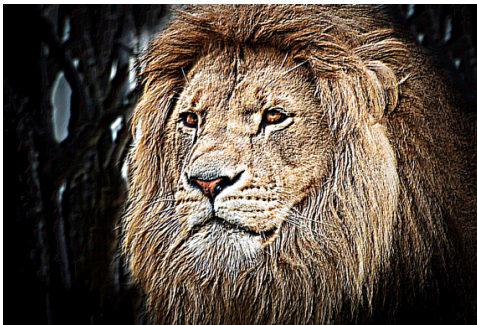


# - PICEL -

## Final Project Report



Manager | Chia-Hao Hsu | ch3141

System Architect | Chih-Sheng Wang | cw2952

System Architect | Ruijie Zhang | rz2337

Language Guru | Rui Lu | rl2784

Testing Jedi | Chang Liu | cl3418

# Contents

<b>1</b>	<b>Introduction.....</b>	<b>4</b>
<b>2</b>	<b>Language Tutorial.....</b>	<b>5</b>
2.1	Environment Setup.....	5
2.2	Using the Compiler.....	5
2.3	The basics PICEL tutorial.....	6
<b>3</b>	<b>Language Reference Manual.....</b>	<b>9</b>
3.1	Lexical Convention.....	9
3.1.1	Comments.....	9
3.1.2	Identifiers.....	9
3.1.3	Keywords.....	9
3.1.4	Types.....	10
3.1.5	Operators & Expression.....	12
3.2	Control Flow.....	15
3.2.1	Block.....	15
3.2.2	Condition.....	15
3.2.3	Loop.....	17
3.3	Program Structure and Scope.....	20
3.3.1	Program Structure.....	20
3.3.2	Function and Variable Scoping.....	21
3.3.3	Function Declaration Syntax.....	21
3.4	PIC and GRAPHIC LIBRARY.....	23
3.4.1	Basic PIC elements.....	23
3.4.2	3rd-party Library.....	23
3.4.3	Picture Manipulation Functions (written in pic).....	25
3.4.4	Other Built-in function.....	27
<b>4</b>	<b>Project Plan.....</b>	<b>28</b>
4.1	Planning & Testing Process.....	28
4.1.1	Planning Process.....	28
4.1.2	Specification Process.....	28
4.1.3	Development Process.....	28
4.1.4	Testing Process.....	29
4.2	Style Guide.....	29
4.2.1	OCaml.....	29
4.2.2	Version Control (Git).....	29
4.2.3	Bash.....	30
4.2.4	LLVM.....	30
4.3	Software development environment:.....	30
4.4	Team Member Github Username / Email:.....	30
4.5	Team Member Responsibility:.....	30
4.6	Git Commit History:.....	30
<b>5</b>	<b>Architecture Design.....</b>	<b>32</b>
5.1	Overview.....	32
5.2	Preprocessor.....	33

5.3	Scanner .....	33
5.4	Parser.....	34
5.5	Semantic Checker .....	34
5.6	Code Generator .....	35
6	Testing Plan.....	38
6.1	Test suite.....	38
6.2	Test automation .....	38
6.3	Source to target.....	39
7	Lesson learned.....	42
7.1	Chia-Hao Hsu .....	42
7.2	Chih-Sheng Wang.....	42
7.3	Ruijie Zhang .....	43
7.4	Rui Lu.....	44
7.5	Chang Liu.....	44
8	Acknowledgements.....	46
9	Appendix .....	47
9.1	Code.....	47
9.2	Github History.....	189

# 1 Introduction

Nowaday, there are many photo editing software around the market, which provides friendly user interface to manipulate and edit the pictures easily. However, there are very few handy picture editing library or programming language. When software engineers want to develop the a photo editing related software, they often need to include several third party library into their project. We want to make a language that make photo editing software development more easier. Our Language --- PICEL provides user some easy way to store the picture. Also, our language has the built-in matrix operation, which make developer easy to build up the image processing function for the desired features. Moreover, PICEL also have the built-in convolution library, which produce some fantastic photo editing effect, for example, sharpen, blur, edge detection, etc. For this language manual we will introduce the grammar and syntax for this language.

## 2 Language Tutorial

### 2.1 Environment Setup

The compiler has been run on an Ubuntu 15.10 virtual machine. A link for downloading VirtualBox and the tutorial for setting up an Ubuntu virtual machine can be found [here](#)<sup>1</sup>. The downloading link for the ISO of Ubuntu 15.10 can be found [here](#)<sup>2</sup>. After the installation, there are some packages for LLVM to be installed before compiling the compiler. Simply run the commands below.

```
$sudo apt-get install m4 clang-3.7 clang 3.7-doc libclang-  
common-3.7-dev libclang-3.7-dev libclang1-3.7 libclang1-3.7-  
dbg libllvm-3.7-ocaml-dev libllvm3.7 libllvm3.7-dbg lldb-3.7  
llvm-3.7 llvm-3.7-dev llvm-3.7-doc llvm-3.7-examples llvm-3.7-  
runtime clang-modernize-3.7 clang-format-3.7 python-clang-3.7  
lldv-3.7-dev liblldb-3.7-dbg opam llvm-runtime
```

Then initialize OCaml's package manager(OPAM) in your home directory:

```
$opam init  
$opam switch 4.02.1  
$eval $(opam config env)  
$opam install core batteries llvm yojson
```

Following that, clone PICEL repository from github (you may also need to install git, whose tutorial can be found [here](#)<sup>3</sup>).

```
$git clone https://github.com/alextrax/PICEL.git
```

### 2.2 Using the Compiler

The following is a sample PICEL program named `example.pic`.

```
int main(){  
    print(42);  
    return 0;  
}
```

Compile this program using the command below.

```
./picel.native < example.pic > example.ll
```

And the output is in the file example.ll which will run using the lli command.

```
$lli example.ll  
42
```

It will return the result “42”.

The sample code shows some features of PICEL:

- The syntax for a main declaration is `int main()`
- Calling the built-in print function
- The function has a return value.

## 2.3 The basics PICEL tutorial

- **Creating your PICEL source file**  
Create a new file called `example.pic`, and open it up.
- **Declaring a primitive variable**  
PICEL supports integer(`int`), boolean(`bool`), character(`char`) types. You can declare a variable in the format: *type identifier*.

```
int a;  
bool if_condition;
```

- **Assigning variables**  
You can assign a legal value to a pre-defined variable.

```
a = 1;  
if_condition = true;  
a = true; /* fail: the value is illegal */  
b = 1; /* fail: undeclared identifier b */
```

- **Declaring & assigning arrays**  
You can define an array using similar method with defining a primitive variable. The length of the array must be pre-defined. The index of an array starts with zero. There are two ways to assign an array. You can do it by assigning value to each element of an array, or use braces.

```

int i[3];      /* declare an array */
i[0] = 1;     /* assign a value to the first element of the array */
i[1] = 2;
i[2] = 3;
i = {1,2,3}; /* assigning can be done in this way as well */

```

- **Declaring & assigning matrices**

Defining a matrix has no difference with defining an array. Please be cautious that the assigning element following the element in the first row and the first column is the element in the first row and the second column.

```

mat m[2][3]; /* declare a matrix */
m[0][0] = 10; /* assign a value to the first element of the matrix */
m = {1,2,3,4,5,6}; /* m[0][0]=1, m[0][1] = 2, m[0][2] = 3, m[1][0] = 4,
m[1][1] = 5 m[1][2] = 6 */

```

One thing to mention is that the kernel we use in convolution is a matrix.

- **Declaring & assigning pictures**

Defining a picture is similar to defining a primitive variable. But usually we assign a picture object with an already-existing image file using *load* function. However, you are still able to assigning the r, g, b value of each pixel in a picture.

```

pic p;
p = load("test.bmp"); /* assign pic p with a bitmap file in current directory */
p.r[0][0] = 255;
p.g[0][0] = 255;
p.b[0][0] = 255;      /* assign the value of the pixel in row 1, column 1 of the
pic p */

```

- **Defining functions**

C-like syntax function defining is easy to understand and use.

```

int func(int arg){
    return arg;
}

```

- **Convolution**

Doing convolution is quite a simple and enjoyable experience in PICEL. There are multiple ways you can complete this.

```

pic p;

```

```
mat kernel_a[5][5];
mat kernel_b[5][5];
mat kernel1[5][5];
mat kernel2[5][5];
convolution(p, kernel a);    /* you can do it in a function way */
p # kernel_b;                /* you can also use a # */
p # kernel1 # kernel2;      /* you can do multiple convolutions in one line */
```

***Now, let's have fun!***



## 3 Language Reference Manual

### 3.1 Lexical Convention

This chapter talks about the lexical conventions in PICEL including comments and tokens. Those tokens are a collection of 4 types: identifiers, keywords, literals and operators. Punctuators, such as white space, is also described in this chapter.

#### 3.1.1 Comments

PICEL supports the comment starting with `/*` and terminating with `*/`. Once the comment starting character `/*` is seen, all the other characters will be ignored until the comment ending character `*/` is seen. Comments do not nest. The characters `/*` and `*/` in a string literal are not considered as comments. The content in `/*` and `*/` can be multiple lines.

```
/* This is a comment.  
And because it's long,  
it spans three lines. */
```

#### 3.1.2 Identifiers

Identifiers are sequences of characters to describe the names of variables and functions. An identifier can include letters, digits and underscores(`_`). The first character of an identifier cannot be a digit. An identifier cannot be the same as a keyword, a boolean literal, an int literal or a null literal. The length of an identifier is limited to 256 characters. Uppercase and lowercase letters are distinct.

```
identifier_name = "[ 'a-z' | 'A-Z' | '_' ] ( [ 'a-z' | 'A-Z' | '_' | '0-9' ] ) *"
```

#### 3.1.3 Keywords

The keywords listed in the chart below are reserved and cannot be used for any other purpose.

if	else	for	while	int	bool
char	void	mat	pic	return	main
true	false	not	and	or	delete

## 3.1.4 Types

### 3.1.4.1 Primitive Data Type

- **Int**

An integer can hold a number in 32 bit. The value of an integer can range from  $-2,147,483,648(-2^{31})$  to  $2,147,483,647(2^{31}-1)$ . For a number used as a string, you should use type char.

- **bool**

Bool type is a binary logical value which can be either true or false. A bool can also be null. A bool takes up 1 bit.

- **char**

The type char is an 8-bit unsigned integer converted to char, i.e., a char object ranges from 0 to 255. Any integer  $i$  other than 0 to 255 will be converted to  $(i \bmod 256)$ . Any other characters will be rejected.

- **string**

A string is represented by two quotes, example: "hello world".

Actually PICEL does not have string type, so there is no variable to contain a string defined by quotes. But our scanner does catch all those static strings. As a result, those static strings can be used as arguments for some built-in functions, like

```
pic a=load("flower.bmp");  
/* Here load is a built-in function which gets a string as the name of the picture,  
load the picture and return a pic structure */
```

### 3.1.4.2 Non-primitive Data Type

- **array**

An array is a data structure that holds one or more literals of the same type. The elements in an array are stored consecutively in the memory. An array can only have one dimension. The identifying number of an array begins at 0, not 1.

```
int NumArray[5];  
NumArray = {1, 2, 3, 4, 5};
```

You can also define an array and assign values to each element of it.

```
int NumArray[5];
NumArray[3] = 4;
```

You can retrieve the element in an array using its identifying number.

```
NumArray[0];          /* 1 */
```

- **mat**

A mat is a two-dimensional array that holds one or more integers. Kernels used for convolution are also mats with 5 rows and 5 columns.

```
mat kernel[5][5];
kernel = { 0,0,0,0,0,
          0,0,0,0,0,
          0,0,1,0,0,
          0,0,0,0,0,
          0,0,0,0,0};
kernel[2][3] = 5;          /* kernel = { 0,0,0,0,0,
          0,0,0,0,0,
          0,0,1,5,0,
          0,0,0,0,0,
          0,0,0,0,0} */

kernel[2][2];          /* 1 */
```

- **pic**

A pic is a data structure we use in PICEL to store an image. The picture is stored in three matrices(arrays) indicating the RGB values of this image. And syntax “.r[x][y]” can be used to refer to the R(ed) value of pixel in position (x,y). In the same way, there are “.g[x][y]” and “.b[x][y]” to refer to the G(reen) and B(lue) value of pixels. Notice that coordinates start from 0. There are some examples:

```
/* Suppose there is a pic A with size 100*100 */
int x=A.r[0][0];      /* get the Red value of point (0,0), the first pixel */
A.g[1][1]=255;       /* set the Green value of point(1,1) to 255 */
int y=A.b[100][0];   /* Invalid since A is 100*100. So the boundary is 99*99 */
```

Picture type supports a series of specific functions such as lengthOf(), widthOf(), etc.

### 3.1.5 Operators & Expression

Operator is the special token that perform a special operation in the program. In this chapter we will introduce our operator in PICEL.

- **Arithmetic Operator**

PICEL provides some basic operators for arithmetic operation: addition, subtraction, multiplication, and division. Also, there are division and negation in PICEL too. Following are some examples to show you the functionality of the operators:

Operator	Description	Example Code
+	addition	<code>x = 4 + 3; /* x = 7 */</code>
-	subtraction	<code>y = 5 - 2; /* y = 3 */</code>
*	multiplication	<code>z = 12 * 10; /* z = 120 */</code>
/	division	<code>w1 = 12 / 3; /* w1 = 4 */</code> <code>w2 = 100 / 3; /* w2 = 33 */</code>
++	self-increase	<code>si++; /* si = si + 1 */</code>
--	self-decrease	<code>sd--; /*sd = si -1 */</code>
-(int var)	negation	<code>a = -5; /* a = -5 */</code> <code>b = -a; /* b = 5 */</code>

- **Array Access Operator:**

In PICEL, array can be accessed with integer index starting from 0 base, the following sample code can show the how to access array in PICEL:

```
int arr[3] = {1, 2, 3};  
int a = arr[1];      /* a = 2 */
```

- **Logical Operators:**

Operator	Description	Example Code
and	conjunction	<code>true and false; /* false */</code>

or	disjunction	true or false; /* true */
not	negation	not false; /* true */

- Comparison Operators:**

Operator	Description	Example Code
>	greater than	5 > 3; /* return true*/ 'a' > 'b'; /* return false */
<	smaller than	4 < 3; /* return false*/ 'a' < 'z'; /* return true */
>=	greater than or equal to	4 >= 4 /*return true */ 5 >= 4 /*return true */
<=	smaller than or equal to	4 <= 4 /* return true */ 4 <= 10 /* return false */
==	equal to	5 == 5 /* return true */ 'a' == 'z' /* return false */
!=	not equal to	'a' != 'z' /* return true */

- Assignment Operators:**

Operator	Description	Example Code
=	assign value	int a; a = 5;

- Matrix Manipulation Operators:**

Operator	Description	Example Code
#	convolution	pic_return = pic # kernel
=	Assign array	int a[100]; a={1,3,5,7,9}; /*now the first 5th elements of a is 1, 3, 5, 7, 9

		<p>The assign operator will assign the int list in the braces to the array from the beginning, namely <math>a[0]</math>.</p> <p>In the length of int list exceeds the size of array, it will lead to an error*/</p>
=	Assign matrix	<pre>mat a[10][10]; a={1,2,3,4,5,6,7,8,9,10,99,999}; /* now the first row of a is 1~10 and a[1][0]=99, a[1][1]=999</pre> <p>The assign operator will assign the int list to the matrix row by row, begin with 0. And in each row, from left to right. It will be more convenient to understand it if we just regard a <math>n*m</math> matrix as an <math>n*m</math> array */</p>

- **Operator Precedence and Associative Property:**

When the program contains multiple operators , then expression will follow the operator precedence and associative property to read our code. For example, if we have a code in our program like “ $a + b * \text{foo}()$ ”, which will follow the rules below, in this part we follow a lot from C Language manual:

1. Function call and membership access operator expression
2. Unary Operator
3. Arithmetic Operator: Multiplication(\*) & Division(/)
4. Matrix Manipulation Operator: #
5. Arithmetic: Addition(+) & Subtraction(-)
6. Logical Operator: Greater-than(>), less-than(<), greater-than-or-equal-to(>=), and less-than-or-equal-to(<=)
7. Logical Operator: Equal-to(==) and not-equal-to(!=) expressions.
8. Logical AND expressions (and)
9. Logical OR expressions (or)
10. Conditional expressions, evaluated from left to right
11. All assignment expressions, evaluated right to left.
12. Comma Expression

## 3.2 Control Flow

In PICEL, there are two types of control flow syntax, namely Condition and Loop, which make program run into different branches according to the corresponding conditions. Control flow enables us to break up the flow of execution by utilizing conditions, looping and branching, making your program more flexible so that your program can conditionally execute particular block of code. With the help of control flow, we can build a complex program with a clear structure.

### 3.2.1 Block

In PICEL, if-block, for/while loop block and function block all begin with “{” and end with “}”. This design is for the unification.

The forms of the block statement are:

```
if ( <condition> ) { <branch1 > } else { <branch2> }  
while ( <condition> ) { <statement> }  
for ( <initialization>; <condition>; <modification> ) { <statement> }
```

### 3.2.2 Condition

Conditional control flow is the basic type. In the conditional control flow, program will either choose among two branches, or choose whether execute a branch or not, according to the given condition. Here is the syntax of conditional control flow.

```
if ( <condition> ) { <branch1> } [ else { <branch2> } ]
```

All conditional control flow must begin with “if ( <condition> ) {” and end with “}”. After “if”, there is <condition>, in which there is the condition of this control flow. All statements in <branch1> will be executed if and only if the condition is true.

And there is also an optional syntax “else”. It can be used to set the second branch of this conditional control flow. If the condition of this “if” control flow is false and there is an “else”, then <branch1> will be ignored and statements in <branch2> will be executed. If the condition of this “if” control flow is false and there is no an “else”, then <branch1> will be ignored and the program will keep executing the remaining statements.

- **<condition>**

The <condition> could be any expression which returns a result in boolean type, including simply a boolean variable and complex boolean expressions with lots of boolean operations.

- **else**

The “else” is an optional syntax of conditional control flow. And one “if” conditional control flow could have at most one “else” syntax. Since a “if” conditional control flow begins with “if” and “else” should follow the first branch which ends with “}”, a “else” control flow will match with the “if” which owns the closet “}”. And if this “if” already has a matching “else”, it is an error. The following code is an example of this error.

```
if(b == 1) {
    if(a > 0) {
        if(c > 0) {
            c = 1;
        }
        a = 0;
    }
    else /* match to the “if (a > 0)” */
    {
        a = -1;
    }
    else /* match to the “if (a > 0)” too, but that “if” already has a “else” */
    {
        /* so it leads to a compile error */
        a = -2;
    }
}
```

Again, the matching cannot cross functions.

- **<branch>**

The <branch> is a series of statements. Actually, <branch1> includes all statements between “{”, which follows <condition>, and “}”, which is followed by “else” (or nothing if there is not “else”). <branch2> includes all statements between “else {” and “}”. As a result, a <branch> could just have a single statement or hundreds of statements with complex structure. Notice that, <branch> could contain all kinds of statements, including all control flow statements. Here is an example.

```
If(a > 0) {
    if(c > 10) {
```



```

        a = a + 1;
        c = c - 10;
    }
    else {
        a = 0;
        c = c * c;
    }
    b = a + c;
}

```

And inside < branch >, user could also define local variable. Those variables would only be available inside the corresponding branch.

### 3.2.3 Loop

In PICEL, we have two kinds of syntax for loop, namely “for” and “while”. Basically, they work in the similar way, but “for” provides a more convenient way to initialize a variable, set a terminal condition and change some variables in every iteration, just like how C++ does. The standard form for for-loop is as following.

```

for ( <initialization>; <condition>; <modification> ) {
    < statement >
}

```

A “for” loop begins with a statement “for ( <initialization>; <condition>; <modification> ) {” and ends with “}”. After “for”, there is a pair of parentheses which contains <initialization>, <condition>, and <modification> separated by semicolon. Between the left brace and right brace, there is the <statement> that loop will execute every iteration. Since the three sections in the parentheses is separated by semicolon, each section could only contain one statement. Here is an example code of for-loop.

```

s = 0;
for (int i = 1; i < 11; i = i + 1) {
    s = s + i; /* accumulate from 1 to 10 */
}

```

- **<initialization>**

The <initialization> section can be regarded as an extra component that will be executed before the program runs into the loop. As a result, we can do some initialization for the loop, like set a variable “i” to 1 to do a loop from 1 to 10.

In <initialization>, we can also define a new local variable which is only available in the for-loop.

- **<condition>**

The <condition> should be an expression which returns a boolean variable. Each time before the program enters the loop, it will check the <condition>. If the <condition> is true, it will begin the new iteration, otherwise it will quit the loop immediately. As a result, if the <condition> is false before the first iteration, the program will just pass the loop, but the <initialization> will be executed since it happens at the beginning. An example is as follows.

```
a = 1;
b = 0;
for (b = 1; a < 1; a = a + 1) { /* here b is set to 1 during <initialization> */
    /* but <condition> is false, so program will not enter the loop and set b to 2 */
    b = 2;
}
/* after those code, b = 1 and a = 1. */
```

- **<modification>**

The <modification> is the code that will be executed after each iteration, in another word, the modification will be executed after each time <statement> is finished. Basically it is used to modify the variable related to <condition>. And it's equivalent to set <modification> empty and put the same code in the <modification> at the end of <statement>. So the following two examples are equivalent.

```
for (int i = 1; i < 11; i = i + 1) {
    s = s + i; /* accumulate from 1 to 10 */
}
for (int i = 1; i < 11;) {
    s = s + i; /* accumulate from 1 to 10 */
    s = i + 1;
}
```

- **while ( <condition> ) { <statement> }**

The “while” is a simple version of “for”. As long as <condition> holds true, a “while” loop will repeat and execute <statement> repeatedly. It is easy to change a for-loop into a while-loop.

```
for ( <initialization>; <condition>; <modification> ) {
    <statement>
}
```

The above for-loop can be converted into while-loop as following.

```
{
    <initialization>
    While ( <condition> ) {
        <statement>
        <modification>
    }
}
```

## 3.3 Program Structure and Scope

### 3.3.1 Program Structure

A PICEL program exist entirely within a single source file. By convention, source file (with a “.pic” extension) contains variable declaration, function declarations, and the corresponding definition.

The layout of a source file with a “.pic” extension is as following.

Global variable declarations
Function Declarations and Definitions
Function Declarations and Definitions (if needed) ...
Main Function Declarations and Definition

For example, if you write a function, and you want this function to be accessible to the main function or other functions, then you would put the function declaration and definition above the all functions that want to call it. Here is an example.

```
int copy_pic (int input) {
    if (input < 0) {
        return 0;
    }
    else If (input > 255) {
        return 255;
    } else {
        return input
    }
}

pic convolution(pic a, mat kernel) {
    ... /* omit some irrelevant code */
    for (int i = 0; i < a.w; i++) {
        for (int j = 0; j < a.h; j++) {
            tempr = tempg = tempb = 0;
            for (int x = 0; x < size; x++) {
```

```

        for (int y = 0; y < size; y++) {
            ... /* omit some irrelevant code */
        }
    }
    temp.r[i][j] = bound(temp_r / tot);
    temp.g[i][j] = bound(temp_g / tot);
    temp.b[i][j] = bound(temp_b / tot);
}
... /* omit some irrelevant code */
}
mat kernel[5][5]
int main() {
    pic a;
    pic b;

    a = load("MARBLES.bmp");

    kernel[2][2] = 5

    convolution(a, kernel);
    save_file("target.bmp", a);

    ... /* omit some irrelevant code */
}

```

### 3.3.2 Function and Variable Scoping

Scope refers to what parts of the program can “see” a declared object, such as variable or function. A declared object can be visible only within a particular block, or within a particular function.

These are some rules for function and variable scoping.

1. All variables should be declared before being referenced.
  - a. Global variables must be declared without initial value will have a default value 0.
2. All functions should be declared(or defined) before being called.
3. The entry point of a PICEL program is the function declared as “main”
4. Variables of the inner block have higher precedence than the outer block.

### 3.3.3 Function Declaration Syntax

```
<return type> <function name>(<parameter list>)  
{  
    <function content>  
}
```

## 3.4 PIC and GRAPHIC LIBRARY

### 3.4.1 Basic PIC elements

- **width and height**

The structure of pic is similar to a structure in C++, so a pic's elements can be accessed by "dot". Syntax ".h" and ".w" are used to refer to the height and width of a pic.

```
/* assume there is a pic a */  
int height=a.h; /* gets the height of a */  
int width=a.w; /* gets the width of a */
```

- **RGB values**

And syntax ".r[x][y]" can be used to refer to the R(ed) value of pixel in position (x,y), which is stored as unsigned char. In the same way, there are ".g[x][y]" and ".b[x][y]" to refer to the G(reen) and B(lue) value of pixels. All coordinates start from 0.

Examples:

```
/* Suppose there is a pic A with size 100*100 */  
int x=A.r[0][0]; /* get the Red value at point (0,0), the first pixel and cast it  
to an integer */  
A.g[1][1]=255; /* set the Gree value of point(1,1) to 255 */  
int y=A.b[100][0]; /* Invalid since A is 100*100. So the boundary is 99*99 */
```

- **Assignment of pic**

For the statement "pic A=B" where "B" is some expressions which returns a pic. This assignment will assign all content of B to a new struct for A, including the height, width and the pointer to the bitmap data. If B is also a pic variable, then change on A's RGB values will also change B's RGB values, vice versa. Thus the assignment of pic must be used carefully.

To support normal "copy" assignment, we have special library function "copy\_pic(A,B)" which will copy the content of pic B to A.

### 3.4.2 3rd-party Library

We leveraged an open-source library<sup>1</sup> to implement the functionality of bitmap read/write.

### 3.4.2.1 Native Built-in Functions (written in C)

These functions are written in C and will be compiled and linked with the PICEL output as an executable.

### 3.4.2.2 The structure to record a pic in C

```
struct pic {
    unsigned int width;
    unsigned int height;
    unsigned int bytes_per_pixel;
    unsigned char* data;
};
```

In the C code, the pic contains 3 unsigned integers and a unsigned char pointer. Variables width and height are the width and height in number of pixel of that picture. bytes\_per\_pixel is an element only used in save\_file and accessing RGB values in code-generation.

And the pointer data points to the memory which stores the RGB values.

### 3.4.2.3 Bitmap file operations:

- **Read**

```
struct pic load(char* filename)
```

The load function will load a picture whose name is the input string. It will allocate the memory for the corresponding RGB values and create a new pic structure for this picture. Th the data pointer of this new pic structure points to the allocated address.

After that, the function returns the pic structure.

Currently load function could only support 24-bit bmp file.

<sup>1</sup>Bitmap Image Reader Writer Library

Author: Arash Partow - 2002

URL: <http://partow.net/programming/bitmap/index.html>



- **Write**

```
int save_file(char* filename, struct pic *src_pic)
```

The function `save_file` takes a string as the name of target file and a `pic` structure needed to be saved. The `pic` structure will be stored in 24-bit bmp format.

#### **3.4.2.4 Picture operations:**

- **New picture**

```
struct pic newpic(unsigned int height, unsigned int width)
```

Function `newpic` will return an empty `pic` structure which refers to a new empty picture of given height and width.

- **Free**

```
int delete_pic(struct pic *src_pic)
```

Since the data pointer of each `pic` structure points to a large memory space storing bitmap data, we need a method to free memory of unused picture. In PICEL, we haven't implemented an automatic garbage collection mechanism, instead, we provide a delete function, `delete_pic`, to free the memory of a `pic` structure. And in our compiler, we use a special syntax to call this function and the `delete_pic` is just an invisible function.

```
delete a; /* free the memory of the pointer for the pic a*/
```

The picture `a`'s bitmap memory will be freed, but its width, height will remain. After the free function, `a`'s pointer points to `NULL`. And if `a` is a new `pic` whose pointer is `NULL`, or its memory has already been freed, it just become a useless statement.

The `delete` keyword must be used carefully especially in the case of two `pic` objects sharing the same bitmap.

#### **3.4.3 Picture Manipulation Functions (written in pic)**

### 3.4.3.1 Convolution

```
pic convolution(pic a, mat kernel)
```

The convolution is the key function of the picture processing part of our language. Given a pic a and a matrix kernel (here kernel must be a 5\*5 matrix), the function will do convolution on the pic a. After modifying the RGB values of a, the function returns the pic a.

The convolution method is widely used in picture processing area. Convolution can achieve blur, sharpen, edge detection and other picture processing methods by define a special kernel matrix.

We also support a special syntax “#” for convolution, the expression “a#kernel” is equivalent to the expression “convolution(a, kernel)”.

Though convolution will return a new pic structure, but its pointer points to the same memory as the input matrix, we could use the return pic as argument for the next convolution function which modifies on the same memory region. Thus we have a convenient way to write convolution:

```
a#kernel_1#kernel_2#kernel_3#kernel_4;  
/*
```

This is equivalent to the following statements:

```
a#kernel_1;  
a#kernel_2;  
a#kernel_3;  
a#kernel_4;
```

Or

```
convolution(a, kernel_1);  
convolution(a, kernel_2);  
convolution(a, kernel_3);  
convolution(a, kernel_4);
```

```
*/
```

### 3.4.3.2 Translate to gray scale picture

```
void to_bw(pic a)
```

The to\_bw function will convert the RGB values of the input pic a into corresponding gray color by some calculation. Namely, for each pixel the

function will convert the RGB values into gray level and then create corresponding RGB values of the gray level.

### **3.4.3.3 Picture copy (different bitmap buffer)**

```
void copy_pic(pic src, pic dst)
```

The `copy_pic` function is used to copy the picture from `src` to `dst`. `dst` must have the same size as `src` and has an available pointer (usually be done by use `newpic` on outside). Then, the RGB values will be copied into `dst` from `src`.

## **3.4.4 Other Built-in function**

### **3.4.4.1 Print integer to console and start a new line**

```
print(int)
/*
```

A char type cannot be used as arguments for this function directly. But it can be assigned to an int type so that its value is extended to an int. So we could use the following statements for some pic x.

```
int temp=x.r[i][j];
print(temp);
*/
```

### **3.4.4.2 Print string to console and start a new line**

```
prints("string")
```

## **4 Project Plan**

### **4.1 Planning & Testing Process**

#### **4.1.1 Planning Process**

Our team met once a week as a team with our T.A. Julie Chien to make sure our process on the right track. Basically we on an as-needed basis at least once a week. Besides, our program manager will report to T.A. about our progress for this week, which make the communication between our team and T.A. very smooth. We use Wechat to synchronize everyone's progress with each teammate. Also, we used Google Drive Shared Folder to co-work on proposal, LRM, and final report. For version control, we used Github to manage every feature we are developing right now. For the development plan, we set up several milestone on weekly basis to check our progress. Also, weekly based plan helps us to estimate and leverage time we have and the goal we have in our mind. In this way, we can avoid waste lot of time on developing some minor features and focus on the most important feature for our language. Also, we have group meeting twice a week to maintain our development process. Also, physical meeting helps us to communicate with each other transparently and smoothly. Everyone can know the progress very clearly.

#### **4.1.2 Specification Process**

We spend a lot of time to figure out what kind of language we should develop. At first, we want to build up a language specific for montage processing and compile it to C or Java. However, we find out the application for montage is very limited and cannot show some interesting thing for a programming language. Besides, a montage language doesn't have enough high level to show the feature of high level language. Therefore, we decide to change the direction from translating a high level language to C or Java to translating some low level language but specific for image processing to some very low level language like FPGA or LLVM. Fortunately, some of our teammate are very familiar with LLVM, which make us decide on this direction very quickly. As the result, we follow this goal and didn't change the direction at all from the beginning to the end of the semester.

#### **4.1.3 Development Process**

We basically set up internal deadline for all the development. Generally we set two weeks earlier before the course deadline and based on the internal deadline to set up

the milestone. For example, we set up the hello world deadline on the end of March, so we need to complete the basic scanner, parser, semantic checking and codegen for the hello world demo. Therefore the roadmap and the milestone is very clear and very easy to know what the team should achieve every week. Before the midterm, we spent most of time on the basic syntax design and took the half of the semester to build up the graphic related structure, library and testing. We keep developing the graphic library until the one week before the demo.

#### **4.1.4 Testing Process**

Throughout the development process we build up several tricky test case by ourselves. We basically develop the testing cases based on MicroC test cases. However, since MicroC doesn't support as much as we want, for example, MicroC doesn't support local variable declaration in random place in the program, we developed our testing case for the feature we want to support. For the syntax part, we basically comes up with several tricky cases first to make sure the semantic and codegen able to generate the right code or to reply the right error message to user. For the graphic part, we basically test the code part and use the human eye to check the picture effect.

## **4.2 Style Guide**

The following outlines our style guide for OCaml, , Bash, and llvm.

### **4.2.1 OCaml**

- 4 space indentation
- 80 character limit lines
- comments included if logic is at all confusing
- pattern matching: no pipe for first pattern case but pipe for all rest of cases
- “then” should follow “if” in same line
- “with” should follow “if” in new line
- “in” should follow “let” in new line

### **4.2.2 Version Control (Git)**

- Generate new branch from master when implementing new feature branch
- Commit often to allow for easy rollback of work if necessary
- Every commit should make sure passing all old test.
- Delete old branch before generate new branch

### **4.2.3 Bash**

- separate actions into discrete statements where possible
- one line per statement
- one space between each token

### **4.2.4 LLVM**

- Since we generate the llvm code, basically we let our generated code follow the llvm coding standard.

### **4.3 Software development environment:**

- Languages: OCaml, LLVM
- Programming Editor: Sublime, vim
- Version Control: Git, Github
- Documentation: Google Drive

### **4.4 Team Member Github Username / Email:**

- Chia Hao Hsu: geniousisme
- Chih-Sheng Wang: alextrax
- Ruijie Zhang: rz2337
- Rui Lu: RuiLu
- Chang Liu: jxsdfzlc

### **4.5 Team Member Responsibility:**

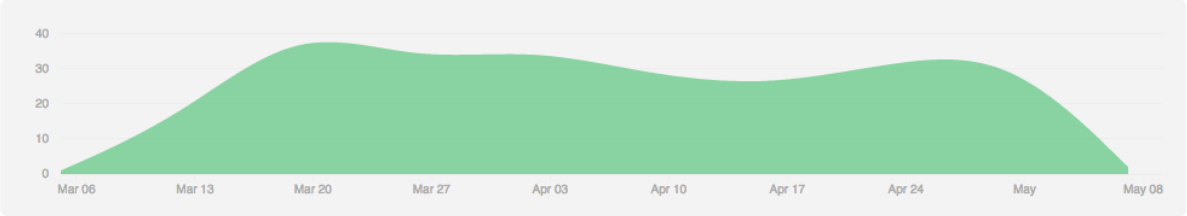
- Chia Hao Hsu: Project Management, Semantic Checking, Development Plan
- Chih-Sheng Wang: Codegen, System Architect, Graphic Library Preprocessor
- Ruijie Zhang: Codegen, AST Development, Graphic Library Implementation
- Rui Lu: Semantic Checking, Language Design, Parser & Scanner Design
- Chang Liu: Automatic Regressive Testing, Final Report Finalization

### **4.6 Git Commit History:**

Mar 6, 2016 – May 10, 2016

Contributions: **Commits** ▾

Contributions to master, excluding merge commits



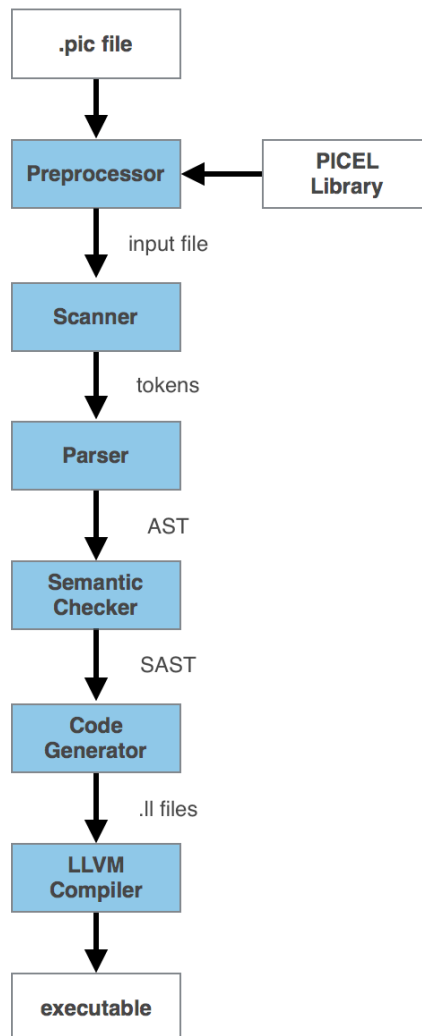
For full commit history please reference from appendix 2.

# 5 Architecture Design

## 5.1 Overview

The PICEL compiler takes a .pic file as source file, and then concatenates it with two PICEL library files, and finally creates a single .pic file as the input file. The compiler then feeds this input file through a series of steps which eventually convert the input file to the llvm code. Finally, the llvm bitcode will be translated into object code and linked with the bitmap library as an executable. All above processes can be performed by using a single script named “buildexe.sh”. After running the “buildexe.sh” script, we will get a resulting executable.

Here is a diagram to illustrate the process that how PICEL compiler works.





To give a quick overview of PICEL compiler, we first show you files we use to achieve the compiling process:

1. **scanner.mll**: Reads the input file and tokenize it to the corresponding tokens.
2. **parser.mly**: Reads in tokens from scanner to produce an AST as representation of the program.
3. **ast.ml**: AST representation of program after parser.
4. **semant.ml**: Semantically checks incoming AST representation of program to determine what the types of various values are, how those types interact in expressions, and whether those interactions are semantically reasonable.
5. **codegen.ml**: Converts a semantically checked AST into working LLVM code.
6. **convolution.pic**: Contains built-in function *copy\_pic* which used to copy pic structures and *convolution* which runs convolution on given pic structure and 5\*5 matrix.
7. **gray.pic**: Contains the function *to\_bw* which convert a picture's color into gray level (but still stored as RGB values).
8. **picel.ml**: Main file (or main module) that calls on other files (or other modules) in the expected order.

## 5.2 Preprocessor

The preprocessor looks for PICEL library files by default and copies two library files, namely *convolution.pic* and *gray.pic*, into one input file. Then the source file is added to the tail of this single input file. We name the resultant file as *source.pic*. Finally, *source.pic* is fed to the compiler.

We achieve this by a script, which is as following:

```
cp ../libpic/convolution.pic source.pic
cat ../libpic/gray.pic >> source.pic
cat ./conv.pic >> source.pic
../picel.native < source.pic > tmp.ll
```

## 5.3 Scanner

The scanner is responsible for scanning through the input file and decomposing code of the input file into a series of prespecified tokens. During scanning, scanner will also discard meaningless token such as whitespace. After scanning through the entire input file, all scanned tokens are passed to the parser. Scanner can reduce the work

of the parser to simply understanding the structure of a program from tokens rather than the original source code.

## 5.4 Parser

The parser scans through the tokens passed by the scanner. The parser will construct an abstract syntax tree based on the parser.mly and the input tokens. If there is no prespecified rule on how to parse some certain tokens, the parser will throw a fatal error and the PICEL compiler stops. Different subtrees of the whole abstract syntax tree represents different component of the program. The parser passes the constructed abstract syntax tree to the semantic checker.

## 5.5 Semantic Checker

The semantic checker takes the abstract syntax tree as input. This abstract syntax tree is passed by the parser. The semantic checker first divides the given abstract syntax tree into two parts, which are globals and functions, which are stored in the format of List. Globals is used to store the declared global variables. And functions is used to store the codes of all functions declared and defined in the input .pic file. There are four subsections in the functions, namely func.typ, func.fname, func.formals, and func.body. Func.typ stores the return types of all functions, func.fname stores the names of all functions, func.formals stores the arguments declared in the prototype of each function, and finally, func.body stores the body of each function. Note that PICEL has built-in function, such as newpic, save, save\_file, load, printb, print, and prints, therefore, the semantic checker will add these built-in functions into functions before the analysis phase.

Semantic checker then begins to semantically analyze on each statement and expression in each block of code in every function. This semantically analysis aims to ensure that functions are defined and don't have duplicate names, the types are consistent in every expression, variables are declared and in the proper scope, and variables are allowed to be declared only once in a certain scope. The semantic checker of PICEL uses five auxiliary data structures to ensure the correctness. First one is a list called local\_hash\_list which stores variables declared before the current block. The second one is a hash table called local\_symbols which stores variables declared in the current block. The third one is a hash table called for\_init\_symbols which stores variables declared in the prototype of for-loop. The fourth one is a hash table called global\_symbols which stores the global variables. The fifth one is a list called pic\_attr which stores four attributes of PICEL-defined object -- pic. Taking

the following code as an example to show how this data structures cooperate with each other.

```
int a;
a = 10;
for (int i = 0; i < 10; i+=1) {
    print(a);
    print(i);
}
print(i);
```

The above code first declares a variable `a`, and assigns 10 to `a`, therefore, the variable `a` is store in `local_symbols`. When dealing with the for-loop, semantic checker finds that `i` is declared in the initialization section of for-loop, semantic checker will add variable `i` into `for_init_symbols`. When semantic checker continue checking `<condition>` and `<modification>` of for, it will first lookup `for_init_symbols`. When semantic checker enters the block of for-loop, all variables in `local_symbols` are added to the head of `local_hash_list`, then `local_symbols` will be cleared. Then all variables in `for_init_symbols` are combined with the current `local_symbols`, then `for_init_symbols` will be cleared. When semantic checker checks `print(a)`, it will first lookup variable `a` in `local_symbols`, then in `global_symbols`, and finally in `local_hash_list`. The same check about variable `i` will be conducted when semantic checker checks `print(i)`. When semantic checker leaves the block, `local_hash_list` returns its head to `local_symbols`, which stores all variables declared before this block. So when semantic checker checks `print(i)` outside the block, the semantic checker will throw an error because it cannot find any declaration information about `i` in all auxiliary data structures.

After the semantic checker checks all codes of the input file, it will return a semantic abstract syntax tree. The purpose of this new data structure is to provide the code generator with data without any syntax errors, so that the code generator can produce LlvM code successfully and flawlessly.

## 5.6 Code Generator

Code Generator will produce the LlvM code of the given syntax tree. According to the struct of abstract syntax tree, in the first level, code generator will evaluate the received declaration lists which contains both variable declarations and function declarations. In each function declaration, code generation will evaluate the statement list. And each statement is either a block of statement list or a syntax of special statement with several expressions needed to be evaluated.

During code generation, there are several features make things challenging.

The first feature is local variable. In PICEL, the definition of local variable is a statement that can appear anywhere a statement could appear. It comes with the challenge to store the mapping of variable name and Llvm address and create a function to look up the address given variable name. Fortunately, we find that Ocaml has a special type Hashtbl which could be assigned and modified instead of create a new binding. Consequently, we could define an empty Hashtbl mapping from string to llvalue and create a lookup function to look up the llvalue of given string. And for each definition of local variables, code generator first check whether the variable name is used and then add the binding into the Hashtbl. The Hashtbl is modified so that the lookup function still works.

And the scope of a local variable brings new troubles. The scope of a local variable begins with the statement of its definition and ends with the end of block it is defined (the statement list of a function is just a huge block) except the scope covered by another local variable in a inner block with the same name.

```
int a;
a=10;
{
    int b=a*10;
    int a=20; /* Here another local variable a is defined
              the first a becomes invisible*/
    int c=a;
    print(c);
}
/* Here the scope of second a ends. So the first a become available again*/
print(a);
```

In the example above, the green part is the scope of the first a and the gray part is the second a. As a result, we could not add the binding directly to the Hashtbl since it will remove the old binding which has larger scope. On the other hand, variables with same name cannot appear in the same block.

To deal with this problem, in Ocaml the function stmt to evaluate statements will receive a list of Hashtbl to store the binding in the outer blocks. And block becomes an intermediate level between function and statements. Another function handle\_block is defined in the same level with stmt (let rec stmt ... and handle\_block .... ) to evaluate blocks.

At the beginning, there is just an empty list and a Hashtbl with the binding of function's parameters. And each Hashtbl will check whether variable's name has already appeared in itself for each binding. When occur a block, `handle_block` will evaluate it by adding the current Hashtbl into the list and evaluating statement list with `stmt` given the modified list and an empty Hashtbl using `fold_left`. Then in each block, the lookup function can turn to Hashtbl list to lookup local variables defined in outer blocks. And each Hashtbl is discarded when its corresponding block ends.

And the final problem happens when we want to access the address of matrix. In Ocaml, there is one-dimension array and two-dimensions matrix while both of them occupy a continuous memory. For a  $n*m$  matrix `a`, to get or store the value of `a[i][j]`, we must access the address of the  $i*m+j$ -th element. As a result, code generator must provide a way to get the type of a given variable name. It becomes complicated when we also consider the local-variable-problems we just talked above. However, we use a small trick to handle this trouble. Since each binding in Llvmlib will assign a local variable a unique llvalue, we add a Hashtbl mapping from llvalue to the types defined in `ast.ml`. Then searching the type becomes two steps, first get the llvalue by using a lookup function we discussed above, then use the llvalue to get the type of the variable. And this method also helps a lot when we deal with type conversion.

## **6 Testing Plan**

### **6.1 Test suite**

We composed our test cases originated from Microc's test cases. These test cases mainly belong to two groups: success tests and fail tests. Success tests are used to ensure that compiled PICEL programs are correctly operational, given the language specifications. Additionally, success tests are used to ensure that the semantic checker does not mistakenly throw errors for semantically correct programs. Fail tests are used to ensure that the semantic checker properly identifies semantic errors in PICEL programs and prohibits semantically incorrect programs from compiling. We named success tests with the prefix "test-" and fail tests with "fail-". All tests and their .out and .err files can be found in the tests/ directory.

Meanwhile we also designed some specific test cases for convolution. These tests are used to ensure some specific functions of convolution work well. The output of these tests are a.out files, which will generate corresponding .bmp files when executed. Those test cases can be found in the conv\_tests/ directory.

### **6.2 Test automation**

We modified some sections of Microc's test script testall.sh to build our test automation.

The main testing script looks through the test directory. It walks through all files and builds a .ll file or return the error message. It then runs the executable (if there is one) and compares the output (or the error message) to a pre-written .out file (or .err file). A difference between expected and actual output means failure. Such differences are recorded in .diff files. Specifying certain flags will clean up intermediate files and will only print the full list of results.

For convolution test cases, we made a buildexe.sh to run these test cases. It will copy conv.pic into a file named source.pic and catenate the source .pic file (e.g., test-conv1.pic) to the source.pic. Then it compiles source.pic and generate tmp.ll file. Run the file and we can have a.out file, which will generate the bitmap file we want when executed.

## 6.3 Source to target

source: source.pic

```
int gcd(int a, int b) {
    while (a != b) {
        if (a > b) a = a - b;
        else b = b - a;
    }
    return a;
}

int main()
{
    print(gcd(2,14));
    print(gcd(3,15));
    print(gcd(99,121));
    return 0;
}
```

target: source.ll

```
; ModuleID = 'PICEL'

@fmti = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmts = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmti1 = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmts2 = private unnamed_addr constant [4 x i8] c"%s\0A\00"

declare i32 @printf(i8*, ...)

declare { i32, i32, i32, i8* } @load(i8*, ...)

declare i32 @save({ i32, i32, i32, i8* }*, ...)

declare i32 @save_file(i8*, { i32, i32, i32, i8* }*, ...)

declare { i32, i32, i32, i8* } @newpic(i32, i32, ...)

declare i32 @delete_pic({ i32, i32, i32, i8* }*, ...)

define i32 @main() {
entry:
    %gcd_result = call i32 @gcd(i32 2, i32 14)
    %printf = call i32 (i8*, ...)* @printf(i8* getelementptr inbounds ([4 x i8]* @fmti, i32 0, i32 0), i32 %gcd_result)
```

```

%gcd_result1 = call i32 @gcd(i32 3, i32 15)
%printf2 = call i32 (i8*, ...)* @printf(i8* getelementptr inbounds ([4 x i8]* @fmt1, i32
0, i32 0), i32 %gcd_result1)
%gcd_result3 = call i32 @gcd(i32 99, i32 121)
%printf4 = call i32 (i8*, ...)* @printf(i8* getelementptr inbounds ([4 x i8]* @fmt1, i32
0, i32 0), i32 %gcd_result3)
ret i32 0
}

```

```

define i32 @gcd(i32 %a, i32 %b) {

```

```

entry:

```

```

%a1 = alloca i32
store i32 %a, i32* %a1
%b2 = alloca i32
store i32 %b, i32* %b2
br label %while

```

```

while:                                     ; preds = %merge, %entry

```

```

%a11 = load i32* %a1
%b12 = load i32* %b2
%tmp13 = icmp ne i32 %a11, %b12
br i1 %tmp13, label %while_body, label %merge14

```

```

while_body:                               ; preds = %while

```

```

%a3 = load i32* %a1
%b4 = load i32* %b2
%tmp = icmp sgt i32 %a3, %b4
br i1 %tmp, label %then, label %else

```

```

merge:                                     ; preds = %else, %then

```

```

br label %while

```

```

then:                                     ; preds = %while_body

```

```

%a5 = load i32* %a1
%b6 = load i32* %b2
%tmp7 = sub i32 %a5, %b6
store i32 %tmp7, i32* %a1
br label %merge

```

```

else:                                     ; preds = %while_body

```

```

%b8 = load i32* %b2
%a9 = load i32* %a1
%tmp10 = sub i32 %b8, %a9
store i32 %tmp10, i32* %b2
br label %merge

```



```
merge14:                                ; preds = %while
    %a15 = load i32*, %a1
    ret i32 %a15
}
```

## **7 Lesson learned**

### **7.1 Chia-Hao Hsu**

For me, there are two challenge to solve in this project: First, how to manage a project that I have no idea to process? Second, how to make sure there are no communication inconsistency between different group?

For the first challenge, it is actually quite hard for me to lead this project since I had no experience in LLVM and compiler. To solve this issue, I try to read several previous projects to figure out system architecture. Also, I asked my friends who took this course in last semester to know what scale we should achieve for this project.

For the second questions, it is actually quite hard to achieve communication inconsistency. Since every group has their own progress and sometime we forget to mention it for other group member, sometimes the semantic will forget to come up with the latest semantic checking with the newest codegen code. I think next time I should try TDD way to make sure every team is on the right track and all the testing case able to pass.

Last but not the least, I actually like ocaml very much. At first it looks very weird and hard to read, but after I am familiar with the syntax, I quite enjoy ocaml after I know all its basic syntax and logic in the language. I found that it is quite similar with the structure we need for building compiler. I am really glad I learned ocaml in this semester and open my horizon to the functional language.

### **7.2 Chih-Sheng Wang**

Design and implement a new language is challenging since the early designed blueprint might not be well considered for later implementation. It's easy to encounter the dilemma between modifying the syntax or changing the plan for implementation. We use LLVM as our backend for code generation which further increases the difficulty of this project. Since LLVM IR is a low-level language, we need to deal with a lot of low-level issues such as type conversion and array index calculation. The picture oriented characteristic of our language implies high memory

usage as well as huge loading of memory debugging. Under such situation, the coding style is very important especially when dealing with memory buffers.

Another important thing I learnt from this project is the value of team working. Indeed, team working always brings extra communication cost and even more bugs due to the misunderstanding between the group members. However, once you find a way to conquer these problems, team working can also be an amazing experience. Working with different people offers you different perspectives and ideas which further enriches our work. I've also noted that it is almost impossible or less efficient to accomplish such a giant project without the help of my teammates.

### **7.3 Ruijie Zhang**

Working a project is always scary to me because it is a long-term work and we need great project plan and time-management. I usually afraid of losing control of the workload and becoming extremely busy near the deadline. But fortunately, we did quite well since we always physical meet twice a week and set internal deadline as motivation. Yet, there are still some features that are abandoned due to the workload, perhaps we should have a better time schedule.

During the project, the first thing I learned is that things may not go as you planned, both good things and bad things. There are some troubles seem challenging at the first sight, but are easily handled after a few minutes. We used to regard local variables as a big deal since we assumed we need to change a bunch of codes and modify the logic of functions. However, it's much easier than we expected. On the other hand, we never thought dealing with array and matrix could be so depressed. When we tried to use array and matrix as arguments of a function, we found that it is impossible to get the size of an array or a matrix in our design unless we change the whole way we store array and matrix. Thus we gave it up and adopted a compromising approach which let the function only accept matrix with 5\*5 size. Actually we could do much better if we had found the design flaw and fixed it at the very beginning. Instead of working recklessly, we should always take the future influence into account.

The second thing I learned a lot is communication. Different from other projects, this project requires us to design something new. Then we need to share our idea with other teammates instead of just following some pre-fixed rules. And it's not an easy work for me to show my thoughts clearly to others. Sometimes what I said is just different from what I thought. Indeed, this project helps me understand the importance of communication in huge projects.

## 7.4 Rui Lu

The PICEL project leaves me three valuable things: the knowledge of starting work early, the experience when cooperating with my brilliant teammates, and the technique to write elegant, functional and powerful programs in OCaml.

Starting working as early as possible gave my teammates and me a plentiful amount of time to fix any coding problems or correct the former mistakes in time. Nobody could exactly predict the problems that will appear during the process of completing a project, especially when we had to use a coding language which is totally different from those languages that we are familiar with. Noted that the LRM is slight different from the final project, because we found something cooler and more powerful after we discussed and experimented again and again, so we kept improving it. This time was earned by the “early start”. Besides, finishing the project early gave our more time to do more tests in order to prevent any potential bugs.

Instead of meeting once a week, we met twice a week, so that we were able to catch the latest progress and share ideas without delay. Even though we have a Wechat group, face-to-face meeting is necessary and more effective. In the beginning, I was unfamiliar with compiler, even some understandings were wrong. But the communication during meetings helped me correct mistakes and consolidate knowledge about compiler structure and OCaml coding skills. Conflicts happened sometimes. There was a time that Chris and I had different solutions for a problem, we discussed and compared each solution. Finally, we found Chris’s solution was way better than mine, so his solution was adopted. We learn from each other; we make progress together. The working experience with my teammate is so wonderful. I never used any functional language like OCaml before, so it was quite painful when I used OCaml to complete the first assignment. However, with increasing knowledge and continual practice, now I find OCaml is so powerful that I can achieve many functionalities within a few lines of code. Learning a functional language like OCaml is interesting.

## 7.5 Chang Liu

Working with great guys is really an amazing experience. An organized meeting plan (two times a week) makes our project plan works fluently. At first, I had no idea what a version control system is and made a lot of mistakes during the project. Fortunately, my teammates are patient enough and helped me out every time. I learned a lot from them, the way they debug, the way they think about and solve a

problem, the way they try to communicate their ideas, etc. Our TA Julie also gave us many useful suggestions on every phase of our project. I appreciate it a lot.

It's painful but interesting to come up with new tricky test cases. Sometimes I even didn't realize there is a problem in the code. One case is that defining and changing the value of variables of the same name in multiple nested blocks. It should prohibit the outer variable's using the value of the inner variable (which is determined by our rule of scoping), but our code actually didn't work in that way and I didn't even realize it. Fortunately, my teammate helped me to figure this out. Another headache is the problem is array and matrix overflow. This also consumed us a plenty of time. I also learned many technical details about what I learned from the course.

One more important lesson I learned from the project is that you should commit and pull timely if you use git as your version control system. And be very careful when doing push. I made mistakes on this and got punished.

## **8 Acknowledgements**

We would like to thank our TA, Julie Chien, for helping us through this semester. She always answered our questions very fast and provided a lot of helpful advices to us. Also, we would like to thank to our professor for offering such a great course and cool project to let us have more understanding about the compiler technology.

## 9 Appendix

### 9.1 Code

scanner.mll

```
(* Ocamllex scanner for PICEL *)

{ open Parser }

rule token = parse
| [' '\t' '\r' '\n'] { token lexbuf }      (* Whitespace *)
| "/"* "          { comment lexbuf }      (* Comments *)
| [0'-9']+ as lxm { LITERAL(int_of_string lxm) }
| "++"           { PPLUS }
| (* | "--"      { MMINUS } *)
| '('           { LPAREN }
| ')'          { RPAREN }
| '{'          { LBRACE }
| '}'          { RBRACE }
| '['          { LBRACKET }
| ']'          { RBRACKET }
| '#'          { CONV }
| ';'          { SEMI }
| ','          { COMMA }
| '+'          { PLUS }
| '-'          { MINUS }
| '*'          { TIMES }
| '/'          { DIVIDE }
| '='          { ASSIGN }
| "=="         { EQ }
| "!="         { NEQ }
| '<'          { LT }
| "<="         { LEQ }
| ">"          { GT }
| ">="         { GEQ }
| "if"         { IF }
| "else"       { ELSE }
```

```

| "for"                { FOR }
| "while"              { WHILE }
| "return"             { RETURN }
| "int"                { INT }
| "bool"               { BOOL }
| "char"               { CHAR }
| "void"               { VOID }
| "pic"                { PIC }
| "mat"                { MATRIX }
| "import"             { IMPORT }
| "break"              { BREAK }
| "continue"           { CONTINUE }
| "sizeof"             { SIZEOF }
| "true"               { TRUE }
| "false"              { FALSE }
| "main"               { MAIN }
| "and"                { AND }
| "or"                 { OR }
| "not"                { NOT }
| "delete"             { DELETE }
| '.'                  { DOT }
| \"_ \" as s { CHARLIT(s.[1]) }
| \" (\"\\\" | [^ \" ])* \" as s { STRINGLIT(String.sub s 1 ((String.length s) -
2))}
| [ '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 }

```



ast.ml

```
(* Abstract Syntax Tree and functions for printing it *)

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

type uop = Neg | Not | Delete

type typ = Int | Bool | Char | Array of typ*int | Pic | Void | Matrix of int*int

type bind = typ * string

type expr = Literal of int
| Id of string
| StringLit of string
| CharLit of char
| BoolLit of bool
| Binop of expr * op * expr
| Unop of uop * expr
| Assign of string * expr
| Call of string * expr list
| Getarr of string * expr
| Assignarr of string * expr * expr
| Getpic of string * string
| GetRGBXY of string * string * expr * expr
| Getmatrix of string * expr * expr
| Assignpic of string * string * expr
| AssignRGBXY of string * string * expr * expr * expr
| Assignmatrix of string * expr * expr * expr
| Convolve of expr * expr
| Noexpr
| Init_array of string * expr list

type initialization = typ * string * expr

type vdecl = Bind of bind

type for_init = F_init of initialization
```

```
| F_expr of expr
```

```
type stmt = Block of stmt list
```

```
| Expr of expr
```

```
| If of expr * stmt * stmt
```

```
| For of for_init * expr * expr * stmt
```

```
| While of expr * stmt
```

```
| Return of expr
```

```
| S_bind of bind
```

```
| S_init of initialization
```

```
type func_decl = {
```

```
  typ: typ;
```

```
  fname: string;
```

```
  formals: bind list;
```

```
  body: stmt list;
```

```
}
```

```
type decl = Vdecl of vdecl
```

```
| Fdecl of func_decl
```

```
type program = decl list
```

```
(* Pretty-printing functions *)
```

```
let rec string_of_typ = function
```

```
  Int -> "int"
```

```
  | Bool -> "bool"
```

```
  | Void -> "void"
```

```
  | Pic -> "pic"
```

```
  | Char -> "char"
```

```
  | Array(typ, i) -> (string_of_typ typ) ^ " array[" ^ (string_of_int i) ^ "]"
```

```
  | Matrix(i1, i2) -> "mat " ^ (string_of_int i1) ^ " " ^ (string_of_int i2)
```

```
let string_of_op = function
```

```
  Add -> "+"
```

```
  | Sub -> "-"
```

```
  | Mult -> "*"
```

```
| Div -> "/"
| Equal -> "=="
| Neq -> "!="
| Less -> "<"
| Leq -> "<="
| Greater -> ">"
| Geq -> ">="
| And -> "&&"
| Or -> "||"
```

```
let string_of_uop = function
```

```
  Neg -> "-"
  | Not -> "!"
  | Delete -> "delete "
```

```
let rec string_of_expr = function
```

```
  Literal(l) -> string_of_int l
  | StringLit(s) -> "\"" ^ s ^ "\""
  | 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 -> ""
  | _ -> "Havn't done yet!!"
```

```
let string_of_for_init = function
```

```
  F_init(t, s, e) -> (string_of_typ t) ^ " " ^ s ^ " " ^ (string_of_expr e) (* for
loop init *)
  | F_expr e -> string_of_expr e
```

```
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
| For(fi, e2, e3, st) ->
  ("for (" ^ (string_of_for_init fi) ^ " ; " ^ (string_of_expr e2) ^ " ; " ^
   (string_of_expr e3) ^ ") " ^ (string_of_stmt st))
| While(e, st) -> ("while (" ^ (string_of_expr e) ^ ") " ^ (string_of_stmt st))
| S_bind(t, s) -> ((string_of_typ t) ^ " " ^ s)
| S_init(t, s, e) -> ((string_of_typ t) ^ " " ^ s ^ " " ^ (string_of_expr e))

```

```

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

```

```

let string_of_vdecl = function
  Bind(bind) -> string_of_bind bind

```

```

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_decl = function
  Vdecl(vdecl) -> string_of_vdecl vdecl ^ "\n"
  |Fdecl(func_decl) -> " " ^ string_of_fdecl func_decl ^ "\n"

```

```

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

```

```

let string_of_hash tbl =
  Hashtbl.fold (fun key value init -> "{" ^ key ^ ": " ^ string_of_typ(value) ^
  "}" ^ init) tbl ""

```

```

let string_of_list string_fun lst =
  "[" ^ (List.fold_left (fun res elem -> res ^ "; " ^ string_fun(elem)) "" lst) ^ "

```

```
]"
```

```
let string_of_program (decls) =  
  String.concat "" (List.map string_of_decl decls)
```

parser.mly

```
(* Abstract Syntax Tree and functions for printing it *)

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

type uop = Neg | Not | Delete

type typ = Int | Bool | Char | Array of typ*int | Pic | Void | Matrix of int*int

type bind = typ * string

type expr = Literal of int
| Id of string
| StringLit of string
| CharLit of char
| BoolLit of bool
| Binop of expr * op * expr
| Unop of uop * expr
| Assign of string * expr
| Call of string * expr list
| Getarr of string * expr
| Assignarr of string * expr * expr
| Getpic of string * string
| GetRGBXY of string * string * expr * expr
| Getmatrix of string * expr * expr
| Assignpic of string * string * expr
| AssignRGBXY of string * string * expr * expr * expr
| Assignmatrix of string * expr * expr * expr
| Convolve of expr * expr
| Noexpr
| Init_array of string * expr list

type initialization = typ * string * expr

type vdecl = Bind of bind

type for_init = F_init of initialization
```

```
| F_expr of expr
```

```
type stmt = Block of stmt list
```

```
| Expr of expr
```

```
| If of expr * stmt * stmt
```

```
| For of for_init * expr * expr * stmt
```

```
| While of expr * stmt
```

```
| Return of expr
```

```
| S_bind of bind
```

```
| S_init of initialization
```

```
type func_decl = {
```

```
  typ: typ;
```

```
  fname: string;
```

```
  formals: bind list;
```

```
  body: stmt list;
```

```
}
```

```
type decl = Vdecl of vdecl
```

```
| Fdecl of func_decl
```

```
type program = decl list
```

```
(* Pretty-printing functions *)
```

```
let rec string_of_typ = function
```

```
  Int -> "int"
```

```
  | Bool -> "bool"
```

```
  | Void -> "void"
```

```
  | Pic -> "pic"
```

```
  | Char -> "char"
```

```
  | Array(typ, i) -> (string_of_typ typ) ^ " array[" ^ (string_of_int i) ^ "]"
```

```
  | Matrix(i1, i2) -> "mat " ^ (string_of_int i1) ^ " " ^ (string_of_int i2)
```

```
let string_of_op = function
```

```
  Add -> "+"
```

```
  | Sub -> "-"
```

```
  | Mult -> "*"
```

```
| Div -> "/"
| Equal -> "=="
| Neq -> "!="
| Less -> "<"
| Leq -> "<="
| Greater -> ">"
| Geq -> ">="
| And -> "&&"
| Or -> "||"
```

```
let string_of_uop = function
```

```
  Neg -> "-"
  | Not -> "!"
  | Delete -> "delete "
```

```
let rec string_of_expr = function
```

```
  Literal(l) -> string_of_int l
  | StringLit(s) -> "\"" ^ s ^ "\""
  | 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 -> ""
  | _ -> "Havn't done yet!!"
```

```
let string_of_for_init = function
```

```
  F_init(t, s, e) -> (string_of_typ t) ^ " " ^ s ^ " " ^ (string_of_expr e) (* for
loop init *)
  | F_expr e -> string_of_expr e
```

```
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
| For(fi, e2, e3, st) ->
  ("for (" ^ (string_of_for_init fi) ^ " ; " ^ (string_of_expr e2) ^ " ; " ^
   (string_of_expr e3) ^ ") " ^ (string_of_stmt st))
| While(e, st) -> ("while (" ^ (string_of_expr e) ^ ") " ^ (string_of_stmt st))
| S_bind(t, s) -> ((string_of_typ t) ^ " " ^ s)
| S_init(t, s, e) -> ((string_of_typ t) ^ " " ^ s ^ " " ^ (string_of_expr e))

```

```

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

```

```

let string_of_vdecl = function
  Bind(bind) -> string_of_bind bind

```

```

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_decl = function
  Vdecl(vdecl) -> string_of_vdecl vdecl ^ "\n"
  |Fdecl(func_decl) -> " " ^ string_of_fdecl func_decl ^ "\n"

```

```

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

```

```

let string_of_hash tbl =
  Hashtbl.fold (fun key value init -> "{" ^ key ^ ": " ^ string_of_typ(value) ^
  "} " ^ init) tbl ""

```

```

let string_of_list string_fun lst =
  "[" ^ (List.fold_left (fun res elem -> res ^ "; " ^ string_fun(elem)) "" lst) ^ "

```

```
]"
```

```
let string_of_program (decls) =  
  String.concat "" (List.map string_of_decl decls)
```

semant.ml

```
(* Semantic checking for the PICEL compiler *)

open Ast

module StringMap = 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 local_symbols = Hashtbl.create 1;;
let for_init_symbols = Hashtbl.create 1;;
let global_symbols = Hashtbl.create 1;;

let pic_attrs = List.fold_left (fun m (t, n) -> StringMap.add n t m)
  StringMap.empty [(Int, "h"); (Int, "w"); (Int, "bpp"); (Void,
"data")]

let check_program =
  (* Split program into globals & functions *)
  let local_hash_list = [] in
  let rec transform p v f =
    match p with
    a::b -> (match a with
      Vdecl(x) -> transform b (x::v) f
      | Fdecl(x) -> transform b v (x::f))
    | [] -> (v,f)
  in
  let (globals, functions) = transform program [] []
  in
  let rec transform_globals g r =
    match g with
    a::b -> (
      match a with
      Bind(x) -> transform_globals b (x::r)
      (*| _ -> transform_globals b r*)
    )
    | [] -> r
  in let globals = transform_globals globals [] in
```

```

(* 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
let string_comp str1 str2 =
  (String.compare str1 str2) == 0
in
let check_assign lvaluet rvaluet err =
  if (string_comp (string_of_typ lvaluet) (string_of_typ rvaluet))
  then lvaluet
  else if ((string_comp (string_of_typ lvaluet) "char") && (string_comp
(string_of_typ rvaluet) "int"))
  then rvaluet
  else if ((string_comp (string_of_typ rvaluet) "char") && (string_comp
(string_of_typ lvaluet) "int"))
  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 = StringMap.add "newpic"
  { typ = Pic; fname = "newpic"; formals = [(Int, "x"); (Int, "y")];
    body = [] } (StringMap.add "save"
  { typ = Int; fname = "save"; formals = [(Pic, "x")];
    body = [] } (StringMap.add "save_file"
  { typ = Int; fname = "save_file"; formals = [(Void, "x"); (Pic, "x")];
    body = [] } (StringMap.add "load"
  { typ = Pic; fname = "load"; formals = [(Void, "x")];
    body = [] } (StringMap.add "printb"
  { typ = Void; fname = "printb"; formals = [(Bool, "x")];
    body = [] } (StringMap.add "print"
  { typ = Void; fname = "print"; formals = [(Int, "x")];
    body = [] } (StringMap.singleton "prints"
  { typ = Void; fname = "prints"; formals = [(Void, "x")];
    body = [] }))))))
in

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

let function_decl s = try StringMap.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 =
  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);

```

```

ignore(List.fold_left (fun tbl (t, n) -> Hashtbl.add tbl n t; tbl)
global_symbols globals);

let rec search_var_in_locals s = function
  hd :: sl -> if (Hashtbl.mem hd s) then Hashtbl.find hd s
             else search_var_in_locals s sl
  | [] -> raise Not_found
in
let type_of_identifier local_hash_list s =
  try Hashtbl.find for_init_symbols s
  with Not_found ->
  try Hashtbl.find local_symbols s
  with Not_found ->
  try search_var_in_locals s local_hash_list
  with Not_found ->
  try Hashtbl.find global_symbols s
  with Not_found -> raise (Failure ("undeclared identifier " ^ s))
in
let pic_attr_checker s =
  try StringMap.find s pic_attrs
  with Not_found -> raise (Failure ("attributes not found in pic:" ^ s))
in
let rgb_attr_checker s =
  let rec exist s = function
    hd :: sl -> if (string_comp hd s) then ()
                else exist s sl
    | [] -> raise Not_found
  in
  try exist s ["r"; "g"; "b"]
  with Not_found -> raise (Failure ("undeclared identifier " ^ s))
in
let check_int_assign e et =
  check_assign Int et
  (Failure ("illegal assignment " ^ string_of_typ Int ^ " = " ^
string_of_typ et ^ " in " ^ string_of_expr e))
in
let check_arr_assign st e et =
  let type_of_arr arr_typ =
    match arr_typ with

```

```

    Array(typ, _) -> typ
    | _ -> raise (Failure ("Not array type!"))
in check_assign (type_of_arr st) et
(Failure ("illegal assignment " ^ string_of_typ st ^ " = " ^
string_of_typ et ^ " in " ^ string_of_expr e))
in
(* Return the type of an expression or throw an exception *)
let rec expr_local_hash_list = function
  Literal _ -> Int
  | BoolLit _ -> Bool
  | StringLit _ -> Void
  | CharLit _ -> Char
  | Id s -> (* print_string ("Id: " ^ s ^ "\n"); *)
             type_of_identifier local_hash_list s
  | Binop(e1, op, e2) as e -> let t1 = (expr_local_hash_list e1) and t2 =
(expr_local_hash_list e2) in
  (match op with
    Add | Sub | Mult | Div when t1 = Int && t2 = Int -> Int
    | Equal | Neq when t1 = t2 -> Bool
    | Less | Leq | Greater | Geq when t1 = Int && t2 = Int -> Bool
    | And | Or when t1 = Bool && t2 = Bool -> Bool
    | _ -> raise (Failure ("illegal binary operator " ^
string_of_typ t1 ^ " " ^ string_of_op op ^ " " ^
string_of_typ t2 ^ " in " ^ string_of_expr e)))
  | Unop(op, e) as ex -> let t = expr_local_hash_list e in
  (match op with
    Neg when t = Int -> Int
    | Not when t = Bool -> Bool
    | Delete when t = Pic -> Pic
    | _ -> raise (Failure ("illegal unary operator " ^ string_of_uop op ^
string_of_typ t ^ " in " ^ string_of_expr ex)))
  | Noexpr -> Void
  | Assign(var, e) as ex -> let lt = type_of_identifier local_hash_list var
and rt = expr_local_hash_list e in
  check_assign lt rt
  (Failure ("illegal assignment " ^ string_of_typ lt ^ " = " ^
string_of_typ rt ^ " in " ^ string_of_expr ex))
  | Convol(e1, e2) -> ignore(expr_local_hash_list e1);
  expr_local_hash_list e2

