Go-- Programming Language

Programming Languages and Translators Final Report

Spring 2021

Chen Chen, Yuyan Ke, Yang Li, Arya Lingyu Zhao

cc4351, yk2822, yl4111, lz2650

Contents

1 Introduction

Go-- is an imperative, statically typed language with C-like syntax and support for concurrency. Inspired by Golang, Go-- also lightweight concurrency with gofunction, and uses channels for communications in between functions. Gofunction and channel enables the users to write concurrent programs with fewer lines of code, neater design, and provides better readability.

1.1 Background

C has long been the fundamental languages for lots of programming learners above the intro level. Yet, using the concurrency in C involves the understanding of pointers and memory management. Goroutine makes it much more user-friendly for writing concurrency in everyday programming. Goroutine is a lightweight thread, and the users don't need to understand pointers or manage runtime memory to use it. However, the overall syntax of Golang is quite different from C.

1.2 Goals

1.2.1 Familiarity

Go-- syntax is very similar to C language. C programmers can easily learn the syntax of Go-- and use it seamlessly. Go-- adds no additional difficulty to the learning curve. As for the concurrency features of Golang. The goroutine, channel and gofunction are syntactically similar to Golang.

1.2.2 Lightweight Concurrency

Go-- provides the lightweight concurrency, including features like channel, goroutine, and gofunction. Compared to C, the syntax of concurrency features are easier to understand, remember, and debug.

1.2.3 Hide Pointers and Memory Management

Go-- users don't need to understand pointers or memory management to use the concurrency. It's a deep relief for lots of programming learners. Using Go--, users can write less and more succinctly.

2 Go-- Tutorial

2.1 Environment Setup

Clone from Go-- git repo and go to the corresponding folder

> git clone https://github.com/keyuyan1145/Go--.git

> cd Go--

Run docker image

> docker run --rm -it -v `pwd`:/home/gmm -w=/home/gmm columbiasedwards/plt

2.2 Compilation Guide

Next, compile the Go-- compiler using the following commands. This will automatically run Go--'s test suite as well

Compile the Go-- compiler

> make all

Run all test cases

> ./testall.sh

Or you can compile the compiler, and then run the test suite in separate steps using the following command in Go-- folder:

> make

When you run either of the two sets of commands above, all the tests that came with the compiler will be run and should pass. If they do not all pass, go back to section 2.1 Environment Setup and ensure that your environment is set up correctly.

If you want to run a single test case in the tests folder:

> ./misc/run.sh ./tests/<name_of_test>.gmm

And to remove the generated files from compiling and running tests

```
> make clean
```

To compile and run a simple program

> ./misc/run.sh ./path/to/<name_of_program>.gmm

And to remove the generated executable and intermediate files

> make clean

2.3 Language Tutorial

We go through the steps here to write, compile, and run a simple program.

sample.gmm

```
function int add(int x, int y)
{
  return x + y;
}
function int main()
{
  print( add(17, 25) );
  return 0;
}
```

```
> ./misc/run.sh ./sample.gmm
```

The program above will output

```
sample...

###### Testing sample

./gmm.native sample.gmm > sample.ll

llc -relocation-model=pic sample.ll > sample.s

cc -o sample.exe sample.s -pthread ./src/builtin.o

./sample.exe

42
```

3 Language Reference Manual

3.1 Comments and Whitespace

3.1.1 Comments

The characters /* introduce a comment, which terminates with the characters */. Comments can be nested as long as the opening /* and closing */ are all matched.

```
/* single-line comment */



/* multi-line comment
```

```
   multi-line comment*/



/* nested /* multi-line */ comment */
```

Regular expressions for comments accepted by Go--

COMMENT = "/* [^ */]* */ "

3.1.2 Whitespace

Whitespace, including newline characters, tabs, and spaces, is used only to separate tokens and is otherwise ignored by our Go-- compiler.

3.2 Data Type

3.2.1 Primitive Data Types

This language has primitive data types including int, bool, float, char and string. Unlike C, this language doesn't have pointers.

3.2.1.1 Integer Types

An integer constant is a sequence of digits. Each integer is 4 bytes.

Integer operations are

```
 ==, < , >, !=, +, -, *, /,  <=, >=
```

Example for declaring a string

```
int foo;

foo = 2;
```

3.2.1.2 Float Types

8-byte data type that represents fraction and floating point numbers in scientific notation.

Float support operations are

```
==, < , >, !=, +, -, *, /,  <=, >=
```

Example for declaring floating point numbers:

```
float foo;

foo = 4115.2;
```

3.2.1.3 Boolean Types

1-byte data type that represents logic values and can has value true and false. Bool supports operations ! (NOT), || (OR), and && (AND).

Example for declaring boolean:

```
bool foo;

foo = true;
```

3.2.1.4 String Types

Data type that contains string literal values. Strings are immutable.
String support operations: +, ==,!=, where string comparisons are based on their values.
Example for declaring a string:

```
string foo;

foo = “hello 4118”;
```

```
foo = “hello ” + “world”; /* the value of foo is “hello world*/
```

3.2.2 The Void Keyword

The type void has no associated value and can only be used as the return type for functions that returns nothing. This is useful for functions which are intended to perform “side-effect” operations only. The return statement can be omitted in this case or can be written as:

```
return;
```

3.3 Variables

3.3.1 Variable Naming and declarations

All variable names must follow [a-z A-Z][a-z A-Z 0-9]* and cannot be any of the reserved words listed below.

Use	Reserved Words				
booleans	true	false			
control flows	if	while	else	for	
data types	int	float	string	bool	
functions	function	gofunction	go	void	return
data structures	array	struct	channel	structdef	new

3.3.2 Scope of Variables

Go-- is a statically scoped language. Variables declared inside functions and struct definitions exist only inside the block in which they are declared, and override any variables of the same name declared before that function within that function only. Variables outside of all functions and struct definitions have global scope and thus can be accessed anywhere in the program following their declaration. Multiple variables and/or functions of the same name, even if their types differ, cannot be declared in the same scope. Variable, struct, and function declarations are not visible to statements that precede them so Go-- does not support recursive or mutually recursive functions or struct type definitions.

3.3.3 Variable Declaration and Assignment

Variables must be declared with a type and a name in the form of

```
type ID;
```

Where :

type: Go-- data types and data structures

ID is the variable name in form string that are accepted by regular expression:

```
[a-z A-Z][a-z A-Z 0-9]*
```

Variables assignments must follow the convention shown above with a type followed by a proper variable name. The variable declaration statement must terminate with a semicolon. Variable types can be any of the types listed in Section 3.2.1, 3.5, 3.6 and 3.7.

Global variables must be declared before function definition and after struct definitions (section 3.7)

3.4 Statement

Statements are executed in sequence.

3.4.1 Expression Statement

Expression statements include assignments and function calls take the form of

```
expr;
```

And all regular expressions for expression statements accepted by Go-- are listed below:

```
expr:
```

```
| expr INC
| expr DEC
| expr PLUS   expr
| expr MINUS  expr
| expr TIMES  expr
| expr DIVIDE expr
| expr EQ     expr
| expr NEQ    expr
| expr LT     expr
| expr LEQ    expr
| expr GT     expr
| expr GEQ    expr
| expr AND    expr
| expr OR     expr
| expr QUEUE expr
| expr QUEUE
| MINUS expr %prec NOT
| NOT expr
| expr ASSIGN expr
| NEW LPAREN ARRAY LT typ GT LSQUARE expr RSQUARE RPAREN
| NEW LPAREN CHANNEL LT typ GT LSQUARE expr RSQUARE RPAREN
| NEW LPAREN styp COMMA args_opt RPAREN
| ID LPAREN args_opt RPAREN
| GO ID LPAREN args_opt RPAREN
| LPAREN expr RPAREN
| accessor
```

Where accessor stands for string,float and int literals; struct and array access (section 6,7) and access to variable values via variable ID :

```
accessor:
```

```
| accessor LSQUARE expr RSQUARE
| accessor DOT ID
| atom
```

3.4.2 Compound Statement

Compound statements are organized into blocks within braces {}, such that an open brace must be matched with a corresponding a closing brace in the form of

```
stmt_list{
        stmt;
        stmt;
        ...
}
```

* stmt and expr stands for statement and expression in the sections below

3.4.3 Conditional Statement

The two forms of conditional statements are

```
if (expr) {stmt_list}
if (expr) {stmt_list} else {stmt}
```

In all cases, expression is evaluated first and the corresponding statement block will execute if the expression results in a non-zero value. If the result of the expression is zero and there exists sequential else if blocks, then the following expression for the else if block will be evaluated the same way as the original if block. This process continues for each else if block in a sequential order. If the result from the expression evaluation is zero and the else block follows next, then the statements within the else block will be executed.

3.4.4 While Statement

The while statement has the form of

```
while (expr) stmt_list
```

The statement within the block is executed repeatedly while the expression is evaluated to be true or non-zero. The expression is re-evaluated after each iteration of the execution of the statement.

3.4.5 For Statement

The for statement has the form of

```
for (expr1;expr2;expr3) stmt_list
```

equivalent to:

```
expr1;
while (expr2) {
        stmt_list;
        expr3;
}
```

such that expression1 denotes the starting value for the loop, expression2 denotes the test made after each iteration, and expression3 denotes an incrementation performed after each iteration. The loop terminates when expression2 evaluates to be zero.

expr1 and expr3 are optional, expr2 is required.

3.4.6 Return Statement

Return statement taken one of the following forms

```
return expr;
return;
```

In the first case, the expression is returned to the caller of the function. In the second case, no value is returned. Since the return expression must be the same type as specified in the function declaration, the no value in case two will be returned as a null object for the specified type.

3.5 Channel

3.5.1 Channel Data Structure

Channel is a special data structure featured in our language meant for communications between computations, especially those executed on different threads. There are two main components in a channel, the first one is a circular array of primitive types or user-defined structs to hold information to be communicated between functions, the second one is a counting semaphore and mutex to guarantee data integrity and consistency of the circular array during enqueue and dequeue.

3.5.2 Channel Creation

Channel variables must be global variables:

```
channel ID;
```

Channel values can be instantiated by new keywords and statement:

```
new(channel<type>[expr]);
```

where type could be any of the primitive types or user defined structs , and expr should evaluate to an integer, and specifies the maximum number of items of matching data types that could be simultaneously stored in the channel. The statement could be called from anywhere (main or other functions), and the naming of which should be unique. The declared channel should be accessible by name anywhere within the program.

3.5.3 Enqueue into channel (->)

One may enqueue data into a channel with matching type using the right arrow (->) operator, with the name of the channel on the right-hand side of the -> operator and the variable to be enqueued on the left. If the channel is at its maximum storage capacity, the thread trying to enqueue data into the channel will be blocked on the enqueue statement, until process termination or there is vacancy in the channel. Note that the enqueue operation is protected by a mutex in the channel such that there could be at most one thread accessing the channel data region at any given time. The channel enqueue statement could be expressed as

```
expr -> CHANNEL_ID;
```

Where expr is the value that is being enqueued and ID stands for the name of an instantiated channel object. The type of the expression must match the data type of the channel at instantiation, for example, we can have :

```
channel some;
int i;
...
i = 4118;
some = new(channel<int>[5]);
i -> some;
```

That enqueues int i into an int channel some of size 5.

3.5.4 Dequeue from channel (->)

One may dequeue data from a channel with the ->operator, with the name of the channel on the left-hand size of the -> operator. Only one item of matching data type could be retrieved and removed from the channel with one -> operation. If the channel is empty, the thread trying to dequeue data from the channel will be blocked on the -> statement, until process termination or there is new data enqueued into the channel. Note that the dequeue operation is protected by the mutex in the channel such that there could be at most one thread accessing the channel at any given time. The variable name on the right-hand side could be omitted when the value of the dequeued item is not evaluated, or directly fed into another function. i.e. Go-- can have:

```
CHANNEL_ID -> expr;
CHANNEL_ID -> ;
```

where expr is usually a variable ID that the programmer wants to assign the dequeued value to. Following the code block in 3.5.3 we can have:

```
i = 4115;
some -> i;
```

And i now will have the value 4118.

3.6 Array

Arrays are containers, with a fixed size to group a number of items of the same type, primitive or a composite type defined by a struct.

3.6.1 Declaring Array

Declaration of arrays need to be in the following form

```
array<type> ID;
```

Where type is the data-types can be contained in the array and ID stands for the variable name of the declared array.

3.6.2 Instantiating and Indexing Array

Array can be instantiated using the keyword new and statement:

```
new(array<type>[expr]);
```

where type could be any of the primitive types or a user-defined struct , and expr should evaluate to an integer, and specifies the maximum number of items of matching data types that could be stored in the array. The statement could be called from within functions (including return statements). Notice that Array elements are not initialized and automatically set to 0. For example;

```
array<int> some;
int i;
some = new(array<int>[5]);
```

Initialize and allocate an int array of size 5 in the memory.

To access elements in the array, we can use

```
ARRAY_ID[expr]
```

such that ARRAY_ID is the name of the array and expr should be of integer type and specifies the desired index of the element, the accessing operation can be both lvalues and rvalues, for example following the code block on page we can have

```
some[0] = 4115;
i = some[0];
```

And now i will have value 4115;

3.7 Structs

3.7.1 Struct Declaration

A struct is a data container type that can contain named fields of primitive type datas which could be any primitive type supported by Go--.Note that we do not allow anonymous declaration of nested structs. In other words, all structs, no matter nested or not, need to be named. Structs are defined with "structdef" keyword followed by the structtype ID, which indicates the name of the struct type. The fields declaration in the struct definition should be binds of type and string, and field names must start with a letter followed by a combination of letters and numbers. Field names need to be unique within the same struct. All field declarations are encapsulated in a pair of braces {}. Struct definition should be before global variable definition and function definitions. Initialization of fields is not allowed during struct declaration. Here is a sample declaration:

```
structdef coordinate {
        int x,
        int y,
};
```

And all struct definitions are accepted are in the form of regular expression, where decl_opt either empty or a list of (type, string) binding for fields:

```
STRUCTDEF ID LBRACE decl_opt RBRACE SEMI;
```

3.7.2 Struct Variable Declaration and Instantiation

One has to declare a struct before creating an instance of it. Otherwise, the compiler would complain about missing type definition when processing the code. The name of the struct is used as the type name of the struct. One can simply declare a struct using the convention

```
struct STRUCTDEFID ID;
```

Where STRUCTDEFID is the name of the struct type defined in struct definition, and `ID` is the name of the declared variable. One has to declare a struct before creating an instance of it. Otherwise, the compiler would complain about missing type definition. Struct initialization is done by using the keyword new and statement:

```
new(struct STRUCTDEFID,expr1,expr2...);
```

Where exprN stands for the value that is assigned to fieldN (sequentially) of the struct initialized, the type of expr must match the type of field in the struct definition. All fields need to be provided.

3.7.3 Reading and Updating Struct Fields

Struct fields could be accessed with the dot operator (.). One would need the name of the struct variable, followed by the dot and then by the name of the field to get the value. Incorrect variable name or field name would result in a compilation error. If a field is not initialized, the type default value would be returned. Same as array access, the dot access of a struct can both be lvalue and rvalue. For example, 0 will be returned if the uninitialized field is of type int. Here is a short example with the same struct definition as specified in 3.7.1 and 3.7.2.

```
struct coordinate point;
int i;
point = new(struct coordinate,1,2);
i = point.y;
```

Now i will have value 2.

3.8 Arithmetic Operators

3.8.1 Order of Evaluation

Our arithmetic operators follow the PEMDAS convention. In other words, () takes priority over *, / and %, over + and -. All operators except the NOT operator (!) are left associative; NOT(!) is right associative. Dot operation has the highest precedence.

3.8.2 Addition (+) and Subtraction (-) Operators

We allow add (+) between two variables of type int or float, or two expressions that would yield int or float. Note that the expressions on the two ends of the operator need to be of the same type, otherwise a compilation error would be thrown. There is no automatic type promotion from int to float. The minus operator also only accepts two variables of type int or float, or expressions that evaluate to int or float, and is used for subtraction in the traditional mathematical sense. We also allow ++ and –- as shorthand for +1 and -1.

```
int a;
int b;
float c;
string h;
string t;
a = 1;
b = 3;
c = 2.0;
a+b;/*int 4*/
a+c;/*error*/
b-c;/*error*/
a++;/*int 2*/
c--;/*error*/
h = “hello ”;
t = “ocaml”;
h+t;/*”hello ocaml”*/
```

3.8.3 Multiplication (*) and Division (/) Operators

The multiplication (*) and division (/) operators are used in the traditional mathematical sense for ints and floats only. When performing int-int division, the quotient is kept and the remainder

is discarded, which is the equivalent of rounding down. If at least one of the numerator and the denominator is of type float, the result would be a float.

```
int a;
int b;
float c;
a = 1;
b = 3;
a*b;/*int 3*/
a*c;/*error*/
b/a;/*int 1*/
```

3.8.5 Boolean Operators (<, >, <=, >=, !=, ==)

Boolean operators in this language are: == , !=, <, >, >=, <=. They operate on ints and floats only. There is automatic type promotion when comparing ints and floats so they can be compared without any problems. == and != can also be used for booleans and boolean expressions that evaluate to true or false and string values to compare their values.

3.8.6 Logical Operators (!, &&, ||)

! is used for NOT in boolean expressions.

&& is used for AND in boolean expressions.

|| is used for OR in boolean expressions.

3.9 Functions

Go-- supports two types of functions, normal functions and functions that can run in concurrent manners.

3.9.1 Function declaration

To define a function, we will have the key word of 'function' or 'gofunction' at the beginning of the definition, and it should be followed by the return type of the function, return type void is allowed in Go--, and after the return type programmers should specify the function name followed by parentheses in which function arguments are specified. Function arguments should follow the format "type name" and multiple function arguments should be separated by comma. A function takes no argument when empty parentheses are present. And inside the braces '{' , '}' programmer should put in the function body.

The function below is an example function called "foo" that takes two integer as arguments and returns their sum as results.

```
function int foo(int x,int y){
        int ret;
        ret = x + y;
        return ret;
}
```

All programs have regular expressions:

fdecl:

isGo typ ID LPAREN formals_opt RPAREN LBRACE vdecl_list stmt_list RBRACE

Where isGo is the keyword function/gofunction, for normal function is function ,typ is the return type of the function and formal opts stands for the argument type list, vdecl_list stands for local variable declarations and stmt_list stands for the function body.

To call the functions in the program, programmers only have to specify the function name and the function arguments, and the argument types should match the ones in the function argument.

For example, to call foo on 1 and 2, one should use the syntax:

```
foo(1,2);
```

3.9.2 Concurrent Functions

To define a concurrent function, users should use the keyword ‘gofunction’ instead of ‘function’ in the function definition to specify that the function can run in a concurrent manner. For example

```
gofunction int foo(int x,int y){
        int ret;
        ret = x + y;
        return ret;
}
```

is a concurrent function called “goo” that takes two integer as arguments and return their sum as results. Unlike normal functions, when calling concurrent functions, the key word go should be specified. For example, to call goo in the program, one should use the syntax

```
        go foo(1,2);
```

And because the gofunctions can run in a concurrent manner. Code like

```
        go foo(1,2);
        go foo(3,4);
```

Runs in a concurrent manner. That is, function in line 2 can finish execution before line 1 finishes execution, depending on the underlying scheduling architecture of the operating system.

* Gofunctions only support up to three arguments and only support void return type for the time being.

3.9.3 Built-in functions

Go-- contains three built-in functions “print”,“printf”, “printb”, and “prints”. The functions have return type void and are normal function that are used to print int, float, boolean, and string, their usage are shown as below

```
        print(1);
```

```
        printf(411.50);
        printb(true);
        prints("ocaml is the best");
```

3.10 Sample Program

The program below shows a map reduce program on word counting using gofunctions and channel in Go-- that demonstrates the main feature of the language:

```
structdef kv{
   string k,
   int count,
};

channel<struct kv> mapper;
channel<struct kv> reducer;
array<string> keys;
int num_keys;

gofunction void search(array<string> arr, int start, int range)
{
   int i;
   int j;
   array<struct kv> ret;

   /* setup the return structure */
   ret = new(array<struct kv>[num_keys]);
   for(i = 0; i < num_keys; i++){
      ret[i] = new(struct kv, keys[i], 0);
   }
```

```
    /* word count */
    for(i=0; i < range; i++){
        for(j = 0; j < num_keys; j++) {
            if(arr[start] == keys[j]){
                ret[j].count++;
                /* prints(keys[j]);
                print(ret[j].count); */
            }
        }
        start++;
    }
    /* pass return values through channel */
    for(i=0; i < num_keys; i++) {
        ret[i]->mapper;
    }
}

gofunction void merge(int num_reduce)
{
    int i;
    int j;
    array<struct kv> counts;
    struct kv tmp;
    counts = new(array<struct kv>[num_keys]);

    for(j=0; j<num_keys;j++){
        counts[j] = new(struct kv, keys[j], 0);
    }
```

```
    /* reduce */
    for(i=0; i< num_reduce; i++){
        mapper->tmp;
        for(j = 0; j<num_keys; j++){
            if(tmp.k == counts[j].k){
                counts[j].count = counts[j].count + tmp.count;
            }
        }
    }

    /* signal to main thread */
    for(j = 0; j < num_keys; j++){
        counts[j] -> reducer;
    }

}

function int main()
{
    array<string> words;
    int i;
    int j;
    int range;
    int threads;
    int gap;
    struct kv tmp;

    range = 1000000;
    threads = 16;
    num_keys = 2;
```

```
words = new(array<string>[range]);
keys = new(array<string>[num_keys]);
mapper = new(channel<struct kv>[threads]);
reducer = new(channel<struct kv>[1]);

keys[0] = "waldo";
keys[1] = "other";

/* fill the array with values */
for(i=0; i < range; i++)
{
   words[i] = keys[1];
}

words[range/10] = keys[0];
words[range/2 + 7] = keys[0];
words[range/5 + 33] = keys[0];
words[83] = keys[0];
words[334] = keys[0];

/* mapper */
gap = range / threads;
for(i=0; i<threads; i = i+1)
{
   go search(words, i*gap, gap);
}

/* reducer */
go merge(threads * num_keys);
```

```
  for(i = 0; i < num_keys; i++){
    reducer->tmp;
    prints("----------");
    prints("word: " + tmp.k);
    print(tmp.count);
  }
}
```

4 Project Plan

4.1 Planning, Specification, Development, and Testing

We met twice a week to check in on our progress and make sure that everyone was on the same page for the next step. One of our weekly meetings is scheduled to be right before our weekly meeting with our T.A., Xijiao Li, so we could consolidate a list of questions to ask during the TA meeting. At the beginning of the semester, we were very ambitious with our language design to include features of concurrency and first-class functions. As the development progresses, we narrowed down the scope of our language to focus on concurrency, specifically the implementation of gofunction and Golang-like channel. During our team meetings, each member of the group would provide an update on what they were working on currently, and collectively, we discussed our next step before the next meeting. We also discussed the finer details of the syntax and semantics of our language and helped each other troubleshoot any issues or bugs that we were having. Additionally, we discuss possible test cases for each feature that is added to ensure that everything was working as we expected from the beginning. Finally, throughout the week, we communicated over Wechat for questions about the language implementation.

4.2 Project Timeline

In developing a timeline, our philosophy was to first implement the central feature in our language, which is concurrency and channel, and then implement additional data structures such as arrays and structs at the end. In practice, we realized that struct and array were actually needed by concurrency and channels, so we implemented them concurrently with the main features. This

allowed us to constantly check that our implementations of other features worked correctly within concurrent threads.

4.2.1 Planned Timeline

(Note: timeline is organized in accordance with the completion of each part.)

February 3: Project Proposal

February 24: LRM, Parse, Scanner, Ast

March 24: Hello World Due

March 31: Gofunction implementation

April 7: Channel

April 17: First-class functions

April 20: Additional data types

April 23: Project reports, slides

4.2.2 Actual Timeline

(Note: timeline is organized in accordance with the completion of each part.)

February 3: Project Proposal

February 24: LRM, Parse, Scanner, Ast

March 24: Hello World Due

March 31: Gofunction implementation, Struct, Array

April 7: Channel, String

April 22: Concurrency with a threadpool

April 23: Project reports, slides

4.3 Team Roles and Responsibilities

Yuyan Ke, Manager

Chen Chen, System Architect

Yang Li, Language Guru

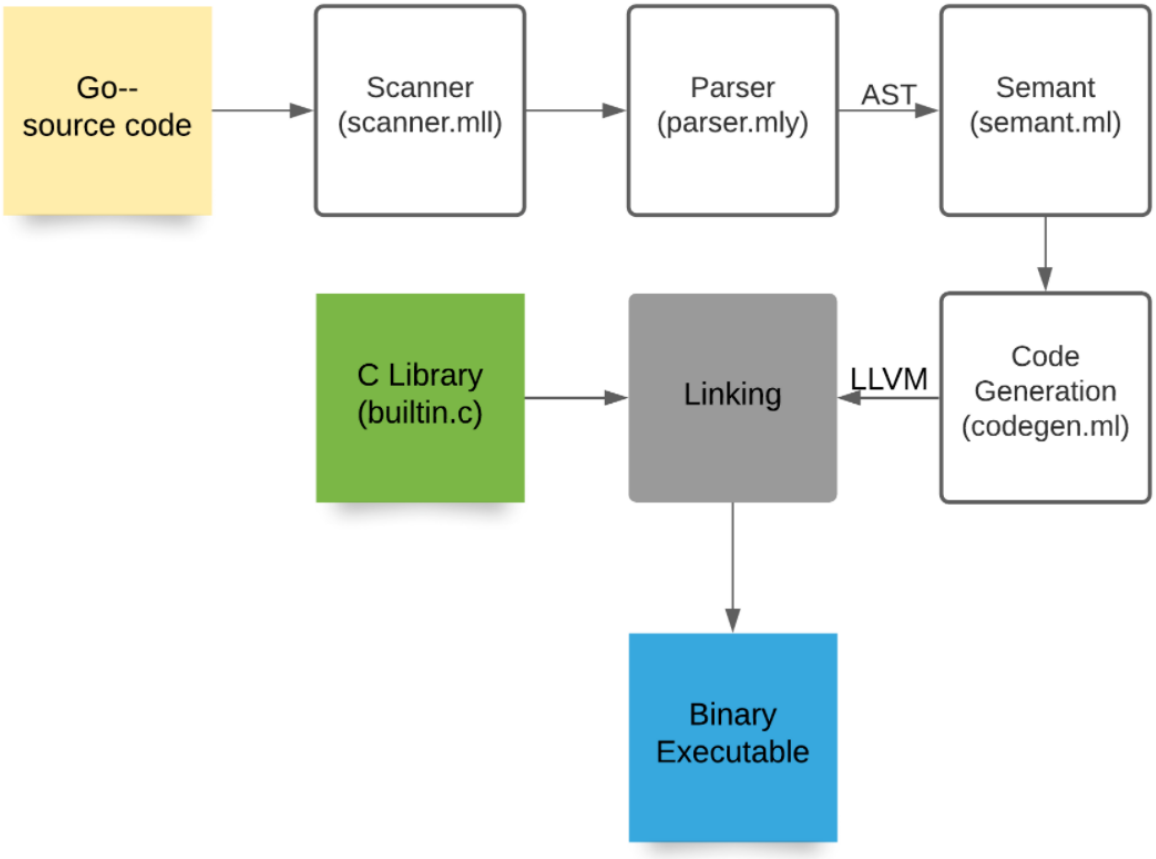
Arya Zhao, Test Designer

Every member of the team touched every file in the compiler and contributed to the test suite as well as the writing of the Language Reference Manual and Final Report.

Scanner, Parser, AST	Arya, Chen, Yang, Yuyan
Struct, String	Yang
Array	Chen
Gofunction	Yang, Chen
Channel	Chen, Yuyan
Testing	Arya (lead), Chen, Yang

LRM, Report, Slides	Arya, Chen, Yang, Yuyan
Project Planning	Yuyan

5 Architecture Design



5.1 Scanner

File: scanner.mll

The scanner takes in a Go-- source program of ASCII characters and translates them into tokens. If any characters in the code are detected to be illegal, lexing errors will be thrown. Characters inside the commented blocks (/**/) will be ignored. The scanner produces tokens that will then be used by the parser in the next step of compilation.

5.2 Parser and AST

File: parser.mly, ast.ml

The parser converts the tokens from the scanner to an abstract syntax tree (AST) based on Go--'s context-free grammar rules for syntax described in the Language Reference Manual. If any violations are detected, parser errors will be thrown.

5.3 Semantic Checking

File: semant.ml, sast.ml

The semantic checker (semant.ml) recursively traverses the AST and converts it to a semantically checked abstract syntax tree (SAST) consisting of objects. Since AST types are generated directly from the user program, which might incur errors, the semantic checker ensures each object in the program makes sense syntactically and conforms to the rules of the language. A string map is used to map a string identifier to an object. If there are typing or scoping errors, messages will be printed to indicate the type of errors. For example, if variables are referenced before initialization or assigned a different type than what was declared, the semantic checker will generate errors. SAST contains a list of semantically corrected data types that have passed the semantic checker and could be used safely by the code generator.

5.4 Code Generation

File: codegen.ml

The code generator interprets and converts Go-- specific language syntax and features into LLVM codes, a lower level machine code that the computer could understand and process during the compilation process. Built-in functions are also defined and specified to be linked with their implementations. All the Go-- specific SAST types, including data types and functions, are converted to their corresponding types in the LLVM land which could be further processed into assembly codes.

6 Design Choices and Language Revolution

In this section, we include the thinking process behind some of the decisions made as we develop the language, and how the key features have evolved over time.

6.1 Gofunction

For Go--'s gofunction implementation, we have chosen the C pthread library and use a 16-thread threadpool implementation to handle gofunction calls. The threadpool is initialized upon first gofunction call, and each gofunction call broken down into a function pointer and arguments pointers is a payload to the threadpool. Our threadpool is built using linkedlist implementation for task queue, pthread conditional variables for signaling and pthread mutex to protect the task queue. The task queue uses a first-come-first-serve policy and there is no priority given to any particular gofunction call. We have referenced one of the homeworks from COMS 4118, OS I, for implementation.

We started off with a vanilla pthread implementation of gofunction, where pthread_create() is called every time a gofunction is called and the thread is detached immediately after. However, this potentially wastes system resources once a large number of gofunction calls are made, which goes against our goal of making the language usable for performance critical and multi-threaded tasks.

We choose 16 threads as the default threadpool size to match with the number of logic CPUs on the testing server in order to maximize CPU utilization and avoid unnecessary overhead. For future work, we would like to dynamically determine the threadpool size based on server architecture.

In codegen.ml, we break up the function pointers and the arguments, group the arguments by type, and reassemble it into a LLVM struct that is subsequently passed into a C function. In C, we allocate space for the arguments passed in from LLVM, and perform the gofunction call. Initially, we didn't sort the arguments by type in LLVM, which has lead to incorrect indexing when mixing int and float as arguments to gofunction calls. We solve this by sorting the arguments by type in codegen.ml and passing pointers to the grouped arguments into C.

Gofunction calls support up to three arguments, which could be any combinations of {int, bool, string, float, array, struct}. We do not allow return values from gofunction because waiting for the return values defeats the purpose of running a task asynchronously. We have also briefly considered using futures, but eventually decided to proceed with channel communications as an alternative to having return values.

6.2 channel

For Go--'s channel implementation, we recognize the similarity between buffered channel (our use case) and the classic producer-consumer problem. As a result, we use a counting semaphore implementation for data integrity, and a circular array implementation to store the data in channel. The channel follows a FIFO pattern and always dequeue the oldest data in the buffer.

The channel data structure is defined as a LLVM struct of {void*; i32_t; mutex_t; sem_t} where mutex_t is a {i64_t, [i8_t * 56]}*, and sem_t is a {i64_t, [i8_t*24]}*. The first field void* is a pointer to the circular array; second field i32_t is the channel capacity that is needed to implement the indexing for circular array; sem_t and mutex_t are used for the counting semaphore.

