GVL Final Report

Yaxin Chen	yc3995	Language Guru
Minhe Zhang	mz2864	System Architect
Jiawen Yan	jy3088	Manager
Aoxue Wei	aw3389	Tester

December 2021

Contents

1 Introduction

Graph-related algorithms are prevalent in networking, navigation, and data analysis. This manual describes GVL (Graph Visualization Language), an imperative language which is specialized to visualize graph data structures and algorithms.
GVL is a strong-typed and indentation-insensitive programming language with C-style syntax. It allows users to show algorithms, such as DFS and BFS, in colorful graphs. GVL compiler is mostly adapted from MicroC designed and implemented by Prof. Stephen A. Edwards.

2 Language Tutorial

2.1 Environment Setup

Before installing GVL, OCaml, LLVM 10, GCC, Clang, and OpenGL need to be installed. On Ubuntu, Ocaml and LLVM 10 can be installed with

```
> sudo apt install ocaml llvm-10 llvm-10-dev llvm-runtime m4 opam
> opam init
> opam install llvm.10.0.0 ocamlfind ocamlbuild
```

Clang and cmake can be installed with

```
> sudo apt install clang cmake make
```

For OpenGL, GLFW and Glad can be installed with

```
> sudo apt update
> sudo apt install libglfw3
> sudo apt install libglfw3-dev
> sudo apt install libglu1-mesa-dev
> sudo apt install xorg-dev
> git clone https://github.com/Dav1dde/glad.git
> cd glad
> cmake ./
> make
> sudo cp -a include /usr/local/
> sudo cp libglad.a /usr/lib/x86_64-linux-gnu/
```

2.2 Compilation Guide

Download GVL source code from https://github.com/MinheZhang/COMSW4115-GVL.git

```
> git clone https://github.com/MinheZhang/COMSW4115-GVL.git
```

The compiler can be built with make. This will also run all the tests.

```
> cd COMSW4115-GVL/src
> make clean
> make
```

To compile and execute a GVL program, you can use run.sh

```
> ./run.sh <gvl file>
```

2.3 Simple Program

GVL uses C-like syntax; please refer to the Language Reference Manual for more details. Below is an example of constructing and showing a graph in GVL language.

```
int main() {
  node n1;
  n1 = create_node(10.0, 20.0, 5.0, 0, 0, 0, 0);
  node n2;
  n2 = create_node(40.0, 50.0, 5.0, 0, 0, 0, 0);
  edge e12;
  e12 = create_edge(n1, n2, 1, 0, 0, 0);

  graph g;
  g = create_graph();
  add_node(g, n1);
  add_node(g, n2);
  add_edge(g, e12);

  show_graph(g);

  destroy_graph(g);

  return 0;
}
```

Save the above code to graph.gvl; compile and run it with run.sh. A graphic window will pop up showing a graph with two black nodes and one edge. The window can be closed by pressing 'Esc' or 'Enter'.

Figure 1: Output of graph.gvl

3 Language Reference Manual

3.1 Lexical conventions

There are six classes of tokens: identifiers, keywords, constants, string literals, operators, and separators. For adjacent identifiers, keywords, and constants, they must be separated with white space. Then white spaces and any characters within comments are ignored by the compiler.

3.1.1 Comments

For single line comment, we use two backslashes `//`. The characters after `//` in the same line will be ignored by compiler.
The characters `/*` introduce a multi-line comment and `*/` end it.
Comments do not nest and they do not occur within string or character literals.

3.1.2 Identifiers (Names)

An identifier is a sequence of letters and digits (the underscore `_` counts as a letter). Letters are case-sensitive and the first character of an identifier must be a letter. Identifiers can have any length. It can represent names of variables, functions, structures, and members of structure. A name has a scope. The same name in different scopes should refer to different data or functions.

3.1.3 Keywords

Following identifiers are reserved as keywords and may not be used otherwise:

```
int         float       char        bool        graph
node        edge        list        list_iterator
if          else        for         while
return      true        false
```

3.1.4 Constants

There are four kinds of constants in GVL, which are integer constant, floating-point number constant, character constant, and string constant.

Integer Constants An integer constant consisting of a sequence of digits is taken to be decimal.

Floating-point number Constants A floating constant consists of an integer part, a decimal point, and a fraction part. It should look like `123.456`.

Character Constants A character constant is a sequence of one of more characters which can only represent ASCII enclosed in single quotes. For example, a character constant might be `'a'`.

3.1.5 Separators

There are four kinds of separators.

- Semicolon `;` means the end of a variable declaration, an expression as a statement. `;` can also separate expressions in parentheses of `for` statement.
- A pair of curly braces `{}` surround and must surround block of statements and expressions. It is used to determine the block of function implementation, block of statements after branch like `if` and `else`, and block of statements after looping like `for` or `while`.
- A pair of parentheses surround and must surround the conditional checking expression after `if` or `while`. It also surround the initialization, condition checking, and updating statements after `for` and the arguments when defining and calling functions. Parentheses also change the precedence explicitly when evaluating an expression.
- Comma `,` separates arguments of function declaration or call.
  For example, `int fun(int a, int b) {...}` or `int a = fun(b, c);`.

3.2 Types

There are two categories of type: basic type and derived type. There are four basic types of which the keywords are:

```
int         float       bool        char
```

There are five derived types.

```
list        function
node        edge            graph
```

3.2.1 Integral Number

Integral numbers are represented by 32 bits and can contain integer from `-2147483648` to `2147483647`. They are declared or initialized using keyword `int`.

```
int a;
a = 1;
```

3.2.2 Floating-point Number

Floating-point numbers are represented by 32 bits and range from `-1.2E-38` to `3.4E+38`. They are declared using keyword `float`.

```
float a;
a = 1.0;
```

3.2.3 Boolean

Boolean type contains boolean value and are represented by 8 bits. It can be either `true` or `false` declared using keyword `bool`.

```
bool a;
a = true;
```

3.2.4 Character

Character type is able to contain a single ASCII character. The value of character is represented by 8 bits. It is declared using keyword `char`.

```
char a;
a = 'a';
```

3.2.5 List

List is a built-in container. It can be declared and constructed by

```
list l;
l = create_list();
```

Currently the data in list can only be pointer type, while our language does not support pointer. However, variables of type node, edge and graph in GVL are actually pointers and therefore list can stores nodes, edges, or graph.

3.2.6 Node

Node is a built-in compound type. It must has coordinate `(x,y)`, radius `(radius)`, and color `(r,g,b)`. It is declared using keyword `node`. For example,

```
node n1;
n1 = create_node(...);
```

Inside the parenthesises are the parameters of `node` constructor. The signature of `node` constructor is

```
create_node(float x, float y, float radius, int r, int g, int b, _);
```

The last argument is an interface provided for extend attributes for node. However, we was not able to implement this feature in GVL. User can simply put a 0 there and it will be ignored.

3.2.7 Edge

Edge is a built-in compound type. It has mandatory attributes containing endpoints(`start, end`), thickness(`t`), and color(`r, g, b`). It is declared using keyword `edge`:

```
edge e1;
e1 = create_edge(...);
```

Inside the parenthesises is the parameters of `edge` constructor which has a signature

```
int create_edge(node n1, node n2, int t, int r, int g, int b);
```

3.2.8 Graph

Graph is a built-in compound type. It has data containing nodes and edges and is declared using keyword `graph`:

```
graph g1;
g1 = create_graph();
```

3.3 Expressions

Expressions are a combination of literals, identifiers, operators, and function calls to be evaluated. The precedence of expression operators follow the order of the subsections in this section. Left/Right associativity is specified in each subsection.

3.3.1 Primary Expressions

Primary expressions are identifier, constant or expressions in parentheses.

```
primary-expression:
    identifier
    constant
    (expression)
```

3.3.2 Postfix Expressions

The operators in postfix expressions group left to right.

```
postfix-expression:
    primary-expression
    postfix-expression ( argument-expression-list-option)

argument-expression-list-option:
    empty
    argument-expression-list

argument-expression-list:
    assignment-expression
    argument-expression-list, assignment-expression
```

3.3.3 Function Calls

A function call is a postfix expression. It is followed by parentheses containing comma-separated list of assignment expressions(arguments of the function). The arguments could be empty. The type of function call is the same as the return type of that function.

Unary Operators Expression with unary operators group right-to-left.

```
unary-expression:
    postfix-expression
    unary-operator expression
unary-operator: one of
    - !
```

Unary Minus Operator The operand of the unary - operator must have arithmetic type, and the result is the negative of the operand.

Logical Negation Operator The operand of the unary `!` operator must be `bool` type, and the result is the negation boolean value of the operand.

3.3.4 Multiplicative Expressions

The multiplicative expressions group left-to-right.

```
expression * expression
expression / expression
```

The `*` operator denotes multiplication of arithmetic type operands.
The `/` operator denotes quotient of the first expression over the second expression.

3.3.5 Additive Expressions

The additive expressions group left-to-right.

```
expression + expression
expression - expression
```

3.3.6 Relational Expressions

The relational expressions, `>` (greater than), `<` (less than), `>=` (greater than or equal to), and `<=` (less than or equal to), group left-to-right. The result of relational expression is `true` if the relation is true, and `false` otherwise.

```
expression > expression
expression < expression
expression >= expression
expression <= expression
```

3.3.7 Equality Operators

Equality Operators `==` (equal to) and `!=` (not equal to) group left-to-right.The result of equality expression is `true` if the relation is true, and `false` otherwise.

```
expression == expression
expression != expression
```

3.3.8 Logical Expressions

Logical expressions group left-to-right.

```
expression && expression
expression || expression
```

The `&&` operator returns 1 if both operands are non-zero, 0 otherwise. The `||` operator returns 1 if one of the operands are non-zero, 0 otherwise. Each of the two operands must have one of the basic types.