```

```

| Getarr(s, e) -> ignore(type_of_identifier local_hash_list s);
    expr local_hash_list e
| Assignarr(s, e1, e2) -> ignore(type_of_identifier local_hash_list s);
    ignore(expr local_hash_list e1);
    (* expr local_hash_list e2 *)
    let st = type_of_identifier local_hash_list s
    and e2t = expr local_hash_list e2
    in
    check_arr_assign st e2 e2t
| Init_array(s, e1) -> List.iter (fun e -> ignore(expr local_hash_list e)) e1;
    type_of_identifier local_hash_list s
| Getmatrix(s, e1, e2) -> ignore(type_of_identifier local_hash_list s);
    ignore(expr local_hash_list e1);
    expr local_hash_list e2
| Assignmatrix(s, e1, e2, e3) -> ignore(type_of_identifier
local_hash_list s);
    ignore(expr local_hash_list e1);
    ignore(expr local_hash_list e2);
    check_int_assign e3 (expr local_hash_list e3)
| GetRGBXY(s1, s2, e1, e2) -> ignore(type_of_identifier
local_hash_list s1);
    rgb_attr_checker s2;
    ignore(expr local_hash_list e1);
    expr local_hash_list e2
| AssignRGBXY(s1, s2, e1, e2, e3) -> ignore(type_of_identifier
local_hash_list s1);
    rgb_attr_checker s2;
    ignore(expr local_hash_list e1);
    ignore(expr local_hash_list e2);
    check_int_assign e3 (expr local_hash_list e3)
| Getpic(s1, s2) -> ignore(type_of_identifier local_hash_list s1);
    ignore(pic_attr_checker s2);
    StringMap.find s2 pic_attrs
| Assignpic(s1, s2, e) -> ignore(type_of_identifier local_hash_list s1);
    ignore(pic_attr_checker s2);
    ignore(expr local_hash_list e);
    check_int_assign e (expr local_hash_list e)
| Call(fname, actuals) as call -> 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 et = (expr local_hash_list e) 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 local_hash_list e = if (expr local_hash_list e) !=
Bool
        then raise (Failure ("expected Boolean expression in " ^
string_of_expr e))
        else ()
    in
    (* Temporarily check for init type and always return expr first *)
    let check_not_void_in_symbols s t =
        check_not_void (fun n -> "illegal void local " ^ n ^ " in " ^ func.fname)
(t, s)
    in
    let check_duplicate_in_symbols s t =
        try
            let types = Hashtbl.find_all local_symbols s
            in
            if List.mem t types then raise (Failure ((fun n -> "duplicate local " ^
n) s))
            else ()
        with Not_found -> ()
    in
    let add_var_into_symbols s t =
        check_not_void_in_symbols s t;
        check_duplicate_in_symbols s t;
        Hashtbl.add local_symbols s t
    in
    let check_for_init local_hash_list e =
        match e with
        F_init(t1, s1, e1) -> check_not_void_in_symbols s1 t1;

```

```

                Hashtbl.add for_init_symbols s1 t1;
                expr local_hash_list e1
            | F_expr e1 -> expr local_hash_list e1
        in
        (* Verify a statement or throw an exception *)
        let combine_hashes from_hash to_hash =
            Hashtbl.iter (fun key value -> Hashtbl.add to_hash key value)
from_hash;
        in
        let enter_block_update_local_hash_list local_hash_list local_symbols =
            let tmp_hash = Hashtbl.copy local_symbols
            in
            Hashtbl.clear local_symbols;
            ignore(List.fold_left (fun tbl (t, n) ->
                Hashtbl.add tbl n t; tbl) local_symbols (func.formals));
            if Hashtbl.length for_init_symbols > 0
            then combine_hashes for_init_symbols local_symbols; Hashtbl.clear
for_init_symbols;
            (tmp_hash :: local_hash_list)
        in
        let leave_block_update_local_hash_list local_hash_list =
            combine_hashes (List.hd local_hash_list) local_symbols;
        in
        let rec stmt local_hash_list = function
            Block sl ->
                let local_hash_list = enter_block_update_local_hash_list
local_hash_list local_symbols
                in
                let rec check_block local_hash_list = function
                    [Return _ as s] -> stmt local_hash_list s
                    | Return _ :: _ -> raise (Failure "nothing may follow a return")
                    | Block sl :: ss -> check_block (enter_block_update_local_hash_list
local_hash_list local_symbols) (sl @ ss)
                    | s :: ss -> stmt local_hash_list s;
                        check_block local_hash_list ss
                    | [] -> leave_block_update_local_hash_list local_hash_list; ()
                in check_block local_hash_list sl
            | Expr e -> ignore (expr local_hash_list e)
            | S_bind(t, s) -> ignore (add_var_into_symbols s t)

```

```

| S_init(t, s, e) -> ignore(add_var_into_symbols s t);
                    ignore(expr local_hash_list e) (* why can this work? *)
| Return e ->
    let t = (expr local_hash_list e) in if t = func.typ then () else
    raise (Failure ("return gives " ^ string_of_typ t ^ " expected " ^
                    string_of_typ func.typ ^ " in " ^ string_of_expr e))
| If(p, b1, b2) -> check_bool_expr local_hash_list p;
                  stmt local_hash_list b1;
                  stmt local_hash_list b2
| For(e1, e2, e3, st) -> ignore (check_for_init local_hash_list e1);
                        check_bool_expr local_hash_list e2;
                        ignore (expr local_hash_list e3);
                        stmt local_hash_list st
| While(p, s) -> check_bool_expr local_hash_list p;
                stmt local_hash_list s

in
stmt local_hash_list (Block func.body)

in
List.iter check_function functions

```

codegen.ml

```
(*
Code generation: translate takes a semantically checked AST and
produces LLVM IR
LLVM tutorial: Make sure to read the OCaml version of the tutorial
http://llvm.org/docs/tutorial/index.html
Detailed documentation on the OCaml LLVM library:
http://llvm.moe/
http://llvm.moe/ocaml/
*)

module L = Llvm
module A = Ast

module StringMap = Map.Make(String)
let named_values:(string, L.llvalue) Hashtbl.t = Hashtbl.create 100
let type_map:(L.llvalue, A.typ) Hashtbl.t =Hashtbl.create 100
exception Error of string

let translate program =
  let rec transform p v f=
    match p with
    a::b -> (match a with
              A.Vdecl(x)-> transform b (x::v) f
              | A.Fdecl(x)-> transform b v (x::f))
    | [] -> (v,f)
  in
  let (globals, functions) = transform program [] [] in
  let rec transform_globals g r =
    match g with
    a::b ->(match a with
              A.Bind(x) -> transform_globals b (x::r)
              (*
                | _ -> transform_globals b r*)
              )
    | [] -> r
  in let globals= transform_globals globals [] in
  let context = L.global_context () in
```

```

let the_module = L.create_module context "PICEL"
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 in
let i8_p = L.pointer_type i8_t in
let pic_t = L.struct_type context [| i32_t; i32_t; i32_t; i8_p|] in (* width,
height, bytes per pixel, data[] *)
let pic_p = L.pointer_type pic_t in
(* let mat_t = L.array_type i32_t 25 in
let mat_p = L.pointer_type mat_t in*)

let cast_or_extend a=match a with
    A.Int -> L.build_zext_or_bitcast
    | A.Char -> L.build_intcast
    | _ -> L.build_bitcast
in
let ltype_of_typ = function
    A.Int -> i32_t
    | A.Bool -> i1_t
    | A.Void -> void_t
    | A.Pic -> pic_t
    | A.Matrix(n,m) -> L.array_type i32_t (n*m)
    | A.Char -> i8_t
    | _ -> i32_t in

(* Declare each global variable; remember its value in a map *)
let global_vars =
    let global_var m (t, n) =
        (*let init = L.const_int (ltype_of_typ t) 0
in StringMap.add n (L.define_global n init the_module) m *)
        match t with
        A.Array(typ, len) ->
            let ainit = L.const_array (ltype_of_typ typ) (Array.make len (
L.const_int (ltype_of_typ typ) 0)) in
                let addr=(L.define_global n ainit the_module) in
                    Hashtbl.add type_map addr t;
                    StringMap.add n addr m;
        | A.Pic -> let init_st = L.const_struct context [| (L.const_int i32_t 0);

```

```

(L.const_int i32_t 0); (L.const_int i32_t 0); (L.const_pointer_null i8_p) []
  in let addr = L.define_global n init_st the_module
  in Hashtbl.add type_map addr t; StringMap.add n addr m;
  |A.Matrix(x, y) ->
    let ainit = L.const_array i32_t (Array.make (x*y) ( L.const_int i32_t 0))
in
  let addr=(L.define_global n ainit the_module) in
  Hashtbl.add type_map addr t;
  StringMap.add n addr m;
  |_ -> let init = L.const_int (ltype_of_typ t) 0 in
  let addr = (L.define_global n init the_module)
  in Hashtbl.add type_map addr t; StringMap.add n addr m;

  (*let leni = L.const_int (ltype_of_typ A.Int) len
  in L.build_array_malloc (ltype_of_typ typ) leni n builder
  |*)
in
  List.fold_left global_var StringMap.empty globals in

(* Declare printf(), which the print built-in function will call *)
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

(* Declare external functions *)
let ext_load_t = L.var_arg_function_type pic_t [| L.pointer_type i8_t |] in
let ext_load_func = L.declare_function "load" ext_load_t the_module in
let ext_save_t = L.var_arg_function_type i32_t [| pic_p|] in
let ext_save_func = L.declare_function "save" ext_save_t the_module in
let ext_save_file_t = L.var_arg_function_type i32_t [| L.pointer_type i8_t
; pic_p|] in
let ext_save_file_func = L.declare_function "save_file" ext_save_file_t
the_module in
let ext_newpic_t = L.var_arg_function_type pic_t [| i32_t ; i32_t |] in
let ext_newpic_func = L.declare_function "newpic" ext_newpic_t
the_module in
let ext_del_t = L.var_arg_function_type i32_t [| pic_p|] in
let ext_del_func = L.declare_function "delete_pic" ext_del_t the_module
in
(*let ext_conv_t = L.var_arg_function_type i32_t [| pic_t ; (L.array_type

```

```

i32_t (25)) [] in
let ext_conv_func = L.declare_function "convolution" ext_conv_t
the_module in*)

(* Define each function (arguments and return type) so we can call it *)
let function_decls =
  let function_decl m fdecl =
    let name = fdecl.A.fname
    and formal_types =
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.A.formals)
    in let ftype = L.function_type (ltype_of_typ fdecl.A.typ) formal_types 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 =
  Hashtbl.clear named_values;
  let (the_function, _) = StringMap.find fdecl.A.fname function_decls in
  let builder = L.builder_at_end context (L.entry_block the_function) in

  let int_format_str = L.build_global_stringptr "%d\n" "fmt_i" builder in
  let str_format_str = L.build_global_stringptr "%s\n" "fmt_s" builder 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 start_formal:(string, L.lvalue) Hashtbl.t=Hashtbl.create 50 in
  (* 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) n builder in
    ignore (L.build_store p local builder);
  Hashtbl.add type_map local t;
  Hashtbl.add start_formal n local; m in
  ignore(List.fold_left2 add_formal start_formal fdecl.A.formals
(Array.to_list (L.params the_function)));
  (*
  let add_local m (t, n) =
    let local_var = L.build_alloca (ltype_of_typ t) n builder
    in StringMap.add n local_var m in

```

```

let formals = List.fold_left2 add_formal StringMap.empty
fdecl.A.formals
  (Array.to_list (L.params the_function)) in
List.fold_left add_local formals [] (* fdecl.A.locals *) 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

let get_pic_index elmt =
  match elmt with
  "w" -> 0
  | "h" -> 1
  | "bpp" -> 2
  | "data" -> 3
  | _ -> -1 in
let get_RGB_offset elmt =
  match elmt with
  "b" -> 0
  | "g" -> 1
  | "r" -> 2
  | _ -> -1 in
(* Build the code for the given statement; return the builder for
  the statement's successor *)
let rec stmt named_values hashlist builder =

(* Return the value for a variable or formal argument *)
let lookup n = (*try StringMap.find n local_vars
  with Not_found -> try StringMap.find n global_vars
  with Not_found -> raise (Failure ("undeclared variable " ^ n))**)
let rec lookup2 n h =
  match h with
  a::b -> (try Hashtbl.find a n
    with Not_found -> lookup2 n b)
  | [] -> (try StringMap.find n global_vars
    with Not_found -> raise (Failure ("undeclared variable " ^ n)))
in (try Hashtbl.find named_values n

```



```

with Not_found ->lookup2 n hashlist)
in
(* Construct code for an expression; return its value *)
let rec expr builder = function
  A.Literal i -> L.const_int i32_t i
| A.StringLit s -> L.build_global_stringptr s ("str_" ^ s) builder
| A.BoolLit b -> L.const_int i1_t (if b then 1 else 0)
| A.Noexpr -> L.const_int i32_t 0
| A.Id s -> L.build_load (lookup s) s builder
| A.Getarr (s, e) -> let e' = expr builder e in
  let addr=lookup s in
  let typ=Hashtbl.find type_map addr in (
  match typ with
    A.Array(t,_) ->
  let arraystar_type = L.pointer_type (ltype_of_type t) in

  let cast_pointer = L.build_bitcast addr arraystar_type "c_ptr"
builder in
  let addr = L.build_in_bounds_gep cast_pointer (Array.make 1
e') "elmt_addr" builder in
  L.build_load addr "elmt" builder
  [_ -> raise (Failure ("Array type is wrong!"))])
| A.Getmatrix (s, x, y) -> let x' = expr builder x and y' = expr builder y in
  let addr=lookup s in

  let typ=Hashtbl.find type_map addr in (
  match typ with
  A.Matrix(_,m) -> (* (x * m) + y *)
  let arraystar_type = L.pointer_type i32_t in
  let x_mul_m = L.build_mul x' (L.const_int i32_t m) "x_mul_m"
builder in
  let xm_add_y = L.build_add x_mul_m y' "xm_add_y" builder
in
  let cast_pointer = L.build_bitcast addr arraystar_type "c_ptr"
builder in
  let addr = L.build_in_bounds_gep cast_pointer (Array.make 1
xm_add_y) "elmt_addr" builder in
  L.build_load addr "elmt" builder
  [_ -> raise (Failure ("Mat type is wrong: " ^ s))])

```

```

| A.Assignarr (s, e1, e2) -> let e1' = expr builder e1 and e2' = expr
builder e2 in
    let addr=lookup s in
    let typ=Hashtbl.find type_map addr in (
    match typ with
        A.Array(t,_) ->
        let arraystar_type = L.pointer_type (ltype_of_type t) in
        let cast_pointer = L.build_bitcast addr arraystar_type "c_ptr"
builder in
            let addr = L.build_in_bounds_gep cast_pointer (Array.make 1
e1') "elmt_addr" builder in
                ignore (L.build_store e2' addr builder); e2'
                |_ -> raise (Failure ("Array type is wrong!")))

| A.Assignmatrix (s, x, y, e) -> let x' = expr builder x and y' = expr
builder y and e' = expr builder e in
    let addr=lookup s in
    let typ=Hashtbl.find type_map addr in (
    match typ with
        A.Matrix(_,m) -> (* (x * m) + y *)
        let arraystar_type = L.pointer_type i32_t in
        let x_mul_m = L.build_mul x' (L.const_int i32_t m) "x_mul_m"
builder in
            let xm_add_y = L.build_add x_mul_m y' "xm_add_y" builder
in
                let cast_pointer = L.build_bitcast addr arraystar_type "c_ptr"
builder in
                    let addr = L.build_in_bounds_gep cast_pointer (Array.make 1
xm_add_y) "elmt_addr" builder in
                        ignore (L.build_store e' addr builder); e'
                        |_ -> raise (Failure ("Array type is wrong!")))

| A.Getpic (pic, elmt) -> let addr = L.build_struct_gep (lookup pic)
(get_pic_index elmt) elmt builder in L.build_load addr elmt builder
| A.GetRGBXY (pic, elmt, y, x) -> let x' = expr builder x and y' = expr
builder y in
    let waddr = L.build_struct_gep (lookup pic) 0 "tmp_w" builder

```

```

in let width = L.build_load waddr "tmp_w" builder in
(*
  let haddr = L.build_struct_gep (lookup pic) 1 "tmp_h"
builder in let height = L.build_load haddr "tmp_h" builder in *)
  let bpp_addr = L.build_struct_gep (lookup pic) 2 "tmp_bpp"
builder in let bpp = L.build_load bpp_addr "tmp_bpp" builder in
  let row_increment = L.build_mul width bpp "row_increment"
builder in
  let y_mul_rinre = L.build_mul y' row_increment
"y_mul_rinre" builder in
  let x_mul_bpp = L.build_mul x' bpp "x_mul_bpp" builder in
  let x_add_y = L.build_add y_mul_rinre x_mul_bpp
"x_add_y" builder in
  let data_index = L.build_add x_add_y (L.const_int i32_t
(get_RGB_offset elmt) ) "data_index" builder in
  (*let charstar_type = L.pointer_type i8_t in *)
  let data_addr = L.build_struct_gep (lookup pic) 3 elmt builder
in
  let data_ptr = L.build_load data_addr "data_ptr" builder in
  (*let cast_pointer = L.build_bitcast data_ptr charstar_type
"c_ptr" builder in*)
  let addr = L.build_in_bounds_gep data_ptr (Array.make 1
data_index) "rgb_addr" builder in
  L.build_load addr "rgb_value" builder

| A.Assignpic (pic, elmt, e) -> let e' = expr builder e in
  let addr = L.build_struct_gep (lookup pic) (get_pic_index
elmt) elmt builder in
  ignore (L.build_store e' addr builder); e'
| A.AssignRGBXY (pic, elmt, y, x, e) -> let x' = expr builder x and y' =
expr builder y and e' = expr builder e in
  let waddr = L.build_struct_gep (lookup pic) 0 "tmp_w" builder
in let width = L.build_load waddr "tmp_w" builder in
(*
  let haddr = L.build_struct_gep (lookup pic) 1 "tmp_h"
builder in let height = L.build_load haddr "tmp_h" builder in *)
  let bpp_addr = L.build_struct_gep (lookup pic) 2 "tmp_bpp"
builder in let bpp = L.build_load bpp_addr "tmp_bpp" builder in
  let row_increment = L.build_mul width bpp "row_increment"
builder in
  let y_mul_rinre = L.build_mul y' row_increment

```

```

"y_mul_rincre" builder in
    let x_mul_bpp = L.build_mul x' bpp "x_mul_bpp" builder in
    let x_add_y = L.build_add y_mul_rincre x_mul_bpp
"x_add_y" builder in
    let data_index = L.build_add x_add_y (L.const_int i32_t
(get_RGB_offset elmt) ) "data_index" builder in
    (*let charstar_type = L.pointer_type i8_t in *)
    let data_addr = L.build_struct_gep (lookup pic) 3 elmt builder
in
    let data_ptr = L.build_load data_addr "data_ptr" builder in
    (*let cast_pointer = L.build_bitcast data_ptr charstar_type
"c_ptr" builder in*)
    let char_e = L.build_intcast e' i8_t "char_RGB" builder in
    let addr = L.build_in_bounds_gep data_ptr (Array.make 1
data_index) "rgb_addr" builder in
    ignore (L.build_store char_e addr builder); char_e

```

```

| A.Binop (e1, op, e2) ->
    let e1' = expr builder e1
    and e2' = expr builder e2 in
    (match op with
    | A.Add    -> L.build_add
    | A.Sub    -> L.build_sub
    | A.Mult   -> L.build_mul
    | A.Div    -> L.build_sdiv
    | A.And    -> L.build_and
    | A.Or     -> L.build_or
    | A.Equal  -> L.build_icmp L.lcmp.Eq
    | A.Neq    -> L.build_icmp L.lcmp.Ne
    | A.Less   -> L.build_icmp L.lcmp.Slt
    | A.Leq    -> L.build_icmp L.lcmp.Sle
    | A.Greater -> L.build_icmp L.lcmp.Sgt
    | A.Geq    -> L.build_icmp L.lcmp.Sge
    ) e1' e2' "tmp" builder
| A.Unop(op, e) ->
    let e' = expr builder e in
    (match op with
    | A.Neg    -> L.build_neg e' "tmp" builder
    | A.Not    -> L.build_not e' "tmp" builder

```

```

    | A.Delete -> (match e with A.Id s -> L.build_call ext_del_func [|
(lookup s) |] "delete_pic" builder
    | _ -> raise(Failure ("Delete only accept id!")))
)
| A.Assign (s, e) -> let e' = expr builder e in
    let addr = lookup s in
    let typ = (try Hashtbl.find type_map addr with Not_found ->
raise (Failure ("find type_map failed " ^ s))) in
    (match typ with
    A.Pic -> ignore (L.build_store e' addr builder); e'
    | _ -> let cast_value = (cast_or_extend typ) e' (ltype_of_typ
typ) "casted_value" builder in
        ignore (L.build_store cast_value addr builder); cast_value
    )

| A.Call ("print", [e]) | A.Call ("printb", [e]) ->
    L.build_call printf_func [| int_format_str ; (expr builder e) |]
    "printf" builder
| A.Call ("prints", [e]) ->
L.build_call printf_func [| str_format_str ; (expr builder e) |]
    "printf" builder
| A.Call ("load", [e]) ->
L.build_call ext_load_func [| (expr builder e) |]
    "load" builder
| A.Call ("save", [e]) ->
    (match e with A.Id s ->
L.build_call ext_save_func [| (lookup s) |]
    "save" builder
    | _ -> raise(Failure("save only accept id!")))
| A.Call ("save_file", e) ->
let a = List.hd e in let b = List.hd (List.tl e) in
    (match b with A.Id s ->
L.build_call ext_save_file_func [| (expr builder a) ; (lookup s) |]
    "save_file" builder
    | _ -> raise(Failure("save_file only accept a string and an id!")))
| A.Call ("newpic", e) ->
let a = List.hd e in let b = List.hd (List.tl e) in
L.build_call ext_newpic_func [| (expr builder a) ; (expr builder b) |]
    "newpic" builder

```

```

(*| A.Call ("convolution", e) ->
  let a = List.hd e in let b = List.hd (List.tl e) in
  L.build_call ext_conv_func [] (expr builder a) ; (expr builder b) []
  "convolution" builder*)
| A.Call (f, act) ->
  let (fdef, fdecl) = StringMap.find f function_decls in
  let actuals = List.rev (List.map (expr builder) (List.rev act)) in
  let result = (match fdecl.A.typ with A.Void -> ""
                | _ -> f ^ "_result") in
  L.build_call fdef (Array.of_list actuals) result builder
| A.Convolve (p, m) ->
  let (fdef, fdecl) = StringMap.find "convolution" function_decls in
  let result = (match fdecl.A.typ with A.Void -> ""
                | _ -> "convolution" ^ "_result") in
  L.build_call fdef [] (expr builder p) ; (expr builder m) [] result builder

| A.Init_array(s, a) ->
  let rec loop_assign t num addr a builder=
    match a with
    | x::y -> let temp=(expr builder x) in
               let index=L.const_int i32_t num in
               let arraystar_type=L.pointer_type (ltype_of_type t) in
               let cast_pointer=L.build_bitcast addr arraystar_type "c_ptr"
builder in
               let addr2=L.build_in_bounds_gep cast_pointer (Array.make 1
index) "elmt_addr" builder in
               ignore (L.build_store temp addr2 builder); loop_assign t (num+1)
addr y builder
    | [] -> addr
  in
  let addr=lookup s in
  let typ=Hashtbl.find type_map addr in (
  match typ with
  | A.Array(t,_) ->
    loop_assign t 0 addr a builder
  | A.Matrix(,_) -> loop_assign A.Int 0 addr a builder
  | _ -> raise(Failure ("Wrong type for array/matrix
assignment!"))
  )
)

```

```

| _ -> raise(Failure ("Some expression which should not be evaluated in
expr"));
in

function
  A.Block sl -> handle_block builder (named_values::hashlist) sl
| A.Expr e -> ignore (expr builder e); builder
| A.S_bind (t, n) -> (match t with
  A.Array(atyp, alen) -> let local_arr =
(*L.build_array_alloca (ltype_of_typ atyp) (L.const_int i32_t alen) n
builder *)
  (* L.const_array (ltype_of_typ atyp) (Array.make alen (
L.const_int (ltype_of_typ atyp) 0)) *)
  L.build_alloca (L.array_type (ltype_of_typ atyp) alen) n
builder
  in Hashtbl.add named_values n local_arr ;
Hashtbl.add type_map local_arr t; builder
  | A.Pic -> (*let local_st = L.build_malloc pic_t n builder*)
let local_st = L.build_alloca pic_t n builder
in Hashtbl.add named_values n local_st ; Hashtbl.add
type_map local_st t; builder
  | A.Matrix(x, y) ->
let local_mat = (*L.build_array_alloca i32_t (L.const_int i32_t (x*y)) n
builder in *)
  L.build_alloca (L.array_type i32_t (x*y)) n builder in
  Hashtbl.add named_values n local_mat ; Hashtbl.add type_map
local_mat t; builder
  | _ -> let local_var = L.build_alloca (ltype_of_typ t) n
builder
  in Hashtbl.add named_values n local_var;
Hashtbl.add type_map local_var t ; builder)
| A.S_init (t, n, p) -> let local_var = L.build_alloca (ltype_of_typ t) n
builder
  in let e' = expr builder p in
let cast_value = (cast_or_extend t) e' (ltype_of_typ t)
"casted_value_s_init" builder in
  ignore (L.build_store cast_value local_var builder);
  Hashtbl.add named_values n local_var; Hashtbl.add

```

```

type_map local_var t ; builder
  | A.Return e -> ignore (match fdecl.A.typ with
    A.Void -> L.build_ret_void builder
  | A.Pic -> L.build_ret (expr builder e) builder
    | _ -> let e' = expr builder e in
      let cast_value = (cast_or_extend fdecl.A.typ) e' (ltype_of_type
fdecl.A.typ) "casted_value" builder in
        L.build_ret cast_value builder); builder
  | A.If (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 named_values hashlist (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 named_values hashlist (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

  | A.While (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 named_values hashlist (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) -> let e' = match e1 with
    | A.F_expr(e) -> A.Expr(e)
    | A.F_init(e) -> A.S_init(e) in
    handle_block builder (named_values::hashlist)
      ( [ e' ; A.While (e2, A.Block [body ; A.Expr e3]) ] )
and handle_block builder hashlist s =

  let new_n:(string, L.lvalue) Hashtbl.t = Hashtbl.create 50 in
  List.fold_left (stmt new_n hashlist) builder s
in
(* Build the code for each statement in the function *)
let builder = stmt start_formal [] builder (A.Block fdecl.A.body) in

(* Add a return if the last block falls off the end *)
add_terminal builder (match fdecl.A.typ 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

```

## bmplib.cpp

```
#include <cstring>
#include <cstdio>
#include "bitmap_image.hpp"

extern "C" {

struct pic {
    unsigned int width;
    unsigned int height;
    unsigned int bytes_per_pixel;
    unsigned char* data;
};

int get_width(char* filename)
{
    bitmap_image image(filename);

    if (!image)
    {
        printf("Error - Failed to open: %s\n", filename);
        return 0;
    }

    return image.width();
}

int get_height(char* filename)
{
    bitmap_image image(filename);

    if (!image)
    {
        printf("Error - Failed to open: %s\n", filename);
        return 0;
    }
}
```

```

return image.height();
}

struct pic load(char* filename)
{
    bitmap_image image(filename);
    struct pic new_pic;
    if (!image)
    {
        printf("Error - Failed to open: %s\n", filename);
        return new_pic;
    }

    new_pic.width = image.width();
    new_pic.height = image.height();
    new_pic.bytes_per_pixel = image.bytes_per_pixel();
    new_pic.data = (unsigned char*)malloc(new_pic.width * new_pic.height *
    * new_pic.bytes_per_pixel);
    memcpy(new_pic.data , image.data(), new_pic.width * new_pic.height *
    new_pic.bytes_per_pixel);

    //printf ("file: %s = &%x\n", filename, new_pic.data);

    return new_pic;
}

int save_file(char* filename, struct pic *src_pic)
{
    bitmap_image image(src_pic->width, src_pic->height);
    unsigned int length      = src_pic->width * src_pic->height * src_pic->
    >bytes_per_pixel;
    std::copy(src_pic->data, src_pic->data + length, image.data());
    image.save_image(filename);
    printf("w: %d, h: %d, bpp: %d \n", src_pic->width, src_pic->height,
    src_pic->bytes_per_pixel);
    printf("saved image to %s\n", filename);
    return 0;
}

```

```

}

int save(struct pic *src_pic)
{
    bitmap_image image(src_pic->width, src_pic->height);
    unsigned int length      = src_pic->width * src_pic->height * src_pic->
bytes_per_pixel;
    std::copy(src_pic->data, src_pic->data + length, image.data());
    image.save_image("pic_output.bmp");
    printf("w: %d, h: %d, bpp: %d \n", src_pic->width, src_pic->height,
src_pic->bytes_per_pixel);
    return 0;
}

struct pic newpic(unsigned int height, unsigned int width){
    struct pic new_pic;
    new_pic.width = width;
    new_pic.height = height;
    new_pic.bytes_per_pixel = 3;
    new_pic.data = (unsigned char*)malloc(new_pic.width * new_pic.height
* new_pic.bytes_per_pixel);

    for(int i = 0; i < new_pic.width * new_pic.height *
new_pic.bytes_per_pixel; i++){
        new_pic.data[i] = 0;
    }
    return new_pic;
}

int delete_pic(struct pic *src_pic){
    if (src_pic->data != NULL){
        delete [] src_pic->data;
        src_pic->data = NULL;
        return 0;
    }
    else
        return 1;
}

```

```
}  
}
```

## convolution.pic

```
void copy_pic(pic src, pic dst)
{
    /*pic dst=newpic(src.w,src.h);*/
    int i;
    int j;
    for(i=0;i<src.h;i++)
    {
        for(j=0;j<src.w;j++)
        {
            dst.r[i][j]=src.r[i][j];
            dst.g[i][j]=src.g[i][j];
            dst.b[i][j]=src.b[i][j];
        }
    }
    return;
}

int bound(int input){
    if(input < 0){
        return 0;
    }
    else
    if(input>255)
    {
        return 255;
    }
    else{
        return input;
    }
}

pic convolution(pic a, mat kernel)
{
    /*
    unable to verify whether kernel is a 5*5 matrix
```

```

*/

pic temp=newpic(a.h,a.w);

int size=5;
int s1=size/2;
int r=0;
    int g=0;
    int b=0;
int tempr;
    int tempg;
    int tempb;
int i1;
    int j1;
int i;
int j;
int x;
int y;

int tot=0;
for(i=0;i<size;i++)
{
    for(j=0;j<size;j++)
    {
        tot=tot+kernel[i][j];
    }
}

if(tot==0)
    tot=1;

for(i=0;i<a.h;i++)
{
    for(j=0;j<a.w;j++)
    {
        tempr=tempg=tempb=0;
        for(x=0;x<size;x++)

```

```

{
    for(y=0;y<size;y++)
    {
        i1=i+x-s1;
        j1=j+y-s1;
        if((i1<0) or (i1>=a.h) or (j1<0) or (j1>=a.w))
        {
            r=a.r[i][j];
            g=a.g[i][j];
            b=a.b[i][j];
        }else
        {
            r=a.r[i1][j1];
            g=a.g[i1][j1];
            b=a.b[i1][j1];
        }

        tempr=tempr+r*kernel[x][y];
        tempg=tempg+g*kernel[x][y];
        tempb=tempb+b*kernel[x][y];
    }
}
temp.r[i][j] = bound(tempr / tot);
temp.g[i][j] = bound(tempg / tot);
temp.b[i][j] = bound(tempb / tot);
}
}

```

```

for(i=0;i<a.h;i++)
{
    for(j=0;j<a.w;j++)
    {
        a.r[i][j]=temp.r[i][j];
        a.g[i][j]=temp.g[i][j];
        a.b[i][j]=temp.b[i][j];
    }
}

```



```
}  
}
```

```
delete temp;  
return a;  
}
```

gray.pic

```
void to_bw(pic a)
{
  int i;
  int j;
  int bw;
  int r;
  int g;
  int b;
  for(i=0;i<a.h;i++)
  {
    for(j=0;j<a.w;j++)
    {
      r=a.r[i][j];
      g=a.g[i][j];
      b=a.b[i][j];
      bw=(r*30+g*59+b*11+50)/100;
      if(bw>255)
      {
        bw=255;
      }
      a.r[i][j]=bw;
      a.g[i][j]=bw;
      a.b[i][j]=bw;
    }
  }
}
```

fail-assign1.pic

```
int main()
{
  int i;
  bool b;

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

fail-assign1.err

```
Fatal error: exception Failure("illegal assignment int = bool in i = false")
```

fail-assign2.pic

```
int main()
{
  int i;
  bool b;

  b = 48; /* Fail: assigning an integer to a bool */
}
```

fail-assign2.err

```
Fatal error: exception Failure("illegal assignment bool = int in b = 48")
```

fail-assign3.pic

```
void myvoid()
{
    return;
}

int main()
{
    int i;

    i = myvoid(); /* Fail: assigning a void to an integer */
}
```

fail-assign3.err

```
Fatal error: exception Failure("illegal assignment int = void in i =
myvoid()")
```

fail-assign4.pic

```
int main(){  
    int a;  
    b = 1;  
    return 0;  
}
```

fail-assign4.err

```
Fatal error: exception Failure("undeclared identifier b")
```

fail-dead1.pic

```
int main()
{
  int i;

  i = 15;
  return i;
  i = 32; /* Error: code after a return */
}
```

fail-dead1.err

```
Fatal error: exception Failure("nothing may follow a return")
```

fail-dead2.pic

```
int main()
{
  int i;

  {
    i = 15;
    return i;
  }
  i = 32; /* Error: code after a return */
}
```

fail-dead2.err

```
Fatal error: exception Failure("nothing may follow a return")
```



fail-expr1.pic

```
int a;
bool b;

void foo(int c, bool d)
{
    int dd;
    bool e;
    a + c;
    c - a;
    a * 3;
    c / 2;
    d + a; /* Error: bool + int */
}

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

fail-expr1.err

```
Fatal error: exception Failure("illegal binary operator bool + int in d + a")
```

fail-expr2.pic

```
int a;
bool b;

void foo(int c, bool d)
{
    int d;
    bool e;
    b + a; /* Error: bool + int */
}

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

fail-expr2.err

```
Fatal error: exception Failure("illegal binary operator bool + int in b + a")
```

fail-for1.pic

```
int main()
{
  int i;

  for (i = 0 ; i < 5 ; i = i + 1) {
    print(i);
  }

  for ( ; true ; ) {} /* OK: Forever */

  for (i = 0 ; i < 10 ; i = i + 1) {
    if (i == 3) print(42);
  }

  for (i = 0 ; i < 15 ; i = i + 1) {
    if (i == 3) return 42;
  }

  for (j = 0; i < 10 ; i = i + 1) {} /* j undefined */

  return 0;
}
```

fail-for1.err

```
Fatal error: exception Failure("undeclared identifier j")
```

fail-for2.pic

```
int main()
{
  int i;

  for (i = 0; j < 10 ; i = i + 1) {} /* j undefined */

  return 0;
}
```

fail-for2.err

```
Fatal error: exception Failure("undeclared identifier j")
```

fail-for3.pic

```
int main()
{
  int i;

  for (i = 0; i ; i = i + 1) {} /* i is an integer, not Boolean */

  return 0;
}
```

fail-for3.err

```
Fatal error: exception Failure("expected Boolean expression in i")
```

fail-for4.pic

```
int main()
{
  int i;

  for (i = 0; i < 10 ; i = j + 1) {} /* j undefined */

  return 0;
}
```

fail-for4.err

```
Fatal error: exception Failure("undeclared identifier j")
```

fail-for5.pic

```
int main()
{
  int i;

  for (i = 0; i < 10 ; i = i + 1) {
    foo(); /* Error: no function foo */
  }

  return 0;
}
```

fail-for5.err

```
Fatal error: exception Failure("unrecognized function foo")
```

fail-for6.pic

```
int main() {  
  
int a = 0;  
print(a);  
  
for (int a = 1; a < 10; a++) {  
    int a = 2; /* duplicate local "a" */  
}  
  
}
```

fail-for6.err

```
Fatal error: exception Failure("duplicate local a")
```



fail-func1.pic

```
int foo() {}  
  
int bar() {}  
  
int baz() {}  
  
void bar() {} /* Error: duplicate function bar */  
  
int main()  
{  
    return 0;  
}
```

fail-func1.err

```
Fatal error: exception Failure("duplicate function bar")
```

fail-func2.pic

```
int foo(int a, bool b, int c) { }  
  
void bar(int a, bool b, int a) {} /* Error: duplicate formal a in bar */  
  
int main()  
{  
  return 0;  
}
```

fail-func2.err

```
Fatal error: exception Failure("duplicate formal a in bar")
```

fail-func3.pic

```
int foo(int a, bool b, int c) { }  
  
void bar(int a, void b, int c) {} /* Error: illegal void formal b */  
  
int main()  
{  
  return 0;  
}
```

fail-func3.err

```
Fatal error: exception Failure("illegal void formal b in bar")
```

fail-func4.pic

```
int foo() {}  
void bar() {}  
int print() {} /* Should not be able to define print */  
void baz() {}  
int main()  
{  
  return 0;  
}
```

fail-func4.err

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

fail-func5.pic

```
int foo() {}

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

  return 0;
}

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

fail-func5.err

```
Fatal error: exception Failure("illegal void local b in bar")
```

fail-func6.pic

```
void foo(int a, bool b)
{
}

int main()
{
  foo(42, true);
  foo(42); /* Wrong number of arguments */
}
```

fail-func6.err

```
Fatal error: exception Failure("expecting 2 arguments in foo(42)")
```

fail-func7.pic

```
void foo(int a, bool b)
{
}

int main()
{
    foo(42, true);
    foo(42, true, false); /* Wrong number of arguments */
}
```

fail-func7.err

```
Fatal error: exception Failure("expecting 2 arguments in foo(42, true, false)")
```

fail-func8.pic

```
void foo(int a, bool b)
{
}

void bar()
{
}

int main()
{
  foo(42, true);
  foo(42, bar()); /* int and void, not int and bool */
}
```

fail-func8.err

```
Fatal error: exception Failure("illegal actual argument found void
expected bool in bar()")
```



fail-func9.pic

```
void foo(int a, bool b)
{
}

int main()
{
  foo(42, true);
  foo(42, 42); /* Fail: int, not bool */
}
```

fail-func9.err

```
Fatal error: exception Failure("illegal actual argument found int expected
bool in 42")
```

fail-global1.pic

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

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

fail-global1.err

```
Fatal error: exception Failure("illegal void global a")
```

fail-global2.pic

```
int b;  
bool c;  
int a;  
int b; /* Duplicate global variable */  
  
int main()  
{  
    return 0;  
}
```

fail-global2.err

```
Fatal error: exception Failure("duplicate global b")
```

fail-if1.pic

```
int main()
{
  if (true) {}
  if (false) {} else {}
  if (42) {} /* Error: non-bool predicate */
}
```

fail-if1.err

```
Fatal error: exception Failure("expected Boolean expression in 42")
```

fail-if2.pic

```
int main()
{
  if (true) {
    foo; /* Error: undeclared variable */
  }
}
```

fail-if2.err

```
Fatal error: exception Failure("undeclared identifier foo")
```

fail-if3.pic

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

fail-if3.err

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

fail-nomain.pic



fail-nomain.err

Fatal error: exception Failure("unrecognized function main")

fail-return1.pic

```
int main()
{
  return true; /* Should return int */
}
```

fail-return1.err

```
Fatal error: exception Failure("return gives bool expected int in true")
```



fail-return2.pic

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

int main()
{
  return 42;
}
```

fail-return2.err

```
Fatal error: exception Failure("return gives int expected void in 42")
```

fail-while1.pic

```
int main()
{
  int i;

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

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

}
```

fail-while1.err

```
Fatal error: exception Failure("expected Boolean expression in 42")
```

fail-while2.pic

```
int main()
{
  int i;

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

  while (true) {
    foo(); /* foo undefined */
  }
}
```

fail-while2.err

```
Fatal error: exception Failure("unrecognized function foo")
```

test-arith1.pic

```
int main()
{
  print(39 + 3);
  return 0;
}
```

test-arith1.out

42

test-arith2.pic

```
int main()
{
  print(1 + 2 * 3 + 4);
  return 0;
}
```

test-arith2.out

11

test-arith3.pic

```
int foo(int a)
{
    return a;
}

int main()
{
    int a;
    a = 42;
    a = a + 5;
    print(a);
    return 0;
}
```

test-arith3.out

47

test-array1.pic

```
int main(){
    int a[2];
    a[0] = 1;
    a[1] = 2;
    print(a[0]);
    print(a[1]);

    return 0;
}
```

test-array1.out

```
1
2
```

test-array2.pic

```
int main()
{
    mat a[2][3];
    a[0][0] = 0;
    a[0][1] = 1;
    a[0][2] = 2;
    a[1][0] = 3;
    a[1][1] = 4;
    a[1][2] = 5;
    print(a[0][1]);
    print(a[1][2]);
    return 0;
}
```

test-array2.out

```
1
5
```



test-array3.pic

```
int main()
{
    int a[3];
    a[0] = 0;
    a[1] = 1;
    a[2] = 2;
    print(a[0]);
    a[0] = 3;
    print(a[0]);
    return 0;
}
```

test-array3.out

```
0
3
```

test-array4.pic

```
int main(){  
    int a[3];  
    a = {0,1,2};  
    print(a[1]);  
    return 0;  
}
```

test-array4.out

1

test-assign1.pic

```
int main()
{
    int a = -1;
    print(a);
    return 0;
}
```

test-assign1.out

-1

test-block1.pic

```
int main() {
    int a = 10;
    print(a);
    {
        int a = 11;
        print(a);
        {
            int a = 12;
            print(a);
        }
        print(a);
    }
    print(a);
    return 0;
}
```

test-block1.out

```
10
11
12
11
10
```

test-fib.pic

```
int fib(int x)
{
  if (x < 2) return 1;
  return fib(x-1) + fib(x-2);
}

int main()
{
  print(fib(0));
  print(fib(1));
  print(fib(2));
  print(fib(3));
  print(fib(4));
  print(fib(5));
  return 0;
}
```

test-fib.out

```
1
1
2
3
5
8
```

test-for1.pic

```
int main()
{
  int i;
  for (i = 0 ; i < 5 ; i = i + 1) {
    print(i);
  }
  print(42);
  return 0;
}
```

test-for1.out

```
0
1
2
3
4
42
```

test-for2.pic

```
int main()
{
  int i;
  i = 0;
  for ( ; i < 5; ) {
    print(i);
    i = i + 1;
  }
  print(42);
  return 0;
}
```

test-for2.out

```
0
1
2
3
4
42
```

test-for4.pic

```
int main() {  
    for (int i=0; i<10; i=i+1) {  
        print(i);  
    }  
}
```

test-for4.out

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```



test-for5.pic

```
int main() {
    int count = 0;
    for (int i = 0; i < 2; i=i+1) {
        prints("first loop:");
        print(i);
        count=count+1;
        for (int i = 2; i < 4; i=i+1) {
            prints("second loop:");
            print(i);
            count=count+1;
            for (int i = 4; i < 6; i=i+1) {
                prints("third loop:");
                print(i);
                count=count+1;
            }
        }
    }
    prints("count:");
    print(count);
}
```

test-for5.out

```
first loop:
0
second loop:
2
third loop:
4
third loop:
5
second loop:
3
third loop:
4
```

third loop:

5

first loop:

1

second loop:

2

third loop:

4

third loop:

5

second loop:

3

third loop:

4

third loop:

5

count:

14

test-func1.pic

```
int add(int a, int b)
{
    return a + b;
}

int main()
{
    int a;
    a = add(39, 3);
    print(a);
    return 0;
}
```

test-func1.out

42

test-func2.pic

```
int fun(int x, int y)
{
    return 0;
}

int main()
{
    int i;
    i = 1;

    fun(i = 2, i = i+1);

    print(i);
    return 0;
}
```

test-func2.out

2

test-func3.pic

```
void printem(int a, int b, int c, int d)
{
    print(a);
    print(b);
    print(c);
    print(d);
}

int main()
{
    printem(42,17,192,8);
    return 0;
}
```

test-func3.out

```
42
17
192
8
```

test-func4.pic

```
int add(int a, int b)
{
  int c;
  c = a + b;
  return c;
}

int main()
{
  int d;
  d = add(52, 10);
  print(d);
  return 0;
}
```

test-func4.out

62

test-func5.pic

```
int foo(int a)
{
    return a;
}

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

test-func5.out

test-gcd.pic

```
int gcd(int a, int b) {
  while (a != b) {
    if (a > b) a = a - b;
    else b = b - a;
  }
  return a;
}

int main()
{
  print(gcd(2,14));
  print(gcd(3,15));
  print(gcd(99,121));
  return 0;
}
```

test-gcd.out

```
2
3
11
```



test-gcd2.pic

```
int gcd(int a, int b) {
    while (a != b)
        if (a > b) a = a - b;
        else b = b - a;
    return a;
}

int main()
{
    print(gcd(14,21));
    print(gcd(8,36));
    print(gcd(99,121));
    return 0;
}
```

test-gcd2.out

```
7
4
11
```

test-global1.pic

```
int a;
int b;

void printa()
{
    print(a);
}

void printb()
{
    print(b);
}

void incab()
{
    a = a + 1;
    b = b + 1;
}

int main()
{
    a = 42;
    b = 21;
    printa();
    printb();
    incab();
    printa();
    printb();
    return 0;
}
```

test-global1.out

```
42
21
```

43  
22

test-global2.pic

```
bool i;  
  
int main()  
{  
  int i; /* Should hide the global i */  
  
  i = 42;  
  print(i + i);  
  return 0;  
}
```

test-global2.out

84

test-global3.pic

```
int foo(){
    int a;
    a = 42;
    b = a;
    print(b);
    return b;
}

int b;

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

test-global3.out

42

test-hello1.pic

```
int a;
int b[5];

int main(){

a = 5;

int c[6];
int i;
i = 0;
while (i<5)
{
    b[i] = i;
    i = i+1;
}

for (int a = 0; a < 6; a=a+1){
    c[a] = a + 10;
}

for (int a = 0; a < 6; a=a+1){
    print(c[a]);
}

print(a);

for (int a = 0; a < 5; a=a+1){
    print(b[a]);
}

prints("hello world!!");

}
```

test-hello1.out

```
10  
11  
12  
13  
14  
15  
5  
0  
1  
2  
3  
4  
hello world!!
```

test-glocal4.pic

```
pic a;  
  
int main()  
{  
    {  
        int a;  
        {  
            a=0;  
            print(a);  
        }  
    }  
  
    return 0;  
}
```

test-glocal4.out

0



test-glocal5.pic

```
int main()
{
  {
    pic a;
    {
      int a;
      {
        a=0;
        print(a);
      }
    }
  }
  return 0;
}
```

test-glocal5.out

0

test-if1.pic

```
int main()
{
  if (true) print(42);
  print(17);
  return 0;
}
```

test-if1.out

```
42
17
```

test-if2.pic

```
int main()
{
  if (true) print(42); else print(8);
  print(17);
  return 0;
}
```

test-if2.out

```
42
17
```

test-if3.pic

```
int main()
{
  if (false) print(42);
  print(17);
  return 0;
}
```

test-if3.out

17

test-if4.pic

```
int main()
{
  if (false) print(42); else print(8);
  print(17);
  return 0;
}
```

test-if4.out

```
8
17
```

test-keyword.pic

```
int main(){
    bool b;
    b = (true or false);
    int a;
    if (b){ a = 1; }
    else { a = 0; }
    print(a);
    return 0;
}
```

test-keyword.out

1

test-locall.pic

```
void foo(bool i)
{
  int i; /* Should hide the formal i */

  i = 42;
  print(i + i);
}

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

test-locall.out

84

test-local2.pic

```
void foo()
{
    int i;
    i = 42;
    print(i+3);
}

int main()
{
    int i;
    i = 1;
    print(i);
    foo();
    print(i);
    return 0;
}
```

test-local2.out

```
1
45
1
```



test-local3.pic

```
int i;  
  
int main()  
{  
    i = 42;  
    int i;  
    i = 0;  
    print(i);  
    return 0;  
}
```

test-local3.out

0

test-mat1.pic

```
int main(){
    mat a[2][2];
    a = {0,1,2,3};
    print(a[1][1]);
    return 0;
}
```

test-mat1.out

3

test-minus.pic

```
int main(){  
    char i;  
    i = -1;  
    print(i);  
    return 0;  
}
```

test-minus.out

255

test-ops1.pic

```
int main()
{
    print(1 + 2);
    print(1 - 2);
    print(1 * 2);
    print(100 / 2);
    print(99);
    printb(1 == 2);
    printb(1 == 1);
    print(99);
    printb(1 != 2);
    printb(1 != 1);
    print(99);
    printb(1 < 2);
    printb(2 < 1);
    print(99);
    printb(1 <= 2);
    printb(1 <= 1);
    printb(2 <= 1);
    print(99);
    printb(1 > 2);
    printb(2 > 1);
    print(99);
    printb(1 >= 2);
    printb(1 >= 1);
    printb(2 >= 1);
    return 0;
}
```

test-ops1.out

```
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-ops2.pic

```
int main()
{
  printb(true);
  printb(false);
  printb(true and true);
  printb(true and false);
  printb(false and true);
  printb(false and false);
  printb(true or true);
  printb(true or false);
  printb(false or true);
  printb(false or false);
  printb(not false);
  printb(not true);
  print(-10);
  print(--42);
}
```

test-ops2.out

```
1
0
1
0
0
0
1
1
1
0
1
0
-10
42
```

test-ops3.pic

```
int main(){
    bool b;
    b = (3 <= 3);
    int a;
    if(b)
        { a = 1; }
    else
        { a = 0; }
    print(a);
    return 0;
}
```

test-ops3.out

1

test-var1.pic

```
int main()
{
  int a;
  a = 42;
  print(a);
  return 0;
}
```

test-var1.out

42



test-var2.pic

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

test-var2.out

test-while1.pic

```
int main()
{
  int i;
  i = 5;
  while (i > 0) {
    print(i);
    i = i - 1;
  }
  print(42);
  return 0;
}
```

test-while1.out

```
5
4
3
2
1
42
```

demo\_color.pic

```
int main()
{
  int i;
  int j;
  pic a;
  a=newpic(800,800);
  for(i=0;i<400;i++)
  {
    for(j=0;j<400;j++)
    {
      a.r[i][j]=255*j/400;
    }
    for(j=400;j<800;j++)
    {
      a.r[i][j]=255*(800-j)/400;
    }
  }
  for(i=400;i<800;i++)
  {
    for(j=0;j<400;j++)
    {
      a.b[i][j]=255*(400-j)/400;
    }
    for(j=400;j<800;j++)
    {
      a.b[i][j]=255*(j-400)/400;
    }
  }
  save_file("gradual.bmp",a);
}
```

hello.pic

```
int a;
int b[5];

int main(){

a = 5;

int c[6];
int i=0;
while (i<5)
{
    b[i] = i;
    i = i+1;
}

for (int a = 0; a < 6; a=a+1){
    c[a] = a + 10;
}

for (int a = 0; a < 6; a=a+1){
    print(c[a]);
}

print(a);

for (int a = 0; a < 5; a=a+1){
    print(b[a]);
}

int w = get_width("input.bmp");
int h = get_height("input.bmp");

prints("width = ");
print(w);
```

```
prints("height = ");  
print(h);  
  
}
```

## struct\_test.pic

```
mat kernel[5][5];

int main(){

pic a;
pic b;
pic c;
a = load("input.bmp");
b = load("balloon.bmp");
c = load("lena.bmp");

convolution(a, kernel);

for (int x = 0 ; x < a.w ; x = x+1){
    for (int y = 0 ; y < a.h ; y = y+1){

        a.g[x][y] = 255;

    }
}

for (int x = 0 ; x < b.w ; x = x+1){
    for (int y = 0 ; y < b.h ; y = y+1){

        b.b[x][y] = 128;

    }
}

for (int i = 0 ; i < c.w ; i = i+1){
    for (int j = 0 ; j < c.h ; j = j+1){

        c.r[i][j] = 128;

    }
}
```

```
    }  
}  
  
save_file("a_test.bmp", a);  
save_file("b_test.bmp", b);  
save_file("c_test.bmp", c);  
  
pic d = newpic(c.w, c.h);  
for (int i = 0 ; i < c.w ; i = i+1){  
    for (int j = 0 ; j < c.h ; j = j+1){  
  
        d.r[i][j] = c.r[i][j];  
        d.g[i][j] = c.g[i][j];  
        d.b[i][j] = c.b[i][j];  
    }  
}  
  
save_file("d_new.bmp", d);  
  
}
```

test-conv1.pic

```
int main(){
    pic a;
    a = load("test/balloon.bmp");
    mat b[5][5];
    b[0][0] = 0;
    b[0][1] = 0;
    b[0][2] = 0;
    b[0][3] = 0;
    b[0][4] = 0;
    b[1][0] = 0;
    b[1][1] = 0;
    b[1][2] = 0;
    b[1][3] = 0;
    b[1][4] = 0;
    b[2][0] = 0;
    b[2][1] = 0;
    b[2][2] = 1;
    b[2][3] = 0;
    b[2][4] = 0;
    b[3][0] = 0;
    b[3][1] = 0;
    b[3][2] = 0;
    b[3][3] = 0;
    b[3][4] = 0;
    b[4][0] = 0;
    b[4][1] = 0;
    b[4][2] = 0;
    b[4][3] = 0;
    b[4][4] = 0;
    convolution(a,b);
    save_file("conv_test1.bmp",a);
    return 0;
}
```



test-conv2.pic

```
int main(){
    pic a;
    a = load("./test/lena.bmp");
    mat b[5][5];
    b = { 0,0,0,0,0,
          0,-1,-1,-1,0,
          0,-1,9,-1,0,
          0,-1,-1,-1,0,
          0,0,0,0,0};
    a # b;
    convolution(a,b);
    save_file("test-conv2.bmp",a);
    return 0;
}
```

test-conv3.pic

```
int main(){
    pic a;
    a = load("./test/lena.bmp");
    mat kernel1[5][5];
    mat kernel2[5][5];
    kernel1 = { 0,0,0,0,0,
                0,-1,-1,-1,0,
                0,-1,8,-1,0,
                0,-1,-1,-1,0,
                0,0,0,0,0};
    kernel2 = { 0,0,0,0,0,
                0,0,-1,0,0,
                0,-1,5,-1,0,
                0,0,-1,0,0,
                0,0,0,0,0};
    a # kernel2 # kernel1;
    save_file("test-conv3.bmp",a);
    return 0;
}
```

test-conv4.pic

```
int main(){
    pic a;
    a = load("./test/lena.bmp");
    mat b[5][5];
    mat c[5][5];
    b = { 0,0,0,0,0,
          0,-1,-1,-1,0,
          0,-1,9,-1,0,
          0,-1,-1,-1,0,
          0,0,0,0,0};
    c = { 0,0,0,0,0,
          0,-1,-1,-1,0,
          0,-1,8,-1,0,
          0,-1,-1,-1,0,
          0,0,0,0,0};
    a # b # c;
    save_file("test-conv2.bmp",a);
    return 0;
}
```

conv.pic

```
mat kernel[5][5];

int main(){

pic a;

a = load("balloon.bmp");
pic b = newpic(a.h, a.w);
pic a_blur = newpic(a.h, a.w);
pic a_edge = newpic(a.h, a.w);
pic a_sharp = newpic(a.h, a.w);
copy_pic(a, b);
copy_pic(a, a_blur);
copy_pic(a, a_edge);
copy_pic(a, a_sharp);

/* Emboss */
kernel = {0,0,0,0,0,
          0,-2,-1,0,0,
          0,-1,1,1,0,
          0,0,1,2,0,
          0,0,0,0,0};

a#kernel;
b#kernel;

save_file("a_emboss.bmp", a);
save_file("b_emboss.bmp", b);

/* Blur */
kernel = {0,0,0,0,0,
          0,1,1,1,0,
          0,1,1,1,0,
          0,1,1,1,0,
          0,0,0,0,0};
```

```
a_blur#kernel#kernel#kernel#kernel;
```

```
save_file("a_blur.bmp", a_blur);
```

```
/* Edge detect */
```

```
kernel = {0,0,0,0,0,  
          0,0,1,0,0,  
          0,1,-4,1,0,  
          0,0,1,0,0,  
          0,0,0,0,0};
```

```
a_edge#kernel;
```

```
save_file("a_edge.bmp", a_edge);
```

```
/* Sharpen */
```

```
kernel = {0,0,0,0,0,  
          0,0,-1,0,0,  
          0,-1,5,-1,0,  
          0,0,-1,0,0,  
          0,0,0,0,0};
```

```
a_sharp#kernel;
```

```
save_file("a_sharp.bmp", a_sharp);
```

```
}
```

rotate.pic

```
pic rotate(pic a)
{
    pic b=newpic(a.w,a.h);
    int i;
    int j;
    int x;
    int y;
    for(i=0;i<a.h;i++)
    {
        x=j;
        y=a.h-i;
        for(j=0;j<a.w;j++)
        {
            b.r[x][y]=a.r[i][j];
            b.g[x][y]=a.g[i][j];
            b.b[x][y]=a.b[i][j];
        }
    }
    return b;
}

int main()
{
}
```

## buildexe.sh

```
cp ../libpic/convolution.pic source.pic
cat ../libpic/gray.pic >> source.pic
cat $1 >> source.pic
../picel.native < source.pic > tmp.ll
#llc -filetype=obj tmp.ll
opt -mem2reg -S tmp.ll > tmp_opt.ll
llc -filetype=obj tmp_opt.ll
llvm-g++ tmp_opt.o ../bitmap/bmplib.o
```

testall.sh

```
#!/bin/sh

# Regression testing script for PICEL
# Step through a list of files
# Compile, run, and check the output of each expected-to-work test
# Compile and check the error of each expected-to-fail test

PICEL="./picel.native"
LLI="lli"

# Set time limit for all operations
ulimit -t 30

globallog=testall.log
rm -f $globallog
error=0
globalerror=0

keep=0

Usage() {
    echo "Usage: testall.sh [options] [.pic files]"
    echo "-k   Keep intermediate files"
    echo "-h   Print this help"
    exit 1
}

SignalError() {
    if [ $error -eq 0 ]; then
        echo "FAILED"
        error=1
    fi
    echo " $1"
}

# Compare <outfile> <reffile> <difffile>
# Compares the outfile with reffile. Differences, if any, written to difffile
```



```

Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
        SignalError "$1 differs"
        echo "FAILED $1 differs from $2" 1>&2
    }
}

# Run <args>
# Report the command, run it, and report any errors
Run() {
    echo $* 1>&2
    eval $* || {
        SignalError "$1 failed on $*"
        return 1
    }
}

# RunFail <args>
# Report the command, run it, and expect an error
RunFail() {
    echo $* 1>&2
    eval $* && {
        SignalError "failed: $* did not report an error"
        return 1
    }
    return 0
}

Check() {
    error=0
    basename=`echo $1 | sed 's/.*\///
                s/.pic//'`
    reffile=`echo $1 | sed 's/.pic$//`
    basedir="" echo $1 | sed 's/\/[^\/]*$//'.
    echo -n "$basename..."

    echo 1>&2

```

```

echo "##### Testing $basename" 1>&2

generatedfiles=""

generatedfiles="$generatedfiles ${basename}.ll ${basename}.out" &&
Run "$PICEL" "<" $1 ">" "${basename}.ll" &&
Run "$LLI" "${basename}.ll" ">" "${basename}.out" &&
Compare ${basename}.out ${reffile}.out ${basename}.diff

# Report the status and clean up the generated files

if [ $error -eq 0 ] ; then
    if [ $keep -eq 0 ] ; then
        rm -f $generatedfiles
    fi
    echo "OK"
    echo "##### SUCCESS" 1>&2
else
    echo "##### FAILED" 1>&2
    globalerror=$error
fi
}

CheckFail() {
    error=0
    basename=`echo $1 | sed 's/.*\V//
                s/.pic//'`
    reffile=`echo $1 | sed 's/.pic$//`
    basedir=""echo $1 | sed 's/\V[\V]*$//`/."

    echo -n "$basename..."

    echo 1>&2
    echo "##### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$PICEL" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

```

```

# Report the status and clean up the generated files

if [ $error -eq 0 ]; then
    if [ $keep -eq 0 ]; then
        rm -f $generatedfiles
    fi
    echo "OK"
    echo "##### SUCCESS" 1>&2
else
    echo "##### FAILED" 1>&2
    globalerror=$error
fi
}

while getopts kdpsh c; do
    case $c in
        k) # Keep intermediate files
            keep=1
            ;;
        h) # Help
            Usage
            ;;
    esac
done

shift `expr $OPTIND - 1`

if [ $# -ge 1 ]
then
    files=$@
else
    files="tests/test-*.pic tests/fail-*.pic"
fi

for file in $files
do
    case $file in
        *test-*)
            Check $file 2>> $globallog
    esac
done

```

```
;;
*fail-*)
    CheckFail $file 2>> $globallog
;;
*)
    echo "unknown file type $file"
    globalerror=1
;;
esac
done

exit $globalerror

find . -name "*.mc" -exec mv 's/.mc$/pic/' {} \;
```

## 9.2 Github History

commit 1297dac1829541c9f08ff7756c58eeac672359c6  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue May 10 20:26:06 2016 -0400

Fix type identifier order

commit f0d6582833fa736c250101aff19454101de391a0  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sun May 8 14:23:51 2016 -0400

Add arr, matrix, pic assign val checking

commit b4fbadac2fabcefcabcca3d06936e54f1d1aa99d  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sun May 8 00:05:32 2016 -0400

Add source.pic & conv.pic test

commit f3859e92c7010b271cdede6186d050e339da7a42  
Merge: 660d56e 450a1fc  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sun May 8 00:04:30 2016 -0400

Merge branch 'demo\_merge' of <https://github.com/alextrax/PICEL> into demo\_merge

commit 660d56e17d97c93295c2dac95d4abe94e83c869b  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sun May 8 00:04:18 2016 -0400

Add matrix, pic, rgb assign check

commit 450a1fc9fc10f76f64359707038fc64c45cf7169  
Author: rz2337 <rz2337@columbia.edu>  
Date: Sat May 7 16:01:55 2016 -0400

Added files via upload

32->24

commit 7e6aed397a6b5c560abd0689fc368bbddc7d4ff5  
Author: rz2337 <rz2337@columbia.edu>  
Date: Sat May 7 16:01:00 2016 -0400

Added files via upload

If that's what you want

commit dd723e52e3044557a8bff111d1923a0c91e10ddb  
Merge: c4596dc 1befa45  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Sat May 7 15:59:48 2016 -0400

Merge branch 'demo\_merge' of <https://github.com/alextrax/PICEL> into demo\_merge

commit c4596dc221f94c3cc0909f8d05257cb5c2491b59  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Sat May 7 15:59:31 2016 -0400

Add assign rgbxy & pic

commit 1befa457b6462e3e0774b4372d1a990d17936fb5  
Author: rz2337 <rz2337@columbia.edu>  
Date: Sat May 7 15:55:54 2016 -0400

Added files via upload

Add bmp files for demo

commit 7701aee79af0f74cfb37e211465c578471cb72c5  
Merge: 59dc591 75d9447  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Sat May 7 15:28:46 2016 -0400

Fix conflicts in semant

commit 59dc591f7fa7b314497879ffaad217334d22fc41  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Sat May 7 15:27:26 2016 -0400

Add rgb checker

commit 75d94471383fdd58194665f10b4e50014853a25f  
Author: alextrax <cw2952@columbia.edu>  
Date: Sat May 7 14:42:51 2016 -0400

Fix r g b check for pic

commit 8268fac126c3866a0897c33bc9c90012b5a98839  
Author: alextrax <cw2952@columbia.edu>  
Date: Sat May 7 14:11:51 2016 -0400

Update conv.pic

commit 32fb3ee33e81a4e1dece742f024b4509456cf91d  
Merge: 8857b15 814206f  
Author: alextrax <cw2952@columbia.edu>

Date: Sat May 7 14:06:18 2016 -0400

Merge branch 'demo\_merge' of <https://github.com/alextrax/PICEL> into demo\_merge

Conflicts:

microc-llvm/test/conv.pic

commit 8857b15cf0094878037175b4fade88e924b389f3

Author: alextrax <cw2952@columbia.edu>

Date: Sat May 7 14:02:17 2016 -0400

Add demo test case

commit 814206fc5239aea3073ce0c1635b56874645a88b

Merge: 0ac5fae 96bf54c

Author: Chris Hsu <chia-hao.hsu@aieseec.net>

Date: Sat May 7 13:53:04 2016 -0400

Merge branch 'demo\_merge' of <https://github.com/alextrax/PICEL> into demo\_merge

commit 0ac5faeedd885fafef18c583e83de8c2f84653c2

Author: Chris Hsu <chia-hao.hsu@aieseec.net>

Date: Sat May 7 13:52:50 2016 -0400

Fix char & int cat

commit 96bf54c8510653566411b9493ea0f82fb184cec6

Author: alextrax <cw2952@columbia.edu>

Date: Sat May 7 13:46:12 2016 -0400

Fix convolution function index bug

commit aaa35a8905ba40d1ee50e45b8e851f39c82a205e

Merge: bca2b59 cd0d5b9

Author: alextrax <cw2952@columbia.edu>

Date: Fri May 6 22:00:15 2016 -0400

Merge remote-tracking branch 'origin/semant\_hello' into demo\_merge

Conflicts:

microc-llvm/test/buildexe.sh

microc-llvm/test/conv.pic

commit cd0d5b9776a77b5df5771f93a2ce2276ab699760

Merge: 8bfd38 7c13b54

Author: Chris Hsu <chia-hao.hsu@aieseec.net>

Date: Thu May 5 12:40:52 2016 -0400

fix conflicts

commit 7c13b5494df3560d5936d694d465d06b37591669  
Merge: fcb9194 18149cb  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Thu May 5 09:48:04 2016 -0400

Add Delete

commit 8bfd387da8186395b13b376fba272c406257e1c  
Merge: 18149cb 2c7a02e  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Thu May 5 09:43:38 2016 -0400

Modify the gitignore

commit 18149cb108deed5ec85738531886e2f74a0e5232  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Thu May 5 09:26:09 2016 -0400

Delete Makefile ignore and change microc.native to picel.native

commit 2c7a02e50dc8525af6d52f54863ba924e2d36c26  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Thu May 5 09:08:41 2016 -0400

Fix .gitignore

commit bca2b595d4e454dd603833a4041b4e4d06e9bb4e  
Author: alextra <cw2952@columbia.edu>  
Date: Thu May 5 01:30:45 2016 -0400

Fix get matrix indexing bug

commit 691db7a75fb3680041cd73d5d9a1c471cc6eed19  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Wed May 4 18:31:03 2016 -0400

Add Makefile

commit bcad83f0bb82d0f40398b20cb694f8e529883a38  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Wed May 4 00:39:57 2016 -0400

Add Delete

commit 0d41a37d4989e71c97da5f99fb6335a994e40abb  
Merge: ace3c3b f570cc1  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Wed May 4 00:22:38 2016 -0400



Fix conflicts

commit f570cc1a3fe44bf4ae29c34551abb621acf73e26  
Merge: 272ccfd 2a86504  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Wed May 4 00:19:45 2016 -0400

Merge branch 'master' of <https://github.com/alextrax/PICEL>

commit ace3c3bce8bae0d573a71038989712e335589c11  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Wed May 4 00:19:34 2016 -0400

Correct the format

commit 2a86504b81e9e41baa46632cf880e03efcf0baa2  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 20:24:19 2016 -0400

Delete source.pic

commit 55dd9fb3a205dde2fc7917ca551a0a99beaf1451  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 20:23:42 2016 -0400

llvm

commit d9c97f86597327058afaaa667250b7fc4a18beea  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 20:22:37 2016 -0400

LLvm

commit fcb91945bf562b655385f54064e513b15a6f503a  
Merge: ba7a1b6 06251d4  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 20:13:24 2016 -0400

..

commit ba7a1b61768491699442547a487bb861ff44cbe5  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 17:27:34 2016 -0400

Rui

commit 06251d4f9e2b42baf96d61544a301cfff5a9918a  
Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Tue May 3 17:21:28 2016 -0400

Fix all warning!

commit 87cfa13c485b306005c391219e7b46b01d3ac288  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 16:52:49 2016 -0400

Change matrix reference sequence of width and height

commit 55c2c069ccbae5c639311c8a9a88cfbb49bb4002  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue May 3 16:33:51 2016 -0400

Fix demo

commit 00d8689dcedd0eec67d5809eaabdfcea9fae0b20  
Merge: a4c3b2d 101b18d  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue May 3 16:28:08 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit a4c3b2d63271890a89187df988c853b031c12994  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue May 3 16:26:23 2016 -0400

A small demo

commit cf22d5aa7210ffefb925e8f81ee2e9d0da10ef7e  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue May 3 16:24:55 2016 -0400

A small demo

commit ec56eae13884195b0a4dbfc7730fc3c9e4bb45cc  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 16:02:16 2016 -0400

Fix conflict of branch convol

commit e4760d2503b93a73fa81a267812e1ffbc5c99a06  
Merge: 644af08 101b18d  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 15:54:40 2016 -0400

Merge branch 'convol'

Conflicts:

microc-llvm/ast.ml  
microc-llvm/microc.ml

commit 101b18d1a2fb81920bbee363b7151eb00ebd73f5  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 15:51:17 2016 -0400

Update buildexe.sh

commit bc6d18b35b2ae88d216c53ae7d8299e8bacdda56  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 15:43:53 2016 -0400

Fix warning

commit 5fa4ac90b5054081ba90399746aa2496bb942c29  
Merge: b5b7889 eefb920  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Tue May 3 15:32:46 2016 -0400

Fix ast conflicts with convol

commit eefb9209bb6a285ce2ea9dceb6bb442ba2a39f62  
Merge: 9d899db e17bcc9  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue May 3 15:31:05 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit b5b78899a86ff76aac812294d23b03e68cb64a1a  
Merge: c318766 c07db6c  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Tue May 3 15:30:46 2016 -0400

Fix all conflicts for latest convol & semant

commit 9d899db72df154a5e170a6623bfc0323ab11cced  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue May 3 15:30:21 2016 -0400

Fix warnings of codegen

commit c07db6c51b9c1bc12e85addd868eb104dd341dc1  
Merge: 5eda868 299de1e  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Tue May 3 15:19:26 2016 -0400

Fix all conflicts

commit 5eda8689621fd09827e701c7eb0685db2325ee4c  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue May 3 15:17:11 2016 -0400

Merge with RuiLu branch

commit c3187668df83a676253e60641e9db29c96465a97  
Merge: de00f55 b76fdd2  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue May 3 15:15:57 2016 -0400

Fix convol conflicts

commit b76fdd2705ca1bff2f3af02995898204e55678de  
Merge: e4b445c e17bcc9  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue May 3 15:11:38 2016 -0400

Fix the conflicts

commit e17bcc92f490f9bdebd74da1c4e852970be314c6  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 15:04:46 2016 -0400

Update convolution test case

commit 299de1eeb7691c5187194fb0af122298ff78d59d  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 15:00:23 2016 -0400

Get rid of another warnings

commit 096b96cda3600fd274abfbf330adf50eb100ebb0  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue May 3 14:51:35 2016 -0400

Get rid of some warnings

commit 8d03a207acc935be071f46b234cade98f253e39c  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue May 3 14:50:14 2016 -0400

Change convolution function to return pic

commit de00f559dca2cc573c1b97e68a7068740bf58db7  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Apr 29 12:34:38 2016 -0400

Finish semant checking for source.pic

commit 295936a7476ebcf3f2278af6a960aa3e71842d9d  
Author: alextrax <luciferxiii@gmail.com>  
Date: Fri Apr 29 12:25:21 2016 -0400

Fix minus int problem

commit 9843359e8c635372d53b0f0d3791098a3010f900  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 29 12:14:34 2016 -0400

Old modification influence the use of minus

commit 24a99bdc5bbedc6a99a284f65b276b0b7a1fc0e6  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 29 11:58:16 2016 -0400

OK

commit a66a29ef31833e8289b21830a7695ea230724468  
Author: alextrax <luciferxiii@gmail.com>  
Date: Fri Apr 29 11:09:20 2016 -0400

Try to let convolution return pic

commit b1b2aa3e45b3d1dce695e099a13e2188b820db0c  
Merge: e8fa439 3e4db38  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 29 10:41:05 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit 3e4db383d26b6d313ac5e56cbb3f6032eb9ad2bd  
Author: alextrax <luciferxiii@gmail.com>  
Date: Fri Apr 29 10:40:23 2016 -0400

Implement # for convolution

commit e8fa4392a20f2b91141a4775fbabd136095d65bf  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 29 10:38:34 2016 -0400

Build library to convert color into gray (but still stored as RGB)

commit bc6e4fd51c96ff0c5141cea6ad4cdc2d7f3e6324  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 29 10:31:07 2016 -0400

For debug

commit 3205dc7650c9c5a0b4078dd9314c0bcd8fb6ab93  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 29 10:30:44 2016 -0400

New

commit a524e19c956e60fde3399fa46cf1c1249081f9b8  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 26 17:04:26 2016 -0400

Complied OK. But don't know if it is correct..

commit 52e77c88ea418c481feaa8a1efbb4c377e31279a  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 26 16:51:02 2016 -0400

Source.pic

commit 4e774933b08016c3ddeca118bd4817b36ae6a8d6  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 26 16:48:26 2016 -0400

Modify scanner.mll to support negative numbers in matrix

commit 82aeb374baf64e27f4dd2572d3028a8bce957e1b  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 26 16:32:12 2016 -0400

Implement the assignment of array/matrix, namely use "{" and "}" to form an array that could be assigned to the beginning of an array or a matrix

commit 517667eede2e005046e54af56a43cb51a30d4839  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 26 16:20:22 2016 -0400

Matrix ast

commit a4cffc0493206bd9e9471ebb8e6d53d38ef34508  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 26 15:59:29 2016 -0400

Modify ast.ml and parser.mly to support initialization of matrix and array

commit 16daaed8d5d39c08413eaa8c960017ce9b23711a  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 26 15:37:49 2016 -0400

Makefile

commit 736cf161d162c221a5e8e4046187d56930c31b16  
Merge: 2a7c61e 30e6c02  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 26 15:34:21 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit 30e6c02e8f3ae80a7830010f22db6c5d459e3659  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Apr 26 15:34:08 2016 -0400

Add delete pic

commit 2a7c61e9019f4a7aab9693d62fb30ed44013570e  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 26 15:32:32 2016 -0400

Add "delete temp" to delete temporary memory

commit 927279cafa64237040d03de4c59acd5303c339d0  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 26 15:30:31 2016 -0400

Add convol support

commit 53208d96fb921d160bd80ac6cd5fbe4047b126df  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 26 15:25:59 2016 -0400

Delete microc,l & Makefile

commit d3e957521c4eb7873c71d972dcb31a9f0fac0fab  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 26 15:21:05 2016 -0400

Add matrix & rgb support

commit 92ffb5c82ae8b5ff72468e391113a30d7749c6cd  
Merge: 909bf38 e4b445c  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 26 14:59:47 2016 -0400

Merge branch 'convol' into semant\_hello

commit e4b445c45077573b7edd6ef52189c7569c61e1a4  
Merge: b471aa7 39394e3  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 26 14:59:32 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit 272ccfde6cbc1b743c77036a7231b87c674c1c21  
Author: Chris Hsu <[chia-hao.hsu@aiasec.net](mailto:chia-hao.hsu@aiasec.net)>  
Date: Tue Apr 26 14:59:18 2016 -0400

Add matrix & rgbxy support

commit 39394e32fcc9c792fe1349e59831e0bbe6fc776b  
Author: alextrax <[luciferxiii@gmail.com](mailto:luciferxiii@gmail.com)>  
Date: Tue Apr 26 14:53:56 2016 -0400

Bug fix: type\_map indexing of global variable

commit 644af08bafd28341e0b8cfd66debe36ea255080d  
Merge: c837839 909bf38  
Author: Chris Hsu <[chia-hao.hsu@aiasec.net](mailto:chia-hao.hsu@aiasec.net)>  
Date: Tue Apr 26 14:40:04 2016 -0400

Merge branch 'semant\_hello'

commit 909bf3857d2a6e8b68a350b9c86673d93c8ec8dc  
Author: Chris Hsu <[chia-hao.hsu@aiasec.net](mailto:chia-hao.hsu@aiasec.net)>  
Date: Tue Apr 26 14:38:15 2016 -0400

Modify Makefile

commit 9af2269601e1dad1c4fc6efbe395405ffc87a053  
Author: Chris Hsu <[chia-hao.hsu@aiasec.net](mailto:chia-hao.hsu@aiasec.net)>  
Date: Tue Apr 26 14:32:11 2016 -0400

Check out RuiLu first

commit 29b9dd6437b7ca04aeba4c8f31241f05091bda08  
Merge: 9c3e83d b471aa7  
Author: Chris Hsu <[chia-hao.hsu@aiasec.net](mailto:chia-hao.hsu@aiasec.net)>  
Date: Tue Apr 26 14:25:14 2016 -0400

Merge branch 'convol' into semant\_hello

commit 9c3e83df00b43a1fcff737b61ba464d5ff0b63a6  
Merge: 7ee7f5f cc54e80  
Author: Chris Hsu <[chia-hao.hsu@aiasec.net](mailto:chia-hao.hsu@aiasec.net)>  
Date: Tue Apr 26 14:24:56 2016 -0400

Merge branch 'RuiLu' into semant\_hello

commit 7ee7f5ffa215abcfcdcb69bec1c8a7acd34c193



Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 26 14:24:26 2016 -0400

Check to Rui Lu first

commit 958508f6e2bcd619f21fa481d3cd02bc66d57201  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 26 14:22:51 2016 -0400

Add for tests

commit 97492276c7cf5e6e222cc71e834afa44e20be209  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 26 14:22:04 2016 -0400

Add array tests back

commit cc54e8084cd650431988c6509b9cab1dc768aef6  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 26 14:21:09 2016 -0400

Delete some tests

commit 5749a63f5dd72339a17e442c4b8f6965708df730  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Sun Apr 24 20:26:21 2016 -0400

Support only for-init

commit e19069f8b564a9a356e066d9b8b57457b220447b  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Sun Apr 24 19:45:51 2016 -0400

Revised new version

commit e86e05dc09466637665377e8fc0f03062c64ff65  
Merge: 19fe981 e48cb7d  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Sun Apr 24 17:11:16 2016 -0400

Block done.

commit 19fe98127a23eeef483f18a7b1d97168ff1aec71  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Sun Apr 24 16:59:48 2016 -0400

For merge

commit ab8ea546a50f927dd9f660c938e3f6ad67fb0b65

Merge: c9a170b 9e7ac52  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Sun Apr 24 16:32:56 2016 -0400

My version

commit e48cb7d148c384f4b5105be0e850a548412c8dd8  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 21:20:07 2016 -0400

Add block test

commit c83783991e8c833623eec9fd79a9bad94d819eb0  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 21:17:50 2016 -0400

Add block test

commit b89a265f225fb4755f421caf8fbc0c11a1e61e7e  
Merge: ad0a9ae 681bdee  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 18:54:27 2016 -0400

Merge branch 'master' of <https://github.com/alextrax/PICEL>

commit ad0a9ae0e53d49e3dbf6586efd57828becec0833  
Merge: a73c93c b471aa7  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 18:47:59 2016 -0400

Merge branch 'convol'

commit b471aa7bad45f2c2062649daf630878a30e24f94  
Merge: a3d9efb e9b5d74  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 18:47:17 2016 -0400

Merge with current semant & fix conflicts

commit e9b5d74ffa825498012c059180c01368730886fc  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 18:44:26 2016 -0400

Check convol first

commit 277fd9f2c318e91058d5779df7c2fcbce7aee376  
Merge: 1382525 78b72c5  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Sat Apr 23 17:52:04 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into semant\_hello

commit 138252593e8f7a92300cd405aea66a513a0974d5  
Author: Chris Hsu <[chia-hao.hsu@aiesec.net](mailto:chia-hao.hsu@aiesec.net)>  
Date: Sat Apr 23 17:51:47 2016 -0400

Fix block & for local symbol list problem

commit 78b72c54780866e9837f94d789065b7657d4d256  
Author: jxsdfzlc <[jxsdfzlc@gmail.com](mailto:jxsdfzlc@gmail.com)>  
Date: Fri Apr 22 12:34:14 2016 -0400

add more array test cases

commit a3d9efb3ac1f1aa3311195e69b739dd281710419  
Author: rz2337 <[rz2337@columbia.edu](mailto:rz2337@columbia.edu)>  
Date: Fri Apr 22 12:23:57 2016 -0400

Fix 2 shift/reduce conflict

commit bce831d1f511830de10c91eb95768a0d1f5cd972  
Author: alextrax <[luciferxiii@gmail.com](mailto:luciferxiii@gmail.com)>  
Date: Fri Apr 22 12:06:58 2016 -0400

Add include lib in bitmap.cpp

commit 353e7077394749dd136519d5970da4c4f5006650  
Author: alextrax <[luciferxiii@gmail.com](mailto:luciferxiii@gmail.com)>  
Date: Fri Apr 22 12:01:10 2016 -0400

Modify new\_pic function

commit 28228282931a275923e7ed10b6ac901e42d2d3ca  
Author: rz2337 <[rz2337@columbia.edu](mailto:rz2337@columbia.edu)>  
Date: Fri Apr 22 11:39:02 2016 -0400

Perhaps we solved the cast\_ext problem

commit c9a170bfbe3b7dc1e0744ac6bf6e0838eeb4366d  
Author: RuiLu <[rlu0213@hotmail.com](mailto:rlu0213@hotmail.com)>  
Date: Fri Apr 22 11:08:56 2016 -0400

..

commit 9e7ac52cdc1c6f78cfd38f91264a124c5add7fbc  
Author: Chris Hsu <[chia-hao.hsu@aiesec.net](mailto:chia-hao.hsu@aiesec.net)>  
Date: Fri Apr 22 11:08:10 2016 -0400

Add string\_of\_list, string\_of\_hash to debug

commit 976cc120be573f0be51e0cf13698dec3aa729f04

Author: rz2337 <rz2337@columbia.edu>

Date: Fri Apr 22 11:04:07 2016 -0400

Try to fix return cast problem.  
--need to handle zext and cast

commit c97d2dd258803df0b430731300aea0e7b51159a1

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Fri Apr 22 10:25:47 2016 -0400

Try to fix for bug

commit 9e0db86916617ad485ff43ef20cc01428c43842d

Merge: 13eff7b 002c710

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 22 10:11:17 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into RuiLu

commit 5f83cf0eb26e95717af009f96f732852553ee388

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Wed Apr 20 16:30:25 2016 -0400

Checkout develop to add hashlist into semant

commit 002c7107051d54d2907b7e3146912324890b2d2e

Merge: b2377ce 1ea77b0

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Tue Apr 19 17:44:47 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into semant\_hello

commit b2377cecb0f3303e5727af6ed823c33de90999bc

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Tue Apr 19 17:44:30 2016 -0400

Still have for & while init error

commit da458831c462e993f07faa1f3021f19a879f5baf

Author: rz2337 <rz2337@columbia.edu>

Date: Tue Apr 19 17:26:17 2016 -0400

copy\_pic seems ok

commit 159fa1e3f378da5749e21d9402d7172db27d847a

Author: alextrax <luciferxiii@gmail.com>

Date: Tue Apr 19 17:12:07 2016 -0400

Bug fix: use unsigned extend when assign char to int

commit 3929a975b4c4a62f2dd8ef7ae0837783d7330a3a

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Tue Apr 19 16:49:36 2016 -0400

Start engage the local variable structure into semant

commit d59bf3574a497de040a360f11bc08eadad7b9d64

Author: alextrax <luciferxiii@gmail.com>

Date: Tue Apr 19 16:46:42 2016 -0400

Bugfix: refine bitmap heap allocation

commit 1ea77b0b2e09316ddb742980e50d0a39229d04f7

Author: jxsdfzlc <jxsdfzlc@gmail.com>

Date: Tue Apr 19 16:03:54 2016 -0400

please fix these problems: array index out of bounds, two-dimensional array

commit e150a2173d95d047959b2b822ed1f01869d5d988

Author: jxsdfzlc <jxsdfzlc@gmail.com>

Date: Tue Apr 19 15:58:34 2016 -0400

please fix these problems: array index out of bounds, two-dimensional array

commit 5425d5d68562ba2d954a66b29299a0c5da3bd013

Author: jxsdfzlc <jxsdfzlc@gmail.com>

Date: Tue Apr 19 15:07:13 2016 -0400

add array, local, variable test

commit 91a88f7784d5df530ccd5ac432995825c32fa7f9

Author: rz2337 <rz2337@columbia.edu>

Date: Tue Apr 19 14:57:44 2016 -0400

Add "copy\_pic" function to copy a picture.

--Don't forget to add delete in convolution.

commit ffeafcc670cb6fdea7249f538fb92779120a499b

Author: alextrax <luciferxiii@gmail.com>

Date: Tue Apr 19 14:54:18 2016 -0400

Convolution bug fix

commit 13eff7bb3c804cb9d0279a1809898816c5137860

Author: RuiLu <rlu0213@hotmail.com>

Date: Tue Apr 19 14:43:35 2016 -0400

### Merge

commit 1e0765ae571af96845f67804fb2f4081ddca9882

Author: alextrax <luciferxiii@gmail.com>

Date: Sat Apr 16 15:32:39 2016 -0400

FIX convolution segmentation fault

commit 1ecf798539803fcd24d2ceafbfdcd47e986e8127

Author: alextrax <luciferxiii@gmail.com>

Date: Sat Apr 16 12:59:31 2016 -0400

Change local array declaration from L.build\_array\_alloca to L.alloca  
(array typew)

commit bf18c34b3a01b66c24e82426e6b25f496cbbf257

Author: alextrax <luciferxiii@gmail.com>

Date: Sat Apr 16 00:37:59 2016 -0400

Fix built-in function redeclaration issue, use build\_exe.sh to auto include built-in functions  
FIXME: calling convolution function will cause segmentation fault

commit 8e6cf52f700896c319aecae0075e9be59613cb65

Author: alextrax <luciferxiii@gmail.com>

Date: Sat Apr 16 00:17:47 2016 -0400

1. Support type conversion, now different types can be assigned to each other
2. Successfully compile convolution.pic
3. FIXME: built-in function will be declared twice, which leads to different function name. Ex. \_convolution.1

commit a4da564eedd7c51a93cd0fc4d6c955037e3fb4e4

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 15 12:43:19 2016 -0400

### Makefile

commit 209625dfa8d33c4952ac6e478f6139eb23141b9e

Merge: a861cee 4ffa7e4

Author: jxsdfzlc <jxsdfzlc@gmail.com>

Date: Fri Apr 15 12:40:22 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into semant\_hello

commit 1fb1b921e9ad9630219a6f1ac23c6392a0b2e3ae

Merge: 4ffa7e4 a861cee

Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Apr 15 12:02:30 2016 -0400

Merge branch 'RuiLu' into semant\_hello

commit b8d3ff4900f9ff17e583a7a859fd4851bf3c61d1  
Merge: a861cee 4ffa7e4  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 15 11:58:20 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into RuiLu

commit 4ffa7e49696a164987c6de97b690cdcefb030a9  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Apr 15 11:49:57 2016 -0400

Add checking local variable duplicates

commit 8c47a8ab0e47d922de062a0d04d2919740909254  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 15 11:21:47 2016 -0400

Print input matrix kernel to test

commit 3987206fc85af945e1bf8f766b25960fb85ff602  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 15 11:19:14 2016 -0400

Modify ast.ml and parser.mly to support matrix type as a paramater of function

commit a861ceea297fa9ca291eae81da8870e5f843c5dc  
Merge: d936ec6 199db87  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 15 11:09:39 2016 -0400

Update

commit d785e4dbe411ac42012cba7d46a2bbea6c6158a0  
Merge: 83ecef7 b3cfbf5  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 15 10:57:06 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit 83ecef710a804dab11aca56de8ecf7e11b4c3295  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 15 10:54:59 2016 -0400

The library for convolution.

Havn't delete the template pic yet.

commit b3c9bf5261accd2282de9c965806add99c347cbd  
Author: alextrax <luciferxiii@gmail.com>  
Date: Fri Apr 15 10:50:45 2016 -0400

Add built-in function newpic() for creating new empty pic

commit d936ec6d7abf7821174aaf8a71b44c1986e30a30  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 15 10:23:18 2016 -0400

Buildexe

commit 11d3c5f278b1441a3ba162001c86035ceb85de81  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 15 10:20:29 2016 -0400

Support save

commit 80621c9c0811bef9a7c086793f0cc7f172f7f822  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 12 17:31:41 2016 -0400

Save has not done yet

commit 199db879282037618981b3e3069e38d33f01facb  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 12 16:58:05 2016 -0400

Modify scanner.mll and parser.mly to support C++ feature a++ and a--;

commit cdf3b3c2eac5f620283362fd98b86c52b547a339  
Merge: ef229d5 f098f63  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Apr 12 16:50:00 2016 -0400

Merge branch 'convol' of <https://github.com/alextrax/PICEL> into convol

commit ef229d5a84d9966130bf0411696bf9307343ee40  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Apr 12 16:48:52 2016 -0400

Implement codegen for matrix initialization and reference

commit c62f76b1d95493eb46820c36159a5340cb1ab26a  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 12 16:45:22 2016 -0400



Save has not done yet.

commit f098f63e08315ae2e49ca6c1c3cca08b6628102d  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 12 16:22:10 2016 -0400

Modify ast.ml and parser.mly to support Convolution.

commit 681bdee12a359e5ec032068a83a866e7dcb7868c  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 12 16:13:54 2016 -0400

Update README.md

commit e0b20471447dbaff215c4e8737f5dd7007033e9f  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 12 16:05:34 2016 -0400

Add hashtable to store type of array (and matrix in the future)

commit 69aa03c9afe7652ce8b83b009bc320aff6452953  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 12 15:27:56 2016 -0400

Support load

commit 01007c91f8dbef31cafd0a4390700fe72f63644f  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 12 15:24:32 2016 -0400

Bug fixed!!!!

commit a73c93c3c06fc9f3fc4e4a452c49bf8bff224241  
Merge: 67dd1c1 433117d  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 12 15:05:48 2016 -0400

Merge branch 'master' of <https://github.com/alextrax/PICEL>

commit 67dd1c15865dc5fa617d22057eb28538e4da3a86  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 12 15:05:11 2016 -0400

Add Makefile

commit 2c17b1a6ff710bfa2571103f097809629fe7024a  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Apr 12 15:02:18 2016 -0400

Add and, or, not test

commit 33196c0ab59f31e095a8efef7f54c5d793755521

Author: rz2337 <rz2337@columbia.edu>

Date: Tue Apr 12 14:57:40 2016 -0400

Fix me! Try to add Matrix, but has conflict.

commit c5e9153906e0f98f7483610a96b0f9dc4d5d25f8

Merge: 220279e 622cad5

Author: RuiLu <rlu0213@hotmail.com>

Date: Tue Apr 12 14:29:30 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into RuiLu

commit 220279e315a4355969932eee250aedfdbc53eb1a

Merge: 0eb39db 0028680

Author: RuiLu <rlu0213@hotmail.com>

Date: Tue Apr 12 14:29:21 2016 -0400

..

commit 622cad50f04c3cdf0abd3a553c9dd310fcbffb54

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Tue Apr 12 14:28:17 2016 -0400

Add test err & out

commit fb74aeb68f4dbf36f1c2455644a2eecb8b42d0ad

Author: alextrax <luciferxiii@gmail.com>

Date: Tue Apr 12 14:26:10 2016 -0400

add test data

commit 0eb39dbc70e604972dee88f4872d8ad43ec9a9d4

Author: RuiLu <rlu0213@hotmail.com>

Date: Tue Apr 12 14:01:13 2016 -0400

Temporary

commit 002868052a91e85ecf67a66ee44b022a6526a5d6

Author: Chris Hsu <chia-hao.hsu@aiese.net>

Date: Tue Apr 12 13:39:25 2016 -0400

Add automatical tests

commit 0ff0a66c8e963cb1a3ad32506f05dbf690804b0d

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 22:11:28 2016 -0400

Bug fix:

1. Change passing of pic to save() function from pic\_t to \*pic\_t
2. Fix global and local pic memory corruption issue
3. Now we can use loop to modify pixel value

commit 63936f9f28333fb1ec4a05a41285520010d0faa3

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 15:29:22 2016 -0400

Implement direct access RGB value by pic a; a.r[x][y]

FIXME:

1. loop causes segmentation fault
2. Global pic can not output picture correctly after printing pic attributes

commit 24c58abdb9604ea0cb4208169f462536724d951b

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 13:31:54 2016 -0400

Fix bug of string arg passed with pic

commit 6ed88c91741d51dad16f47919096388c70ac386b

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 12:27:31 2016 -0400

add temporary test folder

commit 2e9be13a965a7f26fbad774c5d4ab628b45e1c1e

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 11:47:06 2016 -0400

Add bmplib Makefile

commit d2ca944d859fefe431ef87c56828b05786a7b0cf

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 11:39:02 2016 -0400

Support save image

commit d527034eb7e21f0b83aac2523a7aa25b72027d3c

Merge: 468ff2e 49a455b

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Fri Apr 8 11:16:41 2016 -0400

Merge remote-tracking branch 'origin/pointer' into semant\_hello

commit 468ff2e6ab4b380d9bccde5fc39cce16e7a11215

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Fri Apr 8 11:15:44 2016 -0400

Checkout for pointer first

commit 3951941a7d83c2f852a82b321e3150e84367b97a  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 8 11:09:37 2016 -0400

Add two tests for pic\_tests for automatic tests

commit 17cadf7e936f267dd963702d2150cfaa593f37c1  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 8 11:06:08 2016 -0400

Delete unnecessary files

commit 6ec3bd0106dbd95dc953630f294215ec2498d73c  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 8 11:05:17 2016 -0400

Delete old\_semant

commit 196c2b6980f8fa2bb331aa18c7af7e0bf7a9d687  
Merge: 473afb3 49a455b  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 8 11:02:07 2016 -0400

Merge branch 'pointer' of <https://github.com/alextrax/PICEL> into RuiLu

commit 29df45bed7b634eea46683428dfc64e85881e733  
Merge: 700266c 473afb3  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 8 11:02:01 2016 -0400

Merge branch 'RuiLu' into semant\_hello

commit 700266c28c6df5f750db6e31ee3fb595914bd731  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 8 11:01:10 2016 -0400

Modify the for init

commit 433117d897261fd8ebd49a24b8712c9aadadb538  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 8 10:58:15 2016 -0400

Update to\_do.md

commit 473afb3123fcdf5e99d75c0c4caf9fb48868083  
Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 8 10:56:23 2016 -0400

Update README.md

commit fb48d91f637db8e49b4de657d1e3adb04e45ac36

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 8 10:46:12 2016 -0400

Test

commit 072a6dfb42e477979dd3f44f25bff163bb2814e0

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Fri Apr 8 10:44:55 2016 -0400

Update to\_do.md

commit ab130c568b19ef3127368a2046df629160adca1b

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Fri Apr 8 10:44:14 2016 -0400

Update to\_do.md

commit f1556b91cb9a320aa370ddc605ecc5a1adf70f28

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Fri Apr 8 10:43:53 2016 -0400

Create to\_do.md

commit 0730517d95555de968ac44c6bef89d9cc1781ac2

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 8 10:40:15 2016 -0400

Test for push without username and password

commit 11bf9e70ecdce9686c6173708741f65fc86ccacf

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 8 10:35:32 2016 -0400

Support initialization in For.

commit c64a2ee34f3a1bab76185209d73c27b59a363495

Merge: 3c93a98 2491edb

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 8 09:45:41 2016 -0400

New version on 04/08/16

commit 3c93a98a6b639884f02fe1b2cc5dc9cdb0f02fab

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 8 09:40:08 2016 -0400

Semant for init and bind

commit 49a455b51352ca26e7d63285a5578281c7cc8417

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 01:06:55 2016 -0400

Implement load function for directly load bitmap into PICEL

commit 078ef31e084166069b25fc9e880837f4c8d86f07

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 00:20:06 2016 -0400

Support local PIC struct

commit 57eff0e9320bb10dfcc121130e2e10edfcfb4b72

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 8 00:08:43 2016 -0400

1. Support Global pic struct reference
2. Modify grammer to support pic.w, pic.h, pic.bpp, pic.data accessing

commit d6d3b0b099762e39a94061b3a87e1249affd3596

Author: alextrax <luciferxiii@gmail.com>

Date: Thu Apr 7 23:04:45 2016 -0400

Make PIC be a struct type { int width, int height, int bytes\_per\_pixel, char\* data }

commit e8192595c77e16b60d5f92570c0e1cce801c4300

Author: alextrax <luciferxiii@gmail.com>

Date: Thu Apr 7 18:57:31 2016 -0400

1. Add bitmap library
2. implement get\_width and get\_height function

commit 25b7509064c1bb43c1e1ba848e4e690313b85bd2

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Wed Apr 6 18:14:14 2016 -0400

Successfully support array in semant with Getarr & Assignarr

commit 2491edb27d727f99888325a8606f512ab9b3fa7c

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Wed Apr 6 18:14:14 2016 -0400

Successfully support array

commit 56fac228040f6f66f0cb9750bad871f292daf6b9

Merge: 8270d5e e0138bd  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Wed Apr 6 17:59:59 2016 -0400

Support for & while, work on array now

commit 8270d5ee831870f1381e839023029d5ecbfff4f6  
Merge: 6a64ee7 6951edc  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 5 16:28:51 2016 -0400

Merge branch 'codegen\_hello' into semant\_hello

commit 6a64ee706ebb0af4f4a7eae27adcf71b95be213a  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 5 16:25:28 2016 -0400

Modify the local initialization test

commit bfb4045212538c230524753c11fbe91da0f117bc  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 5 16:24:34 2016 -0400

Successfully support local initialization

commit bf7f181a2b3fdbaf4717981a8c7e3508c0689394  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 5 16:08:58 2016 -0400

Successfully support local variable binding

commit 80dcbf95ec6485691df8a062e8bc75547ea8e512  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 5 16:05:29 2016 -0400

Bind in semant succeed

commit c071c3b4f7e07324f851ba1ada13a45af56c6d24  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Apr 5 15:34:27 2016 -0400

Start to change symbols from StringMap to Hashtbl

commit 8856bde83100b96c5cfd0e3075b09ca708c45560  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 5 15:32:15 2016 -0400

Before big change for semant.ml

commit e0138bd3de597f5045ab3c83baeaa55f1bb3df7f  
Merge: 4053b40 3de41e3  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 5 15:25:26 2016 -0400

Merge branch 'array' of <https://github.com/alextrax/PICEL> into array

commit 4053b40475cbbd8c02b8664f55a72e5f7b592e30  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Apr 5 15:25:00 2016 -0400

-modify codegen.mly to support formals

commit 3de41e38374c8376cbce0b94d9d4402b7e960e74  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Apr 5 15:21:28 2016 -0400

Support local array declare and access

commit 2020e9a867641a133ede01d6e9c6beb0c2cb0a61  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Apr 5 14:01:02 2016 -0400

Correcting semantic

commit 36a9122fbd6252014eee917b03c4e77b5f03dae3  
Author: rz2337 <rz2337@columbia.edu>  
Date: Sat Apr 2 14:54:20 2016 -0400

-modify codegen.mly to support local variables in different level of block.

commit 2969c6c5de52f4c16721f2c3781a98e51ce23bf1  
Author: rz2337 <rz2337@columbia.edu>  
Date: Sat Apr 2 12:19:39 2016 -0400

-modify ast.ml, parser.mly and codegen.ml to support for statement.  
-rename the tokens used in "for\_init"

commit c967ffc645e91bfb53ca16a2b7bc871df5ae508f  
Author: alextrax <luciferxiii@gmail.com>  
Date: Fri Apr 1 22:11:01 2016 -0400

Support codegen for int array

commit 69fdf112006f37cd1e7a2ac5746f2bc790107140  
Merge: 94327c2 04306dd  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 1 16:28:42 2016 -0400



New version

commit 94327c2cc74302b8af8cca2026a853db7e6ca59c

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 1 16:23:49 2016 -0400

Ready to add bind.

commit 04306dd30cd6b9764f1c03d8fd8cb9d5b7704213

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Fri Apr 1 16:14:59 2016 -0400

Add bind for symbols map, but not solving the assign issue yet

commit b62db2117181e8753b65e00117e68a9122e4612f

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Fri Apr 1 12:36:10 2016 -0400

Need to support bind & initial

commit a821b686b3592e6e7df734bbae1fc4feb00faea9

Author: rz2337 <rz2337@columbia.edu>

Date: Fri Apr 1 12:27:09 2016 -0400

Update README.md

commit f945cf03a0f0cd5be9fd28ac73e9c19aba58cbe0

Author: rz2337 <rz2337@columbia.edu>

Date: Fri Apr 1 12:26:32 2016 -0400

Create README.md

commit d2da447a713dec8ef83411d0b1d7ddfea9097bf9

Author: RuiLu <rlu0213@hotmail.com>

Date: Fri Apr 1 12:25:33 2016 -0400

New

commit adaeaae7f19e8b6d49eb85641ca99a4e7a83ef13

Author: rz2337 <rz2337@columbia.edu>

Date: Fri Apr 1 12:21:42 2016 -0400

-Again modify ast.ml and parser.mly to support assignment of array and get the value of array

commit 8eb68f7dff7c108301f91b5d7f1a7b4029572781

Author: alextrax <luciferxiii@gmail.com>

Date: Fri Apr 1 12:05:25 2016 -0400

Support Array declaration

commit 6e6ada44e2f9c68eeb29cf124beb598cc3c436b4  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Apr 1 11:35:01 2016 -0400

Print hello world with semant & codegen combination

commit f8adaac9b5088b86d90dca8803343752d78d3d5b  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 1 11:10:24 2016 -0400

Current version, 04/01/2016

commit 6951edcd3b45a04a722117f268d1908c2c31860c  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Apr 1 11:01:46 2016 -0400

Add print hello test cases

commit 38946017c291d071fc52d6135f0736cbe0273a55  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Apr 1 10:57:52 2016 -0400

Checkout codegen\_hello first

commit 89e18bc3aaa1ebe9d053d08ff42b3969c05fd601  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Apr 1 10:52:17 2016 -0400

-Modify ast.ml and parser.mly to support array

commit 949556e23b87e56e93aac44b7dbdca7f32bd6987  
Merge: 4093e20 159f2fc  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 1 10:50:16 2016 -0400

Useless

commit 4093e20b26583e564a95e247c841807c78ee8d8a  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 1 10:43:09 2016 -0400

My Parser

commit 77418e71810c76e79876e79e6f56b22cb69388c2  
Merge: df0adcf 3ded196  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Apr 1 10:42:03 2016 -0400

Test

commit 159f2fc06c70b8a9e72e1335b5e17ce70c00b0b3  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:40:02 2016 -0400

Delete llvm

commit 7cc092da0fd40ce2840db6b1cd8a88314d2669  
Merge: feb22d8 d56d02f  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:38:30 2016 -0400

Test new gitignore

commit feb22d874575df05b991354351ed3bbbb2b84bdd  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:35:07 2016 -0400

Add hello test case for semant & codegen

commit d43a29da9b1acf800f7c8612baf46083819050ca  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:34:45 2016 -0400

Combine current codegen & ast

commit 128ceb801299cdb3511e71e8d0f6ccba720e77c7  
Merge: e497174 3ded196  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:23:43 2016 -0400

Merge with Semant\_hello ast & microc

commit e49717417b80619c6019d10020af9dbb8a01beed  
Merge: e99d382 c5bdb2e  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:21:32 2016 -0400

Add print hello test caser

commit c5bdb2ef9cc92b1fb24fff937b794aadb0b49a47  
Merge: e2bb98d 707122e  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Apr 1 10:18:34 2016 -0400

Merge branch 'codegen\_hello' of <https://github.com/alextrax/PICEL> into codegen\_hello

commit 3ded196a9fd0ea141824e481efe1ee219811b58f

Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 29 17:30:32 2016 -0400

Finish first version semantic check

commit 707122ea1b4a2087967313b22c43c72303281a88  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Mar 29 16:57:35 2016 -0400

Add error message for hashtbl lookup

commit df0adcf8dc8e0edd08f3bfdcfcd989c5fc98c146  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Mar 29 16:46:42 2016 -0400

For\_init

commit fa9d683cbfc81e781d07ab97aa404f8188b97878  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 29 16:45:33 2016 -0400

Start add for init

commit 4a02a17202a002eb487a9b8ab19dff6bd5463f3a  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Mar 29 16:44:24 2016 -0400

1. Support codegen for local variable declaration and initialization
2. Use hashtbl to implement variable table

commit 8c0cef9efc355f6903f6a33f28612d0230a7ecd1  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Mar 29 16:17:01 2016 -0400

Sement v1

commit 49a6fd58d60cdcd55bcccd8386eaa30c19694965  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Mar 29 15:43:28 2016 -0400

New version for checking semantics

commit 49099d4a92ec2ef3ac9b1602e43d629df1243f84  
Merge: 2c64121 2db7ea7  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Mar 29 15:36:27 2016 -0400

Merge branch 'semant\_hello' of <https://github.com/alextrax/PICEL> into RuiLu

commit 2c64121c0bff62d7b1fa64564093696851c808da  
Merge: 6c26487 48228f3  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Tue Mar 29 15:35:41 2016 -0400

..

commit 2db7ea79244f54fb07f4d0c8bb848bd6a6f655ef  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 29 15:32:58 2016 -0400

Start editing semant.ml

commit 48228f3e56abec20fafab50893bd2d2aa08e340c  
Author: rz2337 <rz2337@columbia.edu>  
Date: Tue Mar 29 15:09:14 2016 -0400

Modification

- change notification of "Vdecl" <- No initialization in during variable declaration at "program" anymore.
- modify codegen, can set "vdecl" at global and use them now.
- modify microc.ml <-don't print ast, convenient for LLVM test.

commit e2bb98de4aa9c64c2b9985b37314f1fccbe47523  
Merge: 96c0fa6 fb50722  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 29 14:37:26 2016 -0400

Merge branch 'codegen\_hello' of <https://github.com/alextrax/PICEL> into codegen\_hello

commit 96c0fa6e723b95f9065c56663a8aceb3803eb4ec  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 29 14:37:07 2016 -0400

Pull from codegen branch

commit fb50722770199fbe3d3ac5c5aaefabbd26a77cf4  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Mar 29 14:36:51 2016 -0400

Support both print integer and string (user prints)

commit 2a0c43f4ecf53803e2a6975149732a8ca135f2be  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Mar 29 12:04:05 2016 -0400

Support print 'Hello World'

commit 54d937f820929d4a9eb43672f0bfa3ef0bc52555

Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 25 12:32:35 2016 -0400

Add some test case

commit a61eca7173baf08d0bad7a2a96975126a2e5f66  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 25 12:32:21 2016 -0400

Print integer successfully

commit 6c2648769a102faffc3878044a4d8c752e60b90a  
Merge: ef44b01 be513e2  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Mar 25 12:04:10 2016 -0400

WTF

commit 0cec38b5831d07c837703f6652aa47558384e5a3  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Mar 25 12:01:18 2016 -0400

Add statements to print llvm

commit be513e2e8e58bc75fbc7618aac62a284f7e35b9c  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Mar 25 11:55:31 2016 -0400

a small modification

commit d5fd0fd0a06a0f40259117541c0c090f81cbe21c  
Author: rz2337 <rz2337@columbia.edu>  
Date: Fri Mar 25 11:52:13 2016 -0400

try to modify code generation for "Hello World!"

commit ef44b01216fd727a2bc06961582fd4dd5091570d  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Mar 25 11:39:22 2016 -0400

Micro.ml v1

commit 252098a59310a53be1dce0dfdc763df858558fcd  
Author: RuiLu <rlu0213@hotmail.com>  
Date: Fri Mar 25 11:15:49 2016 -0400

Test for branch

commit d56d02f1a9993b4889f99ac76b1e080ed31beda4

Merge: 4ba5307 e99d382  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Fri Mar 25 10:27:08 2016 -0400

Fix the conflicts

commit 4ba530761b54f258c4004c17752ebaa991369b26  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Fri Mar 25 10:24:21 2016 -0400

Clean file first

commit 0844eadb7d97a911b89584e4f5ead37f4cae45db  
Author: alextrax <luciferxiii@gmail.com>  
Date: Wed Mar 23 00:39:04 2016 -0400

Support AST tree printing

commit 91cc8f487aa468baa2bed5a7c7cb029cf1ee21fb  
Author: alextrax <luciferxiii@gmail.com>  
Date: Tue Mar 22 22:47:11 2016 -0400

Fix function declaration grammar by adding new rule for main():

fdecl:

```
typ MAIN LPAREN formals_opt RPAREN LBRACE stmt_list RBRACE
  { Fdecl({ typ = $1;
    fname = "main";
    formals = $4;
    body = List.rev $7 })
```

commit e99d382bab8628a5ec56b2c45824b693d4d1f9b9  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Tue Mar 22 19:40:59 2016 -0400

Delete llvm folder

commit f185398ce30e9db0d1739529e056d47dfd026b99  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Tue Mar 22 19:29:01 2016 -0400

Perfect gitignore!

commit 4aadf22e459ba58f25578d88788284c5032a791b  
Author: Chris Hsu <chia-hao.hsu@aiese.net>  
Date: Tue Mar 22 19:27:57 2016 -0400

WTF

commit 9e694f8a130f481086553cddb63dc0cd9b1e14e3

Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 19:24:55 2016 -0400

Finish compiling

commit e0609825848c9e3f92bfcac0cee9eedcb27f9d4e  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 19:24:36 2016 -0400

Modify Makefile

commit c56337705f98b5ed5482c726657798297c3851fc  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 19:24:20 2016 -0400

Ignore digests, logm microc.ml in \_build

commit 72630270289e1be045b7626c3a78368d0af30a51  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 19:08:58 2016 -0400

try to compile

commit 09be9afd7caba0a7503a3958056a4b274b48aeb8  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 19:03:59 2016 -0400

Fix the conflicts

commit 29eba1a37d4849c37207896db27580c8d61c0703  
Merge: 737c18a 7ab1635  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 19:03:07 2016 -0400

Fix the conflicts

commit 7ab1635c1e70d7213cc538d6b6fa4c6b9b4d5ea6  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 18:59:55 2016 -0400

Delete .DS\_Store

commit 101a1f536d9394831a10694cb626adee212758a5  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 18:59:35 2016 -0400

Delete microc.ml.depends

commit be0d291b4fde7e32ab43e443954cb37b85870bb1



Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:59:28 2016 -0400

Delete microc.ml

commit dd145f81dffe849503319cbfa3d40ca33ab73abd  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:59:21 2016 -0400

Delete \_log

commit d0b87bf967cca6bb8348d4a99d209dee418dcb2d  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:59:14 2016 -0400

Delete \_digests

commit 737c18acbc292c5c1432859fb02a4d132a24af5c  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:45:54 2016 -0400

Final gitignore

commit 961a3ccfed187f946ece2e869a6cc9779529f564  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:45:33 2016 -0400

Move the local folder, decide to ignore in the future

commit f2234ae83da13ff02ad83c51a7c2a96ea6924d2e  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:43:12 2016 -0400

Test gitignore again

commit 2912cee52b77e5a4375a71d9a47cec75580846e2  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:35:43 2016 -0400

Test gitignore

commit 5fc2c01b13bbfb7fef9a3dde5de69252e7b9e25c  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Tue Mar 22 18:15:28 2016 -0400

Test gitignore

commit 36f43367bc44b122aac9e1b6f98607cab5c849a8  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 18:15:16 2016 -0400

Test gitignore

commit 465b6d5ee1a3498b210dd94d91d4d227337a530b

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 18:04:58 2016 -0400

Modify gitignore

commit 366f695a0c895296c89a396d8285d33671dbae51

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 17:52:19 2016 -0400

Compiled files

commit dea7a20e941a08712a12b8c6eb030f435f9db149

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 17:51:56 2016 -0400

Hello World Paring Error

commit 14678eceeac893c366a6288124c32edc4b6b3219

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 16:41:43 2016 -0400

Lalalal

commit 1ffdfbb68e82be9cd2677107f1e6fbe1c46ee948

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 16:41:35 2016 -0400

Test hello world first

commit ef554399d2d029bcf52dfe5e9d3b62b7ad77e9de

Merge: 0267a50 97df537

Author: Chris Hsu <chia-hao.hsu@aiesec.net>

Date: Tue Mar 22 16:09:39 2016 -0400

Merge branch 'master' of <https://github.com/alextrax/PICEL>

commit 97df5370d654d698bdf7c766d42438a1fcee6980

Author: rz2337 <rz2337@columbia.edu>

Date: Tue Mar 22 15:52:54 2016 -0400

change "exp" into "expr"

commit 380b8248e8a7f425058dac66a355e51596b1f3c7

Author: rz2337 <rz2337@columbia.edu>

Date: Tue Mar 22 15:44:30 2016 -0400

add "CHARLIT" and "STRINGLIT" to exp in parser.mly

commit 0267a50728a38cabdd4b923a5cba94dd8c9dbe85

Merge: 27851ee 4d13bf0

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:40:38 2016 -0400

Include the right path for \_build

commit 4d13bf08ddc8ee8f8c04b15719211786516fced8

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:32:24 2016 -0400

Compile again

commit c316d4be9390eb193f50431bfee7ce70dd1343a1

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:31:59 2016 -0400

Finish modify microc-llvm

commit 68fa3016a4197293e1e8e0246704eb81f1750dc4

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:28:23 2016 -0400

Include microc-llcm/\_build

commit 27851ee4cb4244049dd3a444271dc03df6e40c67

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:19:25 2016 -0400

Modify gitignore

commit de21938412d129366cb876f2d2b236f7ef27d116

Merge: 531d731 a86bc11

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:12:59 2016 -0400

Modify gitignore

commit a86bc115f03ad9140e4d900a9cb577a1e99a6d52

Author: Chris Hsu <chia-hao.hsu@aiasec.net>

Date: Tue Mar 22 15:11:28 2016 -0400

Comment all content first to develop Hello Word support only

commit 5da759149f2f3043213761664e876ca1b710967e

Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 15:10:43 2016 -0400

Compile successfully

commit ff63e1f6c6813b41bb45b2714dc08c3d4a8b6022  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Tue Mar 22 15:10:09 2016 -0400

Finish building

commit 9effb55da86f295fa95bb1573f014549e27c6d05  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:52:25 2016 -0400

Add more ignore cases

commit 1794a1e29ebf2653fb14440f014d0d6b55cb4980  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:51:01 2016 -0400

Modify the scanner to test CHAR case

commit 531d7310731ad3f09714198c8ceb1160bec9b9c8  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:31:50 2016 -0400

Add comments

commit c61226dbca57ffc6a1f5adf30d2973b41a6f17a9  
Merge: 96f8745 47f246f  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:24:42 2016 -0400

Delete .DS\_Store

commit 96f8745fff74412a330b7c3acde0513da7f94af4  
Merge: 34e1a3f e2ec6c5  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:23:40 2016 -0400

Fix conflicts

commit e2ec6c5683b26f42dfaa35c42b38ff3b73522237  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:11:57 2016 -0400

Make scanner successfully

commit 08686fcc2f9a030e239b48bdbcae3aa7ad3163cc  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:11:31 2016 -0400

Add parser & test ignore

commit 4076e5f8572a73982bb9047d1506b2fff3f751a0  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:06:11 2016 -0400

Add test output ignore

commit e1da4f87ee8d19a6d5c1a22c597d20a79d5a8900  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 18:05:36 2016 -0400

Make scanner successfully

commit d0418eb49d86f2d8356dec0f09ef91b84d0bc2fd  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 14:45:08 2016 -0400

Add ast.ml

commit 47f246f41d5e9189d854f6018f0199556549bc0a  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 14:22:21 2016 -0400

Delete .DS\_Store

commit 34e1a3fb0e78594e929787ee9b4e6140a557f3cf  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 14:21:39 2016 -0400

Delete llvm & test folder

commit 324005f5b7671257115be01479b6de614b8e6578  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 14:18:36 2016 -0400

Add more cases

commit 7fb5b6a7dc7831c76f0379763139a1e7cd5ec0fa  
Author: Chris Hsu <chia-hao.hsu@aiasec.net>  
Date: Fri Mar 18 14:13:25 2016 -0400

Add tests ignore cases

commit b6660cc7fecc672f2c8d4f15c60df6e800573d99

Merge: 7859802 ddd9473  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 18 14:12:23 2016 -0400

Fix conflicts in Makefile

commit 7859802ce00abe9bfaf35459c4defef143726a3d  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 18 14:09:59 2016 -0400

Add gitignore

commit e6535790d90bef42889c271099b1de8b41ed0eb1  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 18 14:03:45 2016 -0400

Paste scanner.mll

commit 8a182c82489ae531116f01e9367f36583c068f07  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 18 13:15:11 2016 -0400

Finish the llvm compiling

commit ddd9473f1b5104e78e85ac591095c02649bd8d66  
Author: alextrax <luciferxiii@gmail.com>  
Date: Fri Mar 18 10:44:29 2016 -0400

update Makefile

commit 29d5899222d2b1668ac2f34a60fff103bfce2841  
Merge: aecbb6f 12b799d  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 11 13:29:07 2016 -0500

Merge branch 'master' of <https://github.com/alextrax/PICEL>

commit aecbb6fda0261d173e36c54f6f47d7a35a4bcd13  
Author: Chris Hsu <chia-hao.hsu@aiesec.net>  
Date: Fri Mar 11 13:28:39 2016 -0500

Initial commit

commit 12b799d5fca4c1d37a83b536f7293b3ddd9f59ad  
Author: alextrax <luciferxiii@gmail.com>  
Date: Thu Mar 10 21:11:34 2016 -0500

First Commit