The initialization of channel is done in codegen.ml. For enqueue and dequeue operations, an LLVM struct is assembled in codegen.ml to pass relevant information to the C built-in functions, where the logic is implemented. We choose to implement the logic in C because it is easier to compare integers and do branching in C than in codegen, which is necessary for circular array index update.

We design the channel to be blocking when trying to dequeue from an empty channel, or to enqueue into a full channel to compensate for the lack of return value from gofunction calls. The user is encouraged to use channels to pass values between computations, instead of using global variables.

We recognize the potential of using channels for inter-process and inter-server communications, and would like to implement a UNIX domain socket version of channel for inter-process communications and potentially a TCP-socket based implementation for inter-server communications for future work.

7 Testing

7.1 Pretty Print

We had two phases of testing. Before working on code generation, we worked on the parser and the semantic checker, and pretty printing test is performed for the parser and semantic checker to make sure the source code fed to the scanner was parsed correctly by the parser, and would be correct when we got codegen up and running. These tests were kept in ast.ml and sast.ml. However, they are no longer maintained, since we migrated to integration testing once we started to work on code generation.

7.2 Integration Tests

The second phase of testing happened once we implemented code generation for our language. In this phase, we wrote an end-to-end integration test suite to rigorously test the functionality of our language as we implemented it. The test suite also ensured that we did not break previously working features when we added new ones.

For the integration tests, all of the tests that should successfully produce output follow the

naming pattern of using test-*.gmm for the test program and then using the same name but different file extension of the form test-*.out for the expected output of the program.

All of the negative tests similarly used fail-*.gmm for the test program that should fail and used fail-*.err instead of .out for the expected error message.

A complete set of tests can be found in the Appendix section. There are over 80 negative tests, and over 130 positive tests.

These tests were implemented mostly in parallel with the functionality they tested and so they were chosen to test the implemented functionality and failure cases of the implemented functionality. We have also implemented additional tests when reviewing our language reference manual.

7.2.1 Test Automation and Scripts

For our integration tests, we used a modified version of the testall.sh script provided by Professor Edwards for the MicroC compiler. It prints a line to the screen once it finishes running the full test suite summarizing the number of passing test cases. The script works by running all of the test-*.gmm and fail-*.gmm programs and then comparing the outputs or errors respectively to see if they match the expected value. If the output or error fails to match what is expected, the script will print out FAILED next to the test name along with an error message. A more detailed error message can be found by looking in testall.log. If the test is successful, the script prints a '.' instead of printing the test name. Only failing test names are printed. This way the script output is less verbose and easier to read.

7.2.2 Concurrency Test

Due to the concurrency feature of Go--, the original testing script testall.sh doesn't apply exactly to our testing programs. How the original testing script works is that it automatically runs each test file, and compares the actual output with the expected output file. However, due to the concurrency of our gofunction, often the actual output and the expected output have the same contents but are out of order.

For example, the test program below tests the concurrency of gofunction. The gofunction "twoInt" is defined; it takes in two integers, prints numbers from 0 to 99, and then writes a boolean to the channel called "sig". This gofunction is called twice in the main function. To prevent the main function from exiting too early before all gofunction finishes, we read from the channel "sig" twice, so that the main thread will keep waiting for the channel signal from the gofunction. As a result of concurrency of gofunction "twoInt", the output will be some combinations of two times sequence 0 to 99. In short, the output numbers alternate.

```
channel<bool> sig;

gofunction void twoInt(int a, int b) {
```

```
    int i;
    for ( i=0; i < 100; i = i + 1) {
        print(i);
    }
    true->sig;
    return;
}

function int main() {
    int x;
    int y;
    int i;

    x = 5;
    y = 10;
    sig = new(channel<bool>[5]);

    go twoInt(x, y);
    go twoInt(x+y, y);

    sig->;
    sig->;

    return 0;
}
```

Thus, to test the concurrency of gofunction, instead of comparing the exactly expected output file with the actual output. We first sort the pre-determined or expected output file and the actual output file, and then compare it. We find the following command extremely useful and modify the test script testall.sh accordingly.

```
> diff <(sort expected.out) <(sort actual.out)
```

8 Lessons Learned

8.1 Chen Chen

LLVM documentation and clang .ll output are very helpful, and it is important to learn how to read them no later than midway through the project. Reading through source codes of past projects helps too.

It is important to have multiple solid features for a language, but it is also crucial to consider the interactions between features and what is necessary to create a valid use case. It is better to have features that collectively build up to a key character of the language (in our case, concurrency) than having several features that go in different directions. We decided not to implement first class function because we needed more time on channel & gofunction to create an interesting demo, and first class function does not offer interesting interactions with the other two features that are obvious to us.

Doing a one-one mapping on team members to features to implement is probably a good idea, as long as the features do not build on top of each other.

8.2 Yuyan Ke

It's important and useful to learn to read and understand the LLVM language for codegen early on. The clang tool is a helpful complement to understand the LLVM documentation as many methods are described in their corresponding machine codes. The clang outputs for a C program help us decide which specific version of the function to use for our usage. When working with codegen, it is helpful to keep in mind if I was working with Ocaml types or LLVM types and not to mix and match types.

Communication is essential to make sure we are on the same page with the newest version of the langauge as we made changes to the syntax and the features during the implementation process. Scheduling a consistent meeting schedule for two meetings a week was a good idea as we avoided time spent on figuring out a time that works for everyone on a regular basis and we were able to make progress towards each milestone.

8.3 Yang Li

Functional programming language is fun. Throughout our study and implementation of our own language compilers I gradually realize and appreciate that functional programming languages have really powerful features like pattern matching and are perfect for compiler designs, I cannot imagine how messy our code would be if it is written in C++/Java.

Collaboration is the key. Building a compiler from scratch is hard and we were dreaming really big about building a language with all the fancy features built in that is not quite achievable. Throughout the semester we modified our language design and implementation several times and without proper communication we cannot make it happen. Luckily I have teammates that are responsible and easy to communicate with. In a software development cycle communication is important because it is impossible to build a large scale software without the collaboration of many talented people.

Testing is as important as developing. I used to not test my code a lot and this is definitely not a good way to develop a programming language. When working on low level code, you really want your code to be really robust and bug-free, the only way to ensure that is to incorporate large test sets that test both units and integration. It is time-consuming, but only after you go through this procedure you can ensure that the thing you built is reliable.

8.4 Arya Lingyu Zhao

The most valuable things PLT taught me are teamwork, communication, trust and appreciation. Our two weekly meetings really help us to plan ahead and seek help on time from the TAs and Prof. Edwards. It was indeed very tough to stick to the meeting routine during midterm and final weeks, but we all made it. The environment in each meeting was very supportive, though sometimes also stressful. Almost all concepts in PLT are brand new to me and they overwhelmed me so much. But working as a team during COVID is exactly what we need - emotional support, intellectual discussion, and some friendly mean comments on the course, the project, and life in general. And to work more effectively as a team, we were constantly communicating with each other about the progress and any difficulties of our work, and about expectations and revision. And I learned so many things from my teammates. They are super passionate, intelligent, supportive, helpful and resilient. I’m not sure how long I’ll remember the class materials, but I truly believe that the friendship we formed through this team project is deep and lifelong.

9 Appendix

9.1 Source Files

9.1.1 gmm.ml

```
(* Top-level of the Go-- compiler: scan & parse the input,
  check the resulting AST and generate an SAST from it, generate
LLVM IR,
  and dump the module *)

type action = Ast | Sast | LLVM_IR | Compile

let () =
 let action = ref Compile in
 let set_action a () = action := a in
 let speclist = [
   ("-a", Arg.Unit (set_action Ast), "Print the AST");
   ("-s", Arg.Unit (set_action Sast), "Print the SAST");
   ("-l", Arg.Unit (set_action LLVM_IR), "Print the generated
LLVM IR");
   ("-c", Arg.Unit (set_action Compile),
     "Check and print the generated LLVM IR (default)");
 ] in
 let usage_msg = "usage: ./gmm.native [-a|-s|-l|-c] [file.gmm]"
in
 let channel = ref stdin in
 Arg.parse speclist (fun filename -> channel := open_in
filename) usage_msg;
  let lexbuf = Lexing.from_channel !channel in
```

```
 let ast = Parser.program Scanner.token lexbuf in
 match !action with
   Ast -> print_string (Ast.string_of_program ast)
 | _ -> let sast = Semant.check ast in
   match !action with
     Ast     -> ()
   | Sast    -> print_string (Sast.string_of_sprogram sast)
   | LLVM_IR -> print_string (Llvm.string_of_llmodule
(Codegen.translate sast))
   | Compile -> let m = Codegen.translate sast in
 Llvm_analysis.assert_valid_module m;
 print_string (Llvm.string_of_llmodule m)
```

9.1.2 scanner.mll

```
(* Ocamllex scanner for Go-- *)

{ open Parser }

let digit = ['0' - '9']
let digits = digit+

rule token = parse
 [' ' '\t' '\r' '\n'] { token lexbuf } (* Whitespace *)
| "/*"     { comment lexbuf }           (* Comments *)
| '('      { LPAREN }
| ')'      { RPAREN }
| '['      { LSQUARE } (* added for channel type *)
| ']'      { RSQUARE } (* added for channel type *)
```

```
| '{'       { LBRACE }
| '}'       { RBRACE }
| ';'       { SEMI }
| ','       { COMMA }
| "."     { DOT }
| "++"      { INC  }
| '+'       { PLUS }
| "->"      { QUEUE }
| "--"      { DEC  }
| '-'       { MINUS }
| '*'       { TIMES }
| '/'       { DIVIDE }
| '='       { ASSIGN }
| "=="      { EQ }
| "!="      { NEQ }
| '<'       { LT }
| "<="      { LEQ }
| ">"       { GT }
| ">="      { GEQ }
| "&&"      { AND }
| "||"      { OR }
| "!"       { NOT }
| "if"      { IF }
| "else"    { ELSE }
| "for"     { FOR }
| "while"   { WHILE }
| "return" { RETURN }
| "int"     { INT }
```

```
| "bool"   { BOOL }
| "float"  { FLOAT }
| "void"   { VOID }
| "string" { STRING }
| "new"    { NEW }
| "array"  { ARRAY }
| "channel"{ CHANNEL }
| "struct" { STRUCT }
| "structdef" { SDEF }
| "true"   { BLIT(true)  }
| "false"  { BLIT(false) }
| "function" { FUNCTION }
| "gofunction" { GOFUNCTION }
| "go"     { GO   }
| digits as lxm { LITERAL(int_of_string lxm) }
| digits '.'  digit* ( ['e' 'E'] ['+' '-']? digits )? as lxm {
FLIT(lxm) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']*     as lxm {
ID(lxm) }
| '"' { str (Buffer.create 16) lexbuf }
| eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped
char)) }

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

```
and str buf = parse
 '"' { STRLIT(Buffer.contents buf) }
| [^ '"'] { Buffer.add_string buf (Lexing.lexeme lexbuf); str
buf lexbuf }
```

9.1.3 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

type typ = Int | Bool | Float | Void | Channel of typ | Array of
typ | String
      | Struct of string

and bind = typ * string

and sdef = StructDef of string * bind list

and expr =
   Literal of int
 | Fliteral of string
 | BoolLit of bool
 | Id of string
 | Binop of expr * op * expr
```

```
  | Unop of uop * expr
  | Assign of expr * expr
  | FuncCall of string * expr list
  | GofuncCall of string * expr list
  | NArray of typ * expr
  | ArrAccess of expr * expr
  | Dot of expr * string
  | NChan of typ * expr
  | Queue of expr * expr
  | StrLit of string
  | NStruct of typ * expr list
  | Noexpr

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

type func_decl = {
    isGo: bool;
    typ : typ;
    fname : string;
    formals : bind list;
    locals : bind list;
    body : stmt list;
```

```
 }

type program = sdef list * (bind list * func_decl list)

(* Pretty-printing functions *)

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

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

let rec string_of_typ = function
   Int -> "int"
 | Bool -> "bool"
 | Float -> "float"
```

```
 | Void -> "void"
 | Channel(ty) -> "channel<" ^ (string_of_typ ty) ^ ">"
 | Array (ty) -> "array<" ^ (string_of_typ ty) ^ ">"
 | String  -> "string"
 | Struct(s) -> s

let rec string_of_expr = function
   Literal(l) -> string_of_int l
 | Fliteral(l) -> l
 | StrLit(l) -> l
 | BoolLit(true) -> "true"
 | BoolLit(false) -> "false"
 | Id(s) -> s
 | Binop(e1, o, e2) ->
     string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^
string_of_expr e2
 | Unop(o, e) -> string_of_uop o ^ string_of_expr e
 | Assign(v, e) -> string_of_expr v ^ " = " ^ string_of_expr e
 | FuncCall(f, el) ->
     f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^
")"
 | GofuncCall(f, el) ->
    "go " ^ f ^ "(" ^ String.concat ", " (List.map
string_of_expr el) ^ ")"
 | Dot(a,n) -> string_of_expr a ^ "." ^ n
 | NArray(ty, e) -> "new(array<" ^ (string_of_typ ty) ^ ">[" ^
(string_of_expr e) ^ "]"
```

```
 | ArrAccess(e1, e2) -> string_of_expr e1 ^ "[" ^ string_of_expr
e2 ^ "]"
 | NChan(ty, e) -> "new(channel<" ^ string_of_typ ty ^ ">[" ^
string_of_expr e ^ "]"
 | Queue (e1, e2) -> string_of_expr e1 ^ " -> " ^ string_of_expr
e2
 | NStruct (ty,el) -> "new(" ^ string_of_typ ty ^"," ^
String.concat ", " (List.map string_of_expr el) ^ ")"
 | Noexpr -> ""

let rec string_of_stmt = function
   Block(stmts) ->
     "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^
"}\n"
 | Expr(expr) -> string_of_expr expr ^ ";\n";
 | Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
 | If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^
string_of_stmt s
 | If(e, s1, s2) ->  "if (" ^ string_of_expr e ^ ")\n" ^
     string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
 | For(e1, e2, e3, s) ->
     "for (" ^ string_of_expr e1  ^ " ; " ^ string_of_expr e2 ^
" ; " ^
     string_of_expr e3  ^ ") " ^ string_of_stmt s
 | While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^
string_of_stmt s

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

```
let goKey = function
   true -> "gofunction"
 | false -> "function"

let string_of_sdef = function
 StructDef(n,vars) -> "structdef " ^ n ^" {\n"
            ^ String.concat "" (List.map string_of_vdecl vars)
            ^"};\n"

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

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

9.1.4 parser.mly

```
/* Ocamlyacc parser for Go-- */

%{
```

```
open Ast
%}

%token SEMI LPAREN RPAREN LBRACE RBRACE COMMA PLUS MINUS TIMES
DIVIDE ASSIGN /* arithmatic op */
%token NOT EQ NEQ LT LEQ GT GEQ AND OR INC DEC /* logic op */
%token RETURN IF ELSE FOR WHILE /* control flow */
%token INT BOOL FLOAT VOID CHANNEL STRING STRUCT SDEF/* types */
%token LSQUARE RSQUARE NEWCHAN QUEUE /* channel related */
%token NEW ARRAY /* array related */
%token GO FUNCTION GOFUNCTION /* function related */
%token DOT
%token <int> LITERAL
%token <bool> BLIT
%token <string> ID FLIT
%token <string> STRLIT
%token EOF

%start program
%type <Ast.program> program

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN
%left OR
%left AND
%left EQ NEQ
%left LT GT LEQ GEQ
```

```
%left PLUS MINUS
%left TIMES DIVIDE
%right QUEUE
%right NOT
%right GO
%nonassoc INC DEC
%left DOT

%%

program:
 body EOF { $1 }


body:
 sdef_list decls   {    ($1,$2)       }


sdef_list:
 /* nothing */   { [] }
|  sdef sdef_list     {  $1 :: $2   }

sdef:
 SDEF ID LBRACE pdecl_opt RBRACE SEMI { StructDef($2,$4)}

decls:
  /* nothing */ { ([], [])               }
| decls vdecl { (($2 :: fst $1), snd $1) }
```

```
| decls fdecl { (fst $1, ($2 :: snd $1)) }

fdecl:
  isGo typ ID LPAREN formals_opt RPAREN LBRACE vdecl_list
stmt_list RBRACE
  {{
     isGo = $1;
     typ = $2;
     fname = $3;
     formals = List.rev $5;
     locals = List.rev $8;
     body = List.rev $9 } }

isGo:
   FUNCTION   { false }
 | GOFUNCTION { true  }

formals_opt:
   /* nothing */ { [] }
 | formal_list   { $1 }

formal_list:
   typ ID                   { [($1,$2)]     }
 | formal_list COMMA typ ID { ($3,$4) :: $1 }

ptyp:
   INT   { Int   }
 | BOOL  { Bool  }
```

```
 | FLOAT { Float }
 | VOID  { Void  }
 | STRING { String }

styp:
 | STRUCT ID { Struct($2) }

typ:
   ptyp  { $1  }
 | styp    { $1  }
 | ARRAY LT typ GT { Array($3)}
 | CHANNEL LT typ GT { Channel($3) }

vdecl_list:
   /* nothing */    { [] }
 | vdecl_list vdecl { $2 :: $1 }

vdecl:
  typ ID SEMI { ($1, $2) }


pdecl_opt:
 /* nothing */     {[]}
 | pdecl_list      {List.rev $1}


pdecl_list:
   pdecl            { [$1] }
```

```
 | pdecl_list pdecl { $2 :: $1 }

pdecl:
  ptyp ID COMMA { ($1, $2) }


stmt_list:
   /* nothing */  { [] }
 | stmt_list stmt { $2 :: $1 }

stmt:
   expr SEMI                                { Expr $1
}
 | RETURN expr_opt SEMI                     { Return $2
}
 | LBRACE stmt_list RBRACE                  { Block(List.rev $2)
}
 | IF LPAREN expr RPAREN stmt %prec NOELSE { If($3, $5,
Block([])) }
 | IF LPAREN expr RPAREN stmt ELSE stmt     { If($3, $5, $7)
}
 | FOR LPAREN expr_opt SEMI expr SEMI expr_opt RPAREN stmt
                                            { For($3, $5, $7, $9)
}
 | WHILE LPAREN expr RPAREN stmt            { While($3, $5)
}

expr_opt:
   /* nothing */ { Noexpr }
```

```
  | expr          { $1 }

expr:
  | expr INC         { Assign($1, Binop($1, Add, Literal(1)))}
  | expr DEC         { Assign($1, Binop($1, Sub, Literal(1)))}
  | expr PLUS   expr { Binop($1, Add,   $3)   }
  | expr MINUS  expr { Binop($1, Sub,   $3)   }
  | expr TIMES  expr { Binop($1, Mult,  $3)   }
  | expr DIVIDE expr { Binop($1, Div,   $3)   }
  | expr EQ     expr { Binop($1, Equal, $3)   }
  | expr NEQ    expr { Binop($1, Neq,   $3)   }
  | expr LT     expr { Binop($1, Less,  $3)   }
  | expr LEQ    expr { Binop($1, Leq,   $3)   }
  | expr GT     expr { Binop($1, Greater, $3) }
  | expr GEQ    expr { Binop($1, Geq,   $3)   }
  | expr AND    expr { Binop($1, And,   $3)   }
  | expr OR     expr { Binop($1, Or,    $3)   }
  | expr QUEUE expr  { Queue($1, $3)   }
  | expr QUEUE       { Queue($1, Noexpr)} /* channel -> */
  | MINUS expr %prec NOT { Unop(Neg, $2)      }
  | NOT expr         { Unop(Not, $2)          }
  | expr ASSIGN expr { Assign($1, $3)         }
  | NEW LPAREN ARRAY LT typ GT LSQUARE expr RSQUARE RPAREN {
NArray($5, $8) }
  | NEW LPAREN CHANNEL LT typ GT LSQUARE expr RSQUARE RPAREN {
NChan($5, $8) }
  | NEW LPAREN styp COMMA args_opt RPAREN { NStruct($3,$5) }
  | ID LPAREN args_opt RPAREN { FuncCall($1, $3)  }
```

```
  | GO ID LPAREN args_opt RPAREN { GofuncCall($2, $4)   }
  | LPAREN expr RPAREN {       $2              }
  | accessor           { $1 }


accessor:
    accessor LSQUARE expr RSQUARE { ArrAccess($1, $3)    }
  | accessor DOT ID         { Dot($1, $3)      }
  | atom  {$1}


atom:
    LITERAL          { Literal($1)            }
  | FLIT             { Fliteral($1)           }
  | BLIT             { BoolLit($1)            }
  | ID               { Id($1)                 }
  | STRLIT           { StrLit($1)             }


args_opt:
    /* nothing */ { [] }
  | args_list  { List.rev $1 }

args_list:
    expr                    { [$1] }
  | args_list COMMA expr { $3 :: $1 }
```

9.1.5 sast.ml

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

open Ast

type sexpr = typ * sx
and sx =
   SLiteral of int
 | SFliteral of string
 | SBoolLit of bool
 | SStrLit of string
 | SId of string
 | SBinop of sexpr * op * sexpr
 | SUnop of uop * sexpr
 | SAssign of sexpr * sexpr
 | SFuncCall of string * sexpr list
 | SGofuncCall of string * sexpr list
 | SNArray of typ * sexpr
 | SArrAccess of sexpr * sexpr
 | SNChan of typ *sexpr
 (* dequeue & enqueue following the convention of (arr, data) *)
 | SDot of sexpr * string
 | SDequeue of sexpr * sexpr
 | SEnqueue of sexpr * sexpr
 | SNStruct of typ * sexpr list
 | SNoexpr
```

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

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

type sprogram = sdef list * (bind list * sfunc_decl list)

(* Pretty-printing functions *)

let rec string_of_sexpr (t, e) =
 "(" ^ string_of_typ t ^ " : " ^ (match e with
   SLiteral(l) -> string_of_int l
 | SBoolLit(true) -> "true"
 | SBoolLit(false) -> "false"
 | SStrLit(l) -> l
 | SFliteral(l) -> l
```

```
 | SId(s) -> s
 | SBinop(e1, o, e2) ->
     string_of_sexpr e1 ^ " " ^ string_of_op o ^ " " ^
string_of_sexpr e2
 | SUnop(o, e) -> string_of_uop o ^ string_of_sexpr e
 | SAssign(e1, e2) -> string_of_sexpr e1 ^ " = " ^
string_of_sexpr e2
 | SFuncCall(f, el) ->
     f ^ "(" ^ String.concat ", " (List.map string_of_sexpr el)
^ ")"
 | SGofuncCall(f, el) ->
     "go "^ f ^ "(" ^ String.concat ", " (List.map
string_of_sexpr el) ^ ")"
 | SDot(e,s) -> string_of_sexpr e ^ s
 | SNArray(ty, e) -> "new(array<" ^ string_of_typ ty ^ ">[" ^
string_of_sexpr e ^"]"
 | SArrAccess(e1, e2) -> string_of_sexpr e1 ^ "[" ^
string_of_sexpr e2 ^ "]"
 | SNChan(ty, e) -> "new(channel<" ^ string_of_typ ty ^ ">[" ^
string_of_sexpr e ^ "]"
 | SDequeue(e1, e2) | SEnqueue(e1, e2) -> string_of_sexpr e1 ^
"->" ^ string_of_sexpr e2
 | SNStruct(t,el) -> "new (" ^ string_of_typ t^ "," ^
String.concat ", " (List.map string_of_sexpr el) ^ ")"
 | SNoexpr -> ""
        ) ^ ")"

let rec string_of_sstmt = function
   SBlock(stmts) ->
```

```
     "{\n" ^ String.concat "" (List.map string_of_sstmt stmts) ^
"}\n"
 | SExpr(expr) -> string_of_sexpr expr ^ ";\n";
 | SReturn(expr) -> "return " ^ string_of_sexpr expr ^ ";\n";
 | SIf(e, s, SBlock([])) ->
     "if (" ^ string_of_sexpr e ^ ")\n" ^ string_of_sstmt s
 | SIf(e, s1, s2) ->  "if (" ^ string_of_sexpr e ^ ")\n" ^
     string_of_sstmt s1 ^ "else\n" ^ string_of_sstmt s2
 | SFor(e1, e2, e3, s) ->
     "for (" ^ string_of_sexpr e1  ^ " ; " ^ string_of_sexpr e2
^ " ; " ^
     string_of_sexpr e3  ^ ") " ^ string_of_sstmt s
 | SWhile(e, s) -> "while (" ^ string_of_sexpr e ^ ") " ^
string_of_sstmt s

let goKey = function
   true -> "gofunction"
 | false -> "function"

let string_of_sfdecl fdecl =
 goKey fdecl.sisGo ^ " " ^
 string_of_typ fdecl.styp ^ " " ^
 fdecl.sfname ^ "(" ^ String.concat ", " (List.map snd
fdecl.sformals) ^
 ")\n{\n" ^
 String.concat "" (List.map string_of_vdecl fdecl.slocals) ^
 String.concat "" (List.map string_of_sstmt fdecl.sbody) ^
 "}\n"
```

```
let string_of_sprogram (sdefs,(vars, funcs)) =
 String.concat "" (List.map string_of_sdef sdefs) ^ "\n" ^
 String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^
 String.concat "\n" (List.map string_of_sfdecl funcs)
```