3.3.9 Assignment Expressions

Assignment expressions group right-to-left. The left operand is an lvalue and the right operand is an expression. Two operands must have the same type. The result of assignment expression is the value stored in the left operand after assignment.

```
lvalue = expression
```

The above form of assignment expressions assigns the value of the expression on the right hand side to the lvalue on the left hand side.

3.4 Declarations

Declarations in GVL have the following form:

```
declaration:
    type-specifier declarator
```

where the type-specifier are specified type in Section 3 Types and the declarator are defined identifiers in section 2.

3.5 Statements

Statements are a sequence of GVL code that usually end with semicolon `;` or is delimited by braces `{}`.

3.5.1 Expression Statement

An expression statement is formed by an expression followed by a comma. For example, `y = x + 1;` and `y += 5;` are expression statements.

```
expression ;
```

3.5.2 Compound Statement

An compound statement is formed by list of statements delimited by braces.

```
{ statement-list }
```

where

```
statement-list:
    statement
    statement statement-list
```

3.5.3 If/Else Statement

If/Else statement executes statements conditionally. It follows the following forms

```
if ( expression ) statement
if ( expression ) statement else statement
```

For the first form, statement is executed if expression is evaluated as true. For the second form, if expression is evaluated as true, the first statement is executed; otherwise, the second statement is executed. An exception here is that the second statement in the second form can not only be compound statement but also if/else statement.

3.5.4 While Statement

While statement is a looping statement; it repetitively executes statement as long as the expression is evaluated as true.

```
while ( expression ) statement
```

3.5.5 For Statement

For statement is also a looping statement.

```
for ( expression1 ; expression2 ; expression3 ) statement
```

All of the expressions are optional. Before looping, expression1 is evaluated. Statement followed by expression3 are repetitively executed as long as expression2 is evaluated as true.

3.5.6 Return Statement

A function uses return statement to return to its caller, which has the following form:

```
return expression ;
```

An expression is evaluated and the result value is returned to the caller. GVL does not accept the form that no value is returned.

3.6 Built-in Functions and Attributes

3.6.1 Node

Node Constructor Signature of built-in function which return a reference of node variable:

```
node create_node(float x, float y, float radius, int r, int g, int b, int _);
```

1. `x` and `y` are the x-coordinate and y-coordinate of the node when visualization.

2. `radius` is a float type number larger than 0, which decides the radius of the node.

3. `r, g, b` are int type numbers in range $[0, 255]$ used to decide the node's color.

4. `int _` are placeholder for further implementation. It is used to pass in any type of payload that users want this node variable to carry.

The arguments in the constructors are all mandatory.

Node Attributes Use `set_node_{attribute}` to change a node's built-in attribute in x, y, radius, r, g, b. These attributes cannot be removed by users.

```
int set_node_x(node n, float x);
int set_node_y(node n, float y);
int set_node_radius(node n, float radius);
int set_node_r(node n, int r);
int set_node_g(node n, int g);
int set_node_b(node n, int b);
```

There is also a more concise way to set `node` color.

```
int set_node_color(node n, int r, int g, int b);
```

Use `get_node_{attribute}` to get a node's built-in attribute.

```
float get_node_x(node n);
float get_node_y(node n);
float get_node_radius(node n);
int get_node_r(node n);
int get_node_g(node n);
int get_node_b(node n);
```

Destroy Node To avoid memory leak, user can destroy node explicitly.

```
int destroy_node(node n);
```

3.6.2 Edge

Edge Constructor The signature of built-in `edge` constructor.

```
edge create_edge(node start, node end, int bold, int r, int g , int b);
```

Edge Attributes Use `set_edge_{attribute}` to change a edge's built-in attributes. These attributes cannot be removed by users.

```
int set_edge_start(edge e, node n);
int set_edge_end(edge e, node n);
int set_edge_bold(edge e, int bold);
int set_edge_r(edge e, int r);
int set_edge_g(edge e, int g);
int set_edge_b(edge e, int b);
```

There is also a more concise way to set `edge` color.

```
int set_edge_color(edge e, int r, int g, int b);
```

Use `get_edge_{attribute}` to get an `edge`'s built-in attribute.

```
node get_edge_start(edge e);
node get_edge_end(edge e);
int get_edge_bold(edge e);
int get_edge_r(edge e);
int get_edge_g(edge e);
int get_edge_b(edge e);
```

Destroy Edge To avoid memory leak, user can destroy `edge` explicitly.

```
int destroy_edge(edge e);
```

3.6.3 Graph

Graph Constructor Use `create_graph()` to build a new graph. Signature:

```
graph create_graph();
```

Graph Modification The following code are the signature of built-in function related to `graph` modification.

```
int add_node(graph g, node n);
int remove_node(graph g, node n);
int add_edge(graph g, edge e);
int remove_edge(graph g, edge e);
```

Alternatively, you can use operator "+o" to add node and operator "+-" to add edge.

```
g +o n;
g +- e;
```

Graph Attributes The following code are the signature of built-in function related to accessing to `graph` attributes.

```
list get_edges(graph g, node n);
```

Graph Visualization Signature of graph visualization.

```
int show_graph(graph g);
```

Destroy Graph To avoid memory leak, user can destroy `graph` explicitly.

```
int destroy_graph(graph g);
```

3.6.4 List

Construction An empty list can be constructed by calling the built-in function

```
create_list();
```

Insertion and Removal The list only supports inserting node or edge. Insertion and removal can be performed at the beginning or the end of the list.

```
insert_front(l, p); // l is of type list, p is of type node or edge
insert_back(l, p);
p = remove_front(l);
p = remove_end(l);
```

Iterator Iterator is provided to iterate through the list.

```
list l;
list_iterator l_iter;
for (l_iter = list_begin(l); l_iter != list_end(l);
     l_iter = list_iter_next(l_iter)) {
   // do something
}
```

Destruction All lists need to be destructed at the end, otherwise there may be memory leak.

```
destroy_list(l);
```

3.6.5 Print

These functions are implemented in MicroC by Prof. Stephen A. Edwards. They are very convenient for debugging.

```
int printi(int x);
int printb(bool x);
int printf(float x);
int printc(char x);
```

4 Project Plan

4.1 Development Process

At first, we shared our opinions on what kind of language to implement and the feasibility of GVL's features. We scheduled milestones which we should achieve during the semester. We used Github as our version control tool. Members first make change in their own branch and then create a pull request to the main branch after other members' review.

4.2 Software Development Environment

- Operating Systems: Ubuntu 20.04
- Container: Docker
- Languages: Ocaml, C, Bash
- Version control system: Git
- Other: OpenGL

4.3 Project Timeline

Date	**Milestone**
Oct 7	Project Proposal
Oct 30	Scanner and Parser(Basic)
Oct 31	Language Reference Manual
Nov 10	Hello World
Nov 29	Control Flow
Nov 30	Automatic Testing
Dec 14	GVL Built-In Function
Dec 17	Graph Visualization
Dec 20	BFS demo
Dec 21	Final Report and Presentation

4.4 Roles and Responsibilities

- Jiawen Yan - Manager
  Help with implementing Parser/AST, writing reports
- Yaxin Chen - Language Guru
  Scanner, parser, ast, sast, semant, codegen, makefile, test cases, C library, and built-in.ml. Slides, report.
- Minhe Zhang - System Architect
  Scanner, parser, ast, sast, semant, codegen, dockerfile, makefile, automatic testing, test cases, some drudgery in C library(Just a tiny proportion. Most of the meaningful work was implemented by Yaxin Chen), and built-in.ml. Slides, report, and README.md.

- Aoxue Wei - Tester
  Scanner, Parser, AST and Dockerfile with other members. Writing test cases to check GVL compiler's performance. Slides, report.

4.5 Programming Style Guide

We refer to https://www.seas.upenn.edu/~cis341/current/programming_style.shtml for our style guide.

- Each line of code should have less than 100 columns
- Comments should be above the code they reference.
- Use snake case for variables.
- Use two spaces for indentation.

4.6 Project Log

Commit history listed in appendix

5 Test Plan

5.1 Scanner, Parser, and AST

Every time when adding new features, we compiled the scanner, parser, AST to resolve conflicts before code generation.

5.2 Source to Target

5.2.1 Example 1

Source Language Program

```
int main() {

  graph g;
  g = create_graph();

  node n1;
  n1 = create_node(1.0, 2.0, 1.0, 255, 255, 255, 0);

  add_node(g, n1);
  remove_node(g, n1);

  destroy_graph(g);

  return 0;
}
```

Target Language Program

```
; ModuleID = 'GVL'
source_filename = "GVL"

%struct.node_t = type { double, double, double, i32, i32, i32, i8* }
```

```
%struct.edge_t = type { %struct.node_t*, %struct.node_t*, i32, i32, i32, i32 }
%struct.list_t = type { %struct.list_node_t*, %struct.list_node_t* }
%struct.list_node_t = type { i8*, %struct.list_node_t*, %struct.list_node_t* }

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00", align 1
@fmt.1 = private unnamed_addr constant [4 x i8] c"%g\0A\00", align 1
@fmt.2 = private unnamed_addr constant [4 x i8] c"%c\0A\00", align 1
@fmt.3 = private unnamed_addr constant [4 x i8] c"%s\0A\00", align 1

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

declare %struct.node_t* @create_node(double, double, double, i32, i32, i32, i8*)

declare %struct.edge_t* @create_edge(%struct.node_t*, %struct.node_t*, i32, i32,
↪  i32, i32)

declare i32 @edge_change_color(%struct.edge_t*, i32, i32, i32)

declare i32 @destroy_edge(%struct.edge_t*)

declare %struct.node_t* @get_edge_start(%struct.edge_t*)

declare %struct.node_t* @get_edge_end(%struct.edge_t*)

declare i32 @get_edge_bold(%struct.edge_t*)

declare i32 @get_edge_r(%struct.edge_t*)

declare i32 @get_edge_g(%struct.edge_t*)

declare i32 @get_edge_b(%struct.edge_t*)

declare i32 @set_edge_start(%struct.edge_t*, %struct.node_t*)

declare i32 @set_edge_end(%struct.edge_t*, %struct.node_t*)

declare i32 @set_edge_bold(%struct.edge_t*, i32)

declare i32 @set_edge_r(%struct.edge_t*, i32)

declare i32 @set_edge_g(%struct.edge_t*, i32)

declare i32 @set_edge_b(%struct.edge_t*, i32)

declare i32 @destroy_node(%struct.node_t*)

declare double @get_node_x(%struct.node_t*)

declare double @get_node_y(%struct.node_t*)

declare double @get_node_radius(%struct.node_t*)

declare i32 @get_node_r(%struct.node_t*)
```

```
declare i32 @get_node_g(%struct.node_t*)

declare i32 @get_node_b(%struct.node_t*)

declare i32 @set_node_x(%struct.node_t*, double)

declare i32 @set_node_y(%struct.node_t*, double)

declare i32 @set_node_radius(%struct.node_t*, double)

declare i32 @set_node_r(%struct.node_t*, i32)

declare i32 @set_node_g(%struct.node_t*, i32)

declare i32 @set_node_b(%struct.node_t*, i32)

declare i32 @set_node_color(%struct.node_t*, i32, i32, i32)

declare %struct.list_t* @create_graph()

declare i32 @add_node(%struct.list_t*, %struct.node_t*)

declare i32 @remove_node(%struct.list_t*, %struct.node_t*)

declare i32 @add_edge(%struct.list_t*, %struct.edge_t*)

declare i32 @remove_edge(%struct.list_t*, %struct.node_t*)

declare i32 @destroy_graph(%struct.list_t*)

declare %struct.list_t* @get_edges(%struct.list_t*, %struct.node_t*)

declare i32 @show_graph(%struct.list_t*)

declare %struct.list_t* @create_list()

declare i32 @is_empty(%struct.list_t*)

declare i32 @insert_front(%struct.list_t*, i8*)

declare i32 @insert_back(%struct.list_t*, i8*)

declare i8* @remove_front(%struct.list_t*)

declare i8* @remove_back(%struct.list_t*)

declare i32 @destroy_list(%struct.list_t*)

declare %struct.list_node_t* @list_begin(%struct.list_t*)

declare %struct.list_node_t* @list_end()

declare %struct.list_node_t* @list_iter_next(%struct.list_node_t*)
```

```
declare i8* @list_iter_data(%struct.list_node_t*)

define i32 @main() {
entry:
  %n1 = alloca %struct.node_t*
  %g = alloca %struct.list_t*
  %create_graph_result = call %struct.list_t* @create_graph()
  store %struct.list_t* %create_graph_result, %struct.list_t** %g
  %create_node = call %struct.node_t* @create_node(double 1.000000e+00, double
  ↪  2.000000e+00, double 1.000000e+00, i32 255, i32 255, i32 255, i8* null)
  store %struct.node_t* %create_node, %struct.node_t** %n1
  %n11 = load %struct.node_t*, %struct.node_t** %n1
  %g2 = load %struct.list_t*, %struct.list_t** %g
  %add_node_result = call i32 @add_node(%struct.list_t* %g2, %struct.node_t* %n11)
  %n13 = load %struct.node_t*, %struct.node_t** %n1
  %g4 = load %struct.list_t*, %struct.list_t** %g
  %remove_node_result = call i32 @remove_node(%struct.list_t* %g4, %struct.node_t*
  ↪  %n13)
  %g5 = load %struct.list_t*, %struct.list_t** %g
  %destroy_graph_result = call i32 @destroy_graph(%struct.list_t* %g5)
  ret i32 0
}
```

5.2.2 Example 2

Source Language Program

```
int main() {

  node n1;
  n1 = create_node(1.0, 2.0, 1.0, 255, 255, 255, 0);
  node n2;
  n2 = create_node(3.0, 4.0, 1.0, 255, 255, 255, 0);
  node n3;
  n3 = create_node(5.0, 4.0, 1.0, 255, 255, 255, 0);

  edge e12;
  e12 = create_edge(n1, n2, 1, 255, 255, 255);
  edge e13;
  e13 = create_edge(n1, n3, 1, 0, 0, 0);

  graph g;
  g = create_graph();

  g +o n1;
  g +o n2;
  g +o n3;
  g +- e12;
  g +- e13;

  list adjs;
  adjs = get_edges(g, n1);

  list_iterator iter;
  for (iter = list_begin(adjs); iter != list_end(); iter = list_iter_next(iter)) {
```

```
    edge e;
    e = list_iter_data(iter);
    node end_node;
    end_node = get_edge_end(e);
    printf(get_node_x(end_node));
  }

  destroy_graph(g);

  return 0;
}
```

Target Language Program

```
; ModuleID = 'GVL'
source_filename = "GVL"

%struct.node_t = type { double, double, double, i32, i32, i32, i8* }
%struct.edge_t = type { %struct.node_t*, %struct.node_t*, i32, i32, i32, i32 }
%struct.list_t = type { %struct.list_node_t*, %struct.list_node_t* }
%struct.list_node_t = type { i8*, %struct.list_node_t*, %struct.list_node_t* }

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00", align 1
@fmt.1 = private unnamed_addr constant [4 x i8] c"%g\0A\00", align 1
@fmt.2 = private unnamed_addr constant [4 x i8] c"%c\0A\00", align 1
@fmt.3 = private unnamed_addr constant [4 x i8] c"%s\0A\00", align 1

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

declare %struct.node_t* @create_node(double, double, double, i32, i32, i32, i8*)

declare %struct.edge_t* @create_edge(%struct.node_t*, %struct.node_t*, i32, i32,
↪  i32, i32)

declare i32 @edge_change_color(%struct.edge_t*, i32, i32, i32)

declare i32 @destroy_edge(%struct.edge_t*)

declare %struct.node_t* @get_edge_start(%struct.edge_t*)

declare %struct.node_t* @get_edge_end(%struct.edge_t*)

declare i32 @get_edge_bold(%struct.edge_t*)

declare i32 @get_edge_r(%struct.edge_t*)

declare i32 @get_edge_g(%struct.edge_t*)

declare i32 @get_edge_b(%struct.edge_t*)

declare i32 @set_edge_start(%struct.edge_t*, %struct.node_t*)

declare i32 @set_edge_end(%struct.edge_t*, %struct.node_t*)
```

```
declare i32 @set_edge_bold(%struct.edge_t*, i32)

declare i32 @set_edge_r(%struct.edge_t*, i32)

declare i32 @set_edge_g(%struct.edge_t*, i32)

declare i32 @set_edge_b(%struct.edge_t*, i32)

declare i32 @destroy_node(%struct.node_t*)

declare double @get_node_x(%struct.node_t*)

declare double @get_node_y(%struct.node_t*)

declare double @get_node_radius(%struct.node_t*)

declare i32 @get_node_r(%struct.node_t*)

declare i32 @get_node_g(%struct.node_t*)

declare i32 @get_node_b(%struct.node_t*)

declare i32 @set_node_x(%struct.node_t*, double)

declare i32 @set_node_y(%struct.node_t*, double)

declare i32 @set_node_radius(%struct.node_t*, double)

declare i32 @set_node_r(%struct.node_t*, i32)

declare i32 @set_node_g(%struct.node_t*, i32)

declare i32 @set_node_b(%struct.node_t*, i32)

declare i32 @set_node_color(%struct.node_t*, i32, i32, i32)

declare %struct.list_t* @create_graph()

declare i32 @add_node(%struct.list_t*, %struct.node_t*)

declare i32 @remove_node(%struct.list_t*, %struct.node_t*)

declare i32 @add_edge(%struct.list_t*, %struct.edge_t*)

declare i32 @remove_edge(%struct.list_t*, %struct.node_t*)

declare i32 @destroy_graph(%struct.list_t*)

declare %struct.list_t* @get_edges(%struct.list_t*, %struct.node_t*)

declare i32 @show_graph(%struct.list_t*)

declare %struct.list_t* @create_list()
```

```
declare i32 @is_empty(%struct.list_t*)

declare i32 @insert_front(%struct.list_t*, i8*)

declare i32 @insert_back(%struct.list_t*, i8*)

declare i8* @remove_front(%struct.list_t*)

declare i8* @remove_back(%struct.list_t*)

declare i32 @destroy_list(%struct.list_t*)

declare %struct.list_node_t* @list_begin(%struct.list_t*)

declare %struct.list_node_t* @list_end()

declare %struct.list_node_t* @list_iter_next(%struct.list_node_t*)

declare i8* @list_iter_data(%struct.list_node_t*)

define i32 @main() {
entry:
  %n3 = alloca %struct.node_t*
  %n2 = alloca %struct.node_t*
  %n1 = alloca %struct.node_t*
  %iter = alloca %struct.list_node_t*
  %g = alloca %struct.list_t*
  %end_node = alloca %struct.node_t*
  %e13 = alloca %struct.edge_t*
  %e12 = alloca %struct.edge_t*
  %e = alloca %struct.edge_t*
  %adjs = alloca %struct.list_t*
  %create_node = call %struct.node_t* @create_node(double 1.000000e+00, double
  ↪  2.000000e+00, double 1.000000e+00, i32 255, i32 255, i32 255, i8* null)
  store %struct.node_t* %create_node, %struct.node_t** %n1
  %create_node1 = call %struct.node_t* @create_node(double 3.000000e+00, double
  ↪  4.000000e+00, double 1.000000e+00, i32 255, i32 255, i32 255, i8* null)
  store %struct.node_t* %create_node1, %struct.node_t** %n2
  %create_node2 = call %struct.node_t* @create_node(double 5.000000e+00, double
  ↪  4.000000e+00, double 1.000000e+00, i32 255, i32 255, i32 255, i8* null)
  store %struct.node_t* %create_node2, %struct.node_t** %n3
  %n23 = load %struct.node_t*, %struct.node_t** %n2
  %n14 = load %struct.node_t*, %struct.node_t** %n1
  %create_edge_result = call %struct.edge_t* @create_edge(%struct.node_t* %n14,
  ↪  %struct.node_t* %n23, i32 1, i32 255, i32 255, i32 255)
  store %struct.edge_t* %create_edge_result, %struct.edge_t** %e12
  %n35 = load %struct.node_t*, %struct.node_t** %n3
  %n16 = load %struct.node_t*, %struct.node_t** %n1
  %create_edge_result7 = call %struct.edge_t* @create_edge(%struct.node_t* %n16,
  ↪  %struct.node_t* %n35, i32 1, i32 0, i32 0, i32 0)
  store %struct.edge_t* %create_edge_result7, %struct.edge_t** %e13
  %create_graph_result = call %struct.list_t* @create_graph()
  store %struct.list_t* %create_graph_result, %struct.list_t** %g
  %n18 = load %struct.node_t*, %struct.node_t** %n1
```

```
  %g9 = load %struct.list_t*, %struct.list_t** %g
  %add_node_result = call i32 @add_node(%struct.list_t* %g9, %struct.node_t* %n18)
  %n210 = load %struct.node_t*, %struct.node_t** %n2
  %g11 = load %struct.list_t*, %struct.list_t** %g
  %add_node_result12 = call i32 @add_node(%struct.list_t* %g11, %struct.node_t*
  ↪  %n210)
  %n313 = load %struct.node_t*, %struct.node_t** %n3
  %g14 = load %struct.list_t*, %struct.list_t** %g
  %add_node_result15 = call i32 @add_node(%struct.list_t* %g14, %struct.node_t*
  ↪  %n313)
  %e1216 = load %struct.edge_t*, %struct.edge_t** %e12
  %g17 = load %struct.list_t*, %struct.list_t** %g
  %add_edge_result = call i32 @add_edge(%struct.list_t* %g17, %struct.edge_t*
  ↪  %e1216)
  %e1318 = load %struct.edge_t*, %struct.edge_t** %e13
  %g19 = load %struct.list_t*, %struct.list_t** %g
  %add_edge_result20 = call i32 @add_edge(%struct.list_t* %g19, %struct.edge_t*
  ↪  %e1318)
  %n121 = load %struct.node_t*, %struct.node_t** %n1
  %g22 = load %struct.list_t*, %struct.list_t** %g
  %get_edges_result = call %struct.list_t* @get_edges(%struct.list_t* %g22,
  ↪  %struct.node_t* %n121)
  store %struct.list_t* %get_edges_result, %struct.list_t** %adjs
  %adjs23 = load %struct.list_t*, %struct.list_t** %adjs
  %list_begin_result = call %struct.list_node_t* @list_begin(%struct.list_t*
  ↪  %adjs23)
  store %struct.list_node_t* %list_begin_result, %struct.list_node_t** %iter
  br label %while

while:                                            ; preds = %while_body, %entry
  %iter28 = load %struct.list_node_t*, %struct.list_node_t** %iter
  %list_end_result = call %struct.list_node_t* @list_end()
  %tmp = icmp ne %struct.list_node_t* %iter28, %list_end_result
  br i1 %tmp, label %while_body, label %merge

while_body:                                       ; preds = %while
  %iter24 = load %struct.list_node_t*, %struct.list_node_t** %iter
  %list_iter_data_result = call i8* @list_iter_data(%struct.list_node_t* %iter24)
  %cast = bitcast i8* %list_iter_data_result to %struct.edge_t*
  store %struct.edge_t* %cast, %struct.edge_t** %e
  %e25 = load %struct.edge_t*, %struct.edge_t** %e
  %get_edge_end_result = call %struct.node_t* @get_edge_end(%struct.edge_t* %e25)
  store %struct.node_t* %get_edge_end_result, %struct.node_t** %end_node
  %end_node26 = load %struct.node_t*, %struct.node_t** %end_node
  %get_node_x_result = call double @get_node_x(%struct.node_t* %end_node26)
  %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4 x
  ↪  i8]* @fmt.1, i32 0, i32 0), double %get_node_x_result)
  %iter27 = load %struct.list_node_t*, %struct.list_node_t** %iter
  %list_iter_next_result = call %struct.list_node_t*
  ↪  @list_iter_next(%struct.list_node_t* %iter27)
  store %struct.list_node_t* %list_iter_next_result, %struct.list_node_t** %iter
  br label %while

merge:                                            ; preds = %while
```

```
  %g29 = load %struct.list_t*, %struct.list_t** %g
  %destroy_graph_result = call i32 @destroy_graph(%struct.list_t* %g29)
  ret i32 0
}
```

5.3 Test Suites

The test suites are adapted from `MicroC`.
Visualization-related test cases could be found in `src/visual_tests/`, and other test cases could be found in `src/tests/`. Every feature has correlated test cases.
Test cases that start with "fail-" end with and error, while those start with "test-" will be successfully compiled. Test output files end with ".out" or ".err" in the same directory with .gvl test files indicates the right output.
Every time a gvl file is tested, the new output file will be compared with the previous one. Command-line outputs "...OK" if the two outputs are the same, "...FAILED" otherwise. When expected outputs are different from temporary outputs, a ".diff" file will be generated.

```
fail-expr3...OK
fail-builtin...FAILED
  fail-builtin.err differs
fail-func1...OK
fail-func2...OK
fail-func3...OK
```

Figure 2: Test Outcomes

5.4 Test Automation

We modified the script from MicroC to automate testing process for GVL. One can use the following command to run all the tests (except for visualization-related ones) or run specific tests. The visualization-related tests should be ran individually so that the developers can check out-coming displayed on their window. For more

```
> ./testall.sh
> ./testall.sh tests/graph/test-edge1.gvl
> ./testall.sh visual_tests/dfs.gvl
```

5.5 Example Programs

The testing programs mainly include the following features:

- Variables - Check variable declarations and initialization. Implemented by Aoxue Wei, Minhe Zhang.
- Function - Check function declarations, scopes, and print functions. Implemented by Minhe Zhang, Aoxue Wei.
- Statement - Check for loops, while loops, if-else blocks. Implemented by Minhe Zhang.
- List - Check built-in functions such as creating, inserting and removing for list. Implemented by Yaxin Chen.
- Others - Check the legal comment style in GVL. Implemented by Yaxin Chen, Aoxue Wei.
- Graph - Check built-in functions such as creating and manipulating nodes, edges, and graphs. Implemented by Minhe Zhang, Yaxin Chen.

- Visualization - Check that the graphs could be shown in various environments. Implemented by Yaxin Chen.

Here is an example of a successfully compiled test file which outputs edge's boldness and color:

```
int main() {
  node n1;
  n1 = create_node(0.0, 1.0, 1.0, 255, 255, 255, 0);
  node n2;
  n2 = create_node(2.0, 1.0, 1.0, 255, 255, 255, 0);
  edge e;
  e = create_edge(n1, n2, 1, 1, 1, 1);

  set_edge_bold(e, 100);
  set_edge_r(e, 128);
  set_edge_g(e, 129);
  set_edge_b(e, 130);

  printi(get_edge_bold(e));
  printi(get_edge_r(e));
  printi(get_edge_g(e));
  printi(get_edge_b(e));

  return 0;
}
```

Users can compile and run test-edge4.gvl using the following command.

```
> ./testall.sh test-edge4.gvl
```

The expected output of the above program is:

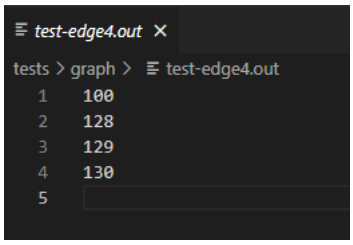

Figure 3: expected output for test-edge4.gvl

Another test example that outputs error due to bad assignment:

```
int main() {
  node n;
  n = create_node(1.0, 1.0, 1.0, 255, 255, 255, 0);
  int a;
  a = n;
}
```

```
> ./testall.sh fail-node1.gvl
```

The expected output of above program is:

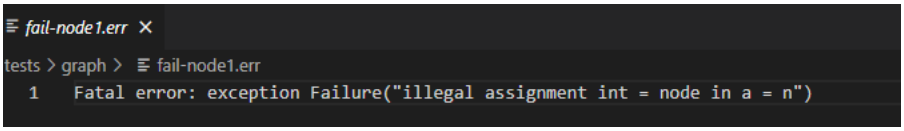

Figure 4: expected output for fail-node1.gvl

6 Architectural Design

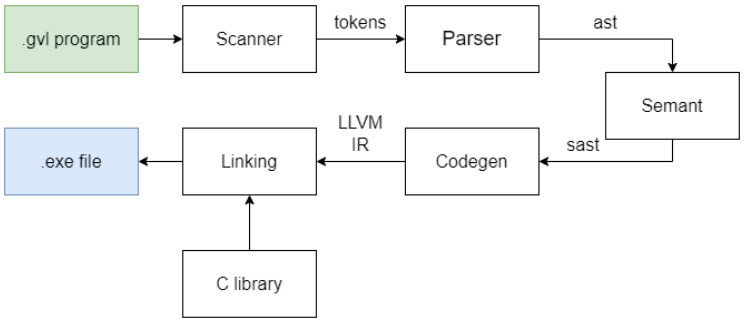

Figure 5: Diagram

GVL's compiler includes the following files:

- scanner.mll - A scanner written in OCamlLex. Generate GVL tokens from the source code or raise Scanner Errors. The tokens are passed to the Parser. White space, Tab, New line, and comment blocks are ignored. Implemented by Aoxue, Jiawen, Yaxin and Minhe.
- parser.mly, ast.ml - A parser written in OCamlYacc. Analyze the tokens from the scanner module, define grammar rules for the Abstract Syntax Tree. The parser make sure that the code is syntactically correct, otherwise, the parser will generate parse errors. Implemented by Aoxue, Jiawen, Yaxin and Minhe.
- builtin.ml - Defines syntax for GVL's built-in functions. Implemented by Yaxin and Minhe.
- semant.ml, sast.ml - The semantic checking files ensure the expression types in the program are correct. Convert AST into Semantically checked AST. Implemented by Yaxin and Minhe.
- codegen.ml - The code generation file contains the instructions on how to use GVL SAST to build up LLVM module. Then LLVM translates IRs into machine code. Implemented by Yaxin and Minhe.
- gvl.ml - Check and print the output of AST, SAST, LLVM IR. Implemented by Yaxin and Minhe.
- lib/list.c, list.h - Include implementation of GVL's list manipulating functions. Implemented by Yaxin.
- lib/graph.c, graph.h - Implement functions related to nodes, edges, and graphs. Implemented by Yaxin and Minhe.
- lib/graph_visualization.c - Include implement of graph visualizing functions. Libraries such as GLAD and GLFW are used in this part to create windows and images on the monitor. Implemented by Yaxin.

7 Lessons Learned

7.1 Jiawen Yan

I came with no knowledge of how compiler works and what is functional programming. But the project push me to understand a language by looking into its inner details. Actually implementing the scanner, parser etc provides me with a mindset of compiler design and I was able to bring all clues together of which I used to see as a "blackbox". Many thanks to my teammates for explaining details and taking the heavy responsibility. This has been a very challenging but rewarding CS course to me.

7.2 Minhe Zhang

These are both what I've learned and what I want to share with the future teams.

1. All these works made me look at programming language and compiler implementation in a new way. From vague feeling to concrete concepts, I forced myself to learn what is statically typed vs. dynamically typed, what is strongly typed vs. weakly typed, and so much other things. After being engaged in the whole project, whenever I stare at lines of code, it's never the same as before.

2. Ocaml is a incredibly efficient tool to implement a compiler compared to C++ or any other ones. The pattern matching and functional feature make coding smoothly.

3. Until tested, everything is garbage.

4. Stick to container if you can. Maintaining a dockerfile might be an overhead as the project goes on. But it's rewarding.

5. Iterations help a lot if you try to build a large program. Not only it makes the goals clear, but also makes your life easier by separating the tasks evenly.

6. MicroC is a priceless example. If you don't know where to start, consult it.

7.3 Yaxin Chen

This project helps me a lot in understanding how a translator works. There are several important things I learned during the implementation process. First, although I was not familiar with Ocaml at the beginning, it is not hard to catch. Ocaml is very powerful in scanning and parsing. Second, design the structure of the compiler carefully before implementation. Learning a bit about llvm before implementing codegen may give you ideas for how to implement it. Third, start early, which is one thing that I learned in every project. We cut a lot of features because we were running out of time. Fourth, it is better to get some integration tool like github action. We did not do it because of laziness, but using this kind of tools can protect your code from unexpecting bugs introduced by some pull requests.

7.4 Aoxue Wei

I learned many lessons from this class and this project. From a technical perspective, I gained a solid understanding of parser, syntax, and code generation, and it was amazing to try a functional programming language. I enjoyed learning about things happening behind code. Moreover, I learned that every bug happens for a reason. It is always important to trace back the log to identify what is going on rather than rely on solutions on the internet anxiously.
The process was challenging and rewarding. Great thanks to my knowledgeable flexible, and patient teammates. They were very supportive when I had difficulties installing packages, understanding the code, and debugging.

8 Appendix

8.1 Source Code

src/scanner.mll (Yaxin Chen, Jiawen Yan, Aoxue Wei, Minhe Zhang)

```
{
open Parser
exception ScannerError of string
}

let whitespace = [' ' '\t' '\r' '\n']
let letter = ['a'-'z' 'A'-'Z']
let digit = ['0'-'9']
let ascii_character = [' '-'!' '#'-'&' '('-'[' ']'-'~']  (* matches ascii
↪  printable char except quotes & backslash *)

rule tokenize = parse
  whitespace { tokenize lexbuf }
(* SEPARATORS *)
| '(' { LPAREN }
| ')' { RPAREN }
| '[' { LBRACKET }
| ']' { RBRACKET }
| '{' { LBRACE }
| '}' { RBRACE }
| ',' { COMMA }
| ';' { SEMI }
(* COMMENTS *)
| "//" { s_comment lexbuf }
| "/*" { m_comment lexbuf }
(* OPERATORS *)
| "==" { EQ }
| "!=" { NEQ }
| ">=" { GEQ }
| "<=" { LEQ }
| "+=" { PLUS_ASSIGN }
| "-=" { MINUS_ASSIGN }
| "*=" { TIMES_ASSIGN }
| "/=" { DIVIDE_ASSIGN }
| "%=" { MOD_ASSIGN }
| "&&" { AND }
| "||" { OR }
| '=' { ASSIGN }
| '>' { GT }
| '<' { LT }
| '+' { PLUS }
| '-' { MINUS }
| '*' { TIMES }
| '/' { DIVIDE }
| '%' { MOD }
(*| '&' { BIT_AND }
| '|' { BIT_OR }*)
| '!' { NOT }
```

```
| '.' { DOT }
| ':' { COLON }
| "++" { PLUSPLUS }
| "--" { MINUSMINUS }
| "+o" { PLUSNODE }
| "+-" { PLUSEDGE }
(* CONTROL FLOW KEYWORDS *)
| "if" { IF }
| "else" { ELSE }
| "while" { WHILE }
| "for" { FOR }
| "break" { BREAK }
| "continue" { CONTINUE }
| "return" { RETURN }
(* TYPES *)
| "bool" { BOOL }
| "int" { INT }
| "float" { FLOAT }
| "char" { CHAR }
| "string" {STRING}
| "struct" { STRUCT }
| "node" { NODE }
| "edge" { EDGE }
| "graph" { GRAPH }
| "list" { LIST }
| "list_iterator" { LISTITER }
(* CONSTANTS *)
| "true" { BOOLLIT(true) }
| "false" { BOOLLIT(false) }
| digit+ as lxm { INTLIT(int_of_string lxm) }
| (digit+) '.' (digit+) as lxm { FLOATLIT(lxm) }
| '"' { STRLIT(string_const "" lexbuf) }
(* CHAR CONSTANTS *)
| ''' ((ascii_character | '"') as lxm) ''' { CHARLIT(lxm) }
| ''' "\\\\" ''' { CHARLIT('\\') }
| ''' "\\n" ''' { CHARLIT('\n') }
| ''' "\\t" ''' { CHARLIT('\t') }
| ''' "\\r" ''' { CHARLIT('\r') }
(* IDENTIFIERS *)
| letter (letter | digit | '_')* as id { ID(id) }
(* EOF *)
| eof { EOF }
(* UNDEFINED *)
| _ { raise (ScannerError "illegal character") }

and s_comment = parse
  '\n' { tokenize lexbuf }
| eof { EOF }
| _ { s_comment lexbuf }

and m_comment = parse
  "*/" { tokenize lexbuf }
| eof { raise (ScannerError "unterminated comment") }
| _ { m_comment lexbuf }
```

```
and string_const result_str = parse
  '"' { result_str }
| "\\\\" { string_const (result_str ^ "\\") lexbuf }
| "\\n" { string_const (result_str ^ "\n") lexbuf }
| "\\t" { string_const (result_str ^ "\t") lexbuf }
| "\\r" { string_const (result_str ^ "\r") lexbuf }
| (ascii_character | ''') as c { string_const (result_str ^ (String.make 1 c))
↪  lexbuf}
| eof { raise (ScannerError "unterminated comment") }
| _ { raise (ScannerError "illegal character") }
```

src/ast.ml (Yaxin Chen, Jiawen Yan, Aoxue Wei, Minhe Zhang)

```
type op = Add | Sub | Mul | Div | Mod | Equal | Neq | Less | Leq | Greater | Geq |
          And | Or

type uop = Neg | Not

type typ = Bool | Int | Float | Char | String | StructID | Node | Edge | Graph |
↪  VoidPtr | GvlList | GvlListIterator

type bind = typ * string

type expr =
    Binop of expr * op * expr
|   Unop of uop * expr
|   Assign of string * expr
|   Id of string
|   IntLit of int
|   BoolLit of bool
|   FloatLit of string
|   CharLit of char
|   StrLit of string
    (* Function Call *)
|   Call of string * expr list
|   Noexpr

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

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

```
type program = bind list * func_decl list

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

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

let string_of_typ = function
    Bool -> "bool"
|   Int -> "int"
|   Float -> "float"
|   Char -> "char"
|   String -> "string"
|   StructID -> "struct" (* *)
|   Node -> "node"
|   Edge -> "edge"
|   Graph -> "graph"
|   VoidPtr -> ""
|   GvlList -> "list"
|   GvlListIterator -> "list_iterator"
    (* typ LBRACKET RBRACKET *)

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

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

let rec string_of_stmt = function
```

```
    Block(stmts) ->
      "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
  | Expr(e) -> string_of_expr e ^ ";\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|   Vdecl(b) ->
      ↪  string_of_bind b
  | 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
  | Return(e) -> "return " ^ string_of_expr e ^ ";\n"

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

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

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

src/parser.mly (Yaxin Chen, Jiawen Yan, Aoxue Wei, Minhe Zhang)

```
%{ open Ast %}
/* operators: logical */
%token EQ NEQ GEQ LEQ GT LT AND OR NOT
/* operators: arithmetic */
%token PLUS MINUS TIMES DIVIDE MOD
/* operators: assignment */
%token ASSIGN PLUS_ASSIGN MINUS_ASSIGN TIMES_ASSIGN DIVIDE_ASSIGN MOD_ASSIGN
/* operators: graph */
%token PLUSPLUS MINUSMINUS PLUSEDGE PLUSNODE
/* controal flow keywords*/
%token IF ELSE WHILE FOR BREAK CONTINUE RETURN
/* separators */
%token LPAREN RPAREN LBRACKET RBRACKET LBRACE RBRACE COMMA SEMI
/* types */
%token BOOL INT FLOAT CHAR STRING STRUCT NODE EDGE GRAPH LIST LISTITER
/* node/edge extension */
%token COLON
/* reference */
%token DOT
/* constants */
%token <bool> BOOLLIT
%token <int> INTLIT
%token <string> FLOATLIT
%token <string> STRLIT
```

```
%token <char> CHARLIT
/* identifiers */
%token <string> ID
/* end of file */
%token EOF

%nonassoc NOELSE
%nonassoc PLUSPLUS MINUSMINUS PLUSEDGE PLUSNODE
%left ELSE
%right ASSIGN PLUS_ASSIGN MINUS_ASSIGN TIMES_ASSIGN DIVIDE_ASSIGN MOD_ASSIGN
%left OR
%left AND
%left EQ NEQ LT GT LEQ GEQ
%left PLUS MINUS
%left TIMES DIVIDE MOD
%right NOT

%start program
// %type <Ast.expr> expr
%type <Ast.program> program

%%

program:
  decls EOF { $1 }

decls:
  /* nothing */ { ([], []) }
| decls vdecl { (($2 :: fst $1), snd $1) }
| decls fdecl { (fst $1, ($2 :: snd $1)) }
// TODO
// struct declaration.
//| decls sdecl {}
// array declaration.
//| decls adecl_assign {}

fdecl:
  typ ID LPAREN formals_opt RPAREN LBRACE stmt_list RBRACE
     { { typ = $1;
   fname = $2;
   formals = List.rev $4;
   body = List.rev $7 } }

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

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

// TODO in struct
// vdecl_list:
// /* nothing */ { [] }
```

```
// | vdecl_list vdecl { $2 :: $1 }

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

// TODO
/*
// Array declaration and assignment.
adecl_assign:
  typ ID ASSIGN array_lit SEMI {}

// Structure declaration and assignment.
sdecl:
  STRUCT ID LBRACE vdecl_list RBRACE SEMI {}
  // Inheritance of node and edge using struct.
| STRUCT ID COLON NODE LBRACE vdecl_list RBRACE SEMI {}
| STRUCT ID COLON EDGE LBRACE vdecl_list RBRACE SEMI {}
*/

typ:
  BOOL        { Bool }
| INT         { Int }
| FLOAT       { Float }
| CHAR        { Char }
| STRING      { String }
| STRUCT ID   { StructID }
| NODE        { Node }
| EDGE        { Edge }
| GRAPH       { Graph }
| LIST        { GvlList }
| LISTITER    { GvlListIterator }
//| typ LBRACKET expr RBRACKET {}  // TODO


/* statements */

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

stmt:
  expr SEMI                                { Expr ($1) }
| vdecl                                    { Vdecl($1) }
| 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_opt SEMI expr_opt RPAREN stmt { For($3, $5, $7,
↪  $9) }
| WHILE LPAREN expr RPAREN stmt           { While($3, $5) }
// | BREAK SEMI                              { Break }
// | CONTINUE SEMI                           { Continue }
| RETURN expr SEMI                         { Return($2) }
```

```
expr_opt:
  /* nothing */ { Noexpr }
| expr { $1 }

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

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

expr:
  expr PLUS   expr            { Binop($1, Add, $3) }
| expr MINUS  expr            { Binop($1, Sub, $3) }
| expr TIMES  expr            { Binop($1, Mul, $3) }
| expr DIVIDE expr            { Binop($1, Div, $3) }
| expr MOD expr               { Binop($1, Mod, $3) }
//| expr PLUSPLUS expr          { }  // TODO
//| expr MINUSMINUS expr        { }
| expr PLUSNODE expr          { Call("add_node", ($1 :: $3 :: [])) }
| expr PLUSEDGE expr          { Call("add_edge", ($1 :: $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) }
| MINUS expr %prec NOT        { Unop(Neg, $2) }
| NOT expr                    { Unop(Not, $2) }
| id ASSIGN expr              { Assign($1, $3) }
/*| id PLUS_ASSIGN expr         {}
| id MINUS_ASSIGN expr        {}
| id DIVIDE_ASSIGN expr       {}
| id TIMES_ASSIGN expr        {}
| id MOD_ASSIGN expr          {}*/ // TODO
| id                          { Id($1) }
| INTLIT                      { IntLit($1) }
| BOOLLIT                     { BoolLit($1) }
| FLOATLIT                    { FloatLit($1) }
| CHARLIT                     { CharLit($1) }
| STRLIT                      { StrLit($1) }
  /* function call */
| ID LPAREN args_opt RPAREN   { Call($1, $3) }
  // Primary expression
| LPAREN expr RPAREN          { $2 }

id:
  ID                          { $1 }
//| id DOT ID                   {} // TODO
  // Array
```

```
//| id LBRACKET expr RBRACKET   {} // TODO

// TODO
/*
array_lit:
  // {1, 2, 3}
  LBRACE args_list RBRACE        {}
  // {{1, 2}, {3, 4}}
| LBRACE array_lit_list RBRACE   {}

array_lit_list:
  array_lit                      {}
| array_lit_list COMMA array_lit {}
*/
```

src/sast.ml (Yaxin Chen, Minhe Zhang)

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

open Ast

type sexpr = typ * sx
and sx =
    SBinop of sexpr * op * sexpr
  | SUnop of uop * sexpr
  | SAssign of string * sexpr
  | SCall of string * sexpr list
  | SId of string
  | SIntLit of int
  | SBoolLit of bool
  | SFloatLit of string
  | SCharLit of char
  | SStrLit of string
  | SNoexpr

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

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

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

```
let rec string_of_sexpr (t, e) =
  "(" ^ string_of_typ t ^ " : " ^ (match e with
      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(v, e) -> v ^ " = " ^ string_of_sexpr e
    | SCall(f, el) ->
        f ^ "(" ^ String.concat ", " (List.map string_of_sexpr el) ^ ")"
    | SId(s) -> s
    | SIntLit(l) -> string_of_int l
    | SBoolLit(true) -> "true"
    | SBoolLit(false) -> "false"
    | SFloatLit(l) -> l
    | SStrLit(l) -> l
    | SCharLit(l) -> Char.escaped l
    | SNoexpr -> ""
                ) ^ ")"

let rec string_of_sstmt = function
    SExpr(expr) -> string_of_sexpr expr ^ ";\n"
  | SVdecl(b) -> string_of_bind b
  | 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
  | _ -> raise (Failure ("sast stmt not implemented"))

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

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

src/semant.ml (Yaxin Chen, Minhe Zhang)

```
open Ast
open Sast
open Builtin

module StringMap = Map.Make(String)

let check (globals, functions) =
        (* Collect function declarations for built-in functions: no bodies *)
  let built_in_decls =
```

```
  let add_bind map (name, formal_list, return_typ) = StringMap.add name {
    typ = return_typ;
    fname = name;
    formals = formal_list;
    body = [] } map
  in List.fold_left add_bind StringMap.empty semant_built_in_funcs
in

(* Add function name to symbol table *)
let add_func map fd =
  let built_in_err = "redefinition of built-in function '" ^ fd.fname ^ "'"
  and dup_err = "redefinition of function '" ^ fd.fname ^ "'"
  and make_err er = raise (Failure er)
  and n = fd.fname (* Name of the function *)
  in match fd with (* No duplicate functions or redefinitions of built-ins *)
       _ when StringMap.mem n built_in_decls -> make_err built_in_err
     | _ when StringMap.mem n map -> make_err dup_err
     | _ ->  StringMap.add n fd map
in

(* Collect all function names into one symbol table *)
let function_decls = List.fold_left add_func built_in_decls functions
in

(* Return a function from our symbol table *)
let find_func s =
  try StringMap.find s function_decls
  with Not_found -> raise (Failure ("unrecognized function " ^ s))
in

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

      let check_function func =
  (* Raise an exception if the given rvalue type cannot be assigned to
     the given lvalue type *)
  let check_assign lvaluet rvaluet err =
    if (lvaluet = rvaluet
        || (lvaluet = VoidPtr)
        || ((lvaluet = GvlListIterator || lvaluet = Node || lvaluet = Edge ||
        ↪  lvaluet = Graph) && rvaluet = VoidPtr))
     then lvaluet else raise (Failure err)
  in

  (* Build local symbol table of variables for this function *)
  let symbols = List.fold_left (fun m (ty, name) -> StringMap.add name ty m)
                      StringMap.empty (globals @ func.formals)
  in

  (* let find_in_locals s' locals' = match locals' with
    l :: ls -> try StringMap.find s' l
                with Not_found -> find_in_locals s' ls *)
        (* Return a variable from local symbol table *)
        (* let type_of_identifier s =
          try StringMap.find s symbols
```

```
        with Not_found -> raise (Failure ("undeclared identifier " ^ s))
      in *)
let type_of_identifier s locals =
        try StringMap.find s symbols
        with Not_found ->
    let rec find_in_locals s' locals' =
      match locals' with
        [] -> raise (Failure ("undeclared identifier " ^ s'))
      | l::ls -> try StringMap.find s' l
                 with Not_found -> find_in_locals s' ls
    in find_in_locals s locals
      in

            (* Return a semantically-checked expression, i.e., with a type *)
            let rec check_expr expr locals = match expr with
                Binop(e1, op, e2) as ex ->
                  let (t1, e1') = check_expr e1 locals
                  and (t2, e2') = check_expr e2 locals in
                  (* All binary operators require operands of the same type *)
                  let same = (t1 = t2) in
                  (* TODO other op & types e.g.
                  ↪  float/bool/char/string/node/graph *)
                  let ty = match op with
                    Add | Sub | Mul | Div | Mod
                      when same && t1 = Int   -> Int
                                    | Add | Sub | Mul | Div when same && t1 =
                                    ↪  Float          -> Float
                  | Equal | Neq            when same
                  ↪                    -> Bool
                  | Less | Leq | Greater | Geq
                      when same && (t1 = Int || t1 = Float)          -> Bool
                  | And | Or               when same && t1 = Bool
                  ↪                    -> Bool
                  | _ -> raise (
                       Failure ("illegal binary operator " ^
                           string_of_typ t1 ^ " " ^ string_of_op op ^ " " ^
                           string_of_typ t2 ^ " in " ^ string_of_expr ex))
                  in (ty, SBinop((t1, e1'), op, (t2, e2')))
              | Unop(op, e) as ex ->
                  let (t, e') = check_expr e locals in
                  let ty = match op with
                  | Neg when (t = Int || t = Float) -> t (* TODO Float *)
                  | Not when t = Bool -> t
                  | _ -> raise (Failure ("illegal unary operator " ^
                                         string_of_uop op ^ string_of_typ t ^
                                         " in " ^ string_of_expr ex))
                  in (ty, SUnop(op, (t, e')))
              | Assign(v, e) as ex ->
                  let v_t = type_of_identifier v locals
                  and (e_t, e') = check_expr e locals in
                  let err = "illegal assignment " ^ string_of_typ v_t ^ " = "
                  ↪  ^
        string_of_typ e_t ^ " in " ^ string_of_expr ex in
      let ty = check_assign v_t e_t err in
```

```
      (ty, SAssign(v, (e_t, e')))
  | Call(fname, args) as call ->
      let fd = find_func fname in
      let param_length = List.length fd.formals in
      if List.length args != param_length then
        raise (Failure ("expecting " ^ string_of_int param_length ^
                        " arguments in " ^ string_of_expr call))
      else let check_call (ft, _) e =
        let (et, e') = check_expr e locals in
        let err = "illegal argument found " ^ string_of_typ et ^
          " expected " ^ string_of_typ ft ^ " in " ^ string_of_expr e
        in (check_assign ft et err, e')
      in
      let args' = List.map2 check_call fd.formals args
      in (fd.typ, SCall(fname, args'))
  | Id s                -> (type_of_identifier s locals, SId s)
  | IntLit l      -> (Int, SIntLit l)
  | BoolLit l     -> (Bool, SBoolLit l)
  | FloatLit l    -> (Float, SFloatLit l)
  | CharLit l     -> (Char, SCharLit l)
  | StrLit l      -> (String, SStrLit l)
  | Noexpr        -> (Int, SNoexpr)
in

let check_bool_expr e locals =
  let (t', e') = check_expr e locals
  and err = "expected Boolean expression in " ^ string_of_expr e
  in if t' != Bool then raise (Failure err) else (t', e')
in

            (* TODO stmt *)
            (* let rec check_stmt (sstmt_list, locals) stmt = match stmt with
                Expr e -> (SExpr (check_expr e locals) :: sstmt_list, locals)
              | Vdecl (t, n) ->
    if StringMap.mem n symbols
      then raise (Failure ("redefinition of " ^ "'" ^ n ^ "'"))
      else if StringMap.mem n locals
            then raise (Failure ("redefinition of " ^ "'" ^ n ^ "'"))
            else (SVdecl (t, n) :: sstmt_list, StringMap.add n t locals)
  | Return e -> let (t, e') = check_expr e locals in
    if t = func.typ then (SReturn (t, e') :: sstmt_list, locals)
    else raise (Failure ("return gives " ^ string_of_typ t
                         ^ " expected " ^ string_of_typ func.typ
                         ^ " in " ^ string_of_expr e))
              | _ -> raise (Failure ("semant stmt not implemented")) *)
let rec check_stmt (sstmt_list, all_locals, curr_locals) stmt = match stmt
↪  with
    Expr e -> (SExpr (check_expr e curr_locals) :: sstmt_list, all_locals,
    ↪  curr_locals)
  | Vdecl (t, n) ->
    if StringMap.mem n symbols
      then raise (Failure ("redefinition of " ^ "'" ^ n ^ "'"))
      else if StringMap.mem n all_locals
            then raise (Failure ("redefinition of " ^ "'" ^ n ^ "'"))
```

```
            else (SVdecl (t, n) :: sstmt_list,
                  StringMap.add n t all_locals,
                  (StringMap.add n t (List.hd curr_locals)) :: List.tl
                  ↪  curr_locals)
  | Return e -> let (t, e') = check_expr e curr_locals in
    if t = func.typ then (SReturn (t, e') :: sstmt_list, all_locals,
    ↪  curr_locals)
    else raise (Failure ("return gives " ^ string_of_typ t
                         ^ " expected " ^ string_of_typ func.typ
                         ^ " in " ^ string_of_expr e))
  | Block sl ->
      let rec check_stmt_list = function
          Return _ :: _ -> raise (Failure "nothing may follow a return")
        | s :: ss -> check_stmt_list ss
        | [] -> () in
      check_stmt_list sl;
      let (blist, new_all_locals, _) =
        List.fold_left check_stmt ([], all_locals, StringMap.empty ::
        ↪  curr_locals) sl
      in
      (SBlock(List.rev blist)::sstmt_list, new_all_locals, curr_locals)
  | If(p, b1, b2) ->
      let sp = check_bool_expr p curr_locals in
      let (b1_sstmt_list, b1_all_locals, _) = check_stmt ([], all_locals,
      ↪  curr_locals) b1 in
      let (b2_sstmt_list, b2_all_locals, _) = check_stmt ([], b1_all_locals,
      ↪  curr_locals) b2 in
      (SIf(sp, List.hd b1_sstmt_list, List.hd b2_sstmt_list) :: sstmt_list,
      ↪  b2_all_locals, curr_locals)
  | For(e1, e2, e3, st) ->
    let (for_sstmt, new_all_locals, _) = check_stmt ([], all_locals,
    ↪  curr_locals) st in
    (SFor(check_expr e1 curr_locals,
          check_bool_expr e2 curr_locals,
          check_expr e3 curr_locals,
          List.hd for_sstmt) :: sstmt_list,
      new_all_locals,
      curr_locals)
  | While(p, s) ->
      let (while_sstmt, new_all_locals, _) = check_stmt ([], all_locals,
      ↪  curr_locals) s in
      (SWhile(check_bool_expr p curr_locals, List.hd while_sstmt) ::
      ↪  sstmt_list,
        new_all_locals,
        curr_locals)
  | _ -> raise (Failure ("semant stmt not implemented"))

      in

let slocals_map_to_list n t slocals_list =
  (t, n) :: slocals_list
in
```

```
    let (sbody, slocals_map, _) = List.fold_left check_stmt ([], StringMap.empty,
    ↪  [StringMap.empty]) func.body (* TODO *)
    in
                { styp = func.typ;
      sfname = func.fname;
      sformals = func.formals;
      (* slocals  = func.locals; *)
      (* TODO: reverse list. *)
      sbody  = List.rev sbody;
      slocals = StringMap.fold slocals_map_to_list slocals_map []
    }
        in (globals, List.map check_function functions)
```

src/builtin.ml (Yaxin Chen, Minhe Zhang)

```
module L = Llvm
open Ast

module StringMap = Map.Make(String)

let built_in_funcs = [
  (* Edge Function *)
  ("create_edge", [
    (Node, "start");
    (Node, "end");
    (Int, "bold");
    (Int, "r");
    (Int, "g");
    (Int, "b")], Edge);
  ("edge_change_color", [
    (Edge, "e");
    (Int, "r");
    (Int, "g");
    (Int, "b")], Int);
  ("destroy_edge", [(Edge, "e")], Int);
  (* Get Edge Attributes *)
  ("get_edge_start", [(Edge, "e")], Node);
  ("get_edge_end", [(Edge, "e")], Node);
  ("get_edge_bold", [(Edge, "e")], Int);
  ("get_edge_r", [(Edge, "e")], Int);
  ("get_edge_g", [(Edge, "e")], Int);
  ("get_edge_b", [(Edge, "e")], Int);
  (* Set Edge Attributes *)
  ("set_edge_start", [(Edge, "e"); (Node, "start")], Int);
  ("set_edge_end", [(Edge, "e"); (Node, "end")], Int);
  ("set_edge_bold", [(Edge, "e"); (Int, "bold")], Int);
  ("set_edge_r", [(Edge, "e"); (Int, "r")], Int);
  ("set_edge_g", [(Edge, "e"); (Int, "g")], Int);
  ("set_edge_b", [(Edge, "e"); (Int, "b")], Int);
  (* Node Functions *)
  ("destroy_node", [(Node, "n")], Int);
  (* Get Node Attributes *)
  ("get_node_x", [(Node, "n")], Float);
  ("get_node_y", [(Node, "n")], Float);
```

```
  ("get_node_radius", [(Node, "n")], Float);
  ("get_node_r", [(Node, "n")], Int);
  ("get_node_g", [(Node, "n")], Int);
  ("get_node_b", [(Node, "n")], Int);
  (* TODO get_node_extra *)
  (* Set Node Attributes *)
  ("set_node_x", [(Node, "n"); (Float, "x")], Int);
  ("set_node_y", [(Node, "n"); (Float, "y")], Int);
  ("set_node_radius", [(Node, "n"); (Float, "radius")], Int);
  ("set_node_r", [(Node, "n"); (Int, "r")], Int);
  ("set_node_g", [(Node, "n"); (Int, "g")], Int);
  ("set_node_b", [(Node, "n"); (Int, "b")], Int);
  (* TODO set_node_extra *)
  (* ("set_node_extra", [(Node, "n"); (VoidPtr, "extra")], Int); *)
  ("set_node_color", [(Node, "n"); (Int, "r"); (Int, "g"); (Int, "b")], Int);
  (* Graph Functions*)
  ("create_graph", [], Graph);
  ("add_node", [(Graph, "g"); (Node, "n")], Int);
  ("remove_node", [(Graph, "g"); (Node, "n")], Int);
  ("add_edge", [(Graph, "g"); (Edge, "e")], Int);
  ("remove_edge", [(Graph, "g"); (Node, "n")], Int);
  ("destroy_graph", [(Graph, "g")], Int);
  ("get_edges", [(Graph, "g"); (Node, "n")], GvlList);
  ("show_graph", [(Graph, "g")], Int);
  (* List Functions *)
  ("create_list", [], GvlList);
  ("is_empty", [(GvlList, "l")], Int);
  ("insert_front", [(GvlList, "l"); (VoidPtr, "data")], Int);
  ("insert_back", [(GvlList, "l"); (VoidPtr, "data")], Int);
  ("remove_front", [(GvlList, "l")], VoidPtr);
  ("remove_back", [(GvlList, "l")], VoidPtr);
  ("destroy_list", [(GvlList, "l")], Int);
  (* List Iterator Functions *)
  ("list_begin", [(GvlList, "l")], GvlListIterator);
  ("list_end", [], GvlListIterator);
  ("list_iter_next", [(GvlListIterator, "iter")], GvlListIterator);
  ("list_iter_data", [(GvlListIterator, "iter")], VoidPtr)
  ]

let semant_built_in_funcs = [
  ("printi", [(Int, "x")], Int);
  ("printb", [(Bool, "x")], Int);
  ("printf", [(Float, "x")], Int);
  ("printc", [(Char, "x")], Int);
  ("prints", [(String, "x")], Int);
  ("create_node", [
    (Float, "x");
    (Float, "y");
    (Float, "radius");
    (Int, "r");
    (Int, "g");
    (Int, "b");
    (VoidPtr, "data")], Node) ] @ built_in_funcs
```

```
let decl_built_in_functions ltype_of_typ the_module =
  (* return llvalue *)
  let decl_built_in_func (name, formal_list, return_typ) =
    let func_t : L.lltype =
      L.function_type (ltype_of_typ return_typ) (Array.of_list (List.map (fun (ty,
      ↪  _) -> ltype_of_typ ty) formal_list))
    in L.declare_function name func_t the_module
  in
  List.map (fun ((name, formal_list, return_typ) as func_sig) -> (name,
  ↪  decl_built_in_func func_sig)) built_in_funcs
```

src/codegen.ml (Yaxin Chen, Minhe Zhang)

```
module L = Llvm
module A = Ast
open Sast
open Builtin

module StringMap = Map.Make(String)

let translate (globals, functions) =
        let context = L.global_context () in
  let llmem = L.MemoryBuffer.of_file "graph" in
  let llm = Llvm_bitreader.parse_bitcode context llmem in
  let llmem_list = L.MemoryBuffer.of_file "list" in
  let llm_list = Llvm_bitreader.parse_bitcode context llmem_list in

        (* Create the LLVM compilation module into which
        we will generate code *)
        let the_module = L.create_module context "GVL" in

        let i32_t      = L.i32_type    context
  and i8_t       = L.i8_type     context
  and i1_t       = L.i1_type     context
  and float_t    = L.double_type context
  and node_t     = L.pointer_type (match L.type_by_name llm "struct.node_t" with
                                     None -> raise (Failure "the node
                                     ↪  type is not defined.")
                                   | Some x -> x)
  and edge_t     = L.pointer_type (match L.type_by_name llm "struct.edge_t" with
                                     None -> raise (Failure "the edge
                                     ↪  type is not defined.")
                                   | Some x -> x)
  and graph_t    = L.pointer_type (match L.type_by_name llm_list "struct.list_t"
  ↪  with
                                     None -> raise (Failure "the graph
                                     ↪  type is not defined.")
                                   | Some x -> x)
  and list_t     = L.pointer_type (match L.type_by_name llm_list "struct.list_t"
  ↪  with
                                     None -> raise (Failure "the list
                                     ↪  type is not defined.")
                                   | Some x -> x)
```

```
and list_iterator_t = L.pointer_type (match L.type_by_name llm_list
↪  "struct.list_node_t" with
                                                None -> raise (Failure "the list
                                                ↪  iterator type is not
                                                ↪  defined.")
                                              | Some x -> x)
and void_ptr_t = L.pointer_type (L.i8_type context)
in

      let ltype_of_typ = function
          A.Int -> i32_t
  | A.Bool  -> i1_t
  | A.Float -> float_t
  | A.Char -> i8_t
  | A.VoidPtr -> void_ptr_t
  | A.Node -> node_t
  | A.Edge -> edge_t
  | A.Graph -> graph_t
  | A.GvlList -> list_t
  | A.GvlListIterator -> list_iterator_t
      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
      | _ -> 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

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

let create_node_t : L.lltype =
  L.function_type node_t [| float_t; float_t; float_t; i32_t; i32_t; i32_t;
  ↪  void_ptr_t |] in
let create_node_func : L.llvalue =
  L.declare_function "create_node" create_node_t the_module in

(* built-in functions *)
let built_in_func_decls : L.llvalue StringMap.t =
  let f map (name, func) = StringMap.add name func map in
  List.fold_left f StringMap.empty (("printf", printf_func) ::
                                    ("create_node", create_node_func) ::
                                    (decl_built_in_functions ltype_of_typ
                                    ↪  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 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) m in
  List.fold_left function_decl built_in_func_decls functions in
(*
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
*)

      let build_function_body fdecl =
        (* TODO *)
  (* let (the_function, _) = StringMap.find fdecl.sfname function_decls in *)
  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 char_format_str = L.build_global_stringptr "%c\n" "fmt" builder
  and string_format_str = L.build_global_stringptr "%s\n" "fmt" builder in

  (* Construct the function's "locals":
      1. formal argurments,
      2. locally decalred 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

    (* Allocate space for any locally declared variables and add the
    * resulting registers to our map *)
    and add_local m (t, n) =
      let local_var = L.build_alloca (ltype_of_typ t) n builder
      in StringMap.add n local_var m
    in
    let formals = List.fold_left2 add_formal StringMap.empