lookupM e2) in
        let mat1 = lookup_matrixid e2 in
        let mat2 = lookup_matrixid e3 in
        let mat3 = lookup_matrixid e4
        in (match typ with
            A.Int -> L.build_call ppmread_func [| (expr builder e1); (get_mptr mat1 builder);
                (get_mptr mat2 builder); (get_mptr mat3 builder); (L.const_int i32_t rows);
                (L.const_int i32_t cols) |] "ppmread" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | S.SCall ("pgmwrite", [e1; e2], t)  ->
    let (typ, rows, cols) = (lookupM e2) in
        let id = lookup_matrixid e2
        in (match typ with
            A.Int -> L.build_call pgmwrite_func [| (expr builder e1); (get_mptr id builder);
                (L.const_int i32_t rows); (L.const_int i32_t cols) |] "pgmwrite" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | S.SCall ("pgmread", [e1; e2], t)  ->
    let (typ, rows, cols) = (lookupM e2) in
        let id = lookup_matrixid e2
        in (match typ with
            A.Int -> L.build_call pgmread_func [| (expr builder e1); (get_mptr id builder);
                (L.const_int i32_t rows); (L.const_int i32_t cols) |] "pgmread" builder
            | _ -> raise(Exceptions.UnsupportedMatrixType)        )
      | _ -> raise(Exceptions.StatementNotSuuported)
    in

    (* Invoke "f builder" if the current block doesn't already
       have a terminal (e.g., a branch). *)
    let add_terminal builder f =
      match L.block_terminator (L.insertion_block builder) with
  Some _ -> ()
      | None -> ignore (f builder) in

  (* ------------------------------------------------- *
   |                Statement Builder                   |
   * ------------------------------------------------- *)
    let rec stmt builder = function
        S.SBlock sl -> List.fold_left stmt builder sl
      | S.SExpr e -> (match e with
          S.SBinop(e1, op, e2, t) -> (match op with
              A.Apply -> snd (build_apply e1 e2 "tmp"  builder)
              |_ -> ignore (expr builder e); builder)
          | _ -> ignore (expr builder e); builder)
      | S.SReturn e -> ignore (match fdecl.S.styp with
          A.Void -> L.build_ret_void builder
          | _ -> L.build_ret (expr builder e) builder); builder
      | S.SIf (predicate, then_stmt, else_stmt) ->
         let bool_val = expr builder predicate in
         let merge_bb = L.append_block context "merge" the_function in

         let then_bb = L.append_block context "then" the_function in
         add_terminal (stmt (L.builder_at_end context then_bb) then_stmt)
           (L.build_br merge_bb);

         let else_bb = L.append_block context "else" the_function in
         add_terminal (stmt (L.builder_at_end context else_bb) else_stmt)
           (L.build_br merge_bb);

         ignore (L.build_cond_br bool_val then_bb else_bb builder);
         L.builder_at_end context merge_bb

      | S.SWhile (predicate, body) ->
          let pred_bb = L.append_block context "while" the_function in
          ignore (L.build_br pred_bb builder);

          let body_bb = L.append_block context "while_body" the_function in
          add_terminal (stmt (L.builder_at_end context body_bb) body)
            (L.build_br pred_bb);

          let pred_builder = L.builder_at_end context pred_bb in
          let bool_val = expr pred_builder predicate in

          let merge_bb = L.append_block context "merge" the_function in
          ignore (L.build_cond_br bool_val body_bb merge_bb pred_builder);
          L.builder_at_end context merge_bb
      (*| A.For (e1, e2, e3, body) -> stmt builder
      ( A.Block [A.Expr e1 ; A.While (e2, A.Block [body ; A.Expr e3]) ] )
      *)
    in

    (* Build the code for each statement in the function *)
```

```
    let builder = stmt builder (S.SBlock fdecl.S.sbody) in

    (* Add a return if the last block falls off the end *)
    add_terminal builder (match fdecl.S.styp with
        A.Void -> L.build_ret_void
      | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))
  in

  List.iter build_function_body functions;
  the_module
```

```
/*
 * File: utils.c
 * Date: 2017-04-23
 *
 * PLT Spring 2017
 * MPL Project
 * Wode "Nimo" Ni     <wn2155@columbia.edu>
 * David Rincon-Cruz <dr2884@columbia.edu>
 * Chi Zhang          <cz2440@columbia.edu>
 * Jiangfeng Wang    <jw3107@columbia.edu>
 */

\\#include <unistd.h>
\#include <stdlib.h>
\#include <stdio.h>

\#define LIVE "o" // a black square
\#define DEAD "."     // a space
\#define clear() printf("\033[H\033[J")
\#define get_symbol(i) (i == 0 ? DEAD : LIVE)
\#define mat_entry(mat, n, i, j) (*((mat + i * n) + j))

// \#define DEBUG

/* Given a board of Conway's Game of life, pretty print it.
 * @mat: the board
 * @m  : the width of the matrix
 * @n  : the height of the matrix
 * @sleep: The time interval between updates, in milliseconds
 */
void print_board(int* mat, int m, int n, int sleep) {
    int len = m * (n + 1) + 1;
    char buf[len];
    char *bp = buf;
    buf[len - 1] = 0;

    for(int i = 0; i < m; ++i) {
        for(int j = 0; j < n; ++j) {
            int entry = mat_entry(mat, n, i, j);
            if(j == n - 1) {
                sprintf(bp, "%s\n", get_symbol(entry));
                bp += 2;
            }
            else {
                sprintf(bp, "%s", get_symbol(entry));
                bp++;
            }
        }
    }
    clear();
    printf("%s", buf);
    fflush(stdout);
    usleep(sleep * 1000); // Sleep for 0.5s
}


/* Pretty-print out an integer matrix to stdout
 * @mat: the pointer to the starting address of the matrix
 * @m  : the width of the matrix
 * @n  : the height of the matrix
 */
void printm_int(int* mat, int m, int n) {
    printf("[\n");
    fflush(stdout);
    for(int i = 0; i < m; ++i) {
        for(int j = 0; j < n; ++j) {
            if(j == n - 1) {
```

```c
                printf("%d;\n", *((mat+i*n) + j));
                fflush(stdout);
            }
            else {
                printf("%d, ", *((mat+i*n) + j));
                fflush(stdout);
            }
        }
    }
    printf("]\n");
}


/* Pretty-print out a float matrix to stdout
 * @mat: the pointer to the starting address of the matrix
 * @m  : the width of the matrix
 * @n  : the height of the matrix
 */
void printm_float(double* mat, int m, int n) {
    printf("[");
    fflush(stdout);
    for(int i = 0; i < m; ++i) {
        for(int j = 0; j < n; ++j) {
            if(j == n - 1) {
                printf("%f; ", *((mat+i*n) + j));
                fflush(stdout);
            }
            else {
                printf("%f, ", *((mat+i*n) + j));
                fflush(stdout);
            }
        }
    }
    printf("]\n");
}

/* Read in an integer matrix from a file designated by a path.
 * The input file should be a linear listing of entries in the
 * matrix in row-major order.
 * @path: the path to the input file
 * @mat : the pointer to the starting address of the output matrix
 * @row : the width of the matrix
 * @col : the height of the matrix
 */
int matread_int (char* path, int* mat, int row, int col){
  FILE* fd = fopen(path,"r");
  if(fd==NULL)
    return -1;
  int count = 0;
  while(fread(mat, 1, 4, fd)==4){
    mat++;
    count++;
    if(count==row*col){
      fclose(fd);
      return 0;
    }
  }
  fclose(fd);
  return -2;
}

/* Read in a float matrix from a file designated by a path.
 * The input file should be a linear listing of entries in the
 * matrix in row-major order.
 * @path   : the path to the input file
 * @mat    : the pointer to the starting address of the output matrix
 * @row    : the width of the matrix
 * @col    : the height of the matrix
 * @return : 0 on success, -1 on file-not-fount error
 */
int matread_float (char* path, double* mat, int row, int col){
  FILE* fd = fopen(path,"r");
  if(fd==NULL)
    return -1;
  int count = 0;
  while(fread(mat, 1,sizeof(double), fd)==sizeof(double)){
    count++;
    mat++;
    if(count==row*col){
      fclose(fd);
      return 0;
    }
  }
  fclose(fd);
  return -2;  // FIXME: file size not enough
}

/* Write to an int matrix from a file designated by a path.
 * The output file should be a linear listing of entries in the
 * matrix in row-major order.
 * @path   : the path to the output file
 * @mat    : the pointer to the starting address of the input matrix
 * @row    : the width of the matrix
 * @col    : the height of the matrix
 * @return : 0 on success, -1 on any I/O error
 */
int matwrite_int (char *path, int *mat, int row, int col) {
  FILE* fd = fopen(path, "w");
  if(fd == NULL)
    return -1;
    int size = row * col * sizeof(int);
    if(fwrite(mat, size, 1, fd) != 1) {
        printf("should not happen\n");
        return -1;
    }
    fclose(fd);
    return 0;
}

/* Write to an float matrix from a file designated by a path.
 * The output file should be a linear listing of entries in the
 * matrix in row-major order.
 * @path   : the path to the output file
 * @mat    : the pointer to the starting address of the input matrix
 * @row    : the width of the matrix
 * @col    : the height of the matrix
 * @return : 0 on success, -1 on any I/O error
 */
int matwrite_float (char *path, double *mat, int row, int col) {
  FILE* fd = fopen(path, "w");
  if(fd == NULL)
    return -1;
    int size = row * col * sizeof(double);
    if(fwrite(mat, size, 1, fd) != 1) {
        printf("should not happen\n");
        return -1;
    }
    fclose(fd);
    return 0;
}

int pbmread(char* path, int *mat, int row, int col){
    char buff[16];
    int c, x, y;
    FILE *fd = fopen(path,"r");
    //open file descriptor
    if(fd==NULL){
        fclose(fd);
        return -1;
    }
    //read image format
    if(!fgets(buff,sizeof(buff),fd)){//If you fail to read in file
        fclose(fd);
        return -1;
    }
    //Check the image format, pbm must be P1 or P4 (not too sure about the difference)
    if(buff[0]!='P' || (buff[1]!='1' && buff[1]!='4')){
        fclose(fd);
        return -1;
    }
    //Read until you skip the comments
    c = getc(fd);
    while(c == '\#'){
        while(getc(fd) != '\n');
        c = getc(fd);
    }
    ungetc(c,fd);
    //read file size info
    if( fscanf(fd, "%d %d", &x, &y) !=2){ //Invalid file size
        fclose(fd);
        return -1;
    }
    //Check that the x and y match the file
    if(x!=row || y!=col){
        fclose(fd);
        return -1;
    }

    //I think this will move you to when the numbers actually start
    while (fgetc(fd) != '\n') ;

  int count = 0;
  //while(fread(mat, 1, 4, fd)==4){
    while(1){
        fscanf(fd, "%d",mat);
    mat++;
    count++;
    if(count==row*col){
      fclose(fd);
      return 0;
    }
  }
  fclose(fd);
  return -2;

}

int pgmread(char* path, int *mat, int row, int col){
    char buff[16];
    int c, x, y, d;
    FILE *fd = fopen(path,"r");
    //open file descriptor
    if(fd==NULL){
        fclose(fd);
        return -1;
    }
    //read image format
    if(!fgets(buff,sizeof(buff),fd)){//If you fail to read in file
        fclose(fd);
        return -1;
    }
    //Check the image format, pbm must be P1 or P4 (not too sure about the difference)
    if(buff[0]!='P' || (buff[1]!='2' && buff[1]!='5')){
        fclose(fd);
        return -1;
    }
    //Read until you skip the comments
    c = getc(fd);
    while(c == ’\#’){
        while(getc(fd) != ’\n’);
        c = getc(fd);
    }
    ungetc(c,fd);
    //read file size info
    if( fscanf(fd, "%d %d", &x, &y) !=2){ //Invalid file size
        fclose(fd);
        return -1;
    }
    //Check that the x and y match the file
    if(x!=col || y!=row){
        fclose(fd);
        return -1;
    }

    //Gotta read the depth component //For writing, we’ll just assume it’s 255
    if(fscanf(fd, "%d", &d) != 1){ //Failed to read the depth
        fclose(fd);
        return -1;
    }

    //I think this will move you to when the numbers actually start
    while (fgetc(fd) != ’\n’) ;

  int count = 0;
  //while(fread(mat, 1, 4, fd)==4){
    while(1){
        fscanf(fd, "%d",mat);
    mat++;
    count++;
    if(count==row*col){
      fclose(fd);
      return 0;
    }
  }
  fclose(fd);
  return -2;
}

int ppmread(char* path, int *mr, int *mg, int *mb, int row, int col){
    char buff[16];
    int c, x, y, d;
    FILE *fd = fopen(path,"r");
    //open file descriptor
    if(fd==NULL){
        fclose(fd);
        return -1;
    }
    //read image format
    if(!fgets(buff,sizeof(buff),fd)){//If you fail to read in file
        fclose(fd);
        return -1;
    }
    //Check the image format, pbm must be P1 or P4 (not too sure about the difference)
    if(buff[0]!=’P’ || (buff[1]!=’3’ && buff[1]!=’6’)){
        fclose(fd);
        return -1;
    }
    //Read until you skip the comments
    c = getc(fd);
    while(c == ’\#’){
        while(getc(fd) != ’\n’);
        c = getc(fd);
    }
    ungetc(c,fd);
    //read file size info
    if( fscanf(fd, "%d %d", &x, &y) !=2){ //Invalid file size
        fclose(fd);
        return -1;
    }
    //Check that the x and y match the file
    if(x!=row || y!=col){
        fclose(fd);
        return -1;
    }

    //Gotta read the depth component //For writing, we’ll just assume it’s 255
    if(fscanf(fd, "%d", &d) != 1){ //Failed to read the depth
        fclose(fd);
        return -1;
    }

    //I think this will move you to when the numbers actually start
    while (fgetc(fd) != ’\n’) ;

  int count = 0;
  //while((fread(mr, 1, 4, fd)==4)&&(fread(mg,1,4,fd)==4)&&(fread(mb,1,4,fd)==4)){
    while(1){
        fscanf(fd, "%d",mr);
        fscanf(fd, "%d",mg);
        fscanf(fd, "%d",mb);
    mr++;
    mg++;
    mb++;
    count++;
    if(count==row*col){
      fclose(fd);
      return 0;
    }
  }
  fclose(fd);
  return -2;
}

int pbmwrite(char* path, int *mat, int row, int col){
    FILE *fd = fopen(path,"w");
    //open file descriptor
    if(fd==NULL){
        fclose(fd);
        return -1;
    }