9.1.6 codegen.ml

```
(* Code generation for Go-- *)

module L = Llvm
module A = Ast
open Sast

module StringMap = Map.Make(String)

(* translate : Sast.program -> Llvm.module *)
let translate (sdefs,(globals, functions)) =
 let context    = L.global_context () in
  (* Create the LLVM compilation module into which
    we will generate code *)
 let the_module = L.create_module context "Gmm" in

 (* Get types from the context *)
 let i32_t      = L.i32_type    context
 and i64_t      = L.i64_type    context
 and i8_t       = L.i8_type     context
 and i1_t       = L.i1_type     context
```

```
 and float_t    = L.double_type context
 and ptr_t      = L.pointer_type (L.i8_type context)
 and void_t     = L.void_type   context in
 let mutex_t    = L.struct_type context [|i64_t; L.array_type
i8_t 56|]
 and sem_t      = L.struct_type context [| i64_t; L.array_type
i8_t 24|] in
  (* collect struct definition in a StringMap hash table in
Ocaml *)
 (* Map structure { key:sID ,value: (idx in the struct,
(type,name)) } *)
 let rec seq n = if n >= 0 then (n :: (seq (n-1))) else [] in
 let sdecl_map =
 let process_def m sdef = (match sdef with
   A.StructDef(id,binds) ->
   let len = List.length binds in
   let seqlist = List.rev (seq (len-1)) in
   let values = List.combine seqlist binds in
   StringMap.add id values m)
 in
   List.fold_left process_def StringMap.empty sdefs
 in
 (* Return the LLVM type for a MicroC type *)
 let rec ltype_of_typ = function
     A.Int   -> i32_t
   | A.Bool  -> i1_t
   | A.Float -> float_t
   | A.Void  -> void_t
```

```
    | A.Array(ty) -> L.pointer_type (ltype_of_typ ty)
    | A.String -> ptr_t
    | A.Channel(ty) ->
      let ptrs = List.map L.pointer_type [ltype_of_typ ty; i32_t;
mutex_t; sem_t] in
      let ptrs_arr = Array.of_list ptrs in
      let chan_t = L.struct_type context ptrs_arr in
      let chanptr = L.pointer_type chan_t in
      chanptr
    | A.Struct(s) ->
      let mvalues = StringMap.find s sdecl_map in
      let collect_typ (_,(t,_)) = ltype_of_typ t in
      let typs = List.map collect_typ mvalues in
      let typs_arr = Array.of_list typs in
      let struct_t = L.struct_type context typs_arr in
      let struct_ptr = L.pointer_type struct_t in
      struct_ptr
    in

  (* Create a map of global variables after creating each *)
  let global_vars : L.llvalue StringMap.t =
    let global_var m (t, n) =
      let init = match t with
          A.Float -> L.const_float (ltype_of_typ t) 0.0
        | A.Array(_) | A.Channel (_) | A.String |A.Struct(_) ->
L.const_null (ltype_of_typ t)
        | _ -> L.const_int (ltype_of_typ t) 0
      in StringMap.add n (L.define_global n init the_module) m in
```

```
  List.fold_left global_var StringMap.empty globals in

 (* built-in functions *)
 let printf_t : L.lltype =
     L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
 let printf_func : L.llvalue =
     L.declare_function "printf" printf_t the_module in
 (* semaphore calls for channel *)
 let sem_init_t : L.lltype = L.function_type i32_t
[|L.pointer_type sem_t; i32_t; i32_t|] in
 let sem_init_f : L.llvalue = L.declare_function "sem_init"
sem_init_t the_module in
 (* functions for strings *)
 let strcat_t : L.lltype =
   L.function_type ptr_t [|ptr_t;ptr_t|] in
 let strcat_f : L.llvalue =
   L.declare_function "string_cat" strcat_t the_module in
 let strcmp_t : L.lltype =
     L.function_type i32_t [|ptr_t;ptr_t|] in
 let strcmp_f : L.llvalue =
   L.declare_function "string_cmp" strcmp_t the_module in
 (* Define each function (arguments and return type) so we can
    call it even before we've created its body *)
 let function_decls : (L.llvalue * sfunc_decl) StringMap.t =
   let function_decl m fdecl =
     let name = fdecl.sfname
     and formal_types =
```

```
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t)
fdecl.sformals)
    in
    let ftype = L.function_type (ltype_of_typ fdecl.styp)
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 =
  let (the_function, _) = StringMap.find fdecl.sfname
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"
builder
  and float_format_str = L.build_global_stringptr "%g\n" "fmt"
builder
  and str_format_str = L.build_global_stringptr "%s\n" "fmt"
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 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);
        StringMap.add n local m
      and add_local m (t, n) =
        let local_var = match t with
            _ -> 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.sformals
          (Array.to_list (L.params the_function)) in
      List.fold_left add_local formals fdecl.slocals
    in

    (* Return the value for a variable or formal argument.
       Check local names first, then global names *)
    let lookup n = try StringMap.find n local_vars
                   with Not_found -> StringMap.find n global_vars
    in

    (* some util functions *)
    let construct_chan_t ty =
      let ptrs = List.map L.pointer_type [ltype_of_typ ty; i32_t;
mutex_t; sem_t] in
      let ptrs_arr = Array.of_list ptrs in
      let chan_t = L.struct_type context ptrs_arr in
      chan_t
```

```
in

let lnum i = L.const_int i32_t i in
let insert_value builder agg i v =
  L.build_insertvalue agg v i "tmp__" builder
in

let inc_seq k =
  let rec dec_seq n =
    if n >= 0 then (n :: (dec_seq (n-1)))
    else []
  in List.rev (dec_seq k)
in

(* Construct code for an expression; return its value *)
let rec expr builder ((_, e) : sexpr) = match e with
    SLiteral i  -> L.const_int i32_t i
  | SStrLit s   -> L.build_global_stringptr s "str" builder
  | SBoolLit b  -> L.const_int i1_t (if b then 1 else 0)
  | SFliteral l -> L.const_float_of_string float_t l
  | SNoexpr     -> L.const_int i32_t 0
  | SId s       -> L.build_load (lookup s) s builder
  (* array element assignment *)
  | SAssign((_, SArrAccess(e1, e2)), e3) ->
    let new_v = expr builder e3 in
    let arr = expr builder e1 in
    let idx = expr builder e2 in
    let ptr = L.build_gep arr [|idx|] "" builder in
```

```
        ignore(L.build_store new_v ptr builder); new_v
        (* struct element assignment *)
      | SAssign((_, SDot((A.Struct(sname),e1),fname)),e2) ->
        let new_v = expr builder e2 in
        let stptr = expr builder (A.Struct(sname),e1) in
        let st = L.build_load stptr "structval" builder in
        let binds = StringMap.find sname sdecl_map in
        let rec idx l name = match l with [] -> raise (Failure
"field not found")
                             | (i,(_,n))::tl -> if n = name then i
else idx tl name in
        let index = idx binds fname in
        let new_struct = L.build_insertvalue st new_v index "dot"
builder in
        ignore(L.build_store new_struct stptr builder); new_v
      | SAssign (e1, e2) ->
        let e' = expr builder e2 in
        (match snd e1 with
           SId s -> ignore(L.build_store e' (lookup s) builder);
e'
         | _ -> raise (Failure ("assignment for " ^
(string_of_sexpr e2) ^
                                "not implemented in codegen")))
      (* intializing a new struct*)
      | SNStruct(ty,el) ->
        let fields = List.map (expr builder) el in
        let tuples = match ty with A.Struct(s) -> StringMap.find
s sdecl_map | _ ->raise (Failure "not struct type") in
```

```
        let types = List.map (fun (_,(t,_)) -> ltype_of_typ t)
tuples in
        let length = List.length types in
        let idxs = List.rev (seq (length-1)) in
        let llstruct_t = L.struct_type context (Array.of_list
types) in
        let ptr = L.build_malloc llstruct_t "struct_init" builder
in
        let value = List.fold_left2 (insert_value builder)
(L.const_null llstruct_t) idxs fields in
        let _ = L.build_store value ptr builder in ptr
      | SDot((A.Struct(sname),e),fname) ->
        let stptr = expr builder (A.Struct(sname),e) in
        let binds = StringMap.find sname sdecl_map in
        let rec idx l name = match l with [] -> raise (Failure
"field not found")
                          | (i,(_,n))::tl -> if n = name then i else
idx tl name in
        let index = idx binds fname in
        let svalue = L.build_load stptr "structval" builder in
        L.build_extractvalue svalue index fname builder
      (* array implementation on stack, doesn't need to be on the
heap for now*)
      | SNArray(ty, e) ->
        let arr_size = expr builder e in
        let llarr_typ = ltype_of_typ ty in
        let ptr = L.build_array_alloca llarr_typ arr_size ""
builder in ptr
      (* array access *)
```

```
      | SArrAccess(e1, e2) ->
        let arr = expr builder e1 in
        let idx = expr builder e2 in
        let ptr = L.build_load (L.build_gep arr [|idx|] ""
builder) "" builder in
        ptr
       (* channel initialization *)
      | SNChan(ty, e) ->
        let malloc_chan lsize ty =
          let typs = [ltype_of_typ ty; i32_t; mutex_t; sem_t] in
          let nums = [lsize; lnum 4; lnum 1; lnum 2] in
          let my_arr_malloc a b = L.build_array_malloc a b
"arr_malloc_" builder in
          let ptrs = List.map2 my_arr_malloc typs nums in
          (* assign values to the book-keeping array *)
          let my_store agg idx value=
            let store_ptr = L.build_gep agg [|idx|] "arr_gep_"
builder in
            ignore(L.build_store value store_ptr builder);
store_ptr
          in
          let type_id = match ty with
              A.Int -> lnum 0
            | A.Float -> lnum 1
            | A.Bool -> lnum 2
            | A.String -> lnum 3
            | A.Struct(_) -> lnum 4
```

```
            | _ -> raise (Failure "type is not supported by
channel")
          in
          let bkptr = List.nth ptrs 1 in
          let _ = List.map2 (my_store bkptr) [(lnum 0); (lnum 3)]
[lsize; type_id] in
          (* sem_init on the two semaphore *)
          let sem_arr = List.nth ptrs 3 in
          let sem_empty = L.build_gep sem_arr [|lnum 0|]
"sem_access_" builder in
          let _ = L.build_call sem_init_f [|sem_empty; (lnum 0);
(lnum 0)|] "sem_empty" builder in
          let sem_full = L.build_gep sem_arr [|lnum 1|]
"sem_access_" builder in
          let _ = L.build_call sem_init_f [|sem_full; (lnum 0);
lsize |] "sem_empty" builder in
          ptrs
        in
        let chan_size = expr builder e in
        let chan_t = construct_chan_t ty in
        let vals = malloc_chan chan_size ty in
        let seq = inc_seq ((List.length vals)-1) in
        let llvals = List.fold_left2
         (fun agg i v -> insert_value builder agg i v)
         (L.const_null chan_t) seq vals in
        let chanptr = L.build_malloc chan_t "chanInit" builder in
        ignore(L.build_store llvals chanptr builder);chanptr
        (* channel enqueue/dequeue op *)
```

```
      | SEnqueue(e1, e2) ->
       let new_v = expr builder e2 in
       let chanptr = expr builder e1 in
       let chan_v = L.build_load chanptr "chan_val" builder in
       let (tyy, _ ) = e2 in
       (* get pointer to value to be enqueued --> copy by value
*)
       let val_ptr = L.build_malloc (ltype_of_typ tyy)
"val_ptr_" builder in
       let _ = L.build_store new_v val_ptr builder in
       (* construct the function call *)
       let enc_t : L.lltype =
         let lltype = ltype_of_typ tyy in
         let tylist = [lltype; i32_t; mutex_t; sem_t; lltype] in
         let typtr = List.map L.pointer_type tylist in
         let tyarr = Array.of_list typtr in
         L.function_type i32_t tyarr in
       let enc_f : L.llvalue = L.declare_function "enqueue_f"
enc_t the_module in
       (* extract value *)
       let my_extract agg idx = L.build_extractvalue agg idx
"extrac_" builder in
       let vals = List.map (my_extract chan_v) (inc_seq 3) in
       let _ = L.build_call enc_f (Array.of_list (List.append
vals [val_ptr])) "enc_ff" builder in
       new_v
     | SDequeue(e1, e2) ->
       let chanptr = expr builder e1 in
```

```
        let chan_v = L.build_load chanptr "chan_val" builder in
        let tyy =  match fst e1 with
            A.Channel(tyy) -> tyy
          | _ -> raise (Failure "dequeue from incorrect type")
        in
        let val_ptr = L.build_alloca (ltype_of_typ tyy)
"val_ptr_" builder in
        (* construct the function call *)
        let dec_t : L.lltype =
          let lltype = ltype_of_typ tyy in
          let tylist = [lltype; i32_t; mutex_t; sem_t; lltype] in
          let typtr = List.map L.pointer_type tylist in
          let tyarr = Array.of_list typtr in
          L.function_type i32_t tyarr in
        let dec_f : L.llvalue = L.declare_function "dequeue_f"
dec_t the_module in
        (* extract value *)
        let my_extract agg idx = L.build_extractvalue agg idx
"extrac_" builder in
        let vals = List.map (my_extract chan_v) (inc_seq 3) in
        let _ = L.build_call dec_f (Array.of_list (List.append
vals [val_ptr])) "enc_ff" builder in
        let new_v = L.build_load val_ptr "new_v_dec_" builder in
        let eval = match e2 with
          (A.Void, _) -> new_v
          |(_, SId s) -> ignore(L.build_store new_v (lookup s)
builder); new_v
```

```
        | _ -> raise (Failure ("assignment for " ^
(string_of_sexpr e2) ^
                                  "not implemented in codegen"))
      in eval
    | SBinop ((A.Float,_ ) as e1, op, e2) ->
      let e1' = expr builder e1
      and e2' = expr builder e2 in
      (match op with
        A.Add     -> L.build_fadd
      | A.Sub     -> L.build_fsub
      | A.Mult    -> L.build_fmul
      | A.Div     -> L.build_fdiv
      | A.Equal   -> L.build_fcmp L.Fcmp.Oeq
      | A.Neq     -> L.build_fcmp L.Fcmp.One
      | A.Less    -> L.build_fcmp L.Fcmp.Olt
      | A.Leq     -> L.build_fcmp L.Fcmp.Ole
      | A.Greater -> L.build_fcmp L.Fcmp.Ogt
      | A.Geq     -> L.build_fcmp L.Fcmp.Oge
      | A.And | A.Or ->
        raise (Failure "internal error: semant should have
rejected and/or on float")
      ) e1' e2' "tmp" builder
    | SBinop ((A.String,_ ) as e1, op, e2) ->
      let e1' = expr builder e1
      and e2' = expr builder e2 in
      (match op with
        A.Add     -> L.build_call strcat_f [| e1';e2' |]
"string_cat" builder
```

```
      | A.Equal   -> (L.build_icmp L.Icmp.Eq) (L.const_int
i32_t 0)
          (L.build_call strcmp_f [| e1';e2' |] "string_cmp"
builder) "tmp" builder
      | A.Neq     -> (L.build_icmp L.Icmp.Ne) (L.const_int
i32_t 0)
          (L.build_call strcmp_f [| e1';e2' |] "string_cmp"
builder) "tmp" builder
      | _         ->
        raise (Failure "internal error: semant should have
rejected and/or on float")
      )
    | SBinop (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.Icmp.Eq
      | A.Neq     -> L.build_icmp L.Icmp.Ne
      | A.Less    -> L.build_icmp L.Icmp.Slt
      | A.Leq     -> L.build_icmp L.Icmp.Sle
      | A.Greater -> L.build_icmp L.Icmp.Sgt
      | A.Geq     -> L.build_icmp L.Icmp.Sge
```

```
      ) e1' e2' "tmp" builder
    | SUnop(op, ((t, _) as e)) ->
      let e' = expr builder e in
      (match op with
          A.Neg when t = A.Float -> L.build_fneg
        | A.Neg                  -> L.build_neg
        | A.Not                  -> L.build_not
      ) e' "tmp" builder
    | SFuncCall ("print", [e]) | SFuncCall ("printb", [e]) ->
      L.build_call printf_func [| int_format_str ; (expr
builder e) |]
      "printf" builder
    | SFuncCall ("printf", [e]) ->
      L.build_call printf_func [| float_format_str ; (expr
builder e) |]
      "printf" builder
    | SFuncCall ("prints",[e]) ->
      L.build_call printf_func [| str_format_str ; (expr
builder e) |]
      "printf" builder
    | SFuncCall (f, args)  ->
      let (fdef, fdecl) = StringMap.find f function_decls in
      let llargs = List.rev (List.map (expr builder) (List.rev
args)) in
      let result = (match fdecl.styp with
                        A.Void -> ""
                      | _ -> f ^ "_result") in
        L.build_call fdef (Array.of_list llargs) result builder
```

```
      | SGofuncCall(f, args) ->
        (* TODO: Now Gocall only supports up to 3 float/int
arguments*)
        let (fdef, fdecl) = StringMap.find f function_decls in
        (* convert number to a desc. sequence *)
        let rec seq n = if n >= 0 then (n :: (seq (n-1))) else []
in
        (* match primitive types to unique number to allow
external C functions
           to identify them.*)
        let match_type = function
             A.Int   -> 1
           | A.Float -> 2
           | _       -> 3   in (* pointer types: string, array,
struct *)
        let offset ty = lnum (match_type ty) in
        (* get a list of length *)
        let typ_offsets alist (t,_) =((offset t)::alist) in
        let offsets = List.rev (List.fold_left typ_offsets []
fdecl.sformals) in
        (* collect types of each argument in the argument list *)
        let collect_typ alist (t,_) = ((ltype_of_typ t)::alist)
in
        let typs = List.rev (List.fold_left collect_typ []
fdecl.sformals) in
        (* evaluate arguments and put them into (type,value)
tuples *)
        let llargs = List.map (expr builder) args in
        let combine alist typ arg = ((typ,arg)::alist)in
```

```
        let tuples = List.rev (List.fold_left2 combine [] typs
llargs) in
        (* get the argument list length *)
        let arg_size l = List.fold_left (fun count _ -> count +
1) 0 l in
        let az = arg_size args in
        let count = L.const_int i32_t az in
        (* construct a struct called prefix that contains
information
          * about gofunction arguments, the form of the prefix
list is
          * [arg.length,arg1.type,arg2.type.....]
         *)
        let build_arg bty tylist len =
          let ty_idxs =  List.rev (seq ((List.length tylist) -
1)) in
          let ty_typs = Array.make len bty in
          let ty_typ = L.struct_type context ty_typs in
          let ty_ptr = L.build_alloca ty_typ "tmp_2" builder in
          let value = List.fold_left2 (insert_value builder)
(L.const_null ty_typ) ty_idxs tylist in
          let _ = L.build_store value ty_ptr builder in
          (ty_typ, ty_ptr) in
        let prelist = (count::offsets) in
        let (pre_typ, pre) = build_arg i32_t prelist (az+1) in
        let build_nums filter bty =
          let tylist = List.rev (List.fold_left filter [] tuples)
in
          build_arg bty tylist (List.length tylist) in
```

```
        let collect_int alist (t,v) = (if t == i32_t then
(v::alist) else alist) in
        let collect_double alist (t,v) = (if t == float_t then
(v::alist) else alist) in
        let (int_typ, intptr) = build_nums collect_int i32_t
        and (d_typ, dptr) = build_nums collect_double float_t in
        (* get a list of length *)
        let typ_matches alist (t,_) =((match_type t)::alist) in
        let matches = List.rev (List.fold_left typ_matches []
fdecl.sformals) in
        let tuples = List.rev (List.fold_left2 combine [] matches
tuples) in
        let collect_ptrs tlist (o,(_,v)) = (if o = 3 then
(v::tlist) else tlist) in
        let palist = List.rev (List.fold_left collect_ptrs []
tuples) in
        let cast some_p = L.build_bitcast some_p ptr_t "bitcast"
builder in
        let palist = List.map cast palist in
        let (p_typ, pptr) = build_arg ptr_t palist (List.length
palist) in
        let gocall_t : L.lltype =
            let func_ptr = L.pointer_type (L.function_type void_t
(Array.of_list typs)) in
            (* external function takes pointers to 1.prefix list
2.int arg list 3. double arg list 4.struct/string arg list*)
        L.function_type i32_t [| func_ptr; (L.pointer_type
pre_typ);(L.pointer_type int_typ);(L.pointer_type
d_typ);(L.pointer_type p_typ) |] in
```

```
            let gocall_func : L.llvalue =
            L.declare_function "gocall" gocall_t the_module in
            L.build_call gocall_func [| fdef;pre;intptr;dptr;pptr
|] "gocall" builder
  | _ -> raise (Failure "expression not implemented") in


   (* LLVM insists each basic block end with exactly one
"terminator"
      instruction that transfers control.  This function runs
"instr builder"
      if the current block does not already have a terminator.
Used,
      e.g., to handle the "fall off the end of the function"
case. *)
   let add_terminal builder instr =
     match L.block_terminator (L.insertion_block builder) with
     Some _ -> ()
     | None -> ignore (instr builder) in
    (* Build the code for the given statement; return the
builder for
      the statement's successor (i.e., the next instruction will
be built
      after the one generated by this call) *)

   let rec stmt builder = function
       SBlock sl -> List.fold_left stmt builder sl
     | SExpr e -> ignore(expr builder e); builder
     | SReturn e -> ignore(match fdecl.styp with
```

```
                                (* Special "return nothing" instr
*)
                               A.Void -> L.build_ret_void builder
                               (* Build return statement *)
                             | _ -> L.build_ret (expr builder e)
builder );
                      builder
      | SIf (predicate, then_stmt, else_stmt) ->
         let bool_val = expr builder predicate in
         let merge_bb = L.append_block context "merge"
the_function in
         let build_br_merge = L.build_br merge_bb in (* partial
function *)
         let then_bb = L.append_block context "then" the_function
in
            add_terminal (stmt (L.builder_at_end context
then_bb) then_stmt)
            build_br_merge;
        let else_bb = L.append_block context "else" the_function
in
            add_terminal (stmt (L.builder_at_end context
else_bb) else_stmt)
            build_br_merge;
            ignore(L.build_cond_br bool_val then_bb else_bb
builder);
            L.builder_at_end context merge_bb

      | SWhile (predicate, body) ->
```

```
        let pred_bb = L.append_block context "while" the_function
in
        ignore(L.build_br pred_bb builder);

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

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

        let merge_bb = L.append_block context "merge"
the_function in
        ignore(L.build_cond_br bool_val body_bb merge_bb
pred_builder);
        L.builder_at_end context merge_bb

      (* Implement for loops as while loops *)
      | SFor (e1, e2, e3, body) -> stmt builder
      ( SBlock [SExpr e1 ; SWhile (e2, SBlock [body ; SExpr e3])
] )
    in

    (* Build the code for each statement in the function *)
    let builder = stmt builder (SBlock fdecl.sbody) in
```

```
   (* Add a return if the last block falls off the end *)
   add_terminal builder (match fdecl.styp with
       A.Void -> L.build_ret_void
     | A.Float -> L.build_ret (L.const_float float_t 0.0)
     | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))
 in

 List.iter build_function_body functions;
 the_module
```

9.1.7 builtin.c

```
/*
* builtin functions and functionalites for Go--
*/
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <stdarg.h>
#include <stdbool.h>
#include <string.h>
#include <semaphore.h>
#include <sys/stat.h>
#include <pthread.h>
#include <errno.h>

#define N_THREADS 16
```

```
//code adapted from cs4118 fall 2020

int init = 0;

static pthread_t thread_pool[N_THREADS];

struct node {
   void *args; // Payload, in our case a new gocall
   struct node *next; // Next call on the list
};

pthread_mutex_t mp  = PTHREAD_MUTEX_INITIALIZER;

struct queue {
   pthread_mutex_t mutex; // A mutex used to protect the queue
itself
   pthread_cond_t cond;   // A condition variable for threads to
sleep on
   struct node *first; // First message in the queue
   struct node *last;  // Last message in the queue
   unsigned int length;   // Number of elements on the queue
};

static struct queue thread_queue;

static void die(const char *message)
{
   perror(message);
```

```
    exit(1);
}


static void queue_init(struct queue *q)
{
    memset(q, 0, sizeof(*q));

    if (pthread_cond_init(&q->cond, NULL) != 0)
        die("Cannot initialize queue condition variable");

    if (pthread_mutex_init(&q->mutex, NULL) != 0) {
        pthread_cond_destroy(&q->cond);
        die("Cannot initialize queue mutex");
    }
}


static void queue_destroy(struct queue *q)
{
    struct node *tmp;
    // First of all delete all elements currently on the queue
    pthread_mutex_lock(&q->mutex);
    while(q->first) {
        tmp = q->first;
        q->first = q->first->next;
        free(tmp);
    }
```

```
    q->last = NULL;
    q->length = 0;
    pthread_mutex_unlock(&q->mutex);
    // After that destroy the mutex and the condition variable
    pthread_mutex_destroy(&q->mutex);
    pthread_cond_destroy(&q->cond);
}



static void queue_put(struct queue *q, void *args)
{
    struct node *msg;

    // Create a new message object, initialized to all zeroes and
store the
    // socket file descriptor in its sock attribute
    msg = (struct node *)calloc(1, sizeof(*msg));
    if (msg == NULL)
        die("Out of memory");
    msg->args = args;

    // Append the new message to the queue
    pthread_mutex_lock(&q->mutex);
    if (q->last == NULL) {
        q->last = msg;
        q->first = msg;
    } else {
```

```
        q->last->next = msg;
        q->last = msg;
    }
    // If the queue was previously empty, wakep up any threads
that might be
    // sleeping, waiting for the queue to contain data
    if (q->length == 0)
        pthread_cond_broadcast(&q->cond);
    q->length++;
    pthread_mutex_unlock(&q->mutex);
}


void *queue_get(struct queue *q)
{
    void *args;
    struct node *msg;

    pthread_mutex_lock(&q->mutex);

    /* Since all the sleeping threads content for the data on the
queue, we
     * need to check if the queue is non-empty after we woke up.
If yes we
     * won, if not we go back to sleep on the condition variable.
     * pthread_cond_wait unlocks the give mutex and suspends the
thread.
     */
```

```
    while (q->first == NULL) {
        pthread_cond_wait(&q->cond, &q->mutex);
    }

    // We won this round, remove the data from the queue and
return it.
    msg = q->first;
    q->first = q->first->next;
    q->length--;

    if (q->first == NULL) {
        q->last = NULL;
        q->length = 0;
    }

    args = msg->args;
    free(msg);
    pthread_mutex_unlock(&q->mutex);
    return args;
}




/* ------------------------- string builtins
-------------------------------*/
```

```
char *string_cat(char *a,char *b)
{
 int len = strlen(a) + strlen(b) + 1;
 char* result = malloc(len);
 strcat(result, a);
 strcat(result, b);
 return result;
}

int string_cmp(char *a,char *b)
{
 int result = strcmp(a,b);
 return result;
}

/* funtion to print double in binary,for debugs only*/
void printbin(double x)
{
   union {
        double x;
        char c[sizeof(double)];
   } u;
   u.x = x;

   for (unsigned ofs = 0; ofs < sizeof(double); ofs++) {
        for(int i = 7; i >= 0; i--) {
            printf(((1 << i) & u.c[ofs]) ? "1" : "0");
        }
```

```
        printf(" ");
    }
}

/* ------------------------- gofunction builtins
--------------------------------*/
/* function to execute the gofunction and manange memory */
void* gothread(void* data)
{
    for(;;){
        void *params = queue_get(&thread_queue);
        void **ptr_f =(void **) params;

        void *fptr = *ptr_f; // first one is func ptr
        // ptr to parameters
        int *ptr = ((int *)params)+sizeof(void *)/sizeof(int);
        int size = *(ptr++); // get number of parameters
        int sum;
        int typs[size];
        double *dptr;
        char **sptr;

        for(int i = 0; i < size; i++){
            typs[i] = *(ptr++);
        }


        if (size > 3) {
```

```
        printf("%s\n","argument exceeds maximum -> 3");
    }
    if (size == 0){
        void (*f)() = (void (*)())fptr;
        f();
    }
    if (size == 1){
        if(typs[0] == 1){
            void(*f)(int) = (void (*) (int))fptr;
            f(*ptr);
        } else if (typs[0] == 2){
            void(*f)(double) = (void (*) (double))fptr;
            f(*((double *)ptr));
        } else if (typs[0] == 3){
            void(*f)(char *) = (void (*) (char *))fptr;
            f(*((char **)ptr));
        }
    }
    if (size == 2){
        if(typs[0]==1){
            int a1 = *ptr;
            ptr++;
            if(typs[1]==1){
                int a2 = *ptr;
                void(*f)(int,int) = (void (*) (int,int))fptr;
                f(a1,a2);
            }else if (typs[1]==2){
                dptr = (double *)ptr;
```

```
                    double a2 = *dptr;
                    void(*f)(int,double) = (void (*)
(int,double))fptr;
                    f(a1,a2);
                }else if (typs[1]==3){
                    sptr = (char **)ptr;
                    char *a2 = *sptr;
                    void(*f)(int,char *) = (void (*) (int,char
*))fptr;
                    f(a1,a2);
                }
            }else if(typs[0]==2){
                dptr =(double *)ptr;
                double a1 = *dptr;
                dptr++;
                if(typs[1]==1){
                    ptr = (int *)dptr;
                    int a2 = *ptr;
                    void(*f)(double,int) = (void (*)
(double,int))fptr;
                    f(a1,a2);
                }else if (typs[1]==2){
                    double a2 = *dptr;
                    void(*f)(double,double) = (void (*)
(double,double))fptr;
                    f(a1,a2);
                }else if (typs[1]==3){
                    sptr = (char **)dptr;
```

```
                    char *a2 = *sptr;
                    void(*f)(double,char *) = (void (*)
(double,char *))fptr;
                    f(a1,a2);
                }

            }else if(typs[0]==3){
                sptr =(char **)ptr;
                char  *a1 = *sptr;
                sptr++;
                if(typs[1]==1){
                    ptr = (int *)sptr;
                    int a2 = *ptr;
                    void(*f)(char *,int) = (void (*) (char
*,int))fptr;
                    f(a1,a2);
                }else if (typs[1]==2){
                    dptr = (double *)sptr;
                    double a2 = *dptr;
                    void(*f)(char *,double) = (void (*) (char
*,double))fptr;
                    f(a1,a2);
                }else if (typs[1]==3){
                    char *a2 = *sptr;
                    void(*f)(char *,char *) = (void (*) (char
*,char *))fptr;
                    f(a1,a2);
                }
```

```
            }
        }
        if (size == 3){
            if(typs[0] == 1){
                int a1 = *ptr;
                ptr++;
                if(typs[1] == 1){
                    int a2= *ptr;
                    ptr++;
                    if(typs[2] == 1){
                        int a3 = *ptr;
                        void(*f)(int,int,int) = (void (*)
(int,int,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        dptr = (double *)ptr;
                        double a3 = *dptr;
                        void(*f)(int,int,double) = (void (*)
(int,int,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        sptr = (char **)ptr;
                        char *a3 = *sptr;
                        void(*f)(int,int,char *) = (void (*)
(int,int,char *))fptr;
                        f(a1,a2,a3);
                    }
```

```
                }else if(typs[1] == 2){
                    dptr =(double *)ptr;
                    double a2= *dptr;
                    dptr++;
                    if(typs[2] == 1){
                        ptr =(int *)dptr;
                        int a3 = *ptr;
                        void(*f)(int,double,int) = (void (*)
(int,double,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        double a3 = *dptr;
                        void(*f)(int,double,double) = (void (*)
(int,double,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        sptr = (char **)dptr;
                        char *a3 = *sptr;
                        void(*f)(int,double,char *) = (void (*)
(int,double,char *))fptr;
                        f(a1,a2,a3);
                    }
                }else if(typs[1] == 3){
                    sptr =(char **)ptr;
                    char *a2= *sptr;
                    sptr ++;
                    if(typs[2] == 1){
                        ptr =(int *)sptr;
```

```
                    int a3 = *ptr;
                    void(*f)(int,char *,int) = (void (*)
(int,char *,int))fptr;
                    f(a1,a2,a3);
                }else if (typs[2] == 2){
                    dptr = (double *)sptr;
                    double a3 = *dptr;
                    void(*f)(int,char *,double) = (void (*)
(int,char *,double))fptr;
                    f(a1,a2,a3);
                }else if (typs[2] == 3){
                    sptr = (char **)sptr;
                    char *a3 = *(char **)sptr;
                    void(*f)(int,char *,char *) = (void (*)
(int,char *,char *))fptr;
                    f(a1,a2,a3);
                }
            }
        } else if(typs[0] == 2){
            dptr = (double *)ptr;
            double a1 = *dptr;
            dptr++;
            if(typs[1] == 1){
                ptr = (int *)dptr;
                int a2= *ptr;
                ptr++;
                if(typs[2] == 1){
                    int a3 = *ptr;
```

```
                        void(*f)(double,int,int) = (void (*)
(double,int,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        dptr = (double *)ptr;
                        double a3 = *dptr;
                        void(*f)(double,int,double) = (void (*)
(double,int,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        sptr = (char **)ptr;
                        char *a3 = *sptr;
                        void(*f)(double,int,char *) = (void (*)
(double,int,char *))fptr;
                        f(a1,a2,a3);
                    }
                }else if(typs[1] == 2){
                    double a2= *dptr;
                    dptr++;
                    if(typs[2] == 1){
                        ptr =(int *)dptr;
                        int a3 = *ptr;
                        void(*f)(double,double,int) = (void (*)
(double,double,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        double a3 = *dptr;
```

```
                        void(*f)(double,double,double) = (void
(*) (double,double,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        sptr = (char **)dptr;
                        char *a3 = *sptr;
                        void(*f)(double,double,char *) = (void
(*) (double,double,char *))fptr;
                        f(a1,a2,a3);
                    }
                }else if (typs[1] == 3){
                    sptr = (char **)dptr;
                    char *a2 = *sptr;
                    sptr++;
                    if(typs[2] == 1){
                        ptr =(int *)sptr;
                        int a3 = *ptr;
                        void(*f)(double,char *,int) = (void (*)
(double,char *,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        dptr =(double *)sptr;
                        double a3 = *dptr;
                        void(*f)(double,char *,double) = (void
(*) (double,char *,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        char *a3 = *sptr;
```

```
                        void(*f)(double,char *,char *) = (void
(*) (double,char *,char *))fptr;
                        f(a1,a2,a3);
                    }

                }
            } else if (typs[0] == 3){
                sptr = (char **)ptr;
                char *a1 = *sptr;
                sptr++;
                if(typs[1] == 1){
                    ptr = (int *)sptr;
                    int a2= *ptr;
                    ptr++;
                    if(typs[2] == 1){
                        int a3 = *ptr;
                        void(*f)(char *,int,int) = (void (*)
(char *,int,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        dptr = (double *)ptr;
                        double a3 = *dptr;
                        void(*f)(char *,int,double) = (void (*)
(char *,int,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        sptr = (char **)ptr;
                        char *a3 = *sptr;
```

```
                        void(*f)(char *,int,char *) = (void (*)
(char *,int,char *))fptr;
                        f(a1,a2,a3);
                    }
                }else if(typs[1] == 2){
                    dptr = (double *)sptr;
                    double a2= *dptr;
                    dptr++;
                    if(typs[2] == 1){
                        ptr =(int *)dptr;
                        int a3 = *ptr;
                        void(*f)(char *,double,int) = (void (*)
(char *,double,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        double a3 = *dptr;
                        void(*f)(char *,double,double) = (void
(*) (char *,double,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        sptr = (char **)dptr;
                        char *a3 = *sptr;
                        void(*f)(char *,double,char *) = (void
(*) (char *,double,char *))fptr;
                        f(a1,a2,a3);
                    }
                }else if (typs[1] == 3){
                    char *a2 = *sptr;
```

```
                    sptr++;
                    if(typs[2] == 1){
                        ptr =(int *)sptr;
                        int a3 = *ptr;
                        void(*f)(char *,char *,int) = (void (*)
(char *,char *,int))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 2){
                        dptr =(double *)sptr;
                        double a3 = *dptr;
                        void(*f)(char *,char *,double) = (void
(*) (char *,char *,double))fptr;
                        f(a1,a2,a3);
                    }else if (typs[2] == 3){
                        char *a3 = *sptr;
                        void(*f)(char *,char *,char *) = (void
(*) (char *,char *,char *))fptr;
                        f(a1,a2,a3);
                    }
                }
            }
        }
        free((char *)params);
    }
}
```

```
void gocall(void* func, void * pre,void * ints, void *
doubles,void *sandsts)
{
   pthread_t tid;

   pthread_mutex_lock(&mp);
   // if has already been init'd
   if(init !=0 ){
       goto proceed;
   }
   /* get the underlying structure of argument
    * and wrap it into memory on the c heap
    */
   init = 1;
   queue_init(&thread_queue);
   for(int i = 0; i < N_THREADS; i++) {
       pthread_create(&thread_pool[i], NULL, gothread, NULL);
   }
proceed:
   pthread_mutex_unlock(&mp);
   int *ptr =(int *) pre; //cast pre to int*
   int size = *(ptr++); // get the number of parameters
   int typs[size];
   // sorted pointers to int and double values
   int *intptr = (int *) ints;
   double * dptr = (double *) doubles;
   char **pptr = (char **) sandsts;
   // calculate the space needed to store all params on heap
```

```
int len = 0;
for (int i = 0; i < size; i++){
    typs[i] = *(ptr+i);
    if(typs[i] == 1){ //int
        len+=4;
    }else if (typs[i] == 2){ //float
        len+=8;
    }else if (typs[i] == 3){
        len+=8;
    }
}

char *pt = malloc(sizeof(void *)+(size+1)*sizeof(int)+len);
char *spt = memcpy(pt,&func,sizeof(void *));
spt += sizeof(void *);
spt = memcpy(spt,(void *)pre,(size+1)*sizeof(int));
spt += (size+1)*sizeof(int);
for (int i = 0; i < size; i++){
    typs[i] = *(ptr+i);
}

for (int i =0; i < size ;i ++) {
    if(typs[i] == 1){
        spt = memcpy(spt,intptr,sizeof(int));
        spt += sizeof(int);
        intptr++;
    }else if (typs[i] == 2){
        spt = memcpy(spt,dptr,sizeof(double));
```

```
            spt += sizeof(double);
            dptr++;
        }else if (typs[i] == 3){
            spt = memcpy(spt,pptr,sizeof(char *));
            spt += sizeof(char *);
            pptr++;
        }
    }


  queue_put(&thread_queue, pt);
}


/* -------------------------- channel builtins
----------------------------*/

int *wrapAround(int *idx, int* size){
    (*idx)++;
    if (*idx == *size)
        *idx = 0;
    return idx;
}

/*
@a: pointer to array storing values
@b: pointer to int array for book-keeping [cap, start, end,
type_id]
```

```
@c: pointer to pthread_mutex_t array [data_lk]
@d: pointer to sem_t array [sem_empty, sem_full]
@val: pointer to value to be enqueued
*/
void enqueue_f(void* a, void* b, void* c, void* d, void* val)
{
   // fixed type fields of a channel
   int* bk = (int*)b;
   int* cap = bk;
   int* start = bk + 1;
   int* end = bk + 2;
   int* type_id = bk + 3;
   pthread_mutex_t* lks = (pthread_mutex_t* )c;
   sem_t* sems = (sem_t* )d;
   sem_wait(sems+1);
   pthread_mutex_lock(lks);
   // variable type fields of a channel
   switch(*type_id)
   {
       case 0: //int
           *((int* )a + *end) = *(int* )val;
           break;
       case 1: //float or double
           *((double* )a + *end) = *(double* )val;
           break;
       case 2: //bool
           *((bool* )a + *end) = *(bool* )val;
           break;
```

```
        case 3: //string
            *((char**)a + *end) = *(char**)val;
            break;
        case 4: //struct
            *((void**)a + *end) = *(void**)val;
            break;
        default:
            printf("type_id %d is not supported by enqueue_f\n",
*type_id);
    }
    wrapAround(end, cap);
    pthread_mutex_unlock(lks);
    sem_post(sems);
}


/*
@a: pointer to array storing values
@b: pointer to int array for book-keeping [cap, start, end,
type_id]
@c: pointer to pthread_mutex_t array [data_lk]
@d: pointer to sem_t array [sem_empty, sem_full]
@val: pointer to space to dequeue into
*/
void dequeue_f(void* a, void* b, void* c, void* d, void* val)
{

    int* bk = (int*)b;
```

```
int* cap = bk;
int* start = bk + 1;
int* end = bk + 2;
int* type_id = bk + 3;
pthread_mutex_t* lks = (pthread_mutex_t* )c;
sem_t* sems = (sem_t* )d;

sem_wait(sems);
pthread_mutex_lock(lks);
switch(*type_id)
{
    case 0: //int
        *(int*) val = *((int*)a + *start);
        break;
    case 1: //float or double
        *(double*) val = *((double*)a + *start);
        break;
    case 2: //bool
        *(bool*) val = *((bool*)a + *start);
        break;
    case 3: //string
        *(char**) val = *((char**)a + *start);
        break;
    case 4: //struct
        *(void**)val = *((void**)a + *start);
        break;
    default:
```

```
            printf("type_id %d is not supported by dequeue_f\n",
*type_id);
    }

    wrapAround(start, cap);
    pthread_mutex_unlock(lks);
    sem_post(sems+1);
}
```

9.1.8 Makefile

```
CXXFLAGS = -pthread
SRC = ./src/

# "make test" Compiles everything and runs the regression tests

.PHONY : test
test : all testall.sh
 ./testall.sh


.PHONY : all
all : gmm.native $(SRC)builtin.o

# "make gmm.native" compiles the compiler
#
# The _tags file controls the operation of ocamlbuild, e.g., by
including
```

```
# packages, enabling warnings
#
# See
https://github.com/ocaml/ocamlbuild/blob/master/manual/manual.ad
oc

gmm.native :
 opam config exec -- \
 ocamlbuild -use-ocamlfind $(SRC)gmm.native


test-print.native:
 opam config exec -- \
 ocamlbuild -use-ocamlfind $(SRC)test-print.native

# "make clean" removes all generated files

.PHONY : clean
clean :
 ocamlbuild -clean
 rm -rf testall.log ocamlllvm *.diff
 rm -rf *.err *.ll *.s *.out *.exe $(SRC)*.o *.o
 rm -rf gmm.native

.PHONY : sclean
sclean:
 rm *.ll *.s *.exe
```

```
# compile builtin functions

builtin : $(SRC)builtin.c
 cc -o builtin -DBUILD_TEST -pthread $(SRC)builtin.c

# Build tests

TESTS = \
 add1 arith1 arith2 arith3 array-types array  \
 bool fib float float1 float2 float3 for-pp \
 for1 for2 func1 func2 func3 func4 func5 func6 func7 func8 \
 func9 gcd gcd2 global-array global1 global2 \
 global3 goargs gocall-stress gocall-stress2 gofunc1 gofunc2 \
 gofunc3 hello if1 if2 if3 if4 if5 if6 integer local1 local2 \
 ops1 ops2 ppmm printfib simple-channel simple-string \
 str str2 string-channel struct var1 var2 while1 while2

GOTEST = \
 bool-channel float-channel global-channel struct-channel


FAILS = \
 array-channel array-init array-typ array1 assign1 assign2
assign3 \
 channel-type dead1 dead2 expr1 expr2 expr3 float1 \
 float2 for1 for2 for3 for4 for5 func1 func2 func3 func4 func5
func6 \
```

```
 func7 global1 global2 gofunc if1 if2 if3 nomain ppmm print
printb \
 return1 return2 void while1 while2

TESTFILES = $(TESTS:%=test-%.gmm) $(TESTS:%=test-%.out) \
     $(FAILS:%=fail-%.gmm) $(FAILS:%=fail-%.err)
```

9.1.9 testall.sh

```
#!/bin/sh

# Regression testing script for MicroC
# 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

# Path to the LLVM interpreter
LLI="lli"
#LLI="/usr/local/opt/llvm/bin/lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the microc compiler.  Usually "./microc.native"
```

```
# Try "_build/microc.native" if ocamlbuild was unable to create
a symbolic link.
GMM="./gmm.native"
#MICROC="_build/microc.native"

# Set time limit for all operations
ulimit -t 30

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

keep=0

Usage() {
   echo "Usage: testall.sh [options] [.gmm 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 "<(sort "$1")" "<(sort "$2")" > "$3" 1>&2
    # echo diff -b $1 $2 ">" $3 1>&2
    diff "$1" "$2" > "$3" 2>&1 || {
    # echo diff "<(sort "$1")" "<(sort "$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/.gmm//'`
   reffile=`echo $1 | sed 's/.gmm$//'`
   basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

   echo -n "$basename..."

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

   generatedfiles=""

   generatedfiles="$generatedfiles ${basename}.ll ${basename}.s
${basename}.exe ${basename}.out" &&
   Run "$GMM" "$1" ">" "${basename}.ll" &&
```

```
   Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">"
"${basename}.s" &&
   Run "$CC" "-o" "${basename}.exe" "${basename}.s"
"${src}builtin.o" "-lpthread" &&
   Run "./${basename}.exe" > "${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/.*\\///
                            s/.gmm//'`
   reffile=`echo $1 | sed 's/.gmm$//'`
   basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."
```

```
    echo -n "$basename..."

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

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err
${basename}.diff" &&
    RunFail "$GMM" "<" $1 "2>" "${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`

LLIFail() {
 echo "Could not find the LLVM interpreter \"$LLI\"."
 echo "Check your LLVM installation and/or modify the LLI
variable in testall.sh"
 exit 1
}

which "$LLI" >> $globallog || LLIFail


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

```
for file in $files
do
   case $file in
   *test-*)
       Check $file 2>> $globallog
       ;;
   *fail-*)
       CheckFail $file 2>> $globallog
       ;;
   *)
       echo "unknown file type $file"
       globalerror=1
       ;;
   esac
done

exit $globalerror
```

9.1.10 run.sh

```
#!/bin/bash
# Regression testing script for MicroC
# 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
```

```
# Path to the LLVM interpreter
LLI="lli"
#LLI="/usr/local/opt/llvm/bin/lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the microc compiler.  Usually "./microc.native"
# Try "_build/microc.native" if ocamlbuild was unable to create
a symbolic link.
GMM="./gmm.native"
#MICROC="_build/microc.native"

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

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

```
}

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

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

    echo -n "$basename..."

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

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.ll ${basename}.s
${basename}.exe ${basename}.out" &&
```

```
    Run "$GMM" "$1" ">" "${basename}.ll" &&
    Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">"
"${basename}.s" &&
    Run "$CC" "-o" "${basename}.exe" "${basename}.s" "-pthread"
"${src}builtin.o" &&
    Run "./${basename}.exe"
    # Run "./${basename}.exe" > "${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
}

Check $1
# rm *.ll *.s *.exe
```

9.2 Git Logs

```
commit 9ccb29815adfc30b75735e49a9506aac64b67c18
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Mon Apr 26 18:24:27 2021 -0700

    add script to convert all test files to latex

 compiler2latex.sh | 41 ++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 41 insertions(+)

commit 230e61b39f7a0e179ba5d846b5f0bfb867534daa
Author: cc4351 <cc4351@columbia.edu>
Date:   Mon Apr 26 11:22:30 2021 -0600

    organize repo

 demo/find-waldo-fib.gmm   | 1 +
 build.sh => misc/build.sh | 0
 misc/run.sh               | 2 +-
 todo.txt => misc/todo.txt | 0
 4 files changed, 2 insertions(+), 1 deletion(-)

commit 4bd6ae3b040224417045ebad3ab4548ef9a8ca73
Author: cc4351 <cc4351@columbia.edu>
Date:   Mon Apr 26 11:21:42 2021 -0600

    do not remove .out

 Makefile | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit e41d915e90fd050602104df40c3308f7f4319894
```

Author: cc4351 <cc4351@node0.cc4351-qv97276.lsm-rep-pg0.utah.cloudlab.us>
Date: Mon Apr 26 11:20:29 2021 -0600

    codegen gofunc code trim down

 src/codegen.ml | 122 ++++++++++++++++++++----------------------------------
 1 file changed, 40 insertions(+), 82 deletions(-)

commit d952a34eccebd1cb82db8ac4d63a6142055a20be
Author: cc4351 <cc4351@node0.cc4351-qv97276.lsm-rep-pg0.utah.cloudlab.us>
Date: Mon Apr 26 10:10:47 2021 -0600

    semant check array<void>

 Demo.txt | 1 -
 make-run.sh | 84 --
 src/semant.ml | 2 +-
 3 files changed, 1 insertion(+), 86 deletions(-)

commit aecc1d4d3b850bd00a78a9171ef92d4d573007ac
Merge: df67c52 8d3906b
Author: cc4351 <cc4351@columbia.edu>
Date: Mon Apr 26 09:40:13 2021 -0600

    Merge branch 'clean' of https://github.com/keyuyan1145/Go-- into clean

commit df67c527049f505cbbac585442e1c2d5ee36395d
Author: cc4351 <cc4351@columbia.edu>
Date: Mon Apr 26 09:40:03 2021 -0600

    rm unused marco

 src/builtin.c | 1 -
 1 file changed, 1 deletion(-)

commit 8d3906b94e7866cb607e6bc8a6c3fd4b79b65b9e
Merge: 6abc6e1 4285b2f
Author: Arya Zhao <lz2650@columbia.edu>
Date: Sun Apr 25 09:13:29 2021 -0700

    Merge branch 'clean' of github.com:keyuyan1145/Go-- into clean

commit 6abc6e14d7a12f2f6571cd387ed80c398ba5b785
Author: Arya Zhao <lz2650@columbia.edu>
Date: Sun Apr 25 09:13:21 2021 -0700

    add gofunc arguments failure test; gofunc test nit;

 tests/fail-gofunc3.err | 1 +
 tests/fail-gofunc3.gmm | 23 +++++++++++++++++++++++
 tests/test-gofunc1.gmm | 6 +++---
 3 files changed, 27 insertions(+), 3 deletions(-)

commit 4285b2ffc808e2801de2e702615f389efdfa4bcb
Author: cc4351 <cc4351@columbia.edu>
Date: Sun Apr 25 06:20:37 2021 -0600

    waldo w better use of channel and string

 demo/find-waldo-fib.gmm | 79 +++++++++++++++++++++++++++++--
 demo/find-waldo.gmm | 74 +++++++++++++++++++++++++++--
 2 files changed, 92 insertions(+), 61 deletions(-)

commit 4a31046f468b4a0d8f21af8c18013aad55ac0cdb
Author: cc4351 <cc4351@columbia.edu>
Date: Sun Apr 25 03:36:54 2021 -0600

    shrink test size below test timing thresh

src/semant.ml | 2 +-
tests/test-gocall-stress.gmm | 3 +--
2 files changed, 2 insertions(+), 3 deletions(-)

commit 7b2ddc9851b20f594c025349202ea5c1a81c02bb
Merge: 5ad4933 4929abc
Author: cc4351 <cc4351@columbia.edu>
Date: Sun Apr 25 03:32:37 2021 -0600

Merge branch 'clean' of https://github.com/keyuyan1145/Go-- into clean

commit 5ad493390ffce6b8864edf8011d15be54917bedf
Author: cc4351 <cc4351@columbia.edu>
Date: Sun Apr 25 03:32:24 2021 -0600

misc repo organization

build.sh | 4 ++
run.sh => misc/run.sh | 2 +
testall.txt | 117 ---
3 files changed, 6 insertions(+), 117 deletions(-)

commit c2b009c5b004e57a822683ddcf35df09cc65ddcc
Author: cc4351 <cc4351@columbia.edu>
Date: Sun Apr 25 03:31:11 2021 -0600

updated demo

demo/find-waldo-fib.gmm | 124
++
demo/find-waldo.gmm | 115 ++++++++++++++++++++++----------------------
2 files changed, 183 insertions(+), 56 deletions(-)

```
commit 4929abc9dc042e14afd792f30ca04eaeb0238871
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 22:20:26 2021 -0700

    update readme

 README | 60 ++------------------------------------------------------
 1 file changed, 2 insertions(+), 58 deletions(-)

commit afb0233d1b47c4f8ec09c4d931faa09c87992153
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 22:17:24 2021 -0700

    fix gofunc tests; add gofunc failure tests; pass ALL tests

 tests/fail-gofunc.err        |    2 +-
 tests/fail-gofunc.gmm        |   36 +-
 tests/fail-gofunc2.err       |    1 +
 tests/fail-gofunc2.gmm       |   11 +
 tests/test-gocall-stress.gmm |    2 -
 tests/test-gofunc1.gmm       |    5 +-
 tests/test-gofunc1.out       | 1198 ----------------------------------------
 tests/test-gofunc2.gmm       |   32 --
 tests/test-gofunc2.out       |    0
 tests/test-gofunc3.gmm       |   34 --
 tests/test-gofunc3.out       |    0
 11 files changed, 22 insertions(+), 1299 deletions(-)

commit 7578867e14d886465eaf563a5325634a4b0085c1
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 21:51:48 2021 -0700

    add script diff sort; add new lines to output files
```

```
 README                  |    4 +-
 testall.sh              |    6 +-
 tests/fail-array-types.err    |    2 +-
 tests/fail-channel-type.err   |    2 +-
 tests/fail-ppmm.err           |    2 +-
 tests/fail-struct-field.err   |    2 +-
 tests/fail-struct-types.err   |    2 +-
 tests/fail-void.err           |    2 +-
 tests/test-array-types.out    |    2 +-
 tests/test-array.out          |    2 +-
 tests/test-bool.out           |    2 +-
 tests/test-channel-bool.out   |    2 +-
 tests/test-channel-float.out  |    2 +-
 tests/test-channel-int.out    |    2 +-
 tests/test-channel-string.out |    2 +-
 tests/test-channel-struct.out |    2 +-
 tests/test-global-array.out   |    2 +-
 tests/test-goargs.out         |    2 +-
 tests/test-gocall-stress.out  |    0
 tests/test-gocall-stress2.out |    0
 tests/test-gofunc1.gmm        |   24 +-
 tests/test-gofunc1.out        | 1398 ++++++++++++++++++++++++++++++++++++++++-
 tests/test-gofunc2.out        |    0
 tests/test-gofunc3.out        |    0
 tests/test-integer.out        |    2 +-
 tests/test-ops2.out           |    2 +-
 tests/test-ppmm.out           |    2 +-
 tests/test-simple-string.out  |    2 +-
 tests/test-struct.out         |    2 +-
 29 files changed, 1444 insertions(+), 30 deletions(-)

commit 2acc7f7d633b4377e2ccd31c3feeeaeba02dd9b8
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 16:44:37 2021 -0700
```

```
    add struct failure tests; fix string test; rename test

 README                                          |  6 ++----
 tests/fail-struct-field.err                     |  1 +
 tests/fail-struct-field.gmm                     | 14 ++++++++++++++
 tests/fail-struct-types.err                     |  1 +
 tests/fail-struct-types.gmm                     | 13 +++++++++++++
 tests/{test-struct-channel.gmm => test-channel-struct.gmm} |  0
 tests/{test-struct-channel.out => test-channel-struct.out} |  0
 tests/test-simple-string.out                    |  1 +
 tests/test-struct.out                           |  7 +++++++
 9 files changed, 39 insertions(+), 4 deletions(-)

commit cd8eafd56635d2cabb2c814b35c826177a512b27
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 16:17:55 2021 -0700

    fix channel tests; add array failure test

 README                      |  1 -
 tests/fail-array-types.err  |  1 +
 tests/fail-array-types.gmm  | 10 ++++++++++
 tests/test-global-channel.gmm | 22 ----------------------
 tests/test-global-channel.out |  9 ---------
 5 files changed, 11 insertions(+), 32 deletions(-)

commit e990c0bfd1b4088502af74df3960e4424824fd05
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 16:05:32 2021 -0700

    fix array tests: delete array overflow test, fix global array test

 README                  |  5 +----
```

```
 tests/fail-array1.err       |  0
 tests/fail-array1.gmm       | 11 -----------
 tests/test-global-array.out |  1 -
 4 files changed, 1 insertion(+), 16 deletions(-)

commit 28f3b6fab3c08073c18041024abdeb8ceca02855
Merge: d5d8aba e7bf973
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 15:56:02 2021 -0700

    Merge branch 'clean' of github.com:keyuyan1145/Go-- into clean

commit d5d8aba64a8ace568f4ab28c07d64fbaea2e1269
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 15:55:44 2021 -0700

    test all primitive types for channel

 README                      |   3 --
 tests/test-channel-bool.gmm   |  21 +++++++++
 tests/test-channel-bool.out   | 100 ++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-channel-float.gmm  |  29 ++++++++++++
 tests/test-channel-float.out  | 100 ++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-channel-int.gmm    |  28 ++++++++++++
 tests/test-channel-int.out    | 100 ++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-channel-string.gmm |  47 ++++++++++++++++++++
 tests/test-channel-string.out |  10 +++++
 9 files changed, 435 insertions(+), 3 deletions(-)

commit ff1d89d24fc35af0d6a82e2ace0f3f5e49550e1c
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Sat Apr 24 12:45:32 2021 -0700

    fix channel bool test; rename channel tests; update readme
```

```
 README                     | 23 ++++++++++
 tests/test-bool-channel.gmm   |  29 ------------
 tests/test-bool-channel.out   | 103 -----------------------------------------
 tests/test-float-channel.gmm  |  31 -------------
 tests/test-float-channel.out  | 100 ---------------------------------------
 tests/test-simple-channel.gmm |  33 --------------
 tests/test-string-channel.gmm |  47 -------------------
 tests/test-string-channel.out |  10 ----
 8 files changed, 23 insertions(+), 353 deletions(-)

commit e7bf9731c4bbf98101ec057484d95710839a28c1
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 12:15:52 2021 -0600

    channel type comparison update

 src/semant.ml | 5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)

commit 48a66f5dbbdf3f7178a44b3b1eb390210d131c4a
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 12:15:38 2021 -0600

    demo update with struct channel

 demo/dull-waldo.gmm | 24 ++++++++++++++++++------
 demo/find-waldo.gmm | 51 ++++++++++++++++++++++++++++++++++++---------------
 2 files changed, 54 insertions(+), 21 deletions(-)

commit 1c8676f1715106833a2ffec69e12598a623947f4
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 12:15:25 2021 -0600
```

```
    add test case

 Makefile | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 0c181790750c1c8400e0b44994468643f2c51882
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 12:15:08 2021 -0600

    support for struct channel; clear warning

 src/builtin.c  | 6 ++++++
 src/codegen.ml | 9 +++++----
 2 files changed, 11 insertions(+), 4 deletions(-)

commit 2973fe0a57af61b33f50a74c8dea654276caa6a4
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 18:08:37 2021 +0000

    new test case for struct channel;update stress

 tests/test-gocall-stress2.gmm | 29 ++++++++++++++++++++---------
 tests/test-struct-channel.gmm | 33 +++++++++++++++++++++++++++++++++
 tests/test-struct-channel.out |  5 +++++
 3 files changed, 58 insertions(+), 9 deletions(-)

commit ae42e01d507e61e50c64bb23111cf2da01065c7a
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:25:42 2021 -0600

    save arr overflow for future

 future-tests/fail-array1.gmm | 11 ++++++++++
 1 file changed, 11 insertions(+)
```

```
commit 316b34b1ca3e800860faabae631ae36527c23790
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:25:24 2021 -0600

    update test output and gmm

 tests/test-bool-channel.out   | 103 ++++++++++++++++++++++++++++++++++++++++
 tests/test-float-channel.out  | 109 ++++++++++++++++++++++++++++++++++++++++++----
 tests/test-for-pp.out         |  19 +-------
 tests/test-global-channel.gmm |   1 -
 tests/test-global-channel.out |   9 ++++
 tests/test-goargs.out         |   5 ++
 6 files changed, 218 insertions(+), 28 deletions(-)

commit b14db1633354bebce092b8025d67b174b5c7efe6
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:24:43 2021 -0600

    update src format

 src/codegen.ml | 3 ++-
 src/semant.ml  | 2 +-
 2 files changed, 3 insertions(+), 2 deletions(-)

commit f47be5984f700873ba028dacd5f31073be28117e
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:23:24 2021 -0600

    add script;testall.txt listing of failed cases

 Makefile           |  25 +++++++----
 misc/format-list.sh |   4 ++
 testall.txt        | 117 +++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
 3 files changed, 138 insertions(+), 8 deletions(-)

commit fbe6eade17297367f59c384c26a26f235684bef5
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 10:20:47 2021 +0000

    rm printfib test

 tests/test-printfib.gmm | 6 ------
 1 file changed, 6 deletions(-)

commit 11a399391a92795e42b2bd4450d316c10587ca77
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 10:19:50 2021 +0000

    rm redundant test

 tests/test-anon-chan-dec.gmm | 8 --------
 1 file changed, 8 deletions(-)

commit d7246e4799d421ea0ca2b9b3dae28766ee7c1be9
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:05:14 2021 -0600

    add fail void tests

 tests/fail-void.err | 1 +
 tests/fail-void.gmm | 4 ++++
 2 files changed, 5 insertions(+)

commit c721ea8697cde6b0bf5dd98ca13886ce4f8e2b77
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:04:57 2021 -0600
```

```
    add test-global-array.out

 tests/test-global-array.out | 2 ++
 1 file changed, 2 insertions(+)

commit f7d98530ef2cfb008ab34c1750f10ce5a4504d81
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 04:04:36 2021 -0600

    add type specific operator tests

 tests/fail-array-channel.gmm |  6 ------
 tests/fail-array-init.err    |  1 -
 tests/fail-array-init.gmm    |  9 ---------
 tests/fail-array-typ.gmm     | 35 ---------------------------------
 tests/test-bool.gmm          | 15 +++++++++++++++
 tests/test-bool.out          |  9 +++++++++
 tests/test-float.gmm         | 47 +++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-float.out         | 15 +++++++++++++++
 tests/test-integer.gmm       | 47 +++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-integer.out       | 15 +++++++++++++++
 10 files changed, 148 insertions(+), 51 deletions(-)

commit c9a72558353a0d1fd9177d50ead0846ad38d79bd
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 24 03:58:53 2021 -0600

    channel overflow test case removed

 tests/fail-channel-overflow.gmm | 10 ----------
 1 file changed, 10 deletions(-)

commit 1b4e7507c45043c5847b4d2e7c4cb1d87fcbcb62
Author: cc4351 <cc4351@columbia.edu>
```

Date: Sat Apr 24 03:58:30 2021 -0600

    add missing .err files

 tests/fail-channel-type.err | 1 +
 tests/fail-channel-type.gmm | 12 +++---------
 tests/fail-gofunc.err | 1 +
 3 files changed, 5 insertions(+), 9 deletions(-)

commit c3bfaaaaa0b576cd97cc47b897a1a6b65877c80e
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 24 03:51:18 2021 -0600

    remove printbig test cases

 tests/fail-printbig.err | 1 -
 tests/fail-printbig.gmm | 2 --
 tests/test-printbig.gmm | 25 -------------
 tests/test-printbig.out | 88 --
 4 files changed, 116 deletions(-)

commit 98c5ca4fea6de6f6488485d13eab9baa98395891
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 24 02:58:39 2021 -0600

    rm unnecessary c files

 src/enc-dec-cond-var.c | 50 --
 1 file changed, 50 deletions(-)

commit acd9ac93344dcb62930b762aa29826046242a9f5
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 24 02:51:29 2021 -0600

```
    remove gofunc/func keyworkd from scanner/parser

 src/parser.mly  | 2 +-
 src/scanner.mll | 2 --
 2 files changed, 1 insertion(+), 3 deletions(-)

commit d35f0b9097e4cc11f2253aba040f75ab92a72d55
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Apr 24 03:04:10 2021 -0400

    clean up unnecssary comment

 src/builtin.c     | 18 +------------
 src/codegen.ml    | 27 ++++----------------
 src/gmm.ml        |  2 +-
 src/parser.mly    |  4 +--
 src/printbig.c    | 75 ----------------------------------------------------
 src/scanner.mll   |  3 +--
 src/semant.ml     |  4 +--
 src/test-print.ml |  3 ---
 8 files changed, 11 insertions(+), 125 deletions(-)

commit 98d30c273e11624a2bb20e5ab892b3e0ed290c61
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Apr 24 01:16:12 2021 -0400

    remove printbig

 Makefile       |  9 ++-------
 make-run.sh    |  2 +-
 run.sh         |  2 +-
 src/codegen.ml | 14 --------------
 src/semant.ml  |  4 +---
 testall.sh     |  8 +-------
```

```
 6 files changed, 6 insertions(+), 33 deletions(-)

commit b234ad4962d95b911c780238be523cf8550941c8
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Apr 24 01:02:14 2021 -0400

    move files around

 Go--.tar.gz                                | Bin 19902 -> 0 bytes
 Makefile                                   |  38 +++++++----------
 make-run.sh                                |   3 +-
 misc/fail-array-channel.err                |   1 +
 misc/fail-array-typ.err                    |   1 +
 fail-array1.err => misc/fail-array1.err    |   0
 misc/fail-channel-overflow.err             |   0
 .../fail-channel-type.err                  |   0
 misc/fail-gofunc.err                       |   1 +
 get-ll.sh => misc/get-ll.sh                |   0
 printfib.c                                 |  39 ----------------------
 run.sh                                     |   3 +-
 ast.ml => src/ast.ml                       |   0
 builtin.c => src/builtin.c                 |   0
 codegen.ml => src/codegen.ml               |   0
 enc-dec-cond-var.c => src/enc-dec-cond-var.c |   0
 gmm.ml => src/gmm.ml                       |   0
 parser.mly => src/parser.mly               |   0
 printbig.c => src/printbig.c               |   0
 sast.ml => src/sast.ml                     |   0
 scanner.mll => src/scanner.mll             |   0
 semant.ml => src/semant.ml                 |   0
 test-print.ml => src/test-print.ml         |   0
 testall.sh                                 |   5 +--
 24 files changed, 22 insertions(+), 69 deletions(-)
```

```
commit f7522c6672ab86c91634420dd84580938f295ba6
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 23:24:09 2021 +0800

    gofunction main not allowed

 builtin.c                  |  2 +-
 semant.ml                  |  6 +++++-
 tests/fail-gofunc-main.err |  1 +
 tests/test-for-pp.gmm      |  7 +++++++
 tests/test-for-pp.out      | 40 ++++++++++++++++++++++++++++++++++++----------
 tests/test-gocall-stress2.gmm |  4 ----
 todo.txt                   | 44 ++++++++++++++++++++++++++++++++++++----------
 7 files changed, 65 insertions(+), 39 deletions(-)

commit fb3cb9bbaa3accf76d06e5349471ba1e278a55f1
Merge: d4a1a6a 1129d3d
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 23:04:00 2021 +0800

    Merge branch 'comment' of github.com:keyuyan1145/Go-- into comment

commit d4a1a6a5f21b09ffb3b7962e43ac09dbdcddba77
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 23:03:39 2021 +0800

    ppmm for while test

 tests/test-for-pp.gmm |  6 ++++++
 tests/test-for-pp.out | 41 +++++++++++++++++++++++++++++++++++++++++
 2 files changed, 47 insertions(+)

commit aeaf53bbebf1cd5849b5a00493ae2e969ae35eb2
Author: cc4351 <cc4351@columbia.edu>
```

```
Date:   Fri Apr 23 19:56:20 2021 +0800

    for loop ++/--

 tests/test-for-pp.gmm | 10 ++++++++++
 1 file changed, 10 insertions(+)

commit 1129d3d7907f927ea0acb6e7f5ff873b8de3f96d
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 03:39:25 2021 -0600

    stress test waldo working

 demo/dull-waldo.gmm | 8 +++++++-
 demo/find-waldo.gmm | 2 +-
 2 files changed, 8 insertions(+), 2 deletions(-)

commit 91118e8561f5e2c90fb72e86f49cf765af8278b9
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 17:30:10 2021 +0800

    clean semant

 semant.ml | 23 -----------------------
 1 file changed, 23 deletions(-)

commit af21b83e41db31061ae83f84fa06909cfc611ffc
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 17:25:31 2021 +0800

    stress test waldo

 demo/dull-waldo.gmm | 21 ++++++++++++++++++++-
 demo/find-waldo.gmm | 23 +++++++++++++++++++++--
```

```
 2 files changed, 41 insertions(+), 3 deletions(-)

commit d7783fe8a864cf2153ee58e3c78699dc5001db89
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 17:01:21 2021 +0800

    demo mapreduce find waldo

 demo/dull-waldo.gmm | 33 +++++++++++++++++++++++++++++++++
 demo/find-waldo.gmm |  6 ++++--
 2 files changed, 37 insertions(+), 2 deletions(-)

commit 6c2fbcab20d12986cca08d6da1d89fa23b24d9fc
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 16:50:33 2021 +0800

    add map reduce example

 demo/find-waldo.gmm | 59
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 59 insertions(+)

commit 44244f12a51ebd30fcfea78da8bb7ecb565d002a
Merge: fea4033 d949425
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 16:29:21 2021 +0800

    merge with comment_yang; threadpool impl

commit fea40335ec573ee440a1b90660501e384e7adf82
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 16:23:25 2021 +0800

    commit before merge
```

builtin.c | 2 --
codegen.ml | 25 ++++++++++++-------------
2 files changed, 12 insertions(+), 15 deletions(-)

commit 114f3621e5365559152b8d2f8f224efd1c07ff1b
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Apr 23 16:03:48 2021 +0800

    add string channel out

tests/test-string-channel.out | 10 ++++++++++
1 file changed, 10 insertions(+)

commit 5f9e2871e05b6a3c64ec910f7cdb62329aa47310
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Apr 23 16:01:36 2021 +0800

    string channel test

tests/test-string-channel.gmm | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit cfa3ba0c23d51110adfb8d345263496c5bbef228
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Apr 23 16:01:26 2021 +0800

    tab formatting

semant.ml | 85 ++++++++++++++++++++++++++++++-------------------------------
1 file changed, 42 insertions(+), 43 deletions(-)

commit 3c6d890317f9ed9a601875d51b197f06ef312297
Author: cc4351 <cc4351@columbia.edu>

```
Date:   Fri Apr 23 16:01:12 2021 +0800

    bugfix string channel

 builtin.c  |  8 ++++----
 codegen.ml | 25 ++++++++++++----------------
 2 files changed, 15 insertions(+), 18 deletions(-)

commit d94942573566567707c213663f5ba143b94848e4
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 00:58:11 2021 -0600

    trivial stress

 tests/test-gocall-stress2.gmm | 22 ++++++++++++++++++++++
 1 file changed, 22 insertions(+)

commit 040707640317ca59a6e3d71ebf47b55c53711783
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 00:54:42 2021 -0600

    second stress test

 tests/test-gocall-stress2.gmm | 22 ++++++++++++++++++++++
 1 file changed, 22 insertions(+)

commit 4fb01ee36b4d51d87c0a87576c25abb40f21b36a
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 00:49:02 2021 -0600

    do 100k op

 tests/test-gocall-stress.gmm | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 8c5afe1d13d1540067ae403fc737db4613b0ed05
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 00:39:03 2021 -0600

    stress test on server comment br

 builtin.c                   |  6 +++---
 tests/test-gocall-stress.gmm | 45 +++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 48 insertions(+), 3 deletions(-)

commit 234368763f0b4167ebd2b1c5afb0a0cc892a3af5
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 06:19:10 2021 +0000

    stress test on server

 builtin.c                   |  8 ++++----
 chn_feature.c               |  6 ------
 tests/test-gocall-stress.gmm | 23 +++++++++++------------
 3 files changed, 15 insertions(+), 22 deletions(-)

commit db47e0d93bad7ba30f165a9a8d5e022564263e9e
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 13:36:33 2021 +0800

    stress test mod

 builtin.c                   | 540 ++++++++++++++++++++++---------------------
 tests/test-gocall-stress.gmm |   9 +-
 2 files changed, 278 insertions(+), 271 deletions(-)

commit 0f9a96150aa997e7251dce24d840bfe5258783d9
Author: cc4351 <cc4351@columbia.edu>
```

Date: Fri Apr 23 12:27:21 2021 +0800

    stress gocall

 tests/test-gocall-stress.gmm | 41 ++
 1 file changed, 41 insertions(+)

commit 17a9fb889f6e98c1fb2e22e8c42cd38ce32f5178
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Apr 23 11:52:16 2021 +0800

    test channel data loss

 builtin.c | 4 +++-
 tests/test-float-channel.gmm | 26 ++++++++++++++++++++++---
 tests/test-string-channel.gmm | 4 ++--
 3 files changed, 28 insertions(+), 6 deletions(-)

commit efa6c8f7f5a58e8bfc2404ecb97b8b94e37815f4
Author: samlee815 <yl4111@columbia.edu>
Date: Thu Apr 22 23:51:13 2021 -0400

    use thread pool implementation

 builtin.c | 159 ++-
 codegen.ml | 2 +-
 semant.ml | 42 ++++++++++++-------
 tests/test-struct.gmm | 2 -
 4 files changed, 178 insertions(+), 27 deletions(-)

commit 5ce64843341ebd3b56f961d1213198d499dc7bfe
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Apr 23 02:16:27 2021 +0800

```
    add array and channel tests

 builtin.c                  |  7 +++----
 tests/fail-array-init.err  |  1 +
 tests/fail-array-init.gmm  |  9 +++++++++
 tests/fail-array-typ.gmm   | 35 +++++++++++++++++++++++++++++++++++
 tests/test-array.gmm       |  4 +---
 tests/test-string-channel.gmm | 47 +++++++++++++++++++++++++++++++++++++++++++++++
 6 files changed, 96 insertions(+), 7 deletions(-)

commit 99c140f3f6e056614b6b92e0bea623635ddc4c5e
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 02:09:35 2021 +0800

    add channel support for string

 builtin.c  | 17 ++++++++++-------
 codegen.ml | 39 +++++++++++++++++++++++++--------------
 2 files changed, 30 insertions(+), 26 deletions(-)

commit 7fedf8ea6257275f088f535503bfcce2a5c1e495
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 01:46:48 2021 +0800

    add float channel out in debug mode

 tests/test-float-channel.gmm | 3 ---
 tests/test-float-channel.out | 9 +++++++++
 2 files changed, 9 insertions(+), 3 deletions(-)

commit 0592f92b577e8db74c07e56ed4e706411a74e5d8
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 01:44:38 2021 +0800
```

```
    float channel working

 builtin.c | 14 +++++++-------
 1 file changed, 7 insertions(+), 7 deletions(-)

commit 253cb3d45c76cf1f9459129af076f63c7fe64a66
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 01:39:36 2021 +0800

    array all type test case

 tests/test-array-types.gmm | 42 ++++++++++++++++++++++++++++++++++++++++++
 tests/test-array-types.out |  9 +++++++++
 2 files changed, 51 insertions(+)

commit 6ae70b6c64786c30e9ed5494b23a1712a9030eb5
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 23 01:29:41 2021 +0800

    wrong test case;use printf works

 tests/fail-array2.err |  1 -
 tests/fail-array2.gmm | 11 -----------
 2 files changed, 12 deletions(-)

commit 22e57face4e7cb9b340b9c447f118416819acc70
Author: samlee815 <yl4111@columbia.edu>
Date:   Mon Apr 19 16:06:57 2021 -0400

    add support for gofunction argtypes: array,struct and string

 builtin.c              | 226 +++++++++++++++++++++++++++++++++++++++++++++-----
 codegen.ml             |  35 +++++---
 tests/test-goargs.gmm |  35 ++++++++
```

```
 tests/test-struct.gmm |  10 +++
 4 files changed, 274 insertions(+), 32 deletions(-)

commit bbfb789ada166466fe83d6ddb75f78015f708bc6
Author: samlee815 <yl4111@columbia.edu>
Date:   Sun Apr 18 13:25:06 2021 -0400

    lhs dot assign completed

 codegen.ml           | 12 ++++++++++++
 tests/test-struct.gmm |  3 ++-
 2 files changed, 14 insertions(+), 1 deletion(-)

commit 0c4033a88fc5afe3eff6acecfac591b70989f62c
Author: samlee815 <yl4111@columbia.edu>
Date:   Sun Apr 18 11:59:15 2021 -0400

    rhs dot access of struct implemented

 ast.ml              |  2 ++
 codegen.ml          | 18 ++++++++++++++-----
 parser.mly          |  5 ++++-
 sast.ml             |  2 ++
 scanner.mll         |  1 +
 semant.ml           | 17 ++++++++++++------
 tests/test-struct.gmm | 13 ++++++++++--
 7 files changed, 44 insertions(+), 14 deletions(-)

commit b1aa37932388085f60808ed17ce7bafac4bca548
Author: samlee815 <yl4111@columbia.edu>
Date:   Sun Apr 18 01:34:22 2021 -0400

    struct init completed
```

```
 ast.ml              | 19 ++++++++++++++++----
 codegen.ml          | 41 +++++++++++++++++++++++++++++++++++++++++-----
 parser.mly          | 47 +++++++++++++++++++++++++++++++++++++++++++++++-------
 sast.ml             |  7 +++++--
 scanner.mll         |  2 ++
 semant.ml           | 39 +++++++++++++++++++++++++++++++++++++---
 tests/test-arith1.gmm |  5 +++++
 tests/test-struct.gmm | 22 ++++++++++++++++++++++
 8 files changed, 161 insertions(+), 21 deletions(-)

commit e7bb3a3da36f149614dddb821009a40ea839e5f4
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Apr 17 14:05:33 2021 -0400

    finish string operations

 builtin.c          | 11 ++++++++---
 codegen.ml         | 17 ++++++++++++------
 make-run.sh        |  0
 tests/test-str.gmm |  5 ++++-
 tests/test-str.out |  2 ++
 tests/test-str2.gmm | 15 +++++++++++++++
 tests/test-str2.out |  2 ++
 7 files changed, 40 insertions(+), 12 deletions(-)

commit 08a46eedef8a64e96daac012f5a60d3f534f6a95
Merge: deca108 b85e9ce
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Apr 17 12:25:59 2021 -0400

    Merge branch 'comment' into comment_yang

commit deca108d8a33ac32e9185357cc8030f55fd4a055
Author: samlee815 <yl4111@columbia.edu>
```

```
Date:   Sat Apr 17 12:10:17 2021 -0400

    add string test

 {future-tests => tests}/test-str.gmm | 6 ++++--
 1 file changed, 4 insertions(+), 2 deletions(-)

commit b85e9ce36567e9f5aa2795360ca4d8c505fe8094
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:51:46 2021 +0800

    update make-run with builtin renaming

 make-run.sh | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit aea96d9b0d92231766dc538974056db84df2277a
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:42:35 2021 +0800

    fix merge conflict;add simple test case

 README                      | 3 +++
 parser.mly                  | 1 +
 tests/test-simple-string.gmm | 6 ++++++
 3 files changed, 10 insertions(+)

commit ded6ba97d6686065752cbdf0f381f783a4dc78d0
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:35:01 2021 +0800

    merge complete

 codegen.ml | 4 ++++
```

```
 1 file changed, 4 insertions(+)

commit af651c191570a9772d91ec31ffff4961a5c6beac
Merge: 685947d 83f871b
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:31:55 2021 +0800

    merge with comment_yang

commit 685947d3c3ffb69990e1ca9c8d2e5cb6f044e78e
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:26:18 2021 +0800

    ideas for demo

 Demo.txt | 1 +
 1 file changed, 1 insertion(+)

commit ee43c72720fa4829f58b672697c55bc95b16fc05
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:20:15 2021 +0800

    update test case names

 README | 11 ++++++-----
 1 file changed, 6 insertions(+), 5 deletions(-)

commit c0f305ead3487e986cdd2709b3f884221b690eb5
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:20:04 2021 +0800

    install clang when setup docker

 Dockerfile         |  4 +++-
```

```
 enc-dec-cond-var.c | 50 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 53 insertions(+), 1 deletion(-)

commit 5aa761877652c923624898e85d24be966390d789
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:19:40 2021 +0800

    rm comments

 semant.ml | 2 --
 1 file changed, 2 deletions(-)

commit 1d13bd1ad4bd9736132b0aa19b72662d4023bca3
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:19:28 2021 +0800

    update gocall for semaphores

 gocall.c | 110 ++++++++++++++++++++++++++++++++++++++----------------------------
 1 file changed, 51 insertions(+), 59 deletions(-)

commit 98b52aec216c1fad1944c4e1ddb696d9d0333e85
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:19:12 2021 +0800

    codegen chan for different data types

 codegen.ml | 10 ++++++++--
 1 file changed, 8 insertions(+), 2 deletions(-)

commit a548aca792089ce1635d293e4acbfeb3a4d4de9c
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:18:49 2021 +0800
```

```
    cleanup comments

 sast.ml | 2 --
 1 file changed, 2 deletions(-)

commit 8002e40668e5e5acc54e2fae41bf69a871f24408
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:18:32 2021 +0800

    fixed shift/reduce error

 parser.mly | 19 +++++++++++++++-----
 1 file changed, 14 insertions(+), 5 deletions(-)

commit 58c5bba1ec6ea0be89a19bc9c9e897ed914fd392
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:18:18 2021 +0800

    float in chan NOT WORKING

 tests/test-float-channel.gmm | 14 ++++++++++++++
 1 file changed, 14 insertions(+)

commit 4b163d80a824e4e16a0f14ee386a9cf19f200793
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 21:18:05 2021 +0800

    failed array channel

 tests/fail-array-channel.gmm | 6 ++++++
 1 file changed, 6 insertions(+)

commit d72a5999645ea78a9a1772f493379fd937b88c63
Author: cc4351 <cc4351@columbia.edu>
```

Date: Sat Apr 17 21:17:45 2021 +0800

    script to build and run all at once

 make-run.sh | 83
++
 1 file changed, 83 insertions(+)

commit d6241fe7e4059ac80cd55efe03e5a4bafcdb9cfe
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 17 21:17:30 2021 +0800

    script to get ll file

 get-ll.sh | 3 +++
 1 file changed, 3 insertions(+)

commit 76cbcef074fdd315bb8190bce22939b8078e201a
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 17 20:57:52 2021 +0800

    add boolean test case for channel

 tests/test-bool-channel.gmm | 29 ++++++++++++++++++++++++++++++
 1 file changed, 29 insertions(+)

commit c540b2be8fc1b80407acc468d4c15c07af64741d
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 17 19:00:45 2021 +0800

    blocking for channel overflow

 codegen.ml | 45 +++++++---------
 gocall.c | 115 ++++++++++++++++++++++++++++++++++++++----------

```
 tests/fail-channel-overflow.gmm |  10 ++++
 tests/test-anon-chan-dec.gmm    |   4 +-
 4 files changed, 115 insertions(+), 59 deletions(-)

commit bfa08190ddb2943d538453cae32de823fe321447
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 17:18:16 2021 +0800

    fix codegen for anon read

 codegen.ml | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)

commit 65136b875e355161e49d78a7813044ae3add19ba
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 17:17:50 2021 +0800

    test case for anon dec from channel

 tests/test-anon-chan-dec.gmm | 10 ++++++++++
 1 file changed, 10 insertions(+)

commit 36532a050431306e06a68af1744f4f346a45cc34
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 17 15:53:45 2021 +0800

    cond_wait working

 codegen.ml                  | 152 ++++++++++++++++----------------------
 tests/test-global-channel.gmm |  23 +++++--
 2 files changed, 76 insertions(+), 99 deletions(-)

commit 3d5ec8b22195a8b12c782f4ac79a91f99242c791
Author: cc4351 <cc4351@columbia.edu>
```

Date: Sat Apr 17 13:37:05 2021 +0800

    type correct codegen

 codegen.ml | 1 +
 1 file changed, 1 insertion(+)

commit a11e3ee0b5d77b015a629b9886696bc5dc7103a4
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Apr 17 12:13:10 2021 +0800

    exp with categorizing inputs

 codegen.ml | 73 ++++++++++++++++++++++++++----------------------------------
 1 file changed, 28 insertions(+), 45 deletions(-)

commit 8e3bf365de72b09e6b5690f0c8eecac944468f2e
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Apr 16 00:51:08 2021 +0800

    not working global channel map

 codegen.ml | 21 +++++++++++++++++++++-
 1 file changed, 20 insertions(+), 1 deletion(-)

commit 2843ef99202de6234a9c1e4f965022f827705eae
Author: cc4351 <cc4351@columbia.edu>
Date: Thu Apr 15 11:43:24 2021 +0000

    bugfix on gofunction call scope

 semant.ml | 12 ++++++------
 1 file changed, 6 insertions(+), 6 deletions(-)

```
commit 83f871b135c4a02cf838580d3cf911bad97ebc5c
Author: samlee815 <yl4111@columbia.edu>
Date:   Thu Apr 15 03:03:06 2021 -0400

    string concat

 Makefile                 |   6 +++---
 gocall.c => builtin.c    |  11 ++++++++++
 codegen.ml               |  23 ++++++++++++++++++++----
 future-tests/test-str.gmm |   2 +-
 printbig                 | Bin 8424 -> 0 bytes
 run.sh                   |   2 +-
 semant.ml                |   1 +
 testall.sh               |   2 +-
 8 files changed, 37 insertions(+), 10 deletions(-)

commit 0dac7baab3c4750a9544185731aba9dddec672bc
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Wed Apr 14 19:18:21 2021 -0700

    minor fix

 README                   |  10 +-
 _build/_digests          |  32 -
 _build/_log              |  82 --
 _build/ast.cmi           | Bin 2991 -> 0 bytes
 _build/ast.cmo           | Bin 5572 -> 0 bytes
 _build/ast.cmx           | Bin 755 -> 0 bytes
 _build/ast.ml            | 125 ---
 _build/ast.ml.depends    |   1 -
 _build/ast.o             | Bin 25232 -> 0 bytes
 _build/codegen.cmi       | Bin 5089 -> 0 bytes
 _build/codegen.cmo       | Bin 18851 -> 0 bytes
 _build/codegen.cmx       | Bin 1611 -> 0 bytes
```

```
_build/codegen.ml        |  522 ----------
_build/codegen.ml.depends |    1 -
_build/codegen.o         |  Bin 95120 -> 0 bytes
_build/gmm.cmi           |  Bin 730 -> 0 bytes
_build/gmm.cmo           |  Bin 1892 -> 0 bytes
_build/gmm.cmx           |  Bin 843 -> 0 bytes
_build/gmm.ml            |   32 -
_build/gmm.ml.depends    |    1 -
_build/gmm.native        |  Bin 1333864 -> 0 bytes
_build/gmm.o             |  Bin 7832 -> 0 bytes
_build/ocamlc.where      |    1 -
_build/parser.cmi        |  Bin 1478 -> 0 bytes
_build/parser.cmx        |  Bin 6573 -> 0 bytes
_build/parser.ml         | 1027 -------------------------
_build/parser.ml.depends |    1 -
_build/parser.mli        |   52 --
_build/parser.mli.depends |    1 -
_build/parser.mly        |  141 ----
_build/parser.o          |  Bin 46224 -> 0 bytes
_build/sast.cmi          |  Bin 2448 -> 0 bytes
_build/sast.cmo          |  Bin 4944 -> 0 bytes
_build/sast.cmx          |  Bin 615 -> 0 bytes
_build/sast.ml           |   98 ---
_build/sast.ml.depends   |    1 -
_build/sast.o            |  Bin 23112 -> 0 bytes
_build/scanner.cmi       |  Bin 805 -> 0 bytes
_build/scanner.cmo       |  Bin 27009 -> 0 bytes
_build/scanner.cmx       |  Bin 25668 -> 0 bytes
_build/scanner.ml        | 1873 ------------------------------------------
_build/scanner.ml.depends |    1 -
_build/scanner.mll       |   64 --
_build/scanner.o         |  Bin 34128 -> 0 bytes
_build/semant.cmi        |  Bin 4962 -> 0 bytes
_build/semant.cmo        |  Bin 9430 -> 0 bytes
```

```
 _build/semant.cmx         | Bin 1453 -> 0 bytes
 _build/semant.ml          | 249 ------
 _build/semant.ml.depends  |   1 -
 _build/semant.o           | Bin 46024 -> 0 bytes
 fail-array1.exe           | Bin 13096 -> 0 bytes
 fail-array1.ll            |  41 -
 fail-array1.s             |  49 --
 gmm.native                |   1 -
 gocall.o                  | Bin 5904 -> 0 bytes
 printbig.o                | Bin 2080 -> 0 bytes
 printfib.o                | Bin 2104 -> 0 bytes
 test-gofunc2.ll           |   0
 tests/fail-array1.gmm     |   1 +
 tests/fail-gofunc.gmm     |  32 +
 tests/test-gofunc2.gmm    |  14 +-
 tests/test-gofunc3.gmm    |  34 +-
 62 files changed, 52 insertions(+), 4436 deletions(-)

commit 26a1d27030fe07cc032aedd7879af41210eef060
Merge: 90e242d c91ce9d
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Wed Apr 14 19:17:13 2021 -0700

    Merge branch 'comment' of github.com:keyuyan1145/Go-- into comment

commit 90e242d90866f869098f4fbe39f8811fbc402d75
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Wed Apr 14 18:45:12 2021 -0700

    minor fix

 README                    |   4 +-
 _build/_digests           |  32 +
 _build/_log               |  82 ++
```

```
_build/ast.cmi           | Bin 0 -> 2991 bytes
_build/ast.cmo           | Bin 0 -> 5572 bytes
_build/ast.cmx           | Bin 0 -> 755 bytes
_build/ast.ml            | 125 +++
_build/ast.ml.depends    |   1 +
_build/ast.o             | Bin 0 -> 25232 bytes
_build/codegen.cmi       | Bin 0 -> 5089 bytes
_build/codegen.cmo       | Bin 0 -> 18851 bytes
_build/codegen.cmx       | Bin 0 -> 1611 bytes
_build/codegen.ml        | 522 ++++++++++++++
_build/codegen.ml.depends |   1 +
_build/codegen.o         | Bin 0 -> 95120 bytes
_build/gmm.cmi           | Bin 0 -> 730 bytes
_build/gmm.cmo           | Bin 0 -> 1892 bytes
_build/gmm.cmx           | Bin 0 -> 843 bytes
_build/gmm.ml            |  32 +
_build/gmm.ml.depends    |   1 +
_build/gmm.native        | Bin 0 -> 1333864 bytes
_build/gmm.o             | Bin 0 -> 7832 bytes
_build/ocamlc.where      |   1 +
_build/parser.cmi        | Bin 0 -> 1478 bytes
_build/parser.cmx        | Bin 0 -> 6573 bytes
_build/parser.ml         | 1027 ++++++++++++++++++++++++++
_build/parser.ml.depends |   1 +
_build/parser.mli        |  52 ++
_build/parser.mli.depends |   1 +
_build/parser.mly        | 141 ++++
_build/parser.o          | Bin 0 -> 46224 bytes
_build/sast.cmi          | Bin 0 -> 2448 bytes
_build/sast.cmo          | Bin 0 -> 4944 bytes
_build/sast.cmx          | Bin 0 -> 615 bytes
_build/sast.ml           |  98 +++
_build/sast.ml.depends   |   1 +
_build/sast.o            | Bin 0 -> 23112 bytes
```

```
 _build/scanner.cmi       | Bin 0 -> 805 bytes
 _build/scanner.cmo       | Bin 0 -> 27009 bytes
 _build/scanner.cmx       | Bin 0 -> 25668 bytes
 _build/scanner.ml        | 1873 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 _build/scanner.ml.depends |    1 +
 _build/scanner.mll       |   64 ++
 _build/scanner.o         | Bin 0 -> 34128 bytes
 _build/semant.cmi        | Bin 0 -> 4962 bytes
 _build/semant.cmo        | Bin 0 -> 9430 bytes
 _build/semant.cmx        | Bin 0 -> 1453 bytes
 _build/semant.ml         | 249 ++++++
 _build/semant.ml.depends |    1 +
 _build/semant.o          | Bin 0 -> 46024 bytes
 fail-array1.exe          | Bin 0 -> 13096 bytes
 fail-array1.ll           |   41 +
 fail-array1.s            |   49 ++
 gmm.native               |    1 +
 gocall.o                 | Bin 0 -> 5904 bytes
 printbig.o               | Bin 0 -> 2080 bytes
 printfib.o               | Bin 0 -> 2104 bytes
 test-gofunc2.ll          |    0
 tests/fail-array1.gmm    |    2 +-
 tests/test-gofunc2.gmm   |    2 +-
 60 files changed, 4401 insertions(+), 4 deletions(-)

commit c91ce9d143a5c2594d6ea84bf55b3cc42a264f86
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Apr 14 21:28:16 2021 -0400

    add wrap_around as a separate function

 chn_feature.c | 6 ++++++
 1 file changed, 6 insertions(+)
```

```
commit 9c85d208bbee36d232bc9e7aa7b3ff3fb1d3c012
Author: Arya Zhao <lz2650@columbia.edu>
Date:   Wed Apr 14 17:55:16 2021 -0700

    test array, gofunc, ppmm

    update readme

 README                                     | 92 ++++++++++++++++++------
 fail-array1.err                            |  0
 fail-channel-type.err                      |  1 +
 future-tests/fail-func8 2.gmm              | 13 ++++
 future-tests/fail-func9 2.err              |  1 +
 future-tests/test-str 2.gmm                |  3 +
 tests/fail-array1.err                      |  0
 tests/fail-array1.gmm                      | 10 +++
 tests/fail-array2.err                      |  1 +
 tests/fail-array2.gmm                      | 11 +++
 tests/fail-ppmm.err                        |  1 +
 tests/fail-ppmm.gmm                        |  7 ++
 tests/test-array.out                       |  3 +
 tests/test-fib 2.out                       |  6 ++
 tests/test-go.gmm                          | 36 ----------
 tests/{test-gofunction.gmm => test-gofunc1.gmm} |  0
 tests/{test-gofunction.out => test-gofunc1.out} |  0
 tests/test-gofunc2.gmm                     | 32 +++++++++
 tests/test-gofunc3.gmm                     | 58 ++++++++++++++++
 19 files changed, 219 insertions(+), 56 deletions(-)

commit 7cccf60262e749944915f9c34026c0cbb79bbaf0
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 14 18:04:33 2021 +0000

    moving func to c
```

```
 codegen.ml | 43 +++++++++++++++++++------------------------------
 gocall.c   | 68 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++----------
 2 files changed, 77 insertions(+), 34 deletions(-)

commit e3f7ee8944858f188f258074c597cb05e58b3ca2
Author: samlee815 <yl4111@columbia.edu>
Date:   Sun Apr 11 19:48:12 2021 -0400

    added string and prints

 ast.ml                  |  6 +++++-
 codegen.ml              | 14 ++++++++++----
 fail-channel-type.err   |  1 +
 future-tests/test-str.gmm | 10 ++++++++--
 parser.mly              |  5 ++++-
 run.sh                  |  0
 sast.ml                 |  2 ++
 scanner.mll             |  7 +++++++
 semant.ml               |  5 ++++-
 9 files changed, 41 insertions(+), 9 deletions(-)

commit 5be6b4a06b6aed4d11e727bd45606dffc57673f8
Author: cc4351 <cc4351@columbia.edu>
Date:   Sun Apr 11 15:15:36 2021 +0800

    cond_var semi functional

 codegen.ml | 162 ++++++++++++++++++++++++++++++++++++++++++++-------------------------------
 1 file changed, 103 insertions(+), 59 deletions(-)

commit a37000d2caa47b4657ab9c4fabd6d704b45e74cd
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 10 21:06:50 2021 +0800
```

```
    lock functionality working; update todo

 codegen.ml | 29 ++++++++++++++++++++++++------
 todo.txt   | 14 +++++++++-----
 2 files changed, 32 insertions(+), 11 deletions(-)

commit 1af62039292f707065cda61d8cc020ad8a594f25
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 10 16:49:17 2021 +0800

    codegen support for channel global vars

 codegen.ml | 26 ++++++++------------------
 1 file changed, 8 insertions(+), 18 deletions(-)

commit 9688817cd1c8f9b59564e947ec800151b80c75d2
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 10 16:49:02 2021 +0800

    init channel & array as global vars

 tests/test-global-array.gmm   |  8 ++++++++
 tests/test-global-channel.gmm | 12 ++++++++++++
 2 files changed, 20 insertions(+)

commit ddc3e995f185e9580ad9e5d7f34e1e907263e72d
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Apr 10 15:21:21 2021 +0800

    basic enc/dec working; circular array tbd

 codegen.ml                  | 65 ++++++++++++++++++++++++++----------------
 tests/test-simple-channel.gmm | 20 ++++++++++++-----
```

```
 2 files changed, 44 insertions(+), 41 deletions(-)

commit 546fff707135b21b01b91b8332e78e659d16cc8c
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Apr 9 18:42:15 2021 +0800

    NewChannel with other fields work

 codegen.ml | 98 ++++++++++++++++++++++++++++++++++++++++++++++------------------
 sast.ml    |  2 ++
 2 files changed, 74 insertions(+), 26 deletions(-)

commit de11b78b6ce0b3fb028ccff6f3a6d9ccb67419d6
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 7 20:05:57 2021 +0800

    add preliminary channel function

 ast.ml     | 37 ++++++++++++++++---------------------
 codegen.ml | 64 ++++++++++++++++++++++++++++++++++++++++------------------------
 parser.mly | 11 ++++++-----
 sast.ml    |  8 +++++++-
 semant.ml  | 34 +++++++++++++++++++++-------------
 5 files changed, 90 insertions(+), 64 deletions(-)

commit aa8cfdf1a307043fabb3406257018e64ea685945
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 7 20:05:15 2021 +0800

    update todos for channel and testing

 todo.txt | 15 ++++++++++++---
 1 file changed, 12 insertions(+), 3 deletions(-)
```

commit a2824077e2d942428ad43946718696b9786c3921
Author: cc4351 <cc4351@columbia.edu>
Date: Wed Apr 7 20:05:02 2021 +0800

    new tests for channel and array

 tests/test-array.gmm | 2 +-
 tests/test-simple-channel.gmm | 17 ++++++++++++-----
 2 files changed, 13 insertions(+), 6 deletions(-)

commit 2c76a7292dde0f47b33a9139365ff2f89c23ba3f
Author: cc4351 <cc4351@columbia.edu>
Date: Wed Apr 7 17:18:16 2021 +0800

    modified channel test case

 tests/test-simple-channel.gmm | 8 ++++++--
 1 file changed, 6 insertions(+), 2 deletions(-)

commit 2a53f76210ac679394fe662d97d532035bab4a15
Author: cc4351 <cc4351@columbia.edu>
Date: Wed Apr 7 17:17:52 2021 +0800

    array with int working;channel not working;parser conflict

 ast.ml | 11 +++---
 codegen.ml | 85 +++++++++++++++++++++++++++++---------------
 parser.mly | 4 +++
 sast.ml | 4 +++
 scanner.mll | 2 ++
 semant.ml | 21 ++++++++++--
 tests/fail-channel-type.gmm | 13 +++++++
 tests/test-array.gmm | 14 ++++++++
 8 files changed, 120 insertions(+), 34 deletions(-)

```
commit bb630bd405aabc3a11005ddcf0dc07753352ea54
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 7 12:55:49 2021 +0800

    add channel simple test case

 tests/test-simple-channel.gmm | 16 ++++++++++++++++
 1 file changed, 16 insertions(+)

commit 80dab9e9b6618e454a6f393be5abc11e9529c00b
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 7 12:55:38 2021 +0800

    work in progress channel

 codegen.ml | 101 +++++++++++++++++++++++++++++++++++++++++++-----------------------
 1 file changed, 72 insertions(+), 29 deletions(-)

commit ced010db83cc4511134fa08a8fdadd1c38851d85
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 7 12:55:20 2021 +0800

    channel basic syntax frontend

 ast.ml      | 32 ++++++++++++++++++++++----------
 parser.mly  | 14 ++++++++++----
 scanner.mll |  5 +++++
 semant.ml   | 19 +++++++++++++++++--
 4 files changed, 54 insertions(+), 16 deletions(-)

commit 3c0dd4540ec7b9b6a8c383581330a3c56c178ed4
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Apr 7 12:54:55 2021 +0800
```

```
    updated readme for channel

 README | 6 ++++++
 1 file changed, 6 insertions(+)

commit 4adafb23470c0fdb32d854ea8846ecf80c4b6a0a
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Apr 3 17:26:49 2021 -0400

    3 params working

 Makefile         |  1 +
 codegen.ml       | 46 ++++++++++++++---------------
 gocall.c         | 93 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-------
 tests/test-go.gmm |  7 +++--
 4 files changed, 106 insertions(+), 41 deletions(-)

commit a30d676ef70fac1e120c474dfa7f15aafad56a7f
Author: samlee815 <yl4111@columbia.edu>
Date:   Fri Apr 2 17:55:21 2021 -0400

    up to two float/int param working

 Dockerfile           |   5 +-
 Makefile             |   2 +-
 codegen.ml           |  86 ++++++++++++++++---
 gocall.c             | 231 ++++++++++++++++++++++++++++++---------------------
 printbig.o           | Bin 2080 -> 0 bytes
 testall.sh           |   2 +-
 tests/test-go.gmm     |  37 ++++++--
 tests/test-two-go.gmm |  19 -----
 8 files changed, 206 insertions(+), 176 deletions(-)
```

```
commit e3a898b86458baa400009d60f22f0227aa83784c
Author: cc4351 <cc4351@columbia.edu>
Date:   Thu Apr 1 02:38:16 2021 +0800

    gocall code update for 2 inputs

 codegen.ml |  10 +--
 gocall.c   | 214 +++++++++++++++++++++++++++++++++++++------------------------------------------------
 2 files changed, 137 insertions(+), 87 deletions(-)

commit 35e08d5d59852855648fb899f992ebb33b551a16
Author: cc4351 <cc4351@columbia.edu>
Date:   Thu Apr 1 02:37:10 2021 +0800

    updated test for gocall

 README                |  7 +++----
 tests/test-two-go.gmm | 19 +++++++++++++++++++
 2 files changed, 22 insertions(+), 4 deletions(-)

commit 4bb9250be3d1aebf430596509088053403 80ca93
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Tue Mar 30 01:00:31 2021 -0400

    attempt to add 2 param for gofunc

 gocall.c | 56 ++++++++++++++++++++++++++++++++++++--------------------
 1 file changed, 36 insertions(+), 20 deletions(-)

commit a8314cda00ecd93cf53b6d6fc26670beb3905cca
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Mon Mar 29 00:19:28 2021 -0400

    added struct for varying param
```

```
 gocall.c | 67
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++--
 1 file changed, 65 insertions(+), 2 deletions(-)

commit a7813e93c0fca6d8e021f3eb3d4ff3bb367cfcb0
Author: cc4351 <cc4351@columbia.edu>
Date:   Sun Mar 28 08:41:16 2021 +0800

    1 param working

 codegen.ml        | 12 +++++++-----
 gocall.c          | 35 ++++++++++++++-------------------
 tests/test-go.gmm | 14 +++++++-------
 3 files changed, 24 insertions(+), 37 deletions(-)

commit a7e6aece02a2b76bfd65914f0a14d26e2f7735d8
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 20:35:15 2021 +0800

    update doc for go keyword

 README   | 8 ++++++++
 todo.txt | 1 +
 2 files changed, 9 insertions(+)

commit 21efa7d9f7d31432cd2aefcc9afb6b5257c862f2
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 20:30:32 2021 +0800

    update test program for gocall

 tests/test-go.gmm | 12 +++++++-----
 1 file changed, 7 insertions(+), 5 deletions(-)
```

```
commit 8884dabbb4c0651135da7e1812e019607518b9fd
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 20:30:06 2021 +0800

    modify gocall to have 1 input

 codegen.ml |  8 +++++---
 gocall.c   | 28 +++++++++++++++++++---------
 todo.txt   | 16 ++++++++++++++--
 3 files changed, 38 insertions(+), 14 deletions(-)

commit 846d1d54ac59c5ef035d2a3f45393db8e09a479a
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 19:12:58 2021 +0800

    backend support for simple gocal

 Makefile   |   9 +++-
 codegen.ml | 146 ++++++++++++++++++++++++++++++++++++++++-------------------------------
 run.sh     |   4 +-
 3 files changed, 96 insertions(+), 63 deletions(-)

commit 5b56bc1f47f6568825251a5ddc69944a5a8a0936
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 19:12:14 2021 +0800

    comment added

 parser.mly | 5 ++++-
 semant.ml  | 2 +-
 2 files changed, 5 insertions(+), 2 deletions(-)

commit b77753978f06a0e2bb164efc1b046749ccd54abd
```

```
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 19:11:58 2021 +0800

    add gocall lib

 gocall.c         | 46 ++++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-go.gmm | 14 ++++++++++++++
 2 files changed, 60 insertions(+)

commit b2b46c8a90cdd5f00e66d831d21034de922207d0
Author: cc4351 <cc4351@columbia.edu>
Date:   Sat Mar 27 10:51:00 2021 +0800

    added demo printfib.c

 Makefile              |  9 +++++++--
 codegen.ml            | 34 ++++++++++++++++++++++++++----------
 printfib.c            | 39 +++++++++++++++++++++++++++++++++++++++
 run.sh                |  2 +-
 semant.ml             |  4 +++-
 tests/test-printfib.gmm |  6 ++++++
 todo.txt              |  7 +++++++-
 7 files changed, 83 insertions(+), 18 deletions(-)

commit e8b4fecc5c9422bd4adff0c1745f1178aa6fc17e
Author: samlee815 <yl4111@columbia.edu>
Date:   Fri Mar 26 01:26:51 2021 -0400

    put go keyword into pipeline except codegen.ml

 ast.ml      |  7 +++++--
 codegen.ml  |  8 ++++----
 parser.mly  |  5 +++--
 sast.ml     |  7 +++++--
```

```
 scanner.mll |  1 +
 semant.ml   | 23 ++++++++++++++++++++----
 6 files changed, 37 insertions(+), 14 deletions(-)

commit 8ec82bdcabb893e118701ffec1718cfe7a01f49d
Author: cc4351 <cc4351@columbia.edu>
Date:   Thu Mar 25 09:16:58 2021 +0800

    rm old files and push tar.gz

 Go--.tar.gz                  | Bin 0 -> 19902 bytes
 Makefile                     |  14 +--
 README.md                    |   5 -
 microc/.ll                   |   0
 microc/Dockerfile            |  32 -----
 microc/Makefile              |  67 ----------
 microc/README                | 257 -----------------------------------------
 microc/_tags                 |   9 --
 microc/arcade-font.pbm       | Bin 344 -> 0 bytes
 microc/ast.ml                | 112 ----------------
 microc/codegen.ml            | 249 -------------------------------------
 microc/fail-func8.diff       |   4 -
 microc/fail-func8.err        |   1 -
 microc/fail-func9.diff       |   4 -
 microc/fail-func9.err        |   1 -
 microc/font2c                |   9 --
 microc/future-tests/test-str.gmm |   3 -
 microc/gmm.ml                |  32 -----
 microc/parser.mly            | 126 ------------------
 microc/printbig              | Bin 8424 -> 0 bytes
 microc/printbig.c            |  75 ----------
 microc/printbig.o            | Bin 2080 -> 0 bytes
 microc/run.sh                |  81 ----------
 microc/sast.ml               |  83 ----------
```

```
microc/scanner.mll              |  56 ---------
microc/semant.ml                | 192 ----------------------------
microc/test-print.ml            |   9 --
microc/testall.sh               | 198 ------------------------------
microc/tests/fail-assign1.err   |   1 -
microc/tests/fail-assign1.gmm   |  11 --
microc/tests/fail-assign2.err   |   1 -
microc/tests/fail-assign2.gmm   |   7 --
microc/tests/fail-assign3.err   |   1 -
microc/tests/fail-assign3.gmm   |  11 --
microc/tests/fail-dead1.err     |   1 -
microc/tests/fail-dead1.gmm     |   8 --
microc/tests/fail-dead2.err     |   1 -
microc/tests/fail-dead2.gmm     |  10 --
microc/tests/fail-expr1.err     |   1 -
microc/tests/fail-expr1.gmm     |  18 ---
microc/tests/fail-expr2.err     |   1 -
microc/tests/fail-expr2.gmm     |  14 ---
microc/tests/fail-expr3.err     |   1 -
microc/tests/fail-expr3.gmm     |  14 ---
microc/tests/fail-float1.err    |   1 -
microc/tests/fail-float1.gmm    |   5 -
microc/tests/fail-float2.err    |   1 -
microc/tests/fail-float2.gmm    |   5 -
microc/tests/fail-for1.err      |   1 -
microc/tests/fail-for1.gmm      |  13 --
microc/tests/fail-for2.err      |   1 -
microc/tests/fail-for2.gmm      |   8 --
microc/tests/fail-for3.err      |   1 -
microc/tests/fail-for3.gmm      |   8 --
microc/tests/fail-for4.err      |   1 -
microc/tests/fail-for4.gmm      |   8 --
microc/tests/fail-for5.err      |   1 -
microc/tests/fail-for5.gmm      |  10 --
```

microc/tests/fail-func1.err | 1 -
microc/tests/fail-func1.gmm | 12 --
microc/tests/fail-func2.err | 1 -
microc/tests/fail-func2.gmm | 8 --
microc/tests/fail-func3.err | 1 -
microc/tests/fail-func3.gmm | 8 --
microc/tests/fail-func4.err | 1 -
microc/tests/fail-func4.gmm | 12 --
microc/tests/fail-func5.err | 1 -
microc/tests/fail-func5.gmm | 14 ---
microc/tests/fail-func6.err | 1 -
microc/tests/fail-func6.gmm | 9 --
microc/tests/fail-func7.err | 1 -
microc/tests/fail-func7.gmm | 9 --
microc/tests/fail-func8.err | 1 -
microc/tests/fail-func8.gmm | 13 --
microc/tests/fail-func9.err | 1 -
microc/tests/fail-func9.gmm | 9 --
microc/tests/fail-global1.err | 1 -
microc/tests/fail-global1.gmm | 9 --
microc/tests/fail-global2.err | 1 -
microc/tests/fail-global2.gmm | 9 --
microc/tests/fail-if1.err | 1 -
microc/tests/fail-if1.gmm | 6 -
microc/tests/fail-if2.err | 1 -
microc/tests/fail-if2.gmm | 6 -
microc/tests/fail-if3.err | 1 -
microc/tests/fail-if3.gmm | 8 --
microc/tests/fail-nomain.err | 1 -
microc/tests/fail-nomain.gmm | 0
microc/tests/fail-print.err | 1 -
microc/tests/fail-print.gmm | 2 -
microc/tests/fail-printb.err | 1 -
microc/tests/fail-printb.gmm | 2 -

```
microc/tests/fail-printbig.err   |   1 -
microc/tests/fail-printbig.gmm   |   2 -
microc/tests/fail-return1.err    |   1 -
microc/tests/fail-return1.gmm    |   4 -
microc/tests/fail-return2.err    |   1 -
microc/tests/fail-return2.gmm    |  10 --
microc/tests/fail-while1.err     |   1 -
microc/tests/fail-while1.gmm     |  13 --
microc/tests/fail-while2.err     |   1 -
microc/tests/fail-while2.gmm     |  13 --
microc/tests/test-add1.gmm       |  10 --
microc/tests/test-add1.out       |   1 -
microc/tests/test-arith1.gmm     |   5 -
microc/tests/test-arith1.out     |   1 -
microc/tests/test-arith2.gmm     |   5 -
microc/tests/test-arith2.out     |   1 -
microc/tests/test-arith3.gmm     |  13 --
microc/tests/test-arith3.out     |   1 -
microc/tests/test-fib.gmm        |  16 ---
microc/tests/test-fib.out        |   6 -
microc/tests/test-float1.gmm     |   7 --
microc/tests/test-float1.out     |   1 -
microc/tests/test-float2.gmm     |  11 --
microc/tests/test-float2.out     |   1 -
microc/tests/test-float3.gmm     |  30 -----
microc/tests/test-float3.out     |  24 ----
microc/tests/test-for1.gmm       |   9 --
microc/tests/test-for1.out       |   6 -
microc/tests/test-for2.gmm       |  11 --
microc/tests/test-for2.out       |   6 -
microc/tests/test-func1.gmm      |  12 --
microc/tests/test-func1.out      |   1 -
microc/tests/test-func2.gmm      |  18 ---
microc/tests/test-func2.out      |   1 -
```

microc/tests/test-func3.gmm | 13 --
microc/tests/test-func3.out | 4 -
microc/tests/test-func4.gmm | 14 ---
microc/tests/test-func4.out | 1 -
microc/tests/test-func5.gmm | 9 --
microc/tests/test-func5.out | 0
microc/tests/test-func6.gmm | 9 --
microc/tests/test-func6.out | 1 -
microc/tests/test-func7.gmm | 13 --
microc/tests/test-func7.out | 1 -
microc/tests/test-func8.gmm | 10 --
microc/tests/test-func8.out | 1 -
microc/tests/test-func9.gmm | 11 --
microc/tests/test-func9.out | 1 -
microc/tests/test-gcd.gmm | 15 ---
microc/tests/test-gcd.out | 3 -
microc/tests/test-gcd2.gmm | 14 ---
microc/tests/test-gcd2.out | 3 -
microc/tests/test-global1.gmm | 30 -----
microc/tests/test-global1.out | 4 -
microc/tests/test-global2.gmm | 10 --
microc/tests/test-global2.out | 1 -
microc/tests/test-global3.gmm | 11 --
microc/tests/test-global3.out | 1 -
microc/tests/test-gofunction.gmm | 13 --
microc/tests/test-gofunction.out | 2 -
microc/tests/test-hello.gmm | 7 --
microc/tests/test-hello.out | 3 -
microc/tests/test-if1.gmm | 6 -
microc/tests/test-if1.out | 2 -
microc/tests/test-if2.gmm | 6 -
microc/tests/test-if2.out | 2 -
microc/tests/test-if3.gmm | 6 -
microc/tests/test-if3.out | 1 -

```
microc/tests/test-if4.gmm        |  6 -
microc/tests/test-if4.out        |  2 -
microc/tests/test-if5.gmm        | 16 ---
microc/tests/test-if5.out        |  2 -
microc/tests/test-if6.gmm        | 18 ---
microc/tests/test-if6.out        |  2 -
microc/tests/test-local1.gmm     | 13 --
microc/tests/test-local1.out     |  1 -
microc/tests/test-local2.gmm     | 14 ---
microc/tests/test-local2.out     |  1 -
microc/tests/test-ops1.gmm       | 28 -----
microc/tests/test-ops1.out       | 24 ----
microc/tests/test-ops2.gmm       | 17 ---
microc/tests/test-ops2.out       | 14 ---
microc/tests/test-ppmm.gmm       |  7 --
microc/tests/test-ppmm.out       |  3 -
microc/tests/test-printbig.gmm   | 25 ----
microc/tests/test-printbig.out   | 88 ----------
microc/tests/test-var1.gmm       |  7 --
microc/tests/test-var1.out       |  1 -
microc/tests/test-var2.gmm       | 13 --
microc/tests/test-var2.out       |  1 -
microc/tests/test-while1.gmm     | 11 --
microc/tests/test-while1.out     |  6 -
microc/tests/test-while2.gmm     | 16 ---
microc/tests/test-while2.out     |  1 -
microc/todo.txt                  | 16 ---
test/test-arr-init.gmm           | 41 -------
test/test-struct-init.gmm        | 39 ------
test/test-types-alg-op.gmm       | 108 ----------
test/test-types-cmp-op.gmm       | 103 ----------
test/test-types.gmm              | 51 --------
192 files changed, 7 insertions(+), 3097 deletions(-)
```

```
commit 541f495ae6d7ee179b6f56aaa6741976ff24b003
Author: cc4351 <cc4351@columbia.edu>
Date:   Thu Mar 25 09:07:42 2021 +0800

    replace mc with gmm

 Dockerfile                  |  32 ++++++
 Makefile                    |  85 ++++++++++-----
 README                      |  36 +++++++
 _tags                       |   9 ++
 arcade-font.pbm             | Bin 0 -> 344 bytes
 ast.ml                      | 165 ++++++++---------------------
 codegen.ml                  | 246 ++++++++++++++++++-------------------------
 font2c                      |   9 ++
 future-tests/fail-func8.err |   1 +
 future-tests/fail-func8.gmm |  13 +++
 future-tests/fail-func9.err |   1 +
 future-tests/fail-func9.gmm |   9 ++
 future-tests/test-str.gmm   |   3 +
 gmm.ml                      |  32 ++++++
 parser.mly                  | 107 +++++--------------
 printbig                    | Bin 0 -> 8424 bytes
 printbig.c                  |  75 ++++++++++++++
 printbig.o                  | Bin 0 -> 2080 bytes
 run.sh                      |  81 +++++++++++++++
 sast.ml                     |  83 +++++++++++++++
 scanner.mll                 |  34 ++----
 semant.ml                   | 192 +++++++++++++++++++++++++++++++++++
 testall.sh                  | 198 +++++++++++++++++++++++++++++++++++
 tests/fail-assign1.err      |   1 +
 tests/fail-assign1.gmm      |  11 ++
 tests/fail-assign2.err      |   1 +
 tests/fail-assign2.gmm      |   7 ++
 tests/fail-assign3.err      |   1 +
```

```
tests/fail-assign3.gmm     |  11 ++
tests/fail-dead1.err       |   1 +
tests/fail-dead1.gmm       |   8 ++
tests/fail-dead2.err       |   1 +
tests/fail-dead2.gmm       |  10 ++
tests/fail-expr1.err       |   1 +
tests/fail-expr1.gmm       |  18 ++++
tests/fail-expr2.err       |   1 +
tests/fail-expr2.gmm       |  14 +++
tests/fail-expr3.err       |   1 +
tests/fail-expr3.gmm       |  14 +++
tests/fail-float1.err      |   1 +
tests/fail-float1.gmm      |   5 +
tests/fail-float2.err      |   1 +
tests/fail-float2.gmm      |   5 +
tests/fail-for1.err        |   1 +
tests/fail-for1.gmm        |  13 +++
tests/fail-for2.err        |   1 +
tests/fail-for2.gmm        |   8 ++
tests/fail-for3.err        |   1 +
tests/fail-for3.gmm        |   8 ++
tests/fail-for4.err        |   1 +
tests/fail-for4.gmm        |   8 ++
tests/fail-for5.err        |   1 +
tests/fail-for5.gmm        |  10 ++
tests/fail-func1.err       |   1 +
tests/fail-func1.gmm       |  12 +++
tests/fail-func2.err       |   1 +
tests/fail-func2.gmm       |   8 ++
tests/fail-func3.err       |   1 +
tests/fail-func3.gmm       |   8 ++
tests/fail-func4.err       |   1 +
tests/fail-func4.gmm       |  12 +++
tests/fail-func5.err       |   1 +
```

```
tests/fail-func5.gmm       | 14 +++
tests/fail-func6.err       |  1 +
tests/fail-func6.gmm       |  9 ++
tests/fail-func7.err       |  1 +
tests/fail-func7.gmm       |  9 ++
tests/fail-global1.err     |  1 +
tests/fail-global1.gmm     |  9 ++
tests/fail-global2.err     |  1 +
tests/fail-global2.gmm     |  9 ++
tests/fail-if1.err         |  1 +
tests/fail-if1.gmm         |  6 ++
tests/fail-if2.err         |  1 +
tests/fail-if2.gmm         |  6 ++
tests/fail-if3.err         |  1 +
tests/fail-if3.gmm         |  8 ++
tests/fail-nomain.err      |  1 +
tests/fail-nomain.gmm      |  0
tests/fail-print.err       |  1 +
tests/fail-print.gmm       |  2 +
tests/fail-printb.err      |  1 +
tests/fail-printb.gmm      |  2 +
tests/fail-printbig.err    |  1 +
tests/fail-printbig.gmm    |  2 +
tests/fail-return1.err     |  1 +
tests/fail-return1.gmm     |  4 +
tests/fail-return2.err     |  1 +
tests/fail-return2.gmm     | 10 ++
tests/fail-while1.err      |  1 +
tests/fail-while1.gmm      | 13 +++
tests/fail-while2.err      |  1 +
tests/fail-while2.gmm      | 13 +++
tests/test-add1.gmm        | 10 ++
tests/test-add1.out        |  1 +
tests/test-arith1.gmm      |  5 +
```

```
tests/test-arith1.out      |  1 +
tests/test-arith2.gmm      |  5 +
tests/test-arith2.out      |  1 +
tests/test-arith3.gmm      | 13 +++
tests/test-arith3.out      |  1 +
tests/test-fib.gmm         | 16 +++
tests/test-fib.out         |  6 ++
tests/test-float1.gmm      |  7 ++
tests/test-float1.out      |  1 +
tests/test-float2.gmm      | 11 ++
tests/test-float2.out      |  1 +
tests/test-float3.gmm      | 30 ++++++
tests/test-float3.out      | 24 +++++
tests/test-for1.gmm        |  9 ++
tests/test-for1.out        |  6 ++
tests/test-for2.gmm        | 11 ++
tests/test-for2.out        |  6 ++
tests/test-func1.gmm       | 12 +++
tests/test-func1.out       |  1 +
tests/test-func2.gmm       | 18 ++++
tests/test-func2.out       |  1 +
tests/test-func3.gmm       | 13 +++
tests/test-func3.out       |  4 +
tests/test-func4.gmm       | 14 +++
tests/test-func4.out       |  1 +
tests/test-func5.gmm       |  9 ++
tests/test-func5.out       |  0
tests/test-func6.gmm       |  9 ++
tests/test-func6.out       |  1 +
tests/test-func7.gmm       | 13 +++
tests/test-func7.out       |  1 +
tests/test-func8.gmm       | 10 ++
tests/test-func8.out       |  1 +
tests/test-func9.gmm       | 11 ++
```

```
tests/test-func9.out       |  1 +
tests/test-gcd.gmm         | 15 +++
tests/test-gcd.out         |  3 +
tests/test-gcd2.gmm        | 14 +++
tests/test-gcd2.out        |  3 +
tests/test-global1.gmm     | 30 ++++++
tests/test-global1.out     |  4 +
tests/test-global2.gmm     | 10 ++
tests/test-global2.out     |  1 +
tests/test-global3.gmm     | 11 ++
tests/test-global3.out     |  1 +
tests/test-gofunction.gmm  | 13 +++
tests/test-gofunction.out  |  2 +
tests/test-hello.gmm       |  7 ++
tests/test-hello.out       |  3 +
tests/test-if1.gmm         |  6 ++
tests/test-if1.out         |  2 +
tests/test-if2.gmm         |  6 ++
tests/test-if2.out         |  2 +
tests/test-if3.gmm         |  6 ++
tests/test-if3.out         |  1 +
tests/test-if4.gmm         |  6 ++
tests/test-if4.out         |  2 +
tests/test-if5.gmm         | 16 +++
tests/test-if5.out         |  2 +
tests/test-if6.gmm         | 18 ++++
tests/test-if6.out         |  2 +
tests/test-local1.gmm      | 13 +++
tests/test-local1.out      |  1 +
tests/test-local2.gmm      | 14 +++
tests/test-local2.out      |  1 +
tests/test-ops1.gmm        | 28 +++++
tests/test-ops1.out        | 24 +++++
tests/test-ops2.gmm        | 17 +++
```

```
 tests/test-ops2.out        |  14 +++
 tests/test-ppmm.gmm        |   7 ++
 tests/test-ppmm.out        |   3 +
 tests/test-printbig.gmm    |  25 +++++
 tests/test-printbig.out    |  88 ++++++++++++++++++
 tests/test-var1.gmm        |   7 ++
 tests/test-var1.out        |   1 +
 tests/test-var2.gmm        |  13 +++
 tests/test-var2.out        |   1 +
 tests/test-while1.gmm      |  11 ++
 tests/test-while1.out      |   6 ++
 tests/test-while2.gmm      |  16 +++
 tests/test-while2.out      |   1 +
 todo.txt                   |  21 ++++
 178 files changed, 2136 insertions(+), 399 deletions(-)

commit 43e4f24db8f685c267cb9635330155829d9c93e0
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Mar 23 14:26:02 2021 +0800

    update run script

 microc/run.sh | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

commit 56a112ba3aaf7db420e31db0874b7b7b156fce83f0
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Mar 23 14:25:16 2021 +0800

    new test case for gofunction

 microc/future-tests/test-str.gmm |  3 +++
 microc/tests/test-gofunction.gmm | 13 +++++++++++++
 microc/tests/test-gofunction.out |  2 ++
```

```
 3 files changed, 18 insertions(+)

commit 6361b70447c465584be168329c96370c8bc65b9b
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Mar 23 12:24:38 2021 +0800

    microc with function keyword

 microc/Makefile                    | 15 ++++++++++-----
 microc/ast.ml                      |  6 ++++++
 microc/fail-func8.diff             |  4 ++++
 microc/fail-func8.err              |  1 +
 microc/fail-func9.diff             |  4 ++++
 microc/fail-func9.err              |  1 +
 microc/gmm.ml                      |  2 +-
 microc/{microcparse.mly => parser.mly} | 26 ++++++++++++++++++++---------
 microc/sast.ml                     |  6 ++++++
 microc/scanner.mll                 |  6 +++++-
 microc/semant.ml                   |  6 +++++-
 microc/test-print.ml               |  9 +++++++++
 microc/todo.txt                    | 11 +++++++++--
 13 files changed, 78 insertions(+), 19 deletions(-)

commit 032cc05f77a8f1804d35c9a5fe11d85334c691ad
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Mar 23 12:22:16 2021 +0800

    modify existing test case to gmm syntax

 microc/tests/fail-assign1.gmm  | 2 +-
 microc/tests/fail-assign2.gmm  | 2 +-
 microc/tests/fail-assign3.gmm  | 4 ++--
 microc/tests/fail-dead1.gmm    | 2 +-
 microc/tests/fail-dead2.gmm    | 2 +-
```

```
microc/tests/fail-expr1.gmm    | 4 ++--
microc/tests/fail-expr2.gmm    | 4 ++--
microc/tests/fail-expr3.gmm    | 4 ++--
microc/tests/fail-float1.gmm   | 2 +-
microc/tests/fail-float2.gmm   | 2 +-
microc/tests/fail-for1.gmm     | 2 +-
microc/tests/fail-for2.gmm     | 2 +-
microc/tests/fail-for3.gmm     | 2 +-
microc/tests/fail-for4.gmm     | 2 +-
microc/tests/fail-for5.gmm     | 2 +-
microc/tests/fail-func1.gmm    | 10 +++++-----
microc/tests/fail-func2.gmm    | 6 +++---
microc/tests/fail-func3.gmm    | 6 +++---
microc/tests/fail-func4.gmm    | 10 +++++-----
microc/tests/fail-func5.gmm    | 6 +++---
microc/tests/fail-func6.gmm    | 4 ++--
microc/tests/fail-func7.gmm    | 4 ++--
microc/tests/fail-func8.gmm    | 6 +++---
microc/tests/fail-func9.gmm    | 4 ++--
microc/tests/fail-global1.gmm  | 2 +-
microc/tests/fail-global2.gmm  | 2 +-
microc/tests/fail-if1.gmm      | 2 +-
microc/tests/fail-if2.gmm      | 2 +-
microc/tests/fail-if3.gmm      | 2 +-
microc/tests/fail-print.gmm    | 2 +-
microc/tests/fail-printb.gmm   | 2 +-
microc/tests/fail-printbig.gmm | 2 +-
microc/tests/fail-return1.gmm  | 2 +-
microc/tests/fail-return2.gmm  | 4 ++--
microc/tests/fail-while1.gmm   | 2 +-
microc/tests/fail-while2.gmm   | 2 +-
microc/tests/test-add1.gmm     | 4 ++--
microc/tests/test-arith1.gmm   | 2 +-
microc/tests/test-arith2.gmm   | 2 +-
```

microc/tests/test-arith3.gmm | 4 ++--
microc/tests/test-fib.gmm | 4 ++--
microc/tests/test-float1.gmm | 2 +-
microc/tests/test-float2.gmm | 2 +-
microc/tests/test-float3.gmm | 4 ++--
microc/tests/test-for1.gmm | 2 +-
microc/tests/test-for2.gmm | 2 +-
microc/tests/test-func1.gmm | 4 ++--
microc/tests/test-func2.gmm | 4 ++--
microc/tests/test-func3.gmm | 4 ++--
microc/tests/test-func4.gmm | 4 ++--
microc/tests/test-func5.gmm | 4 ++--
microc/tests/test-func6.gmm | 6 +++---
microc/tests/test-func7.gmm | 4 ++--
microc/tests/test-func8.gmm | 4 ++--
microc/tests/test-func9.gmm | 4 ++--
microc/tests/test-gcd.gmm | 4 ++--
microc/tests/test-gcd2.gmm | 4 ++--
microc/tests/test-global1.gmm | 8 ++++----
microc/tests/test-global2.gmm | 2 +-
microc/tests/test-global3.gmm | 2 +-
microc/tests/test-hello.gmm | 2 +-
microc/tests/test-if1.gmm | 2 +-
microc/tests/test-if2.gmm | 2 +-
microc/tests/test-if3.gmm | 2 +-
microc/tests/test-if4.gmm | 2 +-
microc/tests/test-if5.gmm | 4 ++--
microc/tests/test-if6.gmm | 4 ++--
microc/tests/test-local1.gmm | 4 ++--
microc/tests/test-local2.gmm | 4 ++--
microc/tests/test-ops1.gmm | 2 +-
microc/tests/test-ops2.gmm | 2 +-
microc/tests/test-ppmm.gmm | 2 +-
microc/tests/test-printbig.gmm | 2 +-

```
 microc/tests/test-var1.gmm     | 2 +-
 microc/tests/test-var2.gmm     | 4 ++--
 microc/tests/test-while1.gmm   | 2 +-
 microc/tests/test-while2.gmm   | 4 ++--
 77 files changed, 126 insertions(+), 126 deletions(-)

commit ca520ec5244aad077376e6626cef2f0a092c9174
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Mar 23 00:48:15 2021 +0800

    change file extension

 microc/Makefile                          |   7 +-
 microc/_build/_digests                   |  32 -
 microc/_build/_log                       |  52 -
 microc/_build/ast.cmi                    | Bin 2554 -> 0 bytes
 microc/_build/ast.cmo                    | Bin 4168 -> 0 bytes
 microc/_build/ast.cmx                    | Bin 757 -> 0 bytes
 microc/_build/ast.ml                     | 106 --
 microc/_build/ast.ml.depends             |   1 -
 microc/_build/ast.o                      | Bin 19448 -> 0 bytes
 microc/_build/codegen.cmi                | Bin 5089 -> 0 bytes
 microc/_build/codegen.cmo                | Bin 7796 -> 0 bytes
 microc/_build/codegen.cmx                | Bin 1611 -> 0 bytes
 microc/_build/codegen.ml                 | 249 ----
 microc/_build/codegen.ml.depends         |   1 -
 microc/_build/codegen.o                  | Bin 42104 -> 0 bytes
 microc/_build/microc.cmi                 | Bin 741 -> 0 bytes
 microc/_build/microc.cmo                 | Bin 1902 -> 0 bytes
 microc/_build/microc.cmx                 | Bin 851 -> 0 bytes
 microc/_build/microc.ml                  |  32 -
 microc/_build/microc.ml.depends          |   1 -
 microc/_build/microc.native              | Bin 1265600 -> 0 bytes
 microc/_build/microc.o                   | Bin 7904 -> 0 bytes
```

```
microc/_build/microcparse.cmi                | Bin 1244 -> 0 bytes
microc/_build/microcparse.cmx                | Bin 4589 -> 0 bytes
microc/_build/microcparse.ml                 | 790 ----------
microc/_build/microcparse.ml.depends         |   1 -
microc/_build/microcparse.mli                |  40 -
microc/_build/microcparse.mli.depends        |   1 -
microc/_build/microcparse.mly                | 118 --
microc/_build/microcparse.o                  | Bin 39512 -> 0 bytes
microc/_build/ocamlc.where                   |   1 -
microc/_build/sast.cmi                       | Bin 1953 -> 0 bytes
microc/_build/sast.cmo                       | Bin 3648 -> 0 bytes
microc/_build/sast.cmx                       | Bin 505 -> 0 bytes
microc/_build/sast.ml                        |  77 --
microc/_build/sast.ml.depends                |   1 -
microc/_build/sast.o                         | Bin 17528 -> 0 bytes
microc/_build/scanner.cmi                    | Bin 815 -> 0 bytes
microc/_build/scanner.cmo                    | Bin 16980 -> 0 bytes
microc/_build/scanner.cmx                    | Bin 15859 -> 0 bytes
microc/_build/scanner.ml                     | 1206 ----------
microc/_build/scanner.ml.depends             |   1 -
microc/_build/scanner.mll                    |  52 -
microc/_build/scanner.o                      | Bin 23800 -> 0 bytes
microc/_build/semant.cmi                     | Bin 4962 -> 0 bytes
microc/_build/semant.cmo                     | Bin 7094 -> 0 bytes
microc/_build/semant.cmx                     | Bin 1453 -> 0 bytes
microc/_build/semant.ml                      | 188 ---
microc/_build/semant.ml.depends              |   1 -
microc/_build/semant.o                       | Bin 36384 -> 0 bytes
microc/codegen.ml                            |   2 +-
microc/{microc.ml => gmm.ml}                 |   2 +-
microc/microc.native                         |   1 -
microc/run.sh                                |   6 +-
microc/test-ops2.exe                         | Bin 8440 -> 0 bytes
microc/test-ops2.ll                          |  28 -
```

```
microc/test-ops2.out                          |   14 -
microc/test-ops2.s                            |   88 --
microc/test-ppmm.exe                          |  Bin 8440 -> 0 bytes
microc/test-ppmm.ll                           |   27 -
microc/test-ppmm.out                          |    3 -
microc/test-ppmm.s                            |   51 -
microc/testall.sh                             |   18 +-
microc/tests/{fail-assign1.mc => fail-assign1.gmm} |    0
microc/tests/{fail-assign2.mc => fail-assign2.gmm} |    0
microc/tests/{fail-assign3.mc => fail-assign3.gmm} |    0
microc/tests/{fail-dead1.mc => fail-dead1.gmm}     |    0
microc/tests/{fail-dead2.mc => fail-dead2.gmm}     |    0
microc/tests/{fail-expr1.mc => fail-expr1.gmm}     |    0
microc/tests/{fail-expr2.mc => fail-expr2.gmm}     |    0
microc/tests/{fail-expr3.mc => fail-expr3.gmm}     |    0
microc/tests/{fail-float1.mc => fail-float1.gmm}   |    0
microc/tests/{fail-float2.mc => fail-float2.gmm}   |    0
microc/tests/{fail-for1.mc => fail-for1.gmm}       |    0
microc/tests/{fail-for2.mc => fail-for2.gmm}       |    0
microc/tests/{fail-for3.mc => fail-for3.gmm}       |    0
microc/tests/{fail-for4.mc => fail-for4.gmm}       |    0
microc/tests/{fail-for5.mc => fail-for5.gmm}       |    0
microc/tests/{fail-func1.mc => fail-func1.gmm}     |    0
microc/tests/{fail-func2.mc => fail-func2.gmm}     |    0
microc/tests/{fail-func3.mc => fail-func3.gmm}     |    0
microc/tests/{fail-func4.mc => fail-func4.gmm}     |    0
microc/tests/{fail-func5.mc => fail-func5.gmm}     |    0
microc/tests/{fail-func6.mc => fail-func6.gmm}     |    0
microc/tests/{fail-func7.mc => fail-func7.gmm}     |    0
microc/tests/{fail-func8.mc => fail-func8.gmm}     |    0
microc/tests/{fail-func9.mc => fail-func9.gmm}     |    0
microc/tests/{fail-global1.mc => fail-global1.gmm} |    0
microc/tests/{fail-global2.mc => fail-global2.gmm} |    0
microc/tests/{fail-if1.mc => fail-if1.gmm}         |    0
```

```
microc/tests/{fail-if2.mc => fail-if2.gmm}         |    0
microc/tests/{fail-if3.mc => fail-if3.gmm}         |    0
microc/{test-op2.ll => tests/fail-nomain.gmm}      |    0
microc/tests/fail-nomain.mc                        |    0
microc/tests/{fail-print.mc => fail-print.gmm}     |    0
microc/tests/{fail-printb.mc => fail-printb.gmm}   |    0
.../tests/{fail-printbig.mc => fail-printbig.gmm}  |    0
microc/tests/{fail-return1.mc => fail-return1.gmm} |    0
microc/tests/{fail-return2.mc => fail-return2.gmm} |    0
microc/tests/{fail-while1.mc => fail-while1.gmm}   |    0
microc/tests/{fail-while2.mc => fail-while2.gmm}   |    0
microc/tests/{test-add1.mc => test-add1.gmm}       |    0
microc/tests/{test-arith1.mc => test-arith1.gmm}   |    0
microc/tests/{test-arith2.mc => test-arith2.gmm}   |    0
microc/tests/{test-arith3.mc => test-arith3.gmm}   |    0
microc/tests/{test-fib.mc => test-fib.gmm}         |    0
microc/tests/{test-float1.mc => test-float1.gmm}   |    0
microc/tests/{test-float2.mc => test-float2.gmm}   |    0
microc/tests/{test-float3.mc => test-float3.gmm}   |    0
microc/tests/{test-for1.mc => test-for1.gmm}       |    0
microc/tests/{test-for2.mc => test-for2.gmm}       |    0
microc/tests/{test-func1.mc => test-func1.gmm}     |    0
microc/tests/{test-func2.mc => test-func2.gmm}     |    0
microc/tests/{test-func3.mc => test-func3.gmm}     |    0
microc/tests/{test-func4.mc => test-func4.gmm}     |    0
microc/tests/{test-func5.mc => test-func5.gmm}     |    0
microc/tests/{test-func6.mc => test-func6.gmm}     |    0
microc/tests/{test-func7.mc => test-func7.gmm}     |    0
microc/tests/{test-func8.mc => test-func8.gmm}     |    0
microc/tests/{test-func9.mc => test-func9.gmm}     |    0
microc/tests/{test-gcd.mc => test-gcd.gmm}         |    0
microc/tests/{test-gcd2.mc => test-gcd2.gmm}       |    0
microc/tests/{test-global1.mc => test-global1.gmm} |    0
microc/tests/{test-global2.mc => test-global2.gmm} |    0
```

```
 microc/tests/{test-global3.mc => test-global3.gmm} |    0
 microc/tests/{test-hello.mc => test-hello.gmm}     |    0
 microc/tests/{test-if1.mc => test-if1.gmm}         |    0
 microc/tests/{test-if2.mc => test-if2.gmm}         |    0
 microc/tests/{test-if3.mc => test-if3.gmm}         |    0
 microc/tests/{test-if4.mc => test-if4.gmm}         |    0
 microc/tests/{test-if5.mc => test-if5.gmm}         |    0
 microc/tests/{test-if6.mc => test-if6.gmm}         |    0
 microc/tests/{test-local1.mc => test-local1.gmm}   |    0
 microc/tests/{test-local2.mc => test-local2.gmm}   |    0
 microc/tests/{test-ops1.mc => test-ops1.gmm}       |    0
 microc/tests/{test-ops2.mc => test-ops2.gmm}       |    0
 microc/tests/test-ops2.out                         |    2 +-
 microc/tests/{test-ppmm.mc => test-ppmm.gmm}       |    0
 .../tests/{test-printbig.mc => test-printbig.gmm}  |    0
 microc/tests/{test-var1.mc => test-var1.gmm}       |    0
 microc/tests/{test-var2.mc => test-var2.gmm}       |    0
 microc/tests/{test-while1.mc => test-while1.gmm}   |    0
 microc/tests/{test-while2.mc => test-while2.gmm}   |    0
 microc/todo.txt                                    |    9 +
 144 files changed, 28 insertions(+), 3181 deletions(-)

commit 223db666df7cd96e5f9c881f122c43a2ebb401ad
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Mar 23 00:27:15 2021 +0800

    microc modified

 microc/.ll                        |    0
 microc/Dockerfile                 |   32 +
 microc/Makefile                   |   61 ++
 microc/README                     |  257 +++++++
 microc/_build/_digests            |   32 +
 microc/_build/_log                |   52 ++
```

```
microc/_build/ast.cmi               | Bin 0 -> 2554 bytes
microc/_build/ast.cmo               | Bin 0 -> 4168 bytes
microc/_build/ast.cmx               | Bin 0 -> 757 bytes
microc/_build/ast.ml                | 106 +++
microc/_build/ast.ml.depends        |   1 +
microc/_build/ast.o                 | Bin 0 -> 19448 bytes
microc/_build/codegen.cmi           | Bin 0 -> 5089 bytes
microc/_build/codegen.cmo           | Bin 0 -> 7796 bytes
microc/_build/codegen.cmx           | Bin 0 -> 1611 bytes
microc/_build/codegen.ml            | 249 +++++++
microc/_build/codegen.ml.depends    |   1 +
microc/_build/codegen.o             | Bin 0 -> 42104 bytes
microc/_build/microc.cmi            | Bin 0 -> 741 bytes
microc/_build/microc.cmo            | Bin 0 -> 1902 bytes
microc/_build/microc.cmx            | Bin 0 -> 851 bytes
microc/_build/microc.ml             |  32 +
microc/_build/microc.ml.depends     |   1 +
microc/_build/microc.native         | Bin 0 -> 1265600 bytes
microc/_build/microc.o              | Bin 0 -> 7904 bytes
microc/_build/microcparse.cmi       | Bin 0 -> 1244 bytes
microc/_build/microcparse.cmx       | Bin 0 -> 4589 bytes
microc/_build/microcparse.ml        | 790 ++++++++++++++++++++++
microc/_build/microcparse.ml.depends |   1 +
microc/_build/microcparse.mli       |  40 ++
microc/_build/microcparse.mli.depends |   1 +
microc/_build/microcparse.mly       | 118 ++++
microc/_build/microcparse.o         | Bin 0 -> 39512 bytes
microc/_build/ocamlc.where          |   1 +
microc/_build/sast.cmi              | Bin 0 -> 1953 bytes
microc/_build/sast.cmo              | Bin 0 -> 3648 bytes
microc/_build/sast.cmx              | Bin 0 -> 505 bytes
microc/_build/sast.ml               |  77 +++
microc/_build/sast.ml.depends       |   1 +
microc/_build/sast.o                | Bin 0 -> 17528 bytes
```

```
microc/_build/scanner.cmi          | Bin 0 -> 815 bytes
microc/_build/scanner.cmo          | Bin 0 -> 16980 bytes
microc/_build/scanner.cmx          | Bin 0 -> 15859 bytes
microc/_build/scanner.ml           | 1206 ++++++++++++++++++++++++++++++++++++++
microc/_build/scanner.ml.depends   |    1 +
microc/_build/scanner.mll          |   52 ++
microc/_build/scanner.o            | Bin 0 -> 23800 bytes
microc/_build/semant.cmi           | Bin 0 -> 4962 bytes
microc/_build/semant.cmo           | Bin 0 -> 7094 bytes
microc/_build/semant.cmx           | Bin 0 -> 1453 bytes
microc/_build/semant.ml            | 188 +++++
microc/_build/semant.ml.depends    |    1 +
microc/_build/semant.o             | Bin 0 -> 36384 bytes
microc/_tags                       |    9 +
microc/arcade-font.pbm             | Bin 0 -> 344 bytes
microc/ast.ml                      | 106 +++
microc/codegen.ml                  | 249 +++++++
microc/font2c                      |    9 +
microc/microc.ml                   |   32 +
microc/microc.native               |    1 +
microc/microcparse.mly             | 118 ++++
microc/printbig                    | Bin 0 -> 8424 bytes
microc/printbig.c                  |   75 ++
microc/printbig.o                  | Bin 0 -> 2080 bytes
microc/run.sh                      |   80 +++
microc/sast.ml                     |   77 +++
microc/scanner.mll                 |   52 ++
microc/semant.ml                   | 188 +++++
microc/test-op2.ll                 |    0
microc/test-ops2.exe               | Bin 0 -> 8440 bytes
microc/test-ops2.ll                |   28 +
microc/test-ops2.out               |   14 +
microc/test-ops2.s                 |   88 +++
microc/test-ppmm.exe               | Bin 0 -> 8440 bytes
```

```
microc/test-ppmm.ll               |  27 +
microc/test-ppmm.out              |   3 +
microc/test-ppmm.s                |  51 ++
microc/testall.sh                 | 198 ++++++
microc/tests/fail-assign1.err     |   1 +
microc/tests/fail-assign1.mc      |  11 +
microc/tests/fail-assign2.err     |   1 +
microc/tests/fail-assign2.mc      |   7 +
microc/tests/fail-assign3.err     |   1 +
microc/tests/fail-assign3.mc      |  11 +
microc/tests/fail-dead1.err       |   1 +
microc/tests/fail-dead1.mc        |   8 +
microc/tests/fail-dead2.err       |   1 +
microc/tests/fail-dead2.mc        |  10 +
microc/tests/fail-expr1.err       |   1 +
microc/tests/fail-expr1.mc        |  18 +
microc/tests/fail-expr2.err       |   1 +
microc/tests/fail-expr2.mc        |  14 +
microc/tests/fail-expr3.err       |   1 +
microc/tests/fail-expr3.mc        |  14 +
microc/tests/fail-float1.err      |   1 +
microc/tests/fail-float1.mc       |   5 +
microc/tests/fail-float2.err      |   1 +
microc/tests/fail-float2.mc       |   5 +
microc/tests/fail-for1.err        |   1 +
microc/tests/fail-for1.mc         |  13 +
microc/tests/fail-for2.err        |   1 +
microc/tests/fail-for2.mc         |   8 +
microc/tests/fail-for3.err        |   1 +
microc/tests/fail-for3.mc         |   8 +
microc/tests/fail-for4.err        |   1 +
microc/tests/fail-for4.mc         |   8 +
microc/tests/fail-for5.err        |   1 +
microc/tests/fail-for5.mc         |  10 +
```

```
microc/tests/fail-func1.err         |   1 +
microc/tests/fail-func1.mc          |  12 +
microc/tests/fail-func2.err         |   1 +
microc/tests/fail-func2.mc          |   8 +
microc/tests/fail-func3.err         |   1 +
microc/tests/fail-func3.mc          |   8 +
microc/tests/fail-func4.err         |   1 +
microc/tests/fail-func4.mc          |  12 +
microc/tests/fail-func5.err         |   1 +
microc/tests/fail-func5.mc          |  14 +
microc/tests/fail-func6.err         |   1 +
microc/tests/fail-func6.mc          |   9 +
microc/tests/fail-func7.err         |   1 +
microc/tests/fail-func7.mc          |   9 +
microc/tests/fail-func8.err         |   1 +
microc/tests/fail-func8.mc          |  13 +
microc/tests/fail-func9.err         |   1 +
microc/tests/fail-func9.mc          |   9 +
microc/tests/fail-global1.err       |   1 +
microc/tests/fail-global1.mc        |   9 +
microc/tests/fail-global2.err       |   1 +
microc/tests/fail-global2.mc        |   9 +
microc/tests/fail-if1.err           |   1 +
microc/tests/fail-if1.mc            |   6 +
microc/tests/fail-if2.err           |   1 +
microc/tests/fail-if2.mc            |   6 +
microc/tests/fail-if3.err           |   1 +
microc/tests/fail-if3.mc            |   8 +
microc/tests/fail-nomain.err        |   1 +
microc/tests/fail-nomain.mc         |   0
microc/tests/fail-print.err         |   1 +
microc/tests/fail-print.mc          |   2 +
microc/tests/fail-printb.err        |   1 +
microc/tests/fail-printb.mc         |   2 +
```

```
microc/tests/fail-printbig.err      |  1 +
microc/tests/fail-printbig.mc       |  2 +
microc/tests/fail-return1.err       |  1 +
microc/tests/fail-return1.mc        |  4 +
microc/tests/fail-return2.err       |  1 +
microc/tests/fail-return2.mc        | 10 +
microc/tests/fail-while1.err        |  1 +
microc/tests/fail-while1.mc         | 13 +
microc/tests/fail-while2.err        |  1 +
microc/tests/fail-while2.mc         | 13 +
microc/tests/test-add1.mc           | 10 +
microc/tests/test-add1.out          |  1 +
microc/tests/test-arith1.mc         |  5 +
microc/tests/test-arith1.out        |  1 +
microc/tests/test-arith2.mc         |  5 +
microc/tests/test-arith2.out        |  1 +
microc/tests/test-arith3.mc         | 13 +
microc/tests/test-arith3.out        |  1 +
microc/tests/test-fib.mc            | 16 +
microc/tests/test-fib.out           |  6 +
microc/tests/test-float1.mc         |  7 +
microc/tests/test-float1.out        |  1 +
microc/tests/test-float2.mc         | 11 +
microc/tests/test-float2.out        |  1 +
microc/tests/test-float3.mc         | 30 +
microc/tests/test-float3.out        | 24 +
microc/tests/test-for1.mc           |  9 +
microc/tests/test-for1.out          |  6 +
microc/tests/test-for2.mc           | 11 +
microc/tests/test-for2.out          |  6 +
microc/tests/test-func1.mc          | 12 +
microc/tests/test-func1.out         |  1 +
microc/tests/test-func2.mc          | 18 +
microc/tests/test-func2.out         |  1 +
```

```
microc/tests/test-func3.mc          |  13 +
microc/tests/test-func3.out         |   4 +
microc/tests/test-func4.mc          |  14 +
microc/tests/test-func4.out         |   1 +
microc/tests/test-func5.mc          |   9 +
microc/tests/test-func5.out         |   0
microc/tests/test-func6.mc          |   9 +
microc/tests/test-func6.out         |   1 +
microc/tests/test-func7.mc          |  13 +
microc/tests/test-func7.out         |   1 +
microc/tests/test-func8.mc          |  10 +
microc/tests/test-func8.out         |   1 +
microc/tests/test-func9.mc          |  11 +
microc/tests/test-func9.out         |   1 +
microc/tests/test-gcd.mc            |  15 +
microc/tests/test-gcd.out           |   3 +
microc/tests/test-gcd2.mc           |  14 +
microc/tests/test-gcd2.out          |   3 +
microc/tests/test-global1.mc        |  30 +
microc/tests/test-global1.out       |   4 +
microc/tests/test-global2.mc        |  10 +
microc/tests/test-global2.out       |   1 +
microc/tests/test-global3.mc        |  11 +
microc/tests/test-global3.out       |   1 +
microc/tests/test-hello.mc          |   7 +
microc/tests/test-hello.out         |   3 +
microc/tests/test-if1.mc            |   6 +
microc/tests/test-if1.out           |   2 +
microc/tests/test-if2.mc            |   6 +
microc/tests/test-if2.out           |   2 +
microc/tests/test-if3.mc            |   6 +
microc/tests/test-if3.out           |   1 +
microc/tests/test-if4.mc            |   6 +
microc/tests/test-if4.out           |   2 +
```

```
 microc/tests/test-if5.mc             |  16 +
 microc/tests/test-if5.out            |   2 +
 microc/tests/test-if6.mc             |  18 +
 microc/tests/test-if6.out            |   2 +
 microc/tests/test-local1.mc          |  13 +
 microc/tests/test-local1.out         |   1 +
 microc/tests/test-local2.mc          |  14 +
 microc/tests/test-local2.out         |   1 +
 microc/tests/test-ops1.mc            |  28 +
 microc/tests/test-ops1.out           |  24 +
 microc/tests/test-ops2.mc            |  17 +
 microc/tests/test-ops2.out           |  14 +
 microc/tests/test-ppmm.mc            |   7 +
 microc/tests/test-ppmm.out           |   3 +
 microc/tests/test-printbig.mc        |  25 +
 microc/tests/test-printbig.out       |  88 +++
 microc/tests/test-var1.mc            |   7 +
 microc/tests/test-var1.out           |   1 +
 microc/tests/test-var2.mc            |  13 +
 microc/tests/test-var2.out           |   1 +
 microc/tests/test-while1.mc          |  11 +
 microc/tests/test-while1.out         |   6 +
 microc/tests/test-while2.mc          |  16 +
 microc/tests/test-while2.out         |   1 +
 234 files changed, 5818 insertions(+)

commit ccedd3fe480c1c8b368c35cdc20a6fc0ee8f6bc2
Author: cc4351 <cc4351@columbia.edu>
Date:   Sun Mar 21 00:22:18 2021 +0800

    commented microc codegen

 codegen.ml | 281
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
 1 file changed, 281 insertions(+)

commit c4166b30028b02543fc65480e3c87160c31c84ab
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Mar 13 21:18:47 2021 -0500

    first version of working prettyprint and parser

 Makefile           | 47 ++++++++++++++++----------------
 ast.ml             | 75 ++++++++++++++++++++++++++++++-----------------------
 scanner.mll        | 22 ++++++++--------
 test-print.ml      |  6 +++--
 test/test-types.gmm | 14 +++++-----
 5 files changed, 89 insertions(+), 75 deletions(-)

commit a3057bc9e80b366406469f8b16dfd131a22b0524
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Mar 13 17:43:32 2021 -0500

    TODO : add operators/tokens struct/array/scanner/channel

 ast.ml     | 131 +++++++++++++++++++++++++++++----------------------------
 parser.mly | 116 ++++++++++++++++++++++++++++++++++++-----------------------
 2 files changed, 130 insertions(+), 117 deletions(-)

commit df396d0884497401f746ba042c516c862fd21c04
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Mar 13 17:03:33 2021 -0500

    delete all irrelvant fiels

 ast.cmi            |  Bin 2177 -> 0 bytes
 ast.mli            |   47 --
 change_log.txt       |   11 -
```

```
 diff_main_struct.txt | 1201 ------------------------------
 parser.cmi           |  Bin 1432 -> 0 bytes
 parser.ml            |  853 ----------------------
 parser.mli           |   51 --
 scanner.ml           | 2020 -------------------------------------------------
 8 files changed, 4183 deletions(-)

commit 600e1efdc7bcd79af31a942ee6e1c104a50af19b
Author: samlee815 <yl4111@columbia.edu>
Date:   Sat Mar 13 16:40:28 2021 -0500

    first cleanup

 ast.cmi    |  Bin 0 -> 2177 bytes
 ast.ml     |   49 +-
 ast.mli    |   47 ++
 parser.cmi |  Bin 0 -> 1432 bytes
 parser.ml  |  853 ++++++++++++++++++++++++++
 parser.mli |   51 ++
 parser.mly |  154 +----
 scanner.ml | 2020
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 8 files changed, 2998 insertions(+), 176 deletions(-)

commit 6dbc9bd55d776e1cd6e7eefcc8e15501fd79a928
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Mar 10 21:48:02 2021 -0500

    debugging for test-print: modified channel declaration

 ast.ml     |  2 +-
 parser.mly | 10 ++++++----
 2 files changed, 7 insertions(+), 5 deletions(-)
```

```
commit a4672445ac1a555224836a22430a4c01f05f416d
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Mon Mar 8 23:44:10 2021 -0500

    fixed bug for struct_typ

 parser.mly | 6 +++---
 1 file changed, 3 insertions(+), 3 deletions(-)

commit 89e2150c52a3dffc9c8bdbeea529dcf6b0b02a71
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Sun Mar 7 20:07:42 2021 -0500

    create struct_typ for list of typ

 parser.mly | 8 +++++++-
 1 file changed, 7 insertions(+), 1 deletion(-)

commit d5072d2d45f2c24a98455207ccae2843580f5fdd
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Sat Mar 6 23:37:06 2021 -0500

    add command to make ast.mli from ast.ml

 Makefile | 6 ++++--
 1 file changed, 4 insertions(+), 2 deletions(-)

commit daa7768be0eb7353aedbcc075c4973698246ef05
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Sat Mar 6 21:08:59 2021 -0500

    add changes to Makefile for test-print

 Makefile                | 19 ++++++++++---------
```

test-print.ml | 7 +++++++
test/test-types-alg-op.gmm | 3 +--
3 files changed, 18 insertions(+), 11 deletions(-)

commit e4feb8b9f061714de5f611ac047de3603002d07a
Merge: 0acb5a8 a3d8ab4
Author: keyuyan1145 <keyuyan01@gmail.com>
Date: Sat Mar 6 18:34:46 2021 -0500

Merge branch 'prettyprint1' into test_arr_struct

commit 0acb5a802919c65402a0d412ea50ecc105254ac0
Author: keyuyan1145 <keyuyan01@gmail.com>
Date: Sat Mar 6 15:49:50 2021 -0500

add tests for data types, array, and struct

test/test-arr-init.gmm | 41 +++++++++++++++++++
test/test-struct-init.gmm | 39 ++++++++++++++++++
test/test-types-alg-op.gmm | 109 ++
test/test-types-cmp-op.gmm | 103 +++
test/test-types.gmm | 51 +++++++++++++++++++++++
5 files changed, 343 insertions(+)

commit a3d8ab42e158612f6c761ebf0956699232ef4b23
Author: cc4351 <cc4351@columbia.edu>
Date: Sat Mar 6 18:39:40 2021 +0800

compiled ast

ast.ml | 108 +++++++++++++++++++++++++++++++++++++++-------------------------------------
1 file changed, 56 insertions(+), 52 deletions(-)

commit 362b7a7e3f6eca79ac9174ae5897fef5f71dc23a

Author: cc4351 <cc4351@columbia.edu>
Date: Sat Mar 6 16:18:16 2021 +0800

    ast func sig and func type print modified

 ast.ml | 17 ++++++++++++-----
 1 file changed, 12 insertions(+), 5 deletions(-)

commit c6eb7e52e8505cdf0d899ee840a3bfa9a0e37ed8
Author: samlee815 <yl4111@columbia.edu>
Date: Fri Mar 5 11:56:02 2021 -0500

    fix array as a type

 ast.ml | 14 ++++++--------
 parser.mly | 10 ++++++++--
 2 files changed, 14 insertions(+), 10 deletions(-)

commit 375471648911f6f46ac9ddc039bb4c717bab7312
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Mar 5 17:44:33 2021 +0800

    cleanup parser

 parser.mly | 34 +++++++---------------------------
 1 file changed, 7 insertions(+), 27 deletions(-)

commit c02b64cd9fb90d4a713c75e8da75744069ef9faf
Author: cc4351 <cc4351@columbia.edu>
Date: Fri Mar 5 17:43:42 2021 +0800

    pretty print part 2 chen commit

 ast.ml | 44 ++----------

```
 1 file changed, 31 insertions(+), 13 deletions(-)

commit 1c6c0e13ca522f0bf9773d8fed883204f2eaf302
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Mar 5 16:15:25 2021 +0800

    add comment and todo's to sync with parser

 ast.ml | 15 ++++++++++-----
 1 file changed, 10 insertions(+), 5 deletions(-)

commit 2dcc11448aa045446891de1ed345a8808d9cff49
Author: cc4351 <cc4351@columbia.edu>
Date:   Fri Mar 5 14:41:17 2021 +0800

    silence charlit

 ast.ml | 7 +++----
 1 file changed, 3 insertions(+), 4 deletions(-)

commit 7dfa729a7ed1b56a563376372f4685ec12501b9a
Author: samlee815 <yl4111@columbia.edu>
Date:   Thu Mar 4 15:08:38 2021 -0500

    pretty print functions ffor expressions

 ast.ml | 23 ++++++++++++++++++++----
 1 file changed, 19 insertions(+), 4 deletions(-)

commit 2dcc2115f371980cef41f21e5ef5adcbfc8606a9
Merge: fcbc2f6 60b5790
Author: keyuyan1145 <66279654+keyuyan1145@users.noreply.github.com>
Date:   Wed Feb 24 22:49:18 2021 -0500
```

```
    Merge pull request #4 from keyuyan1145/array_copy2

    Array copy2

commit 60b5790361096d8a23e2d4bf466ee35a435ce79a
Merge: 00668e9 fcbc2f6
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 22:47:35 2021 -0500

    Merge branch 'main' into array_copy2

commit fcbc2f69a4c7db3f81f1559322bf9b6dc399d7b5
Author: Arya Zhao <44174444+AryaZhao@users.noreply.github.com>
Date:   Wed Feb 24 18:36:32 2021 -0800

    Channel (#2)

    * modified Makefile to test parser (Yuyan)

    * add NEWCHAN token, channel vdecl, deque expr, enque expr

    * change arrow to Uarrow, Larrow, Rarrow for channel

    Co-authored-by: keyuyan1145 <keyuyan01@gmail.com>

 ast.ml      |  6 +++---
 parser.mly  | 12 ++++++++++--
 scanner.mll |  4 +++-
 3 files changed, 16 insertions(+), 6 deletions(-)

commit 00668e9ccb34594bc4b60a293060ec0c2b09ce70
Merge: 9c94d2d cdf96ca
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 20:40:33 2021 -0500
```

```
    Merge branch 'main' into array_copy2

commit cdf96ca38ed0b8eee2ec616dc706d63fb72256dd
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 20:37:58 2021 -0500

    modified Makefile to test parser, scanner

 Makefile | 12 +++---------
 1 file changed, 3 insertions(+), 9 deletions(-)

commit 9c94d2dba820626da25dc5bd5d0179433b9fdead
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 20:29:47 2021 -0500

    modified grammar

 ast.ml    | 13 ++++++++++--
 parser.mly | 10 ++++------
 2 files changed, 15 insertions(+), 8 deletions(-)

commit b848bbeb3317d3291198de6b5c213530596649be
Merge: 655e7fb 2bdf3a5
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 19:54:18 2021 -0500

    Merge branch 'struct' into array_copy2

commit 2bdf3a5a1ca520a5d7cdeaec022eaafe8e98c10b
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 19:46:03 2021 -0500

    modified Makefile
```

```
 Makefile | 11 +++--------
 1 file changed, 3 insertions(+), 8 deletions(-)

commit 655e7fbe16d307a808ca2fc3ca0af7b16618887d
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Wed Feb 24 13:04:40 2021 -0500

    adding array, float, string, and char

 ast.ml      |  9 +++++++--
 parser.mly  | 38 ++++++++++++++++++++++++--------------
 scanner.mll |  7 +++++++
 3 files changed, 38 insertions(+), 16 deletions(-)

commit 51ad0b4046fe564bbbc171f1edfc530cedc58e80
Merge: ae3c4c3 092b066
Author: keyuyan1145 <66279654+keyuyan1145@users.noreply.github.com>
Date:   Wed Feb 24 13:04:00 2021 -0500

    Merge pull request #1 from keyuyan1145/struct

    Struct

commit 092b0669c127b714bda451945fad9a5b53d61161
Author: cc4351 <cc4351@columbia.edu>
Date:   Thu Feb 25 00:31:25 2021 +0800

    commented version

 diff_main_struct.txt | 760 ++++++++++++++++++++++++++++++++++++++++----------
 parser.mly           |   7 +-
 2 files changed, 622 insertions(+), 145 deletions(-)
```

```
commit 4346e3dc3c7f557c7afac29de1c7062ed22ce720
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Feb 24 22:04:56 2021 +0800

    add void as a func_expr return type

 change_log.txt | 1 +
 parser.mly     | 8 ++++----
 2 files changed, 5 insertions(+), 4 deletions(-)

commit af81da7804d1810f08c69376d18a7ec6fca84636
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Feb 24 21:35:00 2021 +0800

    update Makefile;program compile;fix func ambiguity

 Makefile                     |  6 +++---
 README.md                    |  4 ++++
 ast.ml                       | 10 +++++-----
 todo.txt => change_log.txt   |  1 +
 parser.mly                   | 44 ++++++++++++++++++++++-----------------------
 scanner.mll                  |  2 ++
 6 files changed, 36 insertions(+), 31 deletions(-)

commit 201001540153266a5ff646ec18b494035488e7a1
Author: cc4351 <cc4351@columbia.edu>
Date:   Wed Feb 24 17:30:49 2021 +0800

    update diff after checkstyle

 diff_main_struct.txt | 595 +++++++++++++++++++++++++++++++++++++++++++++---------
 1 file changed, 508 insertions(+), 87 deletions(-)

commit f0a1b6d183674bee74002508c66332e47756e91d
```

Author: cc4351 <cc4351@columbia.edu>
Date: Wed Feb 24 17:30:10 2021 +0800

    checkstyle codes

 ast.ml | 22 +++++-----
 parser.mly | 135 ++++++++++++++++++++++++++++++------------------------
 scanner.mll | 2 +-
 todo.txt | 3 +-
 4 files changed, 81 insertions(+), 81 deletions(-)

commit db921455d9df32068482288048fbd68efee18d2c
Author: cc4351 <cc4351@columbia.edu>
Date: Wed Feb 24 17:13:13 2021 +0800

    add diff file between main and struct

 diff_main_struct.txt | 308
+++
 1 file changed, 308 insertions(+)

commit 42b564abf43e9cde399c56ba58e2a0bb7539e637
Author: cc4351 <cc4351@columbia.edu>
Date: Wed Feb 24 17:12:05 2021 +0800

    check operators

 ast.ml | 5 +++--
 parser.mly | 33 +++++++++++++++++++++++----------
 todo.txt | 7 +++----
 3 files changed, 29 insertions(+), 16 deletions(-)

commit 48c255cbacb71e26f3acab33c4a7e840cf5abeff
Author: cc4351 <cc4351@columbia.edu>

Date: Wed Feb 24 16:45:11 2021 +0800

    struct related push

 ast.ml | 50 +++++++++++++++++++++++++++-----------------------
 parser.mly | 62 +++++++++++++++++++++++++++++++++++---------------------------
 scanner.mll | 2 +-
 todo.txt | 9 +++++++++
 4 files changed, 72 insertions(+), 51 deletions(-)

commit ae3c4c357e6866ab04817f2422c0d15dd9385f54
Author: keyuyan1145 <keyuyan01@gmail.com>
Date: Tue Feb 23 22:38:51 2021 -0500

    working on struct

 ast.ml | 1 +
 parser.mly | 15 ++++++++++++++-
 2 files changed, 15 insertions(+), 1 deletion(-)

commit 5038c41a65213ef47cd911b7b040d0cfe1cf0515
Author: samlee815 <yl4111@columbia.edu>
Date: Wed Feb 24 01:24:20 2021 +0800

    starting to parse struct declaration

 ast.ml | 2 ++
 parser.mly | 18 ++++++++++++++++++
 2 files changed, 20 insertions(+)

commit 5edbdbc17fb80d9bbe1b9e6f228581e65a7f0584
Author: cc4351 <cc4351@columbia.edu>
Date: Tue Feb 23 17:34:24 2021 +0800

parser func part done;verification needed

Makefile | 42 ++
ast.ml | 16 ++++++++++------
parser.mly | 45 +++++++++++++++++++++++++++++++++++----------
scanner.mll | 2 +-
4 files changed, 88 insertions(+), 17 deletions(-)

commit 7c966d4d946fb9929a83470bee557216b3fa6b23
Author: cc4351 <cc4351@columbia.edu>
Date: Tue Feb 23 16:55:11 2021 +0800

add func parts to parser

ast.ml | 11 +++++++++--
parser.mly | 39 +++++++++++++++++++++++++++++++++++----
2 files changed, 44 insertions(+), 6 deletions(-)

commit 667f34a0fbc688e4167f298f72184fd4674b4416
Author: cc4351 <cc4351@columbia.edu>
Date: Tue Feb 23 16:08:42 2021 +0800

anon func expression added

parser.mly | 26 ++++++++++++++++++++++----
1 file changed, 22 insertions(+), 4 deletions(-)

commit 9ca0c2809fa9a69e7509572f04ff50e5df5adc44
Author: cc4351 <cc4351@columbia.edu>
Date: Tue Feb 23 15:53:45 2021 +0800

parser comments

parser.mly | 57 ++++++++++++++++++++++++++++++---------------------------

```
 1 file changed, 30 insertions(+), 27 deletions(-)

commit e271db57c3f090578bc90a1836a7801c5d84550a
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Feb 23 14:47:48 2021 +0800

    added dot operator; added assoc in parser

 parse.mly   | 136 ----------------------------------------------------------
 parser.mly  |   9 ++--
 scanner.mll |   1 +
 3 files changed, 7 insertions(+), 139 deletions(-)

commit 628cc6274575c638d0bae08edab2de36a5880409
Author: cc4351 <cc4351@columbia.edu>
Date:   Tue Feb 23 14:28:46 2021 +0800

    mv parse to parser; add token to parser

 parser.mly | 139
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 139 insertions(+)

commit 765377de768d100c241c8ca3839657efd90c17ec
Author: samlee815 <yl4111@columbia.edu>
Date:   Tue Feb 23 13:45:10 2021 +0800

    modify function parsing in parse.mly ast.ml

 ast.ml    | 16 +++++++---------
 parse.mly | 30 ++++++++++++++++++++++++++------
 2 files changed, 31 insertions(+), 15 deletions(-)

commit fe7502abbe15275349c6dd7e47c8c8c1ce8da31d
```

```
Author: keyuyan1145 <keyuyan01@gmail.com>
Date:   Mon Feb 22 09:04:12 2021 -0500

    initial commit for ast, parser, scanner

 ast.ml      | 132
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 parse.mly   | 118 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 scanner.mll |  62 +++++++++++++++++++++++++++++++
 3 files changed, 312 insertions(+)

commit 69fa1dd640ae4c2922dd79a158976e2626ca638f
Author: keyuyan01 <keyuyan01@gmail.com>
Date:   Thu Feb 18 21:42:04 2021 -0500

    first commit

 README.md | 1 +
 1 file changed, 1 insertion(+)
```

9.3 Test Files

9.3.1 fail-array-types.err

```
Fatal error: exception Failure("illegal assignment int =
    float in a[0] = 4113.1")
```

9.3.2 fail-array-types.gmm

```
function int main()
{

    array<int> a;

    a = new(array<int>[5]);
    a[0] = 4113.1;
    print(a[0]);

}
```

9.3.3 fail-assign1.err

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

9.3.4 fail-assign1.gmm

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

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

9.3.5 fail-assign2.err

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

9.3.6 fail-assign2.gmm

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

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

9.3.7 fail-assign3.err

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

9.3.8 fail-assign3.gmm

```
function void myvoid()
{
  return;
}

function int main()
{
  int i;

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

9.3.9 fail-channel-type.err

```
Fatal error: exception Failure("channel type: float and
   data type: int mismatch")
```

9.3.10 fail-channel-type.gmm

```
function int main()
{
    channel<int> a;
    float b;

    b -> a;
}
```

9.3.11 fail-dead1.err

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

9.3.12 fail-dead1.gmm

```
function int main()
{
  int i;

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

9.3.13 fail-dead2.err

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

9.3.14 fail-dead2.gmm

```
function int main()
{
  int i;

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

9.3.15 fail-expr1.err

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

9.3.16 fail-expr1.gmm

```
int a;
bool b;

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

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

9.3.17 fail-expr2.err

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

9.3.18 fail-expr2.gmm

```
int a;
bool b;

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

function int main()
```

```
{
  return 0;
}
```

9.3.19 fail-expr3.err

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

9.3.20 fail-expr3.gmm

```
int a;
float b;

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

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

9.3.21 fail-float1.err

```
Fatal error: exception Failure("illegal binary operator
    float && int in -3.5 && 1")
```

9.3.22 fail-float1.gmm

```
function int main()
{
  -3.5 && 1; /* Float with AND? */
  return 0;
}
```

9.3.23 fail-float2.err

```
Fatal error: exception Failure("illegal binary operator
    float && float in -3.5 && 2.5")
```

9.3.24 fail-float2.gmm

```
function int main()
{
  -3.5 && 2.5; /* Float with AND? */
  return 0;
}
```

9.3.25 fail-for1.err

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

9.3.26 fail-for1.gmm

```
function int main()
{
  int i;
  for ( ; true ; ) {} /* OK: Forever */

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

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

  return 0;
}
```

9.3.27 fail-for2.err

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

9.3.28 fail-for2.gmm

```
function int main()
{
  int i;

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

  return 0;
}
```

9.3.29 fail-for3.err

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

9.3.30 fail-for3.gmm

```
function int main()
{
  int i;

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

  return 0;
}
```

9.3.31 fail-for4.err

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

9.3.32 fail-for4.gmm

```
function int main()
{
  int i;

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

  return 0;
}
```

9.3.33 fail-for5.err

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

9.3.34 fail-for5.gmm

```
function int main()
{
  int i;
```

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

  return 0;
}
```

9.3.35 fail-func1.err

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

9.3.36 fail-func1.gmm

```
function int foo() {}

function int bar() {}

function int baz() {}

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

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

9.3.37 fail-func2.err

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

9.3.38 fail-func2.gmm

```
function int foo(int a, bool b, int c) { }

function void bar(int a, bool b, int a) {} /* Error:
   duplicate formal a in bar */

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

9.3.39 fail-func3.err

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

9.3.40 fail-func3.gmm

```
function int foo(int a, bool b, int c) { }

function void bar(int a, void b, int c) {} /* Error:
   illegal void formal b */

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

9.3.41 fail-func4.err

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

9.3.42 fail-func4.gmm

```
function int foo() {}

function void bar() {}

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

function void baz() {}

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

9.3.43 fail-func5.err

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

9.3.44 fail-func5.gmm

```
function int foo() {}

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

  return 0;
}

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

9.3.45 fail-func6.err

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

9.3.46 fail-func6.gmm

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

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

9.3.47 fail-func7.err

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

9.3.48 fail-func7.gmm

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

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

9.3.49 fail-global1.err

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

9.3.50 fail-global1.gmm

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


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

9.3.51 fail-global2.err

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

9.3.52 fail-global2.gmm

```
int b;
bool c;
int a;
int b; /* Duplicate global variable */

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

9.3.53 fail-gofunc-main.err

```
Fatal error: exception Failure("main has to be regular
    function")
```

9.3.54 fail-gofunc.err

```
Fatal error: exception Failure("regular call on
    gofunction gof")
```

9.3.55 fail-gofunc.gmm

```
gofunction void gof(string a) {
    for (i = 0 ; i < 3 ; i = i + 1) {
        prints(a);
    }
}

function int main() {
    normalf("aaa");
    gof("bbb");
}
```

9.3.56 fail-gofunc2.err

```
Fatal error: exception Failure("go call on regular
    function f")
```

9.3.57 fail-gofunc2.gmm

```
function void f(string a) {
    for (i = 0 ; i < 3 ; i = i + 1) {
        prints(a);
    }
}

function int main() {
    go f("bbb");
}
```

9.3.58 fail-gofunc3.err

```
Fatal error: exception Failure("no more than three args
   in gofunctioncall")
```

9.3.59 fail-gofunc3.gmm

```
gofunction void add(int a, int b, int c, int d) {
    int i;
    int s;
    for ( i=0; i < 100; i = i + 1) {
        print(i);
    }
    s = a + b + c + d;
    return;
}

function int main() {
    int x;
    int y;
    int i;

    x = 5;
    y = 10;

    go add(x, y, x, y);
    go add(x+y, y, x, y);

    return 0;
}
```

9.3.60 fail-if1.err

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

9.3.61 fail-if1.gmm

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

9.3.62 fail-if2.err

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

9.3.63 fail-if2.gmm

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

9.3.64 fail-if3.err

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

9.3.65 fail-if3.gmm

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

9.3.66 fail-nomain.err

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

9.3.67 fail-nomain.gmm

9.3.68 fail-ppmm.err

```
Fatal error: exception Failure("illegal binary operator
    float - int in x - 1")
```

9.3.69 fail-ppmm.gmm

```
function int main() {
    float x;
    x = 1.2;
    print(x++);
    print(x+1);
    print(x--);
}
```

9.3.70 fail-print.err

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

9.3.71 fail-print.gmm

```
/* Should be illegal to redefine */
function void print() {}
```

9.3.72 fail-printb.err

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

9.3.73 fail-printb.gmm

```
/* Should be illegal to redefine */
function void printb() {}
```

9.3.74 fail-return1.err

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

9.3.75 fail-return1.gmm

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

9.3.76 fail-return2.err

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

9.3.77 fail-return2.gmm

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

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

9.3.78 fail-struct-field.err

```
Fatal error: exception Failure("field not found")
```

9.3.79 fail-struct-field.gmm

```
structdef One{
        int x,
        float y,
        bool z,
        string r,
};

struct One x;

function int main(){
        int z;
        x = new(struct One,4115,1.2,true,"hello");
    print(x.a);
}
```

9.3.80 fail-struct-types.err

```
Fatal error: exception Failure("fields of struct does not
    match bool expected float in 1.0")
```

9.3.81 fail-struct-types.gmm

```
structdef One{
        int x,
        float y,
        bool z,
        string r,
};

struct One x;

function int main(){
        int z;
        x = new(struct One,4115,1.2,1.0,"hello");
}
```

9.3.82 fail-void.err

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

9.3.83 fail-void.gmm

```
function int main()
{
    void a;
}
```

9.3.84 fail-while1.err

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

9.3.85 fail-while1.gmm

```
function int main()
{
  int i;

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

  while (42) { /* Should be boolean */
```

```
    i = i + 1;
  }

}
```

9.3.86 fail-while2.err

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

9.3.87 fail-while2.gmm

```
function int main()
{
  int i;

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

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

}
```

9.3.88 test-add1.gmm

```
function int add(int x, int y)
{
  return x + y;
}

function int main()
{
  print( add(17, 25) );
  return 0;
}
```

9.3.89 test-add1.out

```
42
```

9.3.90 test-arith1.gmm

```
structdef Twoint {
        int a,
        int b,
};

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

9.3.91 test-arith1.out

```
42
```

9.3.92 test-arith2.gmm

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

9.3.93 test-arith2.out

```
11
```

9.3.94 test-arith3.gmm

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

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

9.3.95 test-arith3.out

```
47
```

9.3.96 test-array-types.gmm

```
function int main()
{

    array<int> a;
    array<float> b;
    array<bool> c;
    array<string> d;

    /* int test */
    a = new(array<int>[5]);
    a[0] = 4113;
    a[1] = 4115;
    print(a[1]);
    a[1] = 4119;
    print(a[1]);

    /* float test */
    b = new(array<float>[6]);
    b[0] = 4.3;
    printf(b[0]);
    b[0] = 4.7;
    printf(b[0]);
    printf(b[1] = b[0]);
    b[2] = 44.9;

    /* bool test */
    c = new(array<bool>[7]);
    c[5] = true;
    c[3] = false;
    if (c[5] && c[3]) {
        prints("both true");
    } else {
        prints("at least one false");
    }

    /* string test */
    d = new(array<string>[9]);
    prints(d[0] = "hello");
    prints(d[0] + " Go--");
```

```
    prints(d[5] = "Go--");

}
```

9.3.97 test-array-types.out

```
4115
4119
4.3
4.7
4.7
at least one false
hello
hello Go--
Go--
```

9.3.98 test-array.gmm

```
function int main()
{

    array<int> a;

    a = new(array<int>[5]);
    a[0] = 4113;
    a[1] = 4119;
    print(a[0]);
    print(a[1]);
    print(4115);
}
```

9.3.99 test-array.out

```
4113
4119
4115
```

9.3.100 test-bool.gmm

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

```
    b = true;
    printb(true); /* 1 */
    printb(false); /* 0 */
    printb(b); /* 1 */
    printb(!b); /* 0 */
    printb(b && !b); /* 0 */
    printb(b || !b); /* 1 */
    printb(b && !b && b); /* 0 */
    printb(b || !b && b); /* 1 */
    printb(b||b&&!b); /* 1 */
}
```

9.3.101 test-bool.out

```
1
0
1
0
0
1
0
1
1
```

9.3.102 test-channel-bool.gmm

```
channel<bool> c;

gofunction void consume(int i)
{
    for(; i > 0; i = i - 1)
    {
        true->c;
    }
}

function int main()
{
    int i;
    c = new(channel<bool>[5]);
    go consume(100);
    for(i = 0; i < 100; i = i + 1) {
        c->;
        print(i);
```

```
    }
}
```

9.3.103 test-channel-bool.out

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

9.3.104 test-channel-float.gmm

```
channel<float> b;

gofunction void dump(float start, int end)
{
    int i;

    for(i = 0; i < end; i=i+1)
    {
        start = start + 1.0;
        start->b;
    }

}

function int main()
{
    int num_dump;
    int j;

    b = new(channel<float>[5]);
    num_dump = 100;
    go dump(3.14, num_dump);
    for(j = 0; j < num_dump; j = j+1)
    {
        printf(b->);
    }
}
```

9.3.105 test-channel-float.out

```
4.14
5.14
6.14
7.14
8.14
9.14
10.14
11.14
12.14
13.14
14.14
15.14
16.14
17.14
18.14
19.14
20.14
21.14
22.14
23.14
24.14
25.14
26.14
27.14
28.14
29.14
30.14
31.14
32.14
33.14
34.14
35.14
36.14
37.14
38.14
39.14
40.14
41.14
42.14
43.14
44.14
45.14
46.14
47.14
```

48.14
49.14
50.14
51.14
52.14
53.14
54.14
55.14
56.14
57.14
58.14
59.14
60.14
61.14
62.14
63.14
64.14
65.14
66.14
67.14
68.14
69.14
70.14
71.14
72.14
73.14
74.14
75.14
76.14
77.14
78.14
79.14
80.14
81.14
82.14
83.14
84.14
85.14
86.14
87.14
88.14
89.14
90.14
91.14
92.14
93.14

```
94.14
95.14
96.14
97.14
98.14
99.14
100.14
101.14
102.14
103.14
```

9.3.106 test-channel-int.gmm

```
channel<int> b;

gofunction void dump(int start, int end)
{
    int i;

    for(i = 0; i < end; i=i+1)
    {
        start++;
        start->b;
    }

}

function int main()
{
    int num_dump;
    int j;

    b = new(channel<int>[5]);
    num_dump = 100;
    go dump(3, num_dump);
    for(j = 0; j < num_dump; j = j+1)
    {
        print(b->);
    }
}
```

9.3.107 test-channel-int.out

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

96
97
98
99
100
101
102
103

9.3.108 test-channel-string.gmm

```
channel<string> b;
channel<bool> sig;

gofunction void dec(int i)
{
    int j;

    sig = new(channel<bool>[1]);
    for(j = 0; j < i; j = j+1)
    {
        prints(b->);
    }
    true->sig;
}

function int main()
{
    string j;
    int i;
    array<string> a;

    a = new(array<string>[10]);
    b = new(channel<string>[5]);

    /* pre-fill the array */
    a[0] = "hello";
    a[1] = "Go";
    a[2] = "--";
    a[3] = "team";
    a[4] = "testing";
    a[5] = "second";
    a[6] = "round";
    a[7] = "might";
    a[8] = "hang";
```

```
    a[9] = "careful";

    go dec(10);
    for(i = 0; i < 10; i = i+1)
    {
        a[i]->b;
    }
    sig->;


}
```

9.3.109 test-channel-string.out

```
hello
Go
---
team
testing
second
round
might
hang
careful
```

9.3.110 test-channel-struct.gmm

```
structdef Kv{
        string k,
        int v,
};

channel<struct Kv> quit;

gofunction void demo()
{
    struct Kv cd;
    cd = new(struct Kv, "goodbye", 4115);
    cd.v++;
    cd -> quit;
    prints("returning");
}

function int main()
```

```
{
    struct Kv kv;
    struct Kv ab;
    int i;

    i = 4995;
    kv = new(struct Kv, "hello", i);
    quit = new(channel<struct Kv>[1]);
    go demo();
    kv -> quit;
    for(i = 0; i < 2; i++) {
        quit-> ab;
        prints(ab.k);
        print(ab.v);
    }
}
```

9.3.111 test-channel-struct.out

```
hello
4995
returning
goodbye
4116
```

9.3.112 test-fib 2.out

9.3.113 test-fib.gmm

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

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

```
  return 0;
}
```

9.3.114 test-fib.out

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

9.3.115 test-float.gmm

```
/* ==, <, <=, <=, < operator */
function void compare(float a, float b) {
    if (a != b) {
        prints("!=");
    }
    if (a == b){
        prints("==");
    } else if( a <= b){
        prints("<");
        if (a == b) {
            prints("a <= b should not happen in this br")
                ;
        }
    } else if (a >= b) {
        prints(">");
        if (a == b) {
            prints("a>=b should not happen in this br");
        }
    }

}

/* arithmatics with integer */
function void mul_div(float a, float b)
{
    printf(a*b);
    printf(a*b/a*b);
    printf(a + b/a);
    printf((a+b)/a);
}
```

```
function int main()
{
    float a;
    float b;

    a = 1.6;
    b = 3.2;

    compare(a, b);
    compare(b, a);
    compare(a, a);
    compare(-b, -a);
    mul_div(a, b);
    mul_div(-a, -b);
}
```

9.3.116 test-float.out

```
!=
<
!=
>
==
!=
<
5.12
10.24
3.6
3
5.12
10.24
0.4
3
```

9.3.117 test-float1.gmm

```
function int main()
{
  float a;
  a = 3.14159267;
  printf(a);
```

```
  return 0;
}
```

9.3.118 test-float1.out

```
3.14159
```

9.3.119 test-float2.gmm

```
function int main()
{
  float a;
  float b;
  float c;
  a = 3.14159267;
  b = -2.71828;
  c = a + b;
  printf(c);
  return 0;
}
```

9.3.120 test-float2.out

```
0.423313
```

9.3.121 test-float3.gmm

```
function void testfloat(float a, float b)
{
  printf(a + b);
  printf(a - b);
  printf(a * b);
  printf(a / b);
  printb(a == b);
  printb(a == a);
  printb(a != b);
  printb(a != a);
  printb(a > b);
  printb(a >= b);
  printb(a < b);
  printb(a <= b);
}
```

```
function int main()
{
  float c;
  float d;

  c = 42.0;
  d = 3.14159;

  testfloat(c, d);

  testfloat(d, d);

  return 0;
}
```

9.3.122 test-float3.out

```
45.1416
38.8584
131.947
13.369
0
1
1
0
1
1
0
0
6.28318
0
9.86959
1
1
1
0
0
0
1
0
1
```

9.3.123 test-for-pp.gmm

```
function int main()
{
    int i;

    for(i = 0; i < 5; i++){
        print(i);
    }
    for(i=7; i > 0; i--){
        print(i);
    }
    while(i++ < 4){
        print(i);
    }
    while(i--> -1){
        print(i);
    }
    while(i<10){
        print(i++);
    }
    while(i>-2){
        print(i--);
    }
}
```

9.3.124 test-for-pp.out

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

```
0
1
2
3
4
5
6
7
8
9
10
9
8
7
6
5
4
3
2
1
0
-1
-2
```

9.3.125 test-for1.gmm

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

9.3.126 test-for1.out

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

9.3.127 test-for2.gmm

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

9.3.128 test-for2.out

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

9.3.129 test-func1.gmm

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

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

9.3.130 test-func1.out

```
42
```

9.3.131 test-func2.gmm

```
/* Bug noticed by Pin-Chin Huang */

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

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

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

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

9.3.132 test-func2.out

```
2
```

9.3.133 test-func3.gmm

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

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

9.3.134 test-func3.out

```
42
17
```

```
192
8
```

9.3.135 test-func4.gmm

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

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

9.3.136 test-func4.out

```
62
```

9.3.137 test-func5.gmm

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

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

9.3.138 test-func5.out

9.3.139 test-func6.gmm

```
function void foo() {}

function int bar(int a, bool b, int c) { return a + c; }

function int main()
{
  print(bar(17, false, 25));
  return 0;
}
```

9.3.140 test-func6.out

```
42
```

9.3.141 test-func7.gmm

```
int a;

function void foo(int c)
{
  a = c + 42;
}

function int main()
{
  foo(73);
  print(a);
  return 0;
}
```

9.3.142 test-func7.out

```
115
```

9.3.143 test-func8.gmm

```
function void foo(int a)
{
  print(a + 3);
}
```

```
function int main()
{
  foo(40);
  return 0;
}
```

9.3.144 test-func8.out

```
43
```

9.3.145 test-func9.gmm

```
function void foo(int a)
{
  print(a + 3);
  return;
}

function int main()
{
  foo(40);
  return 0;
}
```

9.3.146 test-func9.out

```
43
```

9.3.147 test-gcd.gmm

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

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

```
  return 0;
}
```

9.3.148 test-gcd.out

```
2
3
11
```

9.3.149 test-gcd2.gmm

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

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

9.3.150 test-gcd2.out

```
7
4
11
```

9.3.151 test-global-array.gmm

```
array<int> arr;

function int main()
{
    arr = new(array<int>[5]);
    arr[1] = 555;
    print(arr[1]);
}
```

9.3.152 test-global-array.out

```
555
```

9.3.153 test-global1.gmm

```
int a;
int b;

function void printa()
{
  print(a);
}

function void printbb()
{
  print(b);
}

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

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

9.3.154 test-global1.out

```
42
21
43
22
```

9.3.155 test-global2.gmm

```
bool i;

function int main()
{
  int i; /* Should hide the global i */

  i = 42;
  print(i + i);
  return 0;
}
```

9.3.156 test-global2.out

```
84
```

9.3.157 test-global3.gmm

```
int i;
bool b;
int j;

function int main()
{
  i = 42;
  j = 10;
  print(i + j);
  return 0;
}
```

9.3.158 test-global3.out

```
52
```

9.3.159 test-goargs.gmm

```
structdef One{
        int y,
        string s,
};

structdef Two{
```

```
        string s,
        int i,
        float m,
};

gofunction void printfields(struct Two m,struct One i){
        prints(m.s);
        print(m.i);
        printf(m.m);
        print(i.y);
        prints(i.s);
}

gofunction void whi(struct One i){
    print(i.y);
}

function int main(){
        struct One m;
        struct Two two;
        int i;
        i = 0;
        m = new(struct One,1,"hello");
        two = new(struct Two,"struct two la",2,3.86);
        go printfields(two,m);
        while(i< 9999999999999){
                i = i+1;
        }
}
```

9.3.160 test-goargs.out

```
struct two la
2
3.86
1
hello
```

9.3.161 test-gocall-stress.gmm

```
channel<bool> quit;


function int fibs(int i){
```

```
    int j;
    int k;
    int start;
    int tmp;
    int ret;

    j = 1;
    k = 1;
    start = 2;
    for(;start < i; start = start + 1){
        tmp = j;
        j = k + j;
        k = tmp;
    }
    return j;
}

gofunction void stress(int i, int term)
{

    fibs(i);
    true -> quit;
}

function int main()
{
    int i;
    int num_calls;

    quit = new(channel<bool>[1]);
    num_calls = 10000;
    for(i = 0; i < num_calls; i = i + 1)
    {
        go stress(i, num_calls-1);
        quit->;
    }

}
```

9.3.162 test-gocall-stress.out

9.3.163 test-gocall-stress2.gmm

```
channel<bool> quit;
channel<bool> term;

gofunction void stress(int i)
{
    int j;
    for(j=0; j < i; j++) {
        true -> quit;
    }
}

gofunction void relief(int i)
{
    int k;
    for(k=0; k<i; k++){
        quit->;
    }
    true->term;
}

function int main()
{
    int i;
    int num_calls;

    quit = new(channel<bool>[16]);
    term = new(channel<bool>[1]);
    num_calls = 100000;
    go stress(num_calls);
    go relief(num_calls);
    term->;

}
```

9.3.164 test-gocall-stress2.out

9.3.165 test-gofunc1.gmm

```
channel<bool> sig;

gofunction void twoInt(int a, int b) {
    int i;
    for ( i=0; i < 100; i = i + 1) {
```

```
        print(i);
    }
    true->sig;
    return;
}

function int main() {
    int x;
    int y;
    int i;

    x = 5;
    y = 10;
    sig = new(channel<bool>[5]);

    go twoInt(x, y);
    go twoInt(x+y, y);

    sig->;
    sig->;

    return 0;
}
```

9.3.166 test-gofunc1.out

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
0
1
2
3
4
5
6
7
8
9
10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

9.3.167 test-hello.gmm

```
function int main()
{
  print(42);
  print(71);
  print(1);
  return 0;
}
```

9.3.168 test-hello.out

```
42
71
1
```

9.3.169 test-if1.gmm

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

9.3.170 test-if1.out

```
42
17
```

9.3.171 test-if2.gmm

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

9.3.172 test-if2.out

```
42
17
```

9.3.173 test-if3.gmm

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

9.3.174 test-if3.out

```
17
```

9.3.175 test-if4.gmm

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

9.3.176 test-if4.out

```
8
17
```

9.3.177 test-if5.gmm

```
function int cond(bool b)
{
  int x;
  if (b)
    x = 42;
  else
    x = 17;
  return x;
}

function int main()
{
 print(cond(true));
 print(cond(false));
```

```
 return 0;
}
```

9.3.178 test-if5.out

```
42
17
```

9.3.179 test-if6.gmm

```
function int cond(bool b)
{
  int x;
  x = 10;
  if (b)
    if (x == 10)
      x = 42;
  else
    x = 17;
  return x;
}

function int main()
{
 print(cond(true));
 print(cond(false));
 return 0;
}
```

9.3.180 test-if6.out

```
42
10
```

9.3.181 test-integer.gmm

```
/* ==, <, <=, <=, < operator */
function void compare(int a, int b) {
    if (a != b) {
        prints("!=");
    }
    if (a == b){
        prints("==");
```

```
    } else if( a <=  b){
        prints("<");
        if (a == b) {
            prints("a <= b should not happen in this br")
                ;
        }
    } else if (a >= b) {
        prints(">");
        if (a == b) {
            prints("a>=b should not happen in this br");
        }
    }

}

/* arithmatics with integer */
function void mul_div(int a, int b)
{
    print(a*b);
    print(a*b/a*b);
    print(a + b/a);
    print((a+b)/a);
}



function int main()
{
    int a;
    int b;

    a = 3;
    b = 2;

    compare(a, b);
    compare(b, a);
    compare(a, a);
    compare(-b, -a);
    mul_div(a, b);
    mul_div(-a, -b);
}
```

9.3.182 test-integer.out

```
!=
```

```
>
!=
<
==
!=
>
6
4
3
1
6
4
-3
1
```

9.3.183 test-local1.gmm

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

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

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

9.3.184 test-local1.out

```
84
```

9.3.185 test-local2.gmm

```
function int foo(int a, bool b)
{
  int c;
  bool d;

  c = a;
```

```
  return c + 10;
}

function int main() {
 print(foo(37, false));
 return 0;
}
```

9.3.186 test-local2.out

```
47
```

9.3.187 test-ops1.gmm

```
function 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;
}
```

9.3.188 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
```

9.3.189 test-ops2.gmm

```
function int main()
{
  printb(true);
  printb(false);
  printb(true && true);
  printb(true && false);
  printb(false && true);
  printb(false && false);
  printb(true || true);
  printb(true || false);
  printb(false || true);
  printb(false || false);
  printb(!false);
  printb(!true);
  print(-10);
  print(42+1);
```

```
}
```

9.3.190 test-ops2.out

```
1
0
1
0
0
0
1
1
1
0
1
0
-10
43
```

9.3.191 test-ppmm.gmm

```
function int main() {
    int x;
    x = 1;
    print(x++);
    print(x+1);
    print(x--);
}
```

9.3.192 test-ppmm.out

```
2
3
1
```

9.3.193 test-simple-string.gmm

```
function int main()
{
    string s;
    s = "hello 4115";
    prints(s);
}
```

9.3.194 test-simple-string.out

```
hello 4115
```

9.3.195 test-str.gmm

```
string s;
function int main() {
        string e;
    string f;
        s = "hello ";
        e = "this is ";
        f = "me";
        prints(s+e+f);
        prints("i am here");
}
```

9.3.196 test-str.out

```
hello this is me
i am here
```

9.3.197 test-str2.gmm

```
function int main(){
        if("hello" == "hello"){
                prints("should be printed");
        }
        if("hello" == "hi") {
                prints("should not be printed");
        }
        if("hello" != "hi"){
        prints("should be printed");
    }
    if("hello" != "hello") {
        prints("should not be printed");
    }

}
```

9.3.198 test-str2.out

```
should be printed
should be printed
```

9.3.199 test-struct.gmm

```
structdef One{
        int x,
        int y,
};

structdef Two{
        int x,
        float y,
        bool z,
        string r,
};

struct One x;

function void printaccess(array<int> a){
        print(a[0]);
        a[0] = 1;
        prints("array accessed");
}

function void printfield (struct Two i){
        print(i.x);
        printf(i.y);
        prints(i.r);
        i.x = 4118;
}

function int main(){
    array<int> a;
        struct Two y;
        int z;
        x = new(struct One,1,2);
        y = new(struct Two,4115,1.2,true,"hello");
        a = new(array<int>[5]);
        a[0] = 6;
        printaccess(a);
        print(a[0]);
        printfield(y);
        print(y.x);
}
```

9.3.200 test-struct.out

```
6
array accessed
1
4115
1.2
hello
4118
```

9.3.201 test-var1.gmm

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

9.3.202 test-var1.out

```
42
```

9.3.203 test-var2.gmm

```
int a;

function void foo(int c)
{
  a = c + 42;
}

function int main()
{
  foo(73);
  print(a);
  return 0;
}
```

9.3.204 test-var2.out

```
115
```

9.3.205 test-while1.gmm

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

9.3.206 test-while1.out

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

9.3.207 test-while2.gmm

```
function int foo(int a)
{
  int j;
  j = 0;
  while (a > 0) {
    j = j + 2;
    a = a - 1;
  }
  return j;
}

function int main()
{
  print(foo(7));
  return 0;
}
```

9.3.208 test-while2.out

```
14
```