    //write the image format
    fprintf(fd, "P1\n");


    //Write a comment
    fprintf(fd, "\#Autogenerated by MPL\n");

    //Write the file size
    fprintf(fd, "%d %d\n", row, col);

    //Write the actual bits in
    int count = 0;
    while(1){
        if(count%row != 0){
            fprintf(fd,"%d ", *mat);
        }
        else if(count%row==0){
            fprintf(fd,"%d\n",*mat);
        }
        count++;
        mat++;
        if(count==row*col){
      fclose(fd);
      return 0;
    }
    }

}

int pgmwrite(char* path, int *mat, int row, int col){
    FILE *fd = fopen(path,"w");
    //open file descriptor
    if(fd==NULL){
        fclose(fd);
        return -1;
    }

    //write the image format
    fprintf(fd, "P2\n");


    //Write a comment
    fprintf(fd, "\#Autogenerated by MPL\n");

    //Write the file size
    fprintf(fd, "%d %d\n", col, row);

    //write the depth (assumed to be 255)
    fprintf(fd, "255\n");

    //Write the actual bits in
    int count = 0;
    while(1){
        if(count%row != 0){
            fprintf(fd,"%d ", *mat);
        }
        else if(count%row==0){
            fprintf(fd,"%d\n",*mat);
        }
        count++;
        mat++;
        if(count==row*col){
      fclose(fd);
      return 0;
    }
    }

  fclose(fd);
  return -2;
}

int ppmwrite(char* path, int *mr, int *mg, int *mb, int row, int col){
    FILE *fd = fopen(path,"w");
    //open file descriptor
    if(fd==NULL){
        fclose(fd);
        return -1;
    }

    //write the image format
    fprintf(fd, "P3\n");


    //Write a comment
    fprintf(fd, "\#Autogenerated by MPL\n");

    //Write the file size
    fprintf(fd, "%d %d\n", row, col);

    //write the depth (assumed to be 255)
    fprintf(fd, "255\n");

    //Write the actual bits in
    int count = 0;
    while(1){
        if(count%row != 0){
            fprintf(fd,"%d ", *mr);
            fprintf(fd,"%d ", *mg);
            fprintf(fd,"%d ", *mb);
        }
        else if(count%row==0){
            fprintf(fd,"%d ", *mr);
            fprintf(fd,"%d ", *mg);
            fprintf(fd,"%d\n",*mb);
        }
        count++;
        mr++;
        mg++;
        mb++;
        if(count==row*col){
      fclose(fd);
      return 0;
    }
    }
  fclose(fd);
  return -2;
}

\#ifdef DEBUG
int main() {
    int mat[3][3] = {
        {0, 0, 1},
        {0, 1, 0},
        {1, 0, 0},
    };
    print_board((int *)mat, 3, 3);
    return 0;
}
\#endif
```

9.2 Test Suite

9.2.1 Scanner Test

```
ScannerTest.ml

open Parser

type num =
  | Int_lit of int
  | Float_lit of float

let stringify = function
  (* Punctuation *)
  | LPAREN -> "LPAREN"       | RPAREN -> "RPAREN"
  | LBRACE -> "LBRACE"       | RBRACE -> "RBRACE"
  | COMMA -> "COMMA"         | SEMI -> "SEMI"
  | LBRACKET -> "LBRACKET"   | RBRACKET -> "RBRACKET"


  (* Arithmetic Operators *)
  | PLUS -> "PLUS"      | MINUS -> "MINUS"
  | TIMES -> "TIMES"    | DIVIDE -> "DIVIDE"
  | EQ -> "EQ"     | NEQ -> "NEQ"
  | LEQ -> "LEQ"   | GEQ -> "GEQ"
  | LT -> "LT" | GT -> "GT"

  (* Matrix Operators *)

  | APPLY -> "APPLY" | MATAPP -> "MATAPP"
  | TRANS -> "TRANS" | EMULT -> "EMULT"
  | EDIV -> "EDIV"

  (* Logical Operators & Keywords *)
  | AND -> "AND"    | OR -> "OR"
  | NOT -> "NOT"

  (* Assignment Operator *)
  | ASSIGN -> "ASSIGN"

  (* Conditional Operators *)
  | IF -> "IF"
  | ELSE -> "ELSE"
  | ELSEIF -> "ELSEIF"

  (* Loop ID *)
  | WHILE -> "WHILE"

  (* End-of-File *)
  | EOF -> "EOF"

  (* Identifiers *)
  | ID(string) -> "ID"
  (* | ROWS -> "ROWS" | COLS -> "COLS" | LEN -> "LEN" | TRANSPOSE -> "TRANSPOSE"
  | BAR -> "BAR"
  *)
  (* Literals *)
  | INT -> "INT" | FLOAT -> "FLOAT"
  | BOOL -> "BOOL"
  | TRUE -> "TRUE" | FALSE -> "FALSE"
  | INTLIT(num) -> "INTLIT"
  | FLOATLIT(num) -> "FLOATLIT"
  | FUNC -> "FUNC"
  | NULL -> "NULL"
  | NEW -> "NEW"
  | CENTER -> "CENTER"
  | NORTH -> "NORTH"
  | SOUTH -> "SOUTH"
  | WEST -> "WEST"
  | EAST -> "EAST"
  | NWEST -> "NWEST"
  | NEAST -> "NEAST"
  | SWEST -> "SWEST"
  | SEAST -> "SEAST"
  | IMG -> "IMG"
  | MAT -> "MAT"
  | FMAT -> "FMAT"
  | STRLIT(string) -> "STRLIT"
  (* | INC -> "INC" | DEC -> "DEC"
  | COLON -> "COLON"
  | FOR -> "FOR" *)
  | RETURN -> "RETURN"

  | VOID -> "VOID"

let _ =
  let lexbuf = Lexing.from_channel stdin in
  let rec print_tokens = function
    | EOF -> " "
    | token ->
      print_endline (stringify token);
      print_tokens (Scanner.token lexbuf) in
  print_tokens (Scanner.token lexbuf)
```

```
build.sh

#!/bin/bash

cp ../../../src/scanner.mll ./scanner.mll
cp ../../../src/parser.mly ./parser.mly
cp ../../../src/ast.ml ./ast.ml

ocamllex scanner.mll
ocamlyacc parser.mly
ocamlc -c ast.ml
ocamlc -c parser.mli
ocamlc -c scanner.ml
ocamlc -c parser.ml
ocamlc -c ScannerTest.ml
ocamlc -o ScannerTest parser.cmo scanner.cmo ScannerTest.cmo
```

```
test.sh

cat pass/_base_scanner.test | ./ScannerTest > pass/_base_scanner.res
diff pass/_base_scanner.out pass/_base_scanner.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|      SCANNER: FIRST TEST PASSED       |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|      SCANNER: FIRST TEST FAILED       |"
  echo "-----------------------------------------"
fi

cat pass/_delimiters.test | ./ScannerTest > pass/_delimiters.res
diff pass/_delimiters.out pass/_delimiters.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|   SCANNER: DELIMITERS TEST PASSED     |"
  echo "-----------------------------------------"
else
```

```
  echo -e "\e[0;31m"
  echo "----------------------------------------"
  echo "|   SCANNER: DELIMITERS TEST FAILED    |"
  echo "----------------------------------------"
fi

cat pass/_control_flow.test | ./ScannerTest > pass/_control_flow.res
diff pass/_control_flow.out pass/_control_flow.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "----------------------------------------"
  echo "|  SCANNER: CONTROL FLOW TEST PASSED   |"
  echo "----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "----------------------------------------"
  echo "|  SCANNER: CONTROL FLOW TEST FAILED   |"
  echo "----------------------------------------"
fi

cat pass/_conditionals.test | ./ScannerTest > pass/_conditionals.res
diff pass/_conditionals.out pass/_conditionals.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "----------------------------------------"
  echo "|  SCANNER: CONDITIONALS TEST PASSED   |"
  echo "----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "----------------------------------------"
  echo "|  SCANNER: CONDITIONALS TEST FAILED   |"
  echo "----------------------------------------"
fi

cat pass/_arithmetic.test | ./ScannerTest > pass/_arithmetic.res
diff pass/_arithmetic.out pass/_arithmetic.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "----------------------------------------"
  echo "|   SCANNER: ARITHMETIC TEST PASSED    |"
  echo "----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "----------------------------------------"
  echo "|   SCANNER: ARITHMETIC TEST FAILED    |"
  echo "----------------------------------------"
fi

cat pass/_types.test | ./ScannerTest > pass/_types.res
diff pass/_types.out pass/_types.res > /dev/null
if [ $? = 0 ]; then
  echo -e "test dqwdwqwqdqwdqw"
  echo "----------------------------------------"
  echo "|      SCANNER: TYPES TEST PASSED      |"
  echo "----------------------------------------"
else
  echo -e "test dwqdwqdwqqw"
  echo "----------------------------------------"
  echo "|      SCANNER: TYPES TEST FAILED      |"
  echo "----------------------------------------"
fi

cat pass/_matrix.test | ./ScannerTest > pass/_matrix.res
diff pass/_matrix.out pass/_matrix.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "----------------------------------------"
  echo "|     SCANNER: MATRIX TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "Ignore this"
  echo "-----------------------------------------"
  echo "|     SCANNER: MATRIX TEST FAILED       |"
  echo "-----------------------------------------"
fi

cat pass/_comment.test | ./ScannerTest > pass/_comment.res
diff pass/_comment.out pass/_comment.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|     SCANNER: COMMENTS TEST PASSED     |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|     SCANNER: COMMENTS TEST FAILED     |"
  echo "-----------------------------------------"
fi

cat pass/_identifier.test | ./ScannerTest > pass/_identifier.res
diff pass/_identifier.out pass/_identifier.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|    SCANNER: IDENTIFIER TEST PASSED    |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|    SCANNER: IDENTIFIER TEST FAILED    |"
  echo "-----------------------------------------"
fi

cat pass/_mixed_arithmetic.test | ./ScannerTest > pass/_mixed_arithmetic.res
diff pass/_mixed_arithmetic.out pass/_mixed_arithmetic.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "| SCANNER: MIXED ARITHMETIC TEST PASSED |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "| SCANNER: MIXED ARITHMETIC TEST FAILED  |"
  echo "-----------------------------------------"
fi

cat pass/_literal.test | ./ScannerTest > pass/_literal.res
diff pass/_literal.out pass/_literal.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|     SCANNER: LITERAL TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|     SCANNER: LITERAL TEST FAILED      |"
  echo "-----------------------------------------"
fi

cat pass/_assignment.test | ./ScannerTest > pass/_assignment.res
diff pass/_assignment.out pass/_assignment.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|    SCANNER: ASSIGNMENT TEST PASSED    |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|    SCANNER: ASSIGNMENT TEST FAILED    |"
  echo "-----------------------------------------"
fi

cat pass/_main_function.test | ./ScannerTest > pass/_main_function.res
diff pass/_main_function.out pass/_main_function.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|   SCANNER: MAIN FUNCTION TEST PASSED  |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|   SCANNER: MAIN FUNCTION TEST FAILED  |"
  echo "-----------------------------------------"
fi

cat pass/_function.test | ./ScannerTest > pass/_function.res
diff pass/_function.out pass/_function.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|    SCANNER: FUNCTION TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|    SCANNER: FUNCTION TEST FAILED      |"
  echo "-----------------------------------------"
fi

cat pass/_misc.test | ./ScannerTest > pass/_misc.res
diff pass/_misc.out pass/_misc.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|  SCANNER: MISCELLANEOUS TEST PASSED   |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|  SCANNER: MISCELLANEOUS TEST FAILED   |"
  echo "-----------------------------------------"
fi



cat fail/_illegal_dollar.test | ./ScannerTest >& fail/_illegal_dollar.res
diff fail/_illegal_dollar.out fail/_illegal_dollar.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|      SCANNER: $ FAIL TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|      SCANNER: $ FAIL TEST FAILED      |"
  echo "-----------------------------------------"
fi

```

```
cat fail/_illegal_percent.test | ./ScannerTest >& fail/_illegal_percent.res
diff fail/_illegal_percent.out fail/_illegal_percent.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|      SCANNER: % FAIL TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|      SCANNER: % FAIL TEST FAILED      |"
  echo "-----------------------------------------"
fi

cat fail/_illegal_period.test | ./ScannerTest >& fail/_illegal_period.res
diff fail/_illegal_period.out fail/_illegal_period.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|      SCANNER: . FAIL TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|      SCANNER: . FAIL TEST FAILED      |"
  echo "-----------------------------------------"
fi

cat fail/_illegal_pound.test | ./ScannerTest >& fail/_illegal_pound.res
diff fail/_illegal_pound.out fail/_illegal_pound.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|      SCANNER: # FAIL TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|      SCANNER: # FAIL TEST FAILED      |"
  echo "-----------------------------------------"
fi

cat fail/_illegal_tilde.test | ./ScannerTest >& fail/_illegal_tilde.res
diff fail/_illegal_tilde.out fail/_illegal_tilde.res > /dev/null
if [ $? = 0 ]; then
  echo -e "\e[0;32m"
  echo "-----------------------------------------"
  echo "|      SCANNER: ~ FAIL TEST PASSED      |"
  echo "-----------------------------------------"
else
  echo -e "\e[0;31m"
  echo "-----------------------------------------"
  echo "|      SCANNER: ~ FAIL TEST FAILED      |"
  echo "-----------------------------------------"
fi
```

```
_illegal_carrot.test

^

_illegal_carrot.out

Fatal error: exception Failure("illegal character ^")
```

```
_illegal_dollar.test

$
```

```

_illegal_dollar.out

Fatal error: exception Failure("illegal character $")
```

```
_illegal_percent.test

%

_illegal_percent.out

Fatal error: exception Failure("illegal character %")
```

```
_illegal_period.test

.

_illegal_period.out

Fatal error: exception Failure("illegal character .")
```

```
_illegal_pound.test

#

_illegal_pound.out

Fatal error: exception Failure("illegal character #")
```

```
_illegal_tilde.test

~

_illegal_tilde.out

Fatal error: exception Failure("illegal character ~")
```

```
_arithmetic.test

+ - * / =

_arithmetic.out

PLUS
MINUS
TIMES
DIVIDE
ASSIGN
```

```
_assignment.test

int a = 4

_assignment.out

INT
ID
ASSIGN
INTLIT
```

```
_base_scanner.test

( ) { } [ ]
```

```
if else while return main
== != < > <= >= AND OR NOT
+ - * / = @ .@ ^ .* ./
int float boolean void null true false
; ,
hi 99 "hi" 1.0

_base_scanner.out

LPAREN
RPAREN
LBRACE
RBRACE
LBRACKET
RBRACKET
IF
ELSE
WHILE
RETURN
ID
EQ
NEQ
LT
GT
LEQ
GEQ
ID
ID
ID
PLUS
MINUS
TIMES
DIVIDE
ASSIGN
APPLY
MATAPP
TRANS
EMULT
EDIV
INT
FLOAT
BOOL
VOID
NULL
TRUE
FALSE
SEMI
COMMA
ID
INTLIT
STRLIT
FLOATLIT
```

```
_comment.test

/*
   This is a comment
   "Comment"
   None of this should be tokenized.
   int num = 8;
 */

 _comment.out (empty)


\end{




}
\lstset{escapeinside=}
\begin{lstlisting}
_conditionals.test

== != < > <= >= && || !

_conditionals.out

EQ
NEQ
LT
GT
LEQ
GEQ
ID
ID
ID
```

```
_control_flow.test

if else while return main

_control_flow.out

IF
ELSE
WHILE
RETURN
ID
```

```
_delimiters.test

( ) { } [ ]

_delimiters.out

LPAREN
RPAREN
LBRACE
RBRACE
LBRACKET
RBRACKET
```

```
_function.test

float func (int a, int b) {return 1.0;}

_function.out

FLOAT
FUNC
LPAREN
INT
ID
COMMA
INT
ID
RPAREN
LBRACE
RETURN
FLOATLIT
SEMI
RBRACE
```

```
_literal.test

"string lit"
12.12
true
false
null
"\"quotes string\""
4

_literal.out

STRLIT
FLOATLIT
TRUE
FALSE
NULL
STRLIT
INTLIT
```

```
_main_function.test

int main() {return 0;}

_main_function.out

INT
ID
LPAREN
RPAREN
LBRACE
RETURN
INTLIT
SEMI
RBRACE
```

```

_mixed_arithmetic.test

100 - 50.12 * 0.4 / 5 - 6.0

_mixed_arithmetic.out

INTLIT
MINUS
FLOATLIT
TIMES
FLOATLIT
DIVIDE
INTLIT
MINUS
FLOATLIT
```

```
_types.test

int float boolean void null true false func new #C #N #S #W #E #NW #NE #SW #SE Img Mat FMat

_types.out

INT
FLOAT
BOOL
VOID
NULL
TRUE
FALSE
```

```
FUNC
NEW
CENTER
NORTH
SOUTH
WEST
EAST
NWEST
NEAST
SWEST
SEAST
IMG
MAT
FMAT
```

9.2.2 Program Test

```
test-all1.mpl

int entryf() {
    return 1;
}

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    entryf @ m;
    p = m[1][1];
    print(p);
    return 0;
}
//Should print:
1
```

```
test-all2.mpl

int entryf() {
    return #C;
}

int main() {
    Mat<int>[2][2] m;
    int p;
    int k;
    k = 1;
    m = [1,2;3,4];
    entryf @ m;
    p = m[0][1];
    print(p);
    return 0;
}
//Should print:
2
```

```

test-apply.mpl

int entryf() {
    return 1;
}

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    entryf @ m;
    printm(m);
    return 0;
}
//Should print :
[
1, 1, 1;
1, 1, 1;
1, 1, 1;
]
```

```
test-apply2.mpl

int entryf() {
    return #N;
}

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    entryf @ m;
    printm(m);
    return 0;
}
//Should print :
[
7, 8, 9;
1, 2, 3;
4, 5, 6;
]
```

```
test-func.mpl

int reset() {
  return 1;
}

int main()
{
  Mat<int>[2][2] k;
  int q;
  k = [1,2;3,4];
  q = k[1][1];
  print(q);
  reset @ k;
  q = k[1][1];
  print(q);
  return 0;
}
//Should print :
4
1
```

```
test-if1.mpl

int main()
{
  if (true) prints("succeed");
  prints("finish");
  return 0;
}
//Should print :
succeed
finish

test-if2.mpl

int main()
{
  if (true) prints("succeed"); else prints("fail");
  prints("finish");
  return 0;
}

//Should print :
succeed
finish
```

```
test-if3.mpl

```

```
int main()
{
  if (false) prints("fail");
  prints("finish");
  return 0;
}

//Should print :
finish
```

```
test-if4.mpl

int main()
{
  if (false) prints("fail"); else prints("succeed");
  prints("finish");
  return 0;
}
//Should print :
succeed
finish
```

```
test-if5.mpl

int foo()
{
    int j;
    if (#C == 5)
      j = 42;
    else
        j = 17;
    return j;
}

int main()
{
    Mat<int>[3][3] m;
    m = [1,2,3;4,5,6;7,8,9];
    printm(m);
    foo @ m;
    printm(m);
    return 0;
}
//Should print :
[
1, 2, 3;
4, 5, 6;
7, 8, 9;
]
[
17, 17, 17;
17, 42, 17;
17, 17, 17;
]
```

```
test-imgread.mpl

int main() {
    Mat<int>[512][512] img;
    pgmread("test/testVer1/lena.pgm", img);
    printm(img);
}

//A very huge img matrix, not going to show here to waste space.
```

```
test-local1.mpl

int foo()
{
    int j;
/* Should hide the formal i */
    int k;
    j = 42;
    k = j + j;
    print(k);
    return 42;
}

int main()
{
    Mat<int>[3][3] m;
    int i;
    i = 8;
    print(i);
    m = [1,1,1;1,1,1;1,1,1];
    printm(m);
    foo @ m;
    print(i);
    printm(m);
    return 0;
}

//Should print :
8
[
1, 1, 1;
1, 1, 1;
1, 1, 1;
]
84
84
84
84
84
84
84
84
84
8
[
42, 42, 42;
42, 42, 42;
42, 42, 42;
]
```

```
test-local2.mpl

int foo()
{
    int c;
    int e;
    int w;
    int n;
    int s;
    int ne;
    int se;
    int nw;
    int sw;
    c = #C;
    e = #E;
    w = #W;
    n = #N;
    s = #S;
    ne = #NE;
    se = #SE;
    nw = #NW;
    sw = #SW;

    if (c==5)
  print(c);
    if (c==5)
  print(e);
    if (c==5)
  print(w);
    if (c==5)
  print(n);
    if (c==5)
  print(s);
    if (c==5)
  print(ne);
    if (c==5)
  print(se);
    if (c==5)
  print(nw);
    if (c==5)
  print(sw);
    return c;
}

int main()
{
    Mat<int>[3][3] m;
    m = [1,2,3;4,5,6;7,8,9];
    printm(m);
    foo @ m;
    return 0;
}

//Should print :
[
1, 2, 3;
4, 5, 6;
7, 8, 9;
]
5
6
4
2
8
3
9
1
7
```

```
test-mat.mpl

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    p = m[0][0];
    print(p);
    p = m[0][1];
    print(p);
    p = m[0][2];
    print(p);
    p = m[1][0];
    print(p);
    p = m[1][1];
    print(p);
    p = m[1][2];
    print(p);
    p = m[2][0];
    print(p);
    p = m[2][1];
    print(p);
    p = m[2][2];
    print(p);
    return 0;
}

//Should print :
1
2
3
4
5
6
7
8
9
```

```
test-matall.mpl

int entryf() {
    return 1;
}

int main() {
    Mat<int>[3][3] m;
    int p;
    m = [1,2,3;4,5,6;7,8,9];
    p = m[1][1];
    print(p);
    printm(m);
    entryf @ m;
    p = m[1][1];
    print(p);
    printm(m);
    return 0;
}
//Should print :
5
[
1, 2, 3;
4, 5, 6;
7, 8, 9;
]
1
[
1, 1, 1;
1, 1, 1;
1, 1, 1;
]
```

```
test-matread.mpl

int main() {
    Mat<int>[3][3] m;
    matread("test/testVer1/matexample.bin",m);
    printm(m);
    return 0;
}

//Should print :
[
1, 2, 3;
4, 5, 6;
7, 8, 9;
]
```

```
test-matwrite.mpl

int main() {
    Mat<int>[3][3] m1;
    Mat<int>[3][3] m2;
    m1 = [1,2,3;4,5,6;7,8,9];
    matwrite("test/testVer1/matoutput.bin",m1);
    matread("test/testVer1/matoutput.bin",m2);
    printm(m2);

    return 0;
}

//Should print:
[
1, 2, 3;
4, 5, 6;
7, 8, 9;
]
```

```
test-ops1.mpl

int main()
{
  print(1 + 2);
  print(1 - 2);
  print(1 * 2);
  print(100 / 2);
  print(99);
  print(1 == 2);
  print(1 == 1);

  print(99);
  print(1 != 2);
  print(1 != 1);
  print(99);
  print(1 < 2);
  print(2 < 1);
  print(99);
  print(1 <= 2);
  print(1 <= 1);
  print(2 <= 1);
  print(99);
  print(1 > 2);
  print(2 > 1);
  print(99);
  print(1 >= 2);
  print(1 >= 1);
  print(2 >= 1);
  return 0;
}

//Should print :
3
-1
2
50
99
0
1
99
1
0
```

```
99
1
0
99
1
1
0
99
0
1
99
0
1
1
```

```
test-print-board.mpl

int evolve() {
    int i;
    int sum;
    i = 0;

    sum = #NW + #N + #NE + #W + #E + #SW + #S + #SE;

    if(#C == 1)
        if(sum == 2 || sum == 3)
            return 1;
        else
            return 0;
    else
        if(sum == 3)
            return 1;
        else
            return 0;
}

int main() {
    Mat<int>[100][100] board;
    int i;

    i = matread("test/testVer1/gun.bin", board);
    if(i == 0)
        print_board(board, 100);
    else
        prints("File not found");
}

//Will  be a huge file, not going to show here.
```

```
test-print.mpl

int main()
{
  print(41);
  return 0;
}

//Should print :
41
```

```
test-printm.mpl

int main(){
    Mat<int>[2][2] m;
    m = [1, 2; 3, 4];
    printm(m);
}

//Should print :
[
1, 2;
3, 4;
]
```

```
test-prints.mpl

int main() {
    prints("Hello World1");
    prints("Hello World2");
    prints("Hello World3");
    return 0;
}

//Should print :
Hello World1
Hello World2
Hello World3
```

```
test-var1.mpl

int main()
{
    int a;

  boolean e;
  float b;
  Mat<float>[2][2] d;
  Mat<int>[2][2] c;
    a = 42;


  e =true;
  /*
  FMat f;
  Img g;
  */

  print(a);
    return 0;
}

//Should print :
42
```

```
test-while1.mpl

int main()
{
  int i;
  i = 5;
  while (i > 0) {
    print(i);
    i = i - 1;
  }
  prints("finish");
  return 0;
}

//Should print :
5
4
3
```

```
2
1
finish
```

```
test-while2.mpl

int foo()
{
    int j;
    int k;
    j = 0;
    k = #C;
    while (k > 0) {
      j = j + 2;
      k = k - 1;
    }
    return j;
}


int main()
{
    Mat<int>[3][3] m;
    m = [1,2,3;4,5,6;7,8,9];
    printm(m);
    foo @ m;
    printm(m);
    return 0;
}

//Should print :
[
1, 2, 3;
4, 5, 6;
7, 8, 9;
]
[
2, 4, 6;
8, 10, 12;
14, 16, 18;
]
```

```
fail-apply.mpl

int entry(){
  return 1;

}

int main(){
  int T2;
  T2 = 0;
  entry @ T2;
  return 0;
}
//Should print :
Fatal error: exception Failure("T2 must be a matrix type")
```

```
fail-assign1.mpl

int main()
{
  int i;
  boolean b;

  i = 42;
  i = 10;
  b = true;
  b = false;
  i = false; /* Fail: assigning a bool to an integer */
}

//Should print :
Fatal error: exception Failure("illegal assignment int = bool")
```

```
ail-assign2.mpl
int main()
{
  int i;
  boolean b;

  b = 48; /* Fail: assigning an integer to a bool */
}
//Should print :
Fatal error: exception Failure("illegal assignment bool = int")
```

```
fail-expr1.mpl

int foo()
{
  int dd;
  boolean e;
  dd + e; /* Error: bool + int */
  return 1;
}

int main()
{
  int a;
  boolean b;
  return 0;
}
//Should print :
Fatal error: exception Failure("Invalid type for arithmetic operand")
```

```
fail-func1.mpl

int foo() {}

int bar() {}

int baz() {}

int bar() {} /* Error: duplicate function bar */

int main()
{
  return 0;
}
//Should print :
Fatal error: exception Failure("duplicate function bar")
```

```
fail-func4.mpl

int foo() {}

void bar() {}

int print() {} /* Should not be able to define print */

void baz() {}

```

```
int main()
{
  return 0;
}

//Should print :
Fatal error: exception Failure("function print may not be defined")
```

```

fail-func5.mpl

int foo() {}

int bar() {
  int a;
  void b; /* Error: illegal void local b */
  boolean c;

  return 0;
}

int main()
{
  return 0;
}
//Should print :
Fatal error: exception Failure("illegal void local b in bar")
```

```
fail-global1.mpl

int c;
boolean b;
void a; /* global variables should not be void */

int main()
{
  return 0;
}
//Should print :
Fatal error: exception Parsing.Parse_error
```

```
fail-if1.mpl

int main()
{
  if (true) {}
  if (false) {} else {}
  if (42) {} /* Error: non-bool predicate */
}fail-if1.mpl

//Should print :
Fatal error: exception Failure("expected Boolean expression")
```

```
fail-if2.mpl

int main()
{
  if (true) {
    foo; /* Error: undeclared variable */
  }
}
//Should print :
Fatal error: exception Failure("undeclared identifier foo")
```

```
fail-if3.mpl

int main()
{
  if (true) {
    42;
  } else {
    bar; /* Error: undeclared variable */
  }
}

//Should print :
Fatal error: exception Failure("undeclared identifier bar")
```

```
fail-nomain.mpl(empty file)

//Should print :
Fatal error: exception Failure("unrecognized function main")
```

```

fail-return1.mpl

int main()
{
  return true; /* Should return int */
}
//Should print :
Fatal error: exception Failure("return gives invalid type")
```

```
fail-return2.mpl

float foo()
{
  if (true) return 42; /* Should return void */
  else return 1;
}

int main()
{
  return 42;
}

//Should print :
Fatal error: exception Failure("return gives invalid type")
```

```
fail-while1.mpl

int main()
{
  int i;

  while (true) {
    i = i + 1;
  }

  while (42) { /* Should be boolean */
    i = i + 1;
  }

}
//Should print :
Fatal error: exception Failure("expected Boolean expression")
```

```
fail-while2.mpl
```

```

int main()
{
  int i;

  while (true) {
    i = i + 1;
  }

  while (true) {
    foo(); /* foo undefined */
  }
  return 1;
}
//Should print :
Fatal error: exception Failure("unrecognized function foo")
```

9.3 Project Repository git 'shortlog'

```
wodeni <wn2155@columbia.edu> (33):
      Meeting notes until Spring break and related documents.
      Compiled Scanner and parser with new tokens added
      Added floating point numberin scanner and parser.
      [Sement] Added skeleton code for sement checker
      [Codegen] Trying to add printm
      [Codegen] matrix print needs to be fixed
      [Codegen] printm now works
      [README] added the plan for the next 3 weeks
      [Bug fix] float declaration error
      [FIXME] single grp not enough
      [Apply] Apply operator now works!
      [Entry function] Now have # variables work
      [matwrite] working version with one test case
      [matwrite] added utils.c
      In progress: fix build_apply
      added demo.mpl
      [Demo] First version of demo
      Starting the merge
      [BUG] Apply: segfault when run too many times
      [BUG FIX] The segfault is now fixed
      preping for merge from backend
      Preping for merge from matrix
      Merge remote-tracking branch 'origin/matrix' into merge-sast
      Fixed merged bugs
      Successfully merged the SAST version of semantic checker
      [BUG FIX] neighbor computation
      [Clean up] unneccessary files
      [Demo] added binary files
      [Demo] Now the convolution demo works.
      [Test] matread/write small problem with file input
      [Demo] Added rle files
      [Demo] Clean up and better pics
      [Report] Added git log history tex file


chyzhang <chyzhang@brandeis.edu> (24):
      fixed bugs about matrix and printm builder
      testall modified
      fix codegen.ml
      test:1)if pass 2)varpass 3)while fail
      matrix access fixed
      add @ test
     "04/18/2017 Implement matread function. We implemented matread function by calling C
         function successfully. It works like printm and take 2 parameters, path and the
         destination matrix. Users are assumed to know the size of the matrix they want to
         read."
      test committed and testall.sh modified. Problem now:1:if a mpl file is all commented,
          it cannot be compiled. 2:Something wrong with prints when tested invalid character
      upload golden ref and do some cleaning
      prints works now
      TEST REFINED
      golden ref local2 renew
      changes for test
      basic fmat can be formed now
      modified the apply, now it can first read the matrix and remember all the entries,
          then store all changes
      modify golden ref because of changes made in printm. Now all test pass
      test refined for new sast
      testall.sh
      changes in test for demo
      changes in test for demo 2
      test + gcd demo file
      for test read
      for matwrite test
      fixed matread write test
```

```
DRC9702 <drinconcruz@knox.edu> (14):
      Refactor from scratch. Please don't kill me Nimo
      Adding the functions Jane wants. What's good Nimo?
      Compilation doesn't work because i haven't setup here
      Semantic Checker now accepts the apply operator
      Added in matread and printm into the semantic checker
      matrix access now supports [int][variable] things but doesn't check if the variable is
           an int
      Made an sast and also added string literals.
      Fixed some stuff regarding scopes
      Fixed i+i issue
      Finishing merging
      Dumping work onto Nimo for p*m read and writes
      pmlib now works and Jane did something to semant.ml
      Semant checker can now recognizer pgmread and pgmwrite
      Added support for ppmread and ppmwrite. Still waiting on whether or not to include
          pbmread and pbmwrite


Jiangfeng Wang <jw3107@columbia.edu> (13):
      Edited parser to enable matrix initialization
      Added brackets to scanner; added matrix initialization and access to parser and ast.
      Replace FloatLit with num type
      fixed shift/reduct conflicts
      Revising semantic checker
      Modified semantchecker & sast file
      Revised semantic checker binop
      revised semant helperfunction and binop
      added sample exception test case
      semant checker fixed error ast.typ
      first working version of semant.ml
      working version of semant
      added success tests
```

9.4 Full git Log

The following log was generated using the command:

```
git log --color --stat --no-merges --pretty=format:"%h: %Cblue%aN <%aE>%Creset%nDate: %aD%
    nSubject: %s%nContent: %b"
```

```
1dfd79b: wodeni <wn2155@columbia.edu>
Date: Mon, 8 May 2017 05:24:12 -0400
Subject: [Report] Added git log history tex file
Content:
 doc/gitlog.tex | 113 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/ship.bin   | Bin 257040 -> 792000 bytes
 src/ship.mpl   |   2 +-
 3 files changed, 114 insertions(+), 1 deletion(-)

7b6c411: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Mon, 8 May 2017 05:08:49 -0400
Subject: Update MPL-arch
Content:
 doc/MPL-arch | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

76343dd: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Sun, 7 May 2017 22:25:18 -0400
Subject: Update README.md
Content:
 README.md | 253 -------------------------------
 1 file changed, 253 deletions(-)

b8952dd: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Sun, 7 May 2017 22:25:07 -0400
Subject: Create meeting-notes.md
Content:
 meeting-notes.md | 254 +++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 254 insertions(+)

80331d2: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Sun, 7 May 2017 22:23:39 -0400
Subject: Update README.md
Content:
 README.md | 6 ++++--
 1 file changed, 4 insertions(+), 2 deletions(-)

e39c1a5: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Sun, 7 May 2017 17:40:34 -0400
Subject: Update MPL-arch
Content:
 doc/MPL-arch | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

829b569: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Sun, 7 May 2017 17:40:29 -0400
Subject: Added MPL-arch.svg
Content:
 doc/MPL-arch.svg | 2 ++
 1 file changed, 2 insertions(+)

7805579: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Sun, 7 May 2017 16:57:04 -0400
Subject: Added MPL-arch
Content:
 doc/MPL-arch | 1 +
 1 file changed, 1 insertion(+)

c465fb7: wodeni <wn2155@columbia.edu>
Date: Sun, 7 May 2017 15:53:05 -0400
Subject: [Demo] Clean up and better pics
Content:
 rle-files/gun-long.bin |   Bin 200000 -> 0 bytes
 rle-files/gun.txt      |   Bin 40000 -> 0 bytes
 rle-files/line.txt     |   Bin 40000 -> 0 bytes
 rle-files/parse_rle.py |     2 +-
 rle-files/queen.txt    |   Bin 40000 -> 0 bytes
 rle-files/ship.bin     |   Bin 282240 -> 0 bytes
 src/Makefile           |     2 +-
 src/blur.mpl           |    13 +
 src/edge.mpl           |    15 +
 src/gun-long.mpl       |    29 +
 src/gun.bin            |   Bin 40000 -> 18000 bytes
 src/gun.mpl            |     4 +-
 src/img.sh             |    17 +
 src/lena.pgm           | 22191 +++++++++++++++++++++++++++++++++++++++++++++++
 src/line.bin           |   Bin 40000 -> 800 bytes
 src/line.mpl           |     2 +-
 src/queen.bin          |   Bin 40000 -> 1456 bytes
 src/queen.mpl          |     2 +-
 src/run.sh             |     4 +-
 src/sharpen.mpl        |    13 +
 src/ship.mpl           |    29 +
 src/utils.c            |    22 +-
 22 files changed, 22330 insertions(+), 15 deletions(-)

b98ab43: wodeni <wn2155@columbia.edu>
Date: Sun, 7 May 2017 14:29:28 -0400
Subject: [Demo] Added rle files
Content:
 rle-files/gun-long.bin        | Bin 0 -> 200000 bytes
 rle-files/gun.rle             |   7 ++
 rle-files/gun.txt             | Bin 0 -> 40000 bytes
 rle-files/line.rle            |   5 ++
 rle-files/line.txt            | Bin 0 -> 40000 bytes
 rle-files/parse_rle.py        | 201 ++++++++++++++++++++++++++++++++++++++++++
 rle-files/queen.txt           | Bin 0 -> 40000 bytes
 rle-files/queenbeeshuttle.rle |   6 ++
 rle-files/ship.bin            | Bin 0 -> 282240 bytes
 rle-files/ship.rle            | 145 ++++++++++++++++++++++++++++++
 10 files changed, 364 insertions(+)

1cf3239: chyzhang <chyzhang@brandeis.edu>
Date: Sun, 7 May 2017 01:57:41 -0400
Subject: fixed matread write test
Content:
 test/testVer1/test-matread.mpl  | 2 +-
 test/testVer1/test-matwrite.mpl | 4 ++-
 2 files changed, 3 insertions(+), 3 deletions(-)

86b2e8a: wodeni <wn2155@columbia.edu>
Date: Sun, 7 May 2017 01:53:55 -0400
Subject: [Test] matread/write small problem with file input
Content:
 src/utils.c                        |   2 +-
 test/testVer1/test-print-board.out | 200 ++++++++++++++++++----------
 2 files changed, 101 insertions(+), 101 deletions(-)

936e42e: chyzhang <chyzhang@brandeis.edu>
Date: Sun, 7 May 2017 01:44:56 -0400
Subject: for matwrite test
Content:
 test/testVer1/{matexample.txt => matoutput.bin} | Bin
 test/testVer1/test-matread.mpl                  |   2 +-
 test/testVer1/test-matwrite.mpl                 |   4 ++-
 3 files changed, 3 insertions(+), 3 deletions(-)

478bc27: chyzhang <chyzhang@brandeis.edu>
Date: Sun, 7 May 2017 01:39:26 -0400
Subject: for test read
```

```
Content:
 test/testVer1/gun.bin | Bin 0 -> 40000 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)

02f690c: chyzhang <chyzhang@brandeis.edu>
Date: Sun, 7 May 2017 01:38:34 -0400
Subject: test + gcd demo file
Content:
 src/test-gcd.mpl                     | 19 +++++++++++++++++++
 test/testVer1/test-matwrite.mpl      |  4 ++-
 test/{testVer1 => tmp}/test-ops2.mpl |  0
 test/{testVer1 => tmp}/test-ops2.out |  0
 testall.sh                           |  4 ++-
 5 files changed, 23 insertions(+), 4 deletions(-)

072edba: DRC9702 <drinconcruz@knox.edu>
Date: Sun, 7 May 2017 01:04:31 -0400
Subject: Added support for ppmread and ppmwrite. Still waiting on whether or not to include
    pbmread and pbmwrite
Content:
 src/semant.ml | 38 +++++++++++++++++++++++++++++++++++++-
 1 file changed, 37 insertions(+), 1 deletion(-)

2eff7bf: wodeni <wn2155@columbia.edu>
Date: Sun, 7 May 2017 00:58:54 -0400
Subject: [Demo] Now the convolution demo works.
Content:
 src/Makefile   |   2 +-
 src/codegen.ml |  43 +++++++-
 src/gol.mpl    |   2 +-
 src/pmlib.c    |  23 +++-
 src/ship.bin   | Bin 332336 -> 257040 bytes
 src/utils.c    | 315 +++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 6 files changed, 369 insertions(+), 16 deletions(-)

f04a818: DRC9702 <drinconcruz@knox.edu>
Date: Sun, 7 May 2017 00:16:07 -0400
Subject: Semant checker can now recognizer pgmread and pgmwrite
Content:
 src/semant.ml | 35 +++++++++++++++++++++++++++++---
 1 file changed, 29 insertions(+), 6 deletions(-)

4088ea7: chyzhang <chyzhang@brandeis.edu>
Date: Sat, 6 May 2017 23:38:16 -0400
Subject: changes in test for demo 2
Content:
 test/tmp/fail-func2.err     |  1 +
 test/tmp/fail-func2.mpl     | 12 ++++++++++++
 test/tmp/fail-func3.err     |  1 +
 test/tmp/fail-func3.mpl     | 12 ++++++++++++
 test/tmp/fail-funccheck.err |  1 +
 test/tmp/fail-funccheck.mpl | 11 +++++++++++
 test/tmp/test-entry.mpl     | 21 +++++++++++++++++++++
 test/tmp/test-entry.out     | 45 +++++++++++++++++++++++++++++++++++++++++++++
 test/tmp/test-mat-ops.mpl   |  9 +++++++++
 test/tmp/test-mat-ops.out   |  4 ++++
 10 files changed, 117 insertions(+)

78c3017: chyzhang <chyzhang@brandeis.edu>
Date: Sat, 6 May 2017 23:37:06 -0400
Subject: changes in test for demo
Content:
 test/scanner/scripts/ScannerTest.ml             |  3 ++-
 test/scanner/scripts/pass/_assignment.out       |  2 +-
 test/scanner/scripts/pass/_base_scanner.out     | 10 ++++---
 test/scanner/scripts/pass/_conditionals.out     |  6 +++--
 test/scanner/scripts/pass/_function.out         |  2 +-
 test/scanner/scripts/pass/_literal.out          |  4 ++-
 test/scanner/scripts/pass/_main_function.out    |  2 +-
 test/scanner/scripts/pass/_mixed_arithmetic.out | 10 ++++---
 test/testVer1/test-entry.mpl                    | 18 --------
 test/testVer1/test-entry.out                    |  5 ---
 test/testVer1/test-mat-ops.mpl                  |  9 ----
 test/testVer1/test-mat-ops.out                  |  4 --
 test/testVer1/test-ops1.mpl                     | 28 ++++++++++++-------
 test/testVer2/fail-expr1.err                    |  2 +-
 test/testVer2/fail-expr1.mpl                    |  2 +-
 test/testVer2/fail-func2.err                    |  1 -
 test/testVer2/fail-func2.mpl                    | 12 ------
 test/testVer2/fail-func3.err                    |  1 -
 test/testVer2/fail-func3.mpl                    | 12 ------
 test/testVer2/fail-funcheck.err                 |  1 -
 test/testVer2/fail-funcheck.mpl                 | 11 -----
 testall.sh                                      |  4 ++-
 22 files changed, 38 insertions(+), 111 deletions(-)

47fd16f: DRC9702 <drinconcruz@knox.edu>
Date: Sat, 6 May 2017 23:24:30 -0400
Subject: pmlib now works and Jane did something to semant.ml
Content:
 src/pmlib.c   | 71 ++++++++++++++++++++++++++++++++++---------------------
 src/semant.ml |  4 ++-
 2 files changed, 41 insertions(+), 34 deletions(-)

5d58803: DRC9702 <drinconcruz@knox.edu>
Date: Sat, 6 May 2017 22:38:08 -0400
Subject: Dumping work onto Nimo for p*m read and writes
Content:
 src/pmlib.c | 318 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 318 insertions(+)

4371d23: wodeni <wn2155@columbia.edu>
Date: Sat, 6 May 2017 22:29:57 -0400
Subject: [Demo] added binary files
Content:
 src/gol.mpl    |  29 +++++++++++++++++++++++++++++
 src/gun.bin    | Bin 0 -> 40000 bytes
 src/gun.mpl    |  29 +++++++++++++++++++++++++++++
 src/line.bin   | Bin 0 -> 40000 bytes
 src/line.mpl   |  29 +++++++++++++++++++++++++++++
 src/queen.bin  | Bin 0 -> 40000 bytes
 src/queen.mpl  |  29 +++++++++++++++++++++++++++++
 src/ship.bin   | Bin 0 -> 332336 bytes
 8 files changed, 116 insertions(+)

c7569d9: wodeni <wn2155@columbia.edu>
Date: Sat, 6 May 2017 22:11:14 -0400
Subject: [Clean up] unneccessary files
Content:
 src/buildmat.c                    |  20 --
 src/demo.mpl                      |  54 ----
 src/fmat.c                        |  14 -
 src/matread.c                     |  38 ---
 src/old_semant.ml                 | 300 --------------------
 src/printbig.c                    |  75 -----
 src/printm.c                      |  50 ----
 src/rle-files/gun.txt             | Bin 40000 -> 0 bytes
 src/semant.ml                     |   3 +-
 src/test-mat.mpl                  |  15 -
 src/test-matrix-exception.mpl     |   6 -
 src/test-printm.mpl               |   5 -
 src/test-wrap1.mpl                |  18 --
 src/tokenize.ml                   |  54 ----
 src/utils.c                       |   4 +-
 test/testVer1/test-print-board.mpl |   9 +-
 16 files changed, 10 insertions(+), 655 deletions(-)
```

```
f0e79b4: ZhangChi <ChyZin@dyn-160-39-144-79.dyn.columbia.edu>
Date: Sat, 6 May 2017 22:06:40 -0400
Subject: test for demo
Content:
 test/{testVer2/fail-func8.mpl => testVer1/test-entry.mpl} | 5 +++-
 test/testVer1/test-entry.out                              | 5 +++++
 test/testVer2/fail-func8.err                              | 1 -
 test/{testVer2 => tmp}/fail-dead1.err                     | 0
 test/{testVer2 => tmp}/fail-dead1.mpl                     | 0
 test/{testVer2 => tmp}/fail-dead2.err                     | 0
 test/{testVer2 => tmp}/fail-dead2.mpl                     | 0
 test/{testVer1 => tmp}/test-mat-ops1.mpl                  | 0
 test/{testVer1 => tmp}/test-mat-ops1.out                  | 0
 test/{testVer1 => tmp}/test-mat-ops2.mpl                  | 0
 test/{testVer1 => tmp}/test-mat-ops2.out                  | 0
 test/{testVer1 => tmp}/test-mat-ops3.mpl                  | 0
 test/{testVer1 => tmp}/test-mat-ops3.out                  | 0
 test/{testVer1 => tmp}/test-mat-ops4.mpl                  | 0
 test/{testVer1 => tmp}/test-mat-ops4.out                  | 0
 test/{testVer1 => tmp}/test-mat-ops5.mpl                  | 0
 test/{testVer1 => tmp}/test-mat-ops5.out                  | 0
 test/{testVer1 => tmp}/test-mat-ops6.mpl                  | 0
 test/{testVer1 => tmp}/test-mat-ops6.out                  | 0
 19 files changed, 8 insertions(+), 3 deletions(-)

f0f86ce: ZhangChi <ChyZin@dyn-160-39-144-79.dyn.columbia.edu>
Date: Sat, 6 May 2017 21:19:00 -0400
Subject: test merged from merge-sast
Content:
 test/testVer1/test-all2.mpl     |  6 +++--
 test/testVer1/test-apply.mpl    |  2 +-
 test/testVer1/test-func.mpl     |  2 +-
 test/testVer1/test-local1.mpl   |  9 +++++--
 test/testVer1/test-mat-ops1.mpl |  6 ++++-
 test/testVer1/test-mat-ops2.mpl |  4 +++-
 test/testVer1/test-matall.mpl   |  2 +-
 test/testVer1/test-ops2.mpl     | 24 ++++++++++++------
 test/testVer2/fail-assign1.err  |  2 +-
 test/testVer2/fail-assign2.err  |  2 +-
 test/testVer2/fail-expr1.mpl    |  2 +-
 test/testVer2/fail-func2.mpl    |  6 +++--
 test/testVer2/fail-func8.mpl    |  4 +--
 test/testVer2/fail-funcheck.err |  2 +-
 test/testVer2/fail-funcheck.mpl |  4 ++-
 test/testVer2/fail-global1.err  |  2 +-
 test/testVer2/fail-if1.err      |  2 +-
 test/testVer2/fail-return1.err  |  2 +-
 test/testVer2/fail-return2.err  |  2 +-
 test/testVer2/fail-return2.mpl  |  4 ++-
 test/testVer2/fail-while1.err   |  2 +-
 test/testVer2/fail-while2.mpl   |  2 +-
 testall.sh                      |  5 +--
 23 files changed, 49 insertions(+), 49 deletions(-)

963ec12: DRC9702 <drinconcruz@knox.edu>
Date: Sat, 6 May 2017 05:53:20 -0400
Subject: Fixed i+i issue
Content:
 src/semant.ml | 78 ++++++++++++++++++++++++++++++++++++++++---------
 1 file changed, 54 insertions(+), 24 deletions(-)

464633b: wodeni <wn2155@columbia.edu>
Date: Sat, 6 May 2017 05:45:06 -0400
Subject: [BUG FIX] neighbor computation
Content:
 src/codegen.ml     | 18 ++++++++++++++++--
 src/demo.mpl       | 30 ++++++++++++++++++++++++++++--
 src/exceptions.ml |  1 +
 src/semant.ml     |  2 +-
 src/utils.c       |  2 ++
 5 files changed, 44 insertions(+), 9 deletions(-)

f51bc40: wodeni <wn2155@columbia.edu>
Date: Sat, 6 May 2017 03:36:42 -0400
Subject: Successfully merged the SAST version of semantic checker
Content: - Cleaned up the code for codegen
- Fixed bugs in both codegen and sement
- Test cases are not passing yet

 src/codegen.ml     | 16 ++++++++----
 src/mpl.ml         |  2 +-
 src/test-wrap1.mpl |  9 +++++--
 3 files changed, 14 insertions(+), 13 deletions(-)

e9e0664: chyzhang <chyzhang@brandeis.edu>
Date: Sat, 6 May 2017 03:24:25 -0400
Subject: testall.sh
Content:
 testall.sh | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

06298d4: chyzhang <chyzhang@brandeis.edu>
Date: Sat, 6 May 2017 03:02:19 -0400
Subject: test refined for new sast
Content:
 test/testVer1/test-all1.mpl         |  4 ++-
 test/testVer1/test-all2.mpl         |  4 +++-
 test/testVer1/test-apply.mpl        |  2 +-
 test/testVer1/test-apply.out        |  5 +++++
 test/testVer1/test-func.mpl         |  7 +++--
 test/testVer1/test-mat.mpl          | 18 +++++++++-----
 test/testVer1/test-matall.mpl       |  4 ++-
 test/testVer1/test-print-board.out  |  2 +-
 test/testVer2/fail-assign1.mpl      |  2 +-
 test/testVer2/fail-assign2.mpl      |  2 +-
 test/testVer2/fail-expr1.mpl        | 10 +++++---
 test/testVer2/fail-func1.mpl        |  2 +-
 test/testVer2/fail-func2.mpl        |  8 ++++++-
 test/testVer2/fail-func3.mpl        |  8 ++++++-
 test/testVer2/fail-func5.mpl        |  2 +-
 test/testVer2/fail-func6.mpl        |  3 ++-
 test/testVer2/fail-func7.mpl        |  3 ++-
 test/testVer2/fail-func8.mpl        | 14 ++++++++++--
 test/testVer2/fail-func9.mpl        |  3 ++-
 test/testVer2/fail-funcheck.mpl     | 13 +++++----
 test/testVer2/fail-global1.mpl      |  2 +-
 21 files changed, 69 insertions(+), 49 deletions(-)

8199e36: wodeni <wn2155@columbia.edu>
Date: Sat, 6 May 2017 02:57:32 -0400
Subject: Fixed merged bugs
Content:
 src/codegen.ml | 20 +++++++++++-----
 src/mpl.ml     | 14 ++++-----
 src/semant.ml  | 13 ++++++----
 3 files changed, 21 insertions(+), 26 deletions(-)

2762f92: wodeni <wn2155@columbia.edu>
Date: Sat, 6 May 2017 02:28:47 -0400
Subject: Preping for merge from matrix
Content:
 src/ast.ml        |  10 +-
 src/codegen.ml    | 379 +++++++++++++++++++++++++++--------------
 src/exceptions.ml |  14 +-
 src/mpl.ml        |  14 +-
 src/sast.ml       |  28 ++-
 5 files changed, 227 insertions(+), 218 deletions(-)

5b61f08: DRC9702 <drinconcruz@knox.edu>
Date: Sat, 6 May 2017 02:28:27 -0400
Subject: Fixed some stuff regarding scopes
Content:
 src/ast.ml    |   2 +
 src/sast.ml   |   4 +-
 src/semant.ml | 149 +++++++++++++++++++++++++++++++++++++++++++++++++++++++-----
 3 files changed, 131 insertions(+), 24 deletions(-)

7f5d560: Jiangfeng Wang <jw3107@columbia.edu>
Date: Fri, 5 May 2017 23:11:22 -0400
Subject: added success tests
Content:
 test/testVer1/test-all1.mpl         |  13 +++++
 test/testVer1/test-all1.out         |   1 +
 test/testVer1/test-all2.mpl         |  13 +++++
 test/testVer1/test-all2.out         |   1 +
 test/testVer1/test-apply.mpl        |  12 +++++
 test/testVer1/test-func.mpl         |   4 +-
 test/testVer1/test-if1.mpl          |   2 +-
 test/testVer1/test-if2.mpl          |   2 +-
 test/testVer1/test-if3.mpl          |   2 +-
 test/testVer1e /test-if4.mpl          |   2 +-
 test/testVer1/test-if5.mpl          |   4 +-
 test/testVer1/test-local1.mpl       |   4 +-
 test/testVer1/test-local2.mpl       |   4 +-
 test/testVer1/test-mat-ops.mpl      |   9 ++++
 test/testVer1/test-mat-ops.out      |   4 ++
 test/testVer1/test-mat-ops1.mpl     |   9 ++++
 test/testVer1/test-mat-ops1.out     |   4 ++
 test/testVer1/test-mat-ops2.mpl     |   9 ++++
 test/testVer1/test-mat-ops2.out     |   4 ++
 test/testVer1/test-mat-ops3.mpl     |   9 ++++
 test/testVer1/test-mat-ops3.out     |   4 ++
 test/testVer1/test-mat-ops4.mpl     |   9 ++++
 test/testVer1/test-mat-ops4.out     |   4 ++
 test/testVer1/test-mat-ops5.mpl     |   9 ++++
 test/testVer1/test-mat-ops5.out     |   4 ++
 test/testVer1/test-mat-ops6.mpl     |   9 ++++
 test/testVer1/test-mat-ops6.out     |   4 ++
 test/testVer1/test-mat.mpl          |   6 +-
 test/testVer1/test-matall.mpl       |   6 +-
 test/testVer1/test-matread.mpl      |   2 +-
 test/testVer1/test-matwrite.mpl     |  11 +--
 test/testVer1/test-ops1.mpl         |   2 +-
 test/testVer1/test-ops2.mpl         |   2 +-
 test/testVer1/test-print-board.mpl  |  26 ++++++++++
 test/testVer1/test-print-board.out  | 100 +++++++++++++++++++++++++++++++++++++
 test/testVer1/test-print.mpl        |   7 +-
 test/testVer1/test-printm.mpl       |   2 +-
 test/testVer1/test-prints.mpl       |   2 +-
 test/testVer1/test-var1.mpl         |   2 +-
 test/testVer1/test-while1.mpl       |   2 +-
 test/testVer1/test-while2.mpl       |   4 +-
 41 files changed, 285 insertions(+), 44 deletions(-)

0d63048: ZhangChi <ChyZin@dyn-160-39-144-79.dyn.columbia.edu>
Date: Fri, 5 May 2017 20:36:52 -0400
Subject: related testall.sh updated
Content:
 testall.sh | 19 +++++++++++----
 1 file changed, 11 insertions(+), 8 deletions(-)

fa5e493: ZhangChi <ChyZin@dyn-160-39-144-79.dyn.columbia.edu>
Date: Fri, 5 May 2017 20:29:00 -0400
```

```
Subject: add test err case
Content:
 test/testVer2/fail-assign1.err   |  1 +
 test/testVer2/fail-assign1.mpl   | 11 +++++++++++
 test/testVer2/fail-assign2.err   |  1 +
 test/testVer2/fail-assign2.mpl   |  7 +++++++
 test/testVer2/fail-dead1.err     |  1 +
 test/testVer2/fail-dead1.mpl     |  8 ++++++++
 test/testVer2/fail-dead2.err     |  1 +
 test/testVer2/fail-dead2.mpl     | 10 ++++++++++
 test/testVer2/fail-expr1.err     |  1 +
 test/testVer2/fail-expr1.mpl     | 14 ++++++++++++++
 test/testVer2/fail-func1.err     |  1 +
 test/testVer2/fail-func1.mpl     | 12 ++++++++++++
 test/testVer2/fail-func2.err     |  1 +
 test/testVer2/fail-func2.mpl     |  8 ++++++++
 test/testVer2/fail-func3.err     |  1 +
 test/testVer2/fail-func3.mpl     |  8 ++++++++
 test/testVer2/fail-func4.err     |  1 +
 test/testVer2/fail-func4.mpl     | 12 ++++++++++++
 test/testVer2/fail-func5.err     |  1 +
 test/testVer2/fail-func5.mpl     | 14 ++++++++++++++
 test/testVer2/fail-func6.err     |  1 +
 test/testVer2/fail-func6.mpl     |  9 +++++++++
 test/testVer2/fail-func7.err     |  1 +
 test/testVer2/fail-func7.mpl     |  9 +++++++++
 test/testVer2/fail-func8.err     |  1 +
 test/testVer2/fail-func8.mpl     | 13 +++++++++++++
 test/testVer2/fail-func9.err     |  1 +
 test/testVer2/fail-func9.mpl     |  9 +++++++++
 test/testVer2/fail-funcheck.err |  1 +
 test/testVer2/fail-funcheck.mpl | 14 ++++++++++++++
 test/testVer2/fail-global1.err   |  1 +
 test/testVer2/fail-global1.mpl   |  9 +++++++++
 test/testVer2/fail-if1.err       |  1 +
 test/testVer2/fail-if1.mpl       |  6 ++++++
 test/testVer2/fail-if2.err       |  1 +
 test/testVer2/fail-if2.mpl       |  6 ++++++
 test/testVer2/fail-if3.err       |  1 +
 test/testVer2/fail-if3.mpl       |  8 ++++++++
 test/testVer2/fail-nomain.err    |  1 +
 test/testVer2/fail-nomain.mpl    |  0
 test/testVer2/fail-return1.err   |  1 +
 test/testVer2/fail-return1.mpl   |  4 ++++
 test/testVer2/fail-return2.err   |  1 +
 test/testVer2/fail-return2.mpl   | 10 ++++++++++
 test/testVer2/fail-while1.err    |  1 +
 test/testVer2/fail-while1.mpl    | 13 +++++++++++++
 test/testVer2/fail-while2.err    |  1 +
 test/testVer2/fail-while2.mpl    | 13 +++++++++++++
 48 files changed, 251 insertions(+)

9f1a339: wodeni <wn2155@columbia.edu>
Date: Fri, 5 May 2017 19:24:47 -0400
Subject: preping for merge from backend
Content:
 src/codegen.ml | 9 ++++---
 src/demo.mpl   | 4 ++-
 2 files changed, 6 insertions(+), 7 deletions(-)

b8d6cf2: wodeni <wn2155@columbia.edu>
Date: Fri, 5 May 2017 01:38:32 -0400
Subject: [BUG FIX] The segfault is now fixed
Content: - It was caused by APPLY's repeated alloca on the stack. I changed that
  to malloc and free.
- I am very happy

 src/codegen.ml      | 12 +++++++++--
 src/demo.mpl        | 16 +++-------
 src/mpl.ml          |  2 +-
 src/run.sh          |  4 +++-
 src/test-wrap1.mpl  |  2 +-
 src/utils.c         |  4 ++-
 6 files changed, 19 insertions(+), 21 deletions(-)

caa9ec8: wodeni <wn2155@columbia.edu>
Date: Thu, 4 May 2017 23:45:01 -0400
Subject: [BUG] Apply: segfault when run too many times
Content: - I counted the number of loops until segfault. It seems that the larger
  the matrix, the faster it will occur, in terms of loop counts
- The relationship seems to be almost perfectly linear here
- Suspected memcpy, but really?

 src/test-wrap1.mpl | 17 +++++++++++++++++
 1 file changed, 17 insertions(+)

e89077e: wodeni <wn2155@columbia.edu>
Date: Thu, 4 May 2017 19:45:09 -0400
Subject: [Demo] First version of demo
Content: - We changed the apply operator codegen so that we generate loops rather
  then naively generating a lot of statements
- The neighboring policy for apply now changed to wrap around, for
  simplicity of the code and possibly demo
- Many many bug fixes
- BUG: when we run the demo it will segfault when the thing produced by
  gliber gun hits the boundry of the canvas
        - I tried printing out the indices we have accessed, but I
          didn't find anything out of bounds.

 demo.mpl                |  29 +++++++
 src/codegen.ml          | 235 ++++++++++++++++++++++++++++++++++--------
 src/demo.mpl            |  30 +++--
 src/mpl.ml              |   4 +-
 src/rle-files/gun.txt   | Bin 0 -> 40000 bytes
 src/run.sh              |  15 ++++
 src/script.sh           |   8 -
 src/test-matread.mpl    |  11 +-
 src/utils.c             |  46 ++++++++++
 testall.log             | 173 -------------------
 10 files changed, 270 insertions(+), 281 deletions(-)

db41fbe: DRC9702 <drinconcruz@knox.edu>
Date: Wed, 3 May 2017 20:03:52 -0400
Subject: Made an sast and also added string literals.
Content:
 src/ast.ml                 |   4 +-
 src/newSemanticChecker.ml  | 165 -------------
 src/old_semant.ml          | 300 ++++++++++++++++++++++++++++++++++++++++++++++
 src/parser.mly             |   3 +-
 src/sast.ml                |  40 +++++-
 src/scanner.mll            |   1 +
 src/semant.ml              | 140 +++++++++++++++----
 src/test-matread.mpl       |   4 +-
 8 files changed, 434 insertions(+), 223 deletions(-)

ee72ed7: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 2 May 2017 15:14:38 -0400
Subject: modify golden ref because of changes made in printm. Now all test pass
Content:
 matexample.txt                  | Bin 0 -> 36 bytes
 test/testVer1/test-if5.out      |  12 ++++++++++-
 test/testVer1/test-local1.out   |  12 ++++++++++-
 test/testVer1/test-local2.out   |   6 +++++-
 test/testVer1/test-matall.out   |  12 ++++++++++-
 test/testVer1/test-matread.out  |   6 +++++-
 test/testVer1/test-matwrite.out |   6 +++++-
 test/testVer1/test-ops2.mpl     |   2 --
 test/testVer1/test-ops2.out     |   1 +
 test/testVer1/test-printm.out   |   5 ++++-
 test/testVer1/test-while2.out   |  12 ++++++++++--
 11 files changed, 60 insertions(+), 14 deletions(-)

c177f9c: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 2 May 2017 14:59:05 -0400
Subject: modified the apply, now it can first read the matrix and remember all the entries,
    then store all changes
Content:
 src/codegen.ml | 53 ++++++++++++++++++++++++++++++++++++++---------------
 1 file changed, 38 insertions(+), 15 deletions(-)

d651f9a: wodeni <wn2155@columbia.edu>
Date: Tue, 2 May 2017 10:51:48 -0400
Subject: added demo.mpl
Content:
 src/demo.mpl | 42 ++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 42 insertions(+)

f8eee95: wodeni <wn2155@columbia.edu>
Date: Mon, 1 May 2017 23:38:15 -0400
Subject: In progress: fix build_apply
Content:
 src/codegen.ml  | 53 ++++++++++++++++++++++++++++-------------------------
 src/scanner.mll |  6 +++---
 src/utils.c     |  4 ++--
 3 files changed, 33 insertions(+), 30 deletions(-)

aef0bd0: DRC9702 <drinconcruz@knox.edu>
Date: Mon, 1 May 2017 23:33:16 -0400
Subject: matrix access now supports [int][variable] things but doesn't check if the variable
     is an int
Content:
 src/ast.ml     |  2 +-
 src/parser.mly |  2 +-
 src/semant.ml  | 28 ++++++++++++++++++++++++++--
 3 files changed, 28 insertions(+), 4 deletions(-)

3c12d6a: DRC9702 <drinconcruz@knox.edu>
Date: Mon, 1 May 2017 21:34:34 -0400
Subject: Added in matread and printm into the semantic checker
Content:
 src/semant.ml         | 22 ++++++++++++++++++----
 src/test-matread.mpl  |  9 +++++----
 2 files changed, 23 insertions(+), 8 deletions(-)

cee5726: chyzhang <chyzhang@brandeis.edu>
Date: Fri, 28 Apr 2017 00:02:27 -0400
Subject: basic fmat can be formed now
Content:
 src/codegen.ml  | 45 +++++++++++++++++++++++++++++++++++----------
 src/fmat.c      | 14 ++++++++++++++
 src/parser.mly  |  1 +
 src/scanner.mll |  2 +-
 4 files changed, 51 insertions(+), 11 deletions(-)

41be668: chyzhang <chyzhang@brandeis.edu>
Date: Wed, 26 Apr 2017 10:57:12 -0400
Subject: changes for test
Content:
 test/test-prints.mpl        |  3 ---
 test/testVer1/test-ops1.mpl |  3 +--
 test/testVer1/test-ops1.out | 76 +++++++++-------------------------------------------------------------
 test/testVer1/test-ops2.mpl |  3 +--
 test/testVer1/test-ops2.out | 23 +++++++----------------
 test/testVer1/test-var2.mpl | 13 -------------
 test/testVer1/test-var2.out |  1 -
 7 files changed, 29 insertions(+), 93 deletions(-)

7f9d596: DRC9702 <drinconcruz@knox.edu>
Date: Tue, 25 Apr 2017 16:37:32 -0400
Subject: Semantic Checker now accepts the apply operator
Content:
 src/mpl.ml                    |  2 +-
 src/semant.ml                 | 12 +++++++---
 src/test-matread.mpl          | 12 +++++----
 src/test-matrix-exception.mpl |  2 +-
 4 files changed, 14 insertions(+), 14 deletions(-)

b97dbd5: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 25 Apr 2017 15:36:17 -0400
Subject: golden ref local2 renew
Content:
 test/testVer1/test-local2.out | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

2a95ffa: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 25 Apr 2017 02:18:12 -0400
Subject: TEST REFINED
Content:
 test/testVer1/test-if5.mpl    | 26 ++++++++++-----
 test/testVer1/test-if5.out    |  4 +-
 test/testVer1/test-local1.mpl | 25 ++++++++++++---
 test/testVer1/test-local1.out | 12 +++++++++
 test/testVer1/test-local2.mpl | 59 ++++++++++++++++++++++++++++++++++-----
 test/testVer1/test-local2.out | 11 +++++++-
 test/testVer1/test-matall.out |  4 +-
 test/testVer1/test-ops1.mpl   |  6 +++-
 test/testVer1/test-ops2.mpl   |  7 +++-
 test/testVer1/test-printm.out |  3 +-
 test/testVer1/test-var1.mpl   |  2 -
 test/testVer1/test-var1.out   |  1 -
 test/testVer1/test-while2.mpl | 29 ++++++++++++-----
 test/testVer1/test-while2.out |  2 ++
 testall.sh                    |  8 +++--
 15 files changed, 137 insertions(+), 62 deletions(-)

23f7f46: chyzhang <chyzhang@brandeis.edu>
Date: Mon, 24 Apr 2017 21:09:53 -0400
Subject: prints works now
Content:
 src/codegen.ml                | 7 ++++--
 test/testVer1/test-prints.mpl | 4 +++-
 test/testVer1/test-prints.out | 4 +++-
 testall.sh                    | 2 +-
 4 files changed, 11 insertions(+), 6 deletions(-)

ab63ab8: chyzhang <chyzhang@brandeis.edu>
Date: Mon, 24 Apr 2017 14:23:43 -0400
Subject: upload golden ref and do some cleaning
Content:
 test/fail-assign1.err        |   1 -
 test/fail-assign1.mc         |   1 -
 test/fail-assign2.err        |   1 -
 test/fail-assign2.mc         |   1 -
 test/fail-assign3.err        |   1 -
 test/fail-assign3.mc         |  11 --
 test/fail-dead1.err          |   1 -
 test/fail-dead1.mc           |   8 -
 test/fail-dead2.err          |   1 -
 test/fail-dead2.mc           |  10 --
 test/fail-expr1.err          |   1 -
 test/fail-expr1.mc           |  18 ---
 test/fail-expr2.err          |   1 -
```

```
test/fail-expr2.mc             |  14 --
test/fail-for1.err             |   1 -
test/fail-for1.mc              |  13 --
test/fail-for2.err             |   1 -
test/fail-for2.mc              |   8 -
test/fail-for3.err             |   1 -
test/fail-for3.mc              |   8 -
test/fail-for4.err             |   1 -
test/fail-for4.mc              |   8 -
test/fail-for5.err             |   1 -
test/fail-for5.mc              |  10 --
test/fail-func1.err            |   1 -
test/fail-func1.mc             |  12 --
test/fail-func2.err            |   1 -
test/fail-func2.mc             |   8 -
test/fail-func3.err            |   1 -
test/fail-func3.mc             |   8 -
test/fail-func4.err            |   1 -
test/fail-func4.mc             |  12 --
test/fail-func5.err            |   1 -
test/fail-func5.mc             |  14 --
test/fail-func6.err            |   1 -
test/fail-func6.mc             |   9 --
test/fail-func7.err            |   1 -
test/fail-func7.mc             |   9 --
test/fail-func8.err            |   1 -
test/fail-func8.mc             |  13 --
test/fail-func9.err            |   1 -
test/fail-func9.mc             |   9 --
test/fail-global1.err          |   1 -
test/fail-global1.mc           |   9 --
test/fail-global2.err          |   1 -
test/fail-global2.mc           |   9 --
test/fail-if1.err              |   1 -
test/fail-if1.mc               |   6 -
test/fail-if2.err              |   1 -
test/fail-if2.mc               |   6 -
test/fail-if3.err              |   1 -
test/fail-if3.mc               |   8 -
test/fail-nomain.err           |   1 -
test/fail-nomain.mc            |   0
test/fail-return1.err          |   1 -
test/fail-return1.mc           |   4 -
test/fail-return2.err          |   1 -
test/fail-return2.mc           |  10 --
test/fail-while1.err           |   1 -
test/fail-while1.mc            |  13 --
test/fail-while2.err           |   1 -
test/fail-while2.mc            |  13 --
test/test-add1.mc              |  10 --
test/test-add1.out             |   1 -
test/test-arith1.mc            |   5 -
test/test-arith1.out           |   1 -
test/test-arith2.mc            |   5 -
test/test-arith2.out           |   1 -
test/test-arith3.mc            |  13 --
test/test-arith3.out           |   1 -
test/test-fib.mc               |  16 --
test/test-fib.out              |   6 -
test/test-for1.mc              |   9 --
test/test-for1.out             |   6 -
test/test-for2.mc              |  11 --
test/test-for2.out             |   6 -
test/test-func1.mc             |  12 --
test/test-func1.out            |   1 -
test/test-func2.mc             |  18 ---
test/test-func2.out            |   1 -
test/test-func3.mc             |  13 --
```

```
 test/test-func3.out           |   4 -
 test/test-func4.mc            |  14 --
 test/test-func4.out           |   1 -
 test/test-func5.mc            |   9 --
 test/test-func5.out           |   0
 test/test-func6.mc            |   9 --
 test/test-func6.out           |   1 -
 test/test-func7.mc            |  13 --
 test/test-func7.out           |   1 -
 test/test-func8.mc            |  10 --
 test/test-func8.out           |   1 -
 test/test-gcd.mc              |  15 --
 test/test-gcd.out             |   3 -
 test/test-gcd2.mc             |  14 --
 test/test-gcd2.out            |   3 -
 test/test-global1.mc          |  30 ----
 test/test-global1.out         |   4 -
 test/test-global2.mc          |  10 --
 test/test-global2.out         |   1 -
 test/test-global3.mc          |  11 --
 test/test-global3.out         |   1 -
 test/test-hello.mc            |   7 -
 test/test-hello.out           |   3 -
 test/test-if1.mc              |   6 -
 test/test-if1.out             |   2 -
 test/test-if2.mc              |   6 -
 test/test-if2.out             |   2 -
 test/test-if3.mc              |   6 -
 test/test-if3.out             |   1 -
 test/test-if4.mc              |   6 -
 test/test-if4.out             |   2 -
 test/test-if5.mc              |  16 --
 test/test-if5.out             |   2 -
 test/test-local1.mc           |  13 --
 test/test-local1.out          |   1 -
 test/test-local2.mc           |  14 --
 test/test-local2.out          |   1 -
 test/test-ops1.mc             |  28 ----
 test/test-ops1.out            |  24 ---
 test/test-ops2.mc             |  17 ---
 test/test-ops2.out            |  14 --
 test/test-var1.mc             |   7 -
 test/test-var1.out            |   1 -
 test/test-var2.mc             |  13 --
 test/test-var2.out            |   1 -
 test/test-while1.mc           |  11 --
 test/test-while1.out          |   6 -
 test/test-while2.mc           |  16 --
 test/test-while2.out          |   1 -
 test/testVer1/matexample.txt  | Bin 0 -> 36 bytes
 testall.log                   | 173 ++++++++++++++++++++++++++++++++++++++++++++
 testall.sh                    |  61 +++++++----
 133 files changed, 204 insertions(+), 861 deletions(-)

01a488b: chyzhang <chyzhang@brandeis.edu>
Date: Mon, 24 Apr 2017 14:16:15 -0400
Subject: test committed and testall.sh modified. Problem now:1:if a mpl file is all
    commented, it cannot be compiled. 2:Something wrong with prints when tested invalid
    character
Content:
 result.txt                       | Bin 0 -> 11848 bytes
 test/testVer1/test-func.out      |   2 ++
 test/testVer1/test-if1.out       |   2 ++
 test/testVer1/test-if2.out       |   2 ++
 test/testVer1/test-if3.out       |   1 +
 test/testVer1/test-if4.out       |   2 ++
 test/testVer1/test-if5.out       |   2 ++
 test/testVer1/test-local1.out    |   1 +
 test/testVer1/test-local2.out   |   1 +
 test/testVer1/test-mat.mpl      |  28 ++++++++++++++++
 test/testVer1/test-mat.out      |   9 ++++++
 test/testVer1/test-matall.mpl   |  17 ++++++++++
 test/testVer1/test-matall.out   |   4 +++
 test/testVer1/test-matread.mpl  |   6 ++++
 test/testVer1/test-matread.out  |   1 +
 test/testVer1/test-matwrite.mpl |  17 ++++++++++
 test/testVer1/test-matwrite.out |   1 +
 test/testVer1/test-ops1.out     |  70 ++++++++++++++++++++++++++++++++++++++++
 test/testVer1/test-ops2.out     |  14 ++++++++
 test/testVer1/test-print.mpl    |   1 -
 test/testVer1/test-print.out    |   1 +
 test/testVer1/test-printm.mpl   |   5 +++
 test/testVer1/test-printm.out   |   2 ++
 test/testVer1/test-prints.mpl   |   4 +++
 test/testVer1/test-prints.out   |   1 +
 test/testVer1/test-var1.mpl     |   1 +
 test/testVer1/test-var1.out     |   2 ++
 test/testVer1/test-var2.out     |   1 +
 test/testVer1/test-while1.out   |   6 ++++
 29 files changed, 203 insertions(+), 1 deletion(-)

8383a0c: wodeni <wn2155@columbia.edu>
Date: Mon, 24 Apr 2017 11:07:17 -0400
Subject: [matwrite] added utils.c
Content:
 src/utils.c | 156 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 156 insertions(+)

2d6afd6: wodeni <wn2155@columbia.edu>
Date: Sun, 23 Apr 2017 23:29:54 -0400
Subject: [matwrite] working version with one test case
Content: - Added a function in C
- Put all the C functions inside utils.c

 README.md             |  2 +-
 src/Makefile          | 12 ++++++---
 src/codegen.ml        | 36 +++++++++++++++++++++++++++-----
 src/test-matread.mpl  |  1 +
 4 files changed, 35 insertions(+), 16 deletions(-)

12b5a65: DRC9702 <drinconcruz@knox.edu>
Date: Sun, 23 Apr 2017 22:32:03 -0400
Subject: Compilation doesn't work because i haven't setup here
Content:
 src/semant.ml | 16 ++++++++++++++-
 1 file changed, 14 insertions(+), 2 deletions(-)

a72bd93: Jiangfeng Wang <jw3107@columbia.edu>
Date: Sun, 23 Apr 2017 22:01:29 -0400
Subject: working version of semant
Content:
 src/ast.ml                      | 19 +++++++------
 src/semant.ml                   | 25 +++++++++++++++-----
 src/test-matrix-exception.mpl |  4 +++-
 3 files changed, 25 insertions(+), 23 deletions(-)

5b242e2: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 18 Apr 2017 17:26:33 -0400
Subject: "04/18/2017 Implement matread function. We implemented matread function by calling
    C function successfully. It works like printm and take 2 parameters, path and the
    destination matrix. Users are assumed to know the size of the matrix they want to read."
Content:
 src/Makefile          |  7 ++++--
 src/buildmat.c        | 20 ++++++++++++++++++++
 src/codegen.ml        | 15 +++++++++++++++
 src/matread.c         | 38 ++++++++++++++++++++++++++++++++++++++
 src/test-matread.mpl | 14 ++++++++++++++
 5 files changed, 91 insertions(+), 3 deletions(-)

aec9dc5: wodeni <wn2155@columbia.edu>
Date: Sun, 16 Apr 2017 21:24:37 -0400
Subject: [Entry function] Now have # variables work
Content: - We manually set the arguments of the entry function so that we can
  access neighbors with the entry function using the sharp notation.
- We set all the out-of-bound neighbors to 0.
- TODO:
    - fMat‘- Wait until monday.README.md | 4 ‘++++
 src/Makefile       |  4 ++-
 src/codegen.ml     | 32 +++++++++++++++++++++------
 src/exceptions.ml  |  2 +-
 src/parser.mly     |  9 +++++++++
 5 files changed, 37 insertions(+), 14 deletions(-)

27ae947: Jiangfeng Wang <jw3107@columbia.edu>
Date: Sun, 16 Apr 2017 20:30:57 -0400
Subject: first working version of semant.ml
Content:
 src/semant.ml | 173 +++++++++++++++++++++++++++++++++++------------------
 1 file changed, 105 insertions(+), 68 deletions(-)

f0be81e: Jiangfeng Wang <jw3107@columbia.edu>
Date: Sun, 16 Apr 2017 19:25:35 -0400
Subject: semant checker fixed error ast.typ
Content:
 src/newSemanticChecker.ml | 77 +++++++++++++++++++++++++-----------
 1 file changed, 41 insertions(+), 36 deletions(-)

164b00a: Jiangfeng Wang <jw3107@columbia.edu>
Date: Fri, 14 Apr 2017 16:19:49 -0400
Subject: added sample exception test case
Content:
 src/test-matrix-exception.mpl | 4 ++++
 1 file changed, 4 insertions(+)

4f35a40: Jiangfeng Wang <jw3107@columbia.edu>
Date: Fri, 14 Apr 2017 16:09:40 -0400
Subject: revised semant helperfunction and binop
Content:
 src/newSemanticChecker.ml | 116 +++++++++++++++++++++++------------
 1 file changed, 59 insertions(+), 57 deletions(-)

ed6a2a3: chyzhang <chyzhang@brandeis.edu>
Date: Thu, 13 Apr 2017 22:51:04 -0400
Subject: add @ test
Content:
 test/testVer1/test-func.mpl | 17 +++++++++++++++++
 1 file changed, 17 insertions(+)

2496f60: DRC9702 <drinconcruz@knox.edu>
Date: Thu, 13 Apr 2017 20:51:53 -0400
Subject: Adding the functions Jane wants. What's good Nimo?
Content:
 src/newSemanticChecker.ml | 26 +++++++++++++++++++++++++-
 1 file changed, 25 insertions(+), 1 deletion(-)

6bd3307: wodeni <wn2155@columbia.edu>
Date: Wed, 12 Apr 2017 17:44:30 -0400
Subject: [Apply] Apply operator now works!
Content: - Implemented the apply operator
- Fixed the order of matrix literal in the parser
- Comment: code is ugly
- TODOs
    - Function matrix
 src/codegen.ml    | 105 +++++++++++++++++++++++++++++++++-----------
 src/parser.mly    |   4 +-
 src/printbig.c    |  75 +++++++++++++++++++++++++++++++++++++++
 src/test-mat.mpl  |  13 +++++-
 4 files changed, 150 insertions(+), 47 deletions(-)

1cd1097: Jiangfeng Wang <jw3107@columbia.edu>
Date: Wed, 12 Apr 2017 17:36:55 -0400
Subject: Revised semantic checker binop
Content:
 src/ast.ml                 |   1 +
 src/newSemanticChecker.ml | 167 +++++++++++++++++++++++++-----------
 src/parser.mly             |  14 +++-
 3 files changed, 102 insertions(+), 80 deletions(-)

e8d0923: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 11 Apr 2017 18:59:27 -0400
Subject: matrix access fixed
Content:
 src/codegen.ml    | 4 ++-
 src/test-mat.mpl | 5 +++++
 2 files changed, 7 insertions(+), 2 deletions(-)

da0637e: Jiangfeng Wang <jw3107@columbia.edu>
Date: Tue, 11 Apr 2017 18:20:00 -0400
Subject: Modified semantchecker & sast file
Content:
 src/newSemanticChecker.ml | 79 ++++++++++++++++++++++++++++++++++++++++++---
 src/sast.ml                | 46 +++++++++++++++++++++++++++
 2 files changed, 118 insertions(+), 7 deletions(-)

16bdb6a: wodeni <wn2155@columbia.edu>
Date: Tue, 11 Apr 2017 16:34:32 -0400
Subject: [FIXME] single grp not enough
Content:
 src/codegen.ml | 83 +++++++++++++++++++++++++++++++++++++++++---------
 1 file changed, 57 insertions(+), 26 deletions(-)

d407c38: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 11 Apr 2017 15:09:28 -0400
Subject: test:1)if pass 2)varpass 3)while fail
Content:
 test/testVer1/test-if1.mpl    |  6 ++++++
 test/testVer1/test-if2.mpl    |  6 ++++++
 test/testVer1/test-if3.mpl    |  6 ++++++
 test/testVer1/test-if4.mpl    |  6 ++++++
 test/testVer1/test-if5.mpl    | 17 +++++++++++++++++
 test/testVer1/test-local1.mpl | 14 ++++++++++++++
 test/testVer1/test-local2.mpl | 15 +++++++++++++++
 test/testVer1/test-ops1.mpl   | 28 ++++++++++++++++++++++++++++
 test/testVer1/test-ops2.mpl   | 17 +++++++++++++++++
 test/testVer1/test-print.mpl  | 11 +++++++++++
 test/testVer1/test-var1.mpl   | 21 +++++++++++++++++++++
 test/testVer1/test-var2.mpl   | 13 +++++++++++++
 test/testVer1/test-while1.mpl | 11 +++++++++++
 test/testVer1/test-while2.mpl | 17 +++++++++++++++++
 14 files changed, 188 insertions(+)

99abae2: chyzhang <chyzhang@brandeis.edu>
Date: Mon, 10 Apr 2017 23:11:10 -0400
Subject: fix codegen.ml
Content:
 src/codegen.ml | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

018b381: wodeni <wn2155@columbia.edu>
Date: Mon, 10 Apr 2017 22:58:47 -0400
Subject: [Bug fix] float declaration error
```

```
Content:
 src/codegen.ml   | 12 +++++++---
 src/printm.c     | 16 ++++++++----
 src/scanner.mll |  4 ++-
 3 files changed, 17 insertions(+), 15 deletions(-)

8e580a5: DRC9702 <drinconcruz@knox.edu>
Date: Mon, 10 Apr 2017 22:35:30 -0400
Subject: Refactor from scratch. Please don't kill me Nimo
Content:
 src/newSemanticChecker.ml | 56 +++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 56 insertions(+)

aa187e7: chyzhang <chyzhang@brandeis.edu>
Date: Mon, 10 Apr 2017 20:49:02 -0400
Subject: testall modified
Content:
 testall.sh | 18 +++++++++-----
 1 file changed, 9 insertions(+), 9 deletions(-)

f676265: wodeni <wn2155@columbia.edu>
Date: Mon, 10 Apr 2017 12:48:49 -0400
Subject: [README] added the plan for the next 3 weeks
Content:
 README.md | 17 +++++++++++++++++
 1 file changed, 17 insertions(+)

c2035ea: wodeni <wn2155@columbia.edu>
Date: Wed, 5 Apr 2017 14:23:41 -0400
Subject: [Codegen] printm now works
Content:
 src/Makefile         |  2 +-
 src/codegen.ml       | 10 +++++---
 src/printm.c         |  2 +-
 src/test-mat.mpl     |  3 +++
 src/test-printm.mpl |  5 +++++
 5 files changed, 15 insertions(+), 7 deletions(-)

f0a4c41: wodeni <wn2155@columbia.edu>
Date: Wed, 5 Apr 2017 00:39:43 -0400
Subject: [Codegen] matrix print needs to be fixed
Content:
 src/Makefile   |  2 +-
 src/codegen.ml | 17 +++++++++++++--
 src/mpl.ml     |  2 +-
 src/printm.c   | 50 ++++++++++++++++++++++++++++++++++++++++++++++++++
 4 files changed, 65 insertions(+), 6 deletions(-)

b388f2f: chyzhang <chyzhang@brandeis.edu>
Date: Tue, 4 Apr 2017 21:50:57 -0400
Subject: fixed bugs about matrix and printm builder
Content:
 src/codegen.ml | 41 ++++++++++++++++++++++----------
 1 file changed, 22 insertions(+), 19 deletions(-)

a75c331: wodeni <wn2155@columbia.edu>
Date: Tue, 4 Apr 2017 00:50:00 -0400
Subject: [Codegen] Trying to add printm
Content: - We had rows and cols before we add the value associated with an id
  into the map, BUT when we encounter a function call to "printm", we do
not have row and col data with the id.
- Trying to add in another map to store it.

 src/Makefile   |  8 ++++++
 src/codegen.ml | 85 +++++++++++++++++++++++++++++++++++++++++++++---------
 2 files changed, 69 insertions(+), 24 deletions(-)

c614615: Jiangfeng Wang <jw3107@columbia.edu>
```

```
Date: Mon, 3 Apr 2017 23:04:25 -0400
Subject: Revising semantic checker
Content:
 src/exceptions.ml |  6 ---
 src/semant.ml     | 54 +++++++++++++++++++++++++++++++++++++++++++++-----
 2 files changed, 44 insertions(+), 16 deletions(-)

6c5d0e7: wodeni <wn2155@columbia.edu>
Date: Mon, 3 Apr 2017 10:42:57 -0400
Subject: [Sement] Added skeleton code for sement checker
Content:
 src/semant.ml | 172 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 172 insertions(+)

953ee7c: Nimo Wode Ni <wodeni@users.noreply.github.com>
Date: Mon, 3 Apr 2017 10:25:55 -0400
Subject: Hello world version
Content: - We have the system up and running
- Still missing if and loops
- Need to add matrix as soon as possible


 .travis.yml                                       |   14 +
 hello_world_demo.sh                               |   27 +
 src/Makefile                                      |    4 +-
 src/ast.ml                                        |   33 +-
 src/codegen.ml                                    |  220 +++
 src/exceptions.ml                                 |   74 +
 src/mpl.ml                                        |    9 +-
 src/parser.mly                                    |   97 +-
 src/scanner.mll                                   |   19 +-
 src/script.sh                                     |    8 +
 src/tokenize.ml                                   |   54 +
 test/scanner/scanner.ml                           | 2288 ++++++++++++++++++++++++
 test/scanner/scripts/ScannerTest.ml               |   86 +
 test/scanner/scripts/build.sh                     |   14 +
 test/scanner/scripts/clean.sh                     |    3 +
 test/scanner/scripts/fail/_illegal_carrot.out     |    1 +
 test/scanner/scripts/fail/_illegal_carrot.test    |    1 +
 test/scanner/scripts/fail/_illegal_dollar.out     |    1 +
 test/scanner/scripts/fail/_illegal_dollar.test    |    1 +
 test/scanner/scripts/fail/_illegal_percent.out    |    1 +
 test/scanner/scripts/fail/_illegal_percent.test   |    1 +
 test/scanner/scripts/fail/_illegal_period.out     |    1 +
 test/scanner/scripts/fail/_illegal_period.test    |    1 +
 test/scanner/scripts/fail/_illegal_pound.out      |    1 +
 test/scanner/scripts/fail/_illegal_pound.test     |    1 +
 test/scanner/scripts/fail/_illegal_tilde.out      |    1 +
 test/scanner/scripts/fail/_illegal_tilde.test     |    1 +
 test/scanner/scripts/pass/_arithmetic.out         |    5 +
 test/scanner/scripts/pass/_arithmetic.test        |    1 +
 test/scanner/scripts/pass/_assignment.out         |    4 +
 test/scanner/scripts/pass/_assignment.test        |    1 +
 test/scanner/scripts/pass/_base_scanner.out       |   43 +
 test/scanner/scripts/pass/_base_scanner.test      |    7 +
 test/scanner/scripts/pass/_comment.out            |    0
 test/scanner/scripts/pass/_comment.test           |    6 +
 test/scanner/scripts/pass/_conditionals.out       |    9 +
 test/scanner/scripts/pass/_conditionals.test      |    1 +
 test/scanner/scripts/pass/_control_flow.out       |    5 +
 test/scanner/scripts/pass/_control_flow.test      |    1 +
 test/scanner/scripts/pass/_delimiters.out         |    6 +
 test/scanner/scripts/pass/_delimiters.test        |    1 +
 test/scanner/scripts/pass/_function.out           |   14 +
 test/scanner/scripts/pass/_function.test          |    1 +
 test/scanner/scripts/pass/_identifier.out         |    5 +
 test/scanner/scripts/pass/_identifier.test        |    1 +
 test/scanner/scripts/pass/_literal.out            |    7 +
 test/scanner/scripts/pass/_literal.test          |    7 +
 test/scanner/scripts/pass/_main_function.out     |    9 +
 test/scanner/scripts/pass/_main_function.test    |    1 +
 test/scanner/scripts/pass/_matrix.out            |    6 +
 test/scanner/scripts/pass/_matrix.test           |    1 +
 test/scanner/scripts/pass/_misc.out              |    2 +
 test/scanner/scripts/pass/_misc.test             |    1 +
 test/scanner/scripts/pass/_mixed_arithmetic.out  |    9 +
 test/scanner/scripts/pass/_mixed_arithmetic.test |    1 +
 test/scanner/scripts/pass/_types.out             |   21 +
 test/scanner/scripts/pass/_types.test            |    1 +
 test/scanner/scripts/test.sh                     |  281 +++
 test/test-prints.mpl                             |    3 +
 travis-ci.sh                                     |   23 +
 60 files changed, 3367 insertions(+), 79 deletions(-)

df9e2b5: Jiangfeng Wang <jw3107@columbia.edu>
Date: Mon, 20 Mar 2017 15:35:21 -0400
Subject: fixed shift/reduct conflicts
Content:
 src/ast.ml      |  5 +++-
 src/parser.mly | 29 +++++++++++----------
 2 files changed, 12 insertions(+), 22 deletions(-)

ce23c15: Jiangfeng Wang <jw3107@columbia.edu>
Date: Wed, 15 Mar 2017 15:58:30 -0400
Subject: Replace FloatLit with num type
Content:
 src/ast.ml     | 1 -
 src/parser.mly | 5 ++--
 2 files changed, 2 insertions(+), 4 deletions(-)

5d0f119: Jiangfeng Wang <jw3107@columbia.edu>
Date: Wed, 15 Mar 2017 15:50:10 -0400
Subject: Added brackets to scanner; added matrix initialization and access to parser and ast
    .
Content:
 src/ast.ml      | 18 ++++++++++++++--
 src/parser.mly  | 34 ++++++++++++++++++++--------
 src/scanner.mll |  2 ++
 3 files changed, 35 insertions(+), 19 deletions(-)

ddb2962: wodeni <wn2155@columbia.edu>
Date: Wed, 15 Mar 2017 13:12:35 -0400
Subject: Added floating point numberin scanner and parser.
Content: ISSUE: NEG operator is deleted bacause microc has a solution to this
problem. We need to clean up the ocde to revert to the original
solution, especially the unnecessary BInt type.

 src/ast.ml      |  3 +-
 src/mpl.ml      |  2 ++
 src/parser.mly  | 11 +++++---
 src/scanner.mll |  9 +++++++-
 4 files changed, 15 insertions(+), 10 deletions(-)

e17386a: ZhangChi <ChyZin@ZhangChideMacBook-Pro.local>
Date: Wed, 15 Mar 2017 11:30:28 -0400
Subject: skeleton for test, modified first 3 test
Content:
 test/fail-assign1.err |   1 +
 test/fail-assign1.mc  |   1 +
 test/fail-assign2.err |   1 +
 test/fail-assign2.mc  |   1 +
 test/fail-assign3.err |   1 +
 test/fail-assign3.mc  |  11 +++
 test/fail-dead1.err   |   1 +
 test/fail-dead1.mc    |   8 +++
 test/fail-dead2.err   |   1 +
test/fail-dead2.mc    |  10 +++
test/fail-expr1.err   |   1 +
test/fail-expr1.mc    |  18 +++++
test/fail-expr2.err   |   1 +
test/fail-expr2.mc    |  14 ++++
test/fail-for1.err    |   1 +
test/fail-for1.mc     |  13 ++++
test/fail-for2.err    |   1 +
test/fail-for2.mc     |   8 +++
test/fail-for3.err    |   1 +
test/fail-for3.mc     |   8 +++
test/fail-for4.err    |   1 +
test/fail-for4.mc     |   8 +++
test/fail-for5.err    |   1 +
test/fail-for5.mc     |  10 +++
test/fail-func1.err   |   1 +
test/fail-func1.mc    |  12 ++++
test/fail-func2.err   |   1 +
test/fail-func2.mc    |   8 +++
test/fail-func3.err   |   1 +
test/fail-func3.mc    |   8 +++
test/fail-func4.err   |   1 +
test/fail-func4.mc    |  12 ++++
test/fail-func5.err   |   1 +
test/fail-func5.mc    |  14 ++++
test/fail-func6.err   |   1 +
test/fail-func6.mc    |   9 +++
test/fail-func7.err   |   1 +
test/fail-func7.mc    |   9 +++
test/fail-func8.err   |   1 +
test/fail-func8.mc    |  13 ++++
test/fail-func9.err   |   1 +
test/fail-func9.mc    |   9 +++
test/fail-global1.err |   1 +
test/fail-global1.mc  |   9 +++
test/fail-global2.err |   1 +
test/fail-global2.mc  |   9 +++
test/fail-if1.err     |   1 +
test/fail-if1.mc      |   6 ++
test/fail-if2.err     |   1 +
test/fail-if2.mc      |   6 ++
test/fail-if3.err     |   1 +
test/fail-if3.mc      |   8 +++
test/fail-nomain.err  |   1 +
test/fail-nomain.mc   |   0
test/fail-return1.err |   1 +
test/fail-return1.mc  |   4 ++
test/fail-return2.err |   1 +
test/fail-return2.mc  |  10 +++
test/fail-while1.err  |   1 +
test/fail-while1.mc   |  13 ++++
test/fail-while2.err  |   1 +
test/fail-while2.mc   |  13 ++++
test/test-add1.mc     |  10 +++
test/test-add1.out    |   1 +
test/test-arith1.mc   |   5 ++
test/test-arith1.out  |   1 +
test/test-arith2.mc   |   5 ++
test/test-arith2.out  |   1 +
test/test-arith3.mc   |  13 ++++
test/test-arith3.out  |   1 +
test/test-fib.mc      |  16 +++++
test/test-fib.out     |   6 ++
test/test-for1.mc     |   9 +++
test/test-for1.out    |   6 ++
test/test-for2.mc     |  11 +++
test/test-for2.out    |   6 ++
test/test-func1.mc    |  12 ++++
 test/test-func1.out   |   1 +
 test/test-func2.mc    |  18 +++++
 test/test-func2.out   |   1 +
 test/test-func3.mc    |  13 ++++
 test/test-func3.out   |   4 ++
 test/test-func4.mc    |  14 ++++
 test/test-func4.out   |   1 +
 test/test-func5.mc    |   9 +++
 test/test-func5.out   |   0
 test/test-func6.mc    |   9 +++
 test/test-func6.out   |   1 +
 test/test-func7.mc    |  13 ++++
 test/test-func7.out   |   1 +
 test/test-func8.mc    |  10 +++
 test/test-func8.out   |   1 +
 test/test-gcd.mc      |  15 ++++
 test/test-gcd.out     |   3 +
 test/test-gcd2.mc     |  14 ++++
 test/test-gcd2.out    |   3 +
 test/test-global1.mc  |  30 ++++++++
 test/test-global1.out |   4 ++
 test/test-global2.mc  |  10 +++
 test/test-global2.out |   1 +
 test/test-global3.mc  |  11 +++
 test/test-global3.out |   1 +
 test/test-hello.mc    |   7 ++
 test/test-hello.out   |   3 +
 test/test-if1.mc      |   6 ++
 test/test-if1.out     |   2 +
 test/test-if2.mc      |   6 ++
 test/test-if2.out     |   2 +
 test/test-if3.mc      |   6 ++
 test/test-if3.out     |   1 +
 test/test-if4.mc      |   6 ++
 test/test-if4.out     |   2 +
 test/test-if5.mc      |  16 +++++
 test/test-if5.out     |   2 +
 test/test-local1.mc   |  13 ++++
 test/test-local1.out  |   1 +
 test/test-local2.mc   |  14 ++++
 test/test-local2.out  |   1 +
 test/test-ops1.mc     |  28 ++++++++
 test/test-ops1.out    |  24 +++++++
 test/test-ops2.mc     |  17 +++++
 test/test-ops2.out    |  14 ++++
 test/test-var1.mc     |   7 ++
 test/test-var1.out    |   1 +
 test/test-var2.mc     |  13 ++++
 test/test-var2.out    |   1 +
 test/test-while1.mc   |  11 +++
 test/test-while1.out  |   6 ++
 test/test-while2.mc   |  16 +++++
 test/test-while2.out  |   1 +
 testall.log           |   0
 testall.sh            | 184 ++++++++++++++++++++++++++++++++++++++++++++++++++
 132 files changed, 1015 insertions(+)

9ec4379: Jiangfeng Wang <jw3107@columbia.edu>
Date: Wed, 15 Mar 2017 11:06:42 -0400
Subject: Edited parser to enable matrix initialization
Content:
 src/parser.mly | 26 +++++++++++++++++++++++++-
 1 file changed, 25 insertions(+), 1 deletion(-)

8c0525c: wodeni <wn2155@columbia.edu>
Date: Tue, 14 Mar 2017 16:16:40 -0400
Subject: Compiled Scanner and parser with new tokens added
Content: - Scanner now compiles but could be buggy
```

```
- Parser now has the new tokens but not the correct grammar

 scanner.mll      |  45 -------
 src/Makefile     |  92 +++++++++++++++++++++++++++
 src/ast.ml       | 113 +++++++++++++++++++++++++++++++++
 src/mpl.ml       |  32 ++++++++++
 src/parser.mly   | 129 +++++++++++++++++++++++++++++++++++++
 src/scanner.mll  | 193 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 6 files changed, 559 insertions(+), 45 deletions(-)

3600804: wodeni <wn2155@columbia.edu>
Date: Thu, 26 Jan 2017 15:02:57 -0500
Subject: Meeting notes until Spring break and related documents.
Content: Also skeleton code for scanner.

 README.md                          | 237 +++++++++++++++++++++++++++++++++++++
 doc/language-reference-manual.pdf | Bin 0 -> 289072 bytes
 doc/proposal-graded.pdf            | Bin 0 -> 1347012 bytes
 scanner.mll                        |  45 ++++++++
 4 files changed, 282 insertions(+)

1f51699: DRC9702 <david.rincon.cruz@gmail.com>
Date: Wed, 25 Jan 2017 21:25:38 -0500
Subject: Initial commit
Content:
 README.md | 2 ++
 1 file changed, 2 insertions(+)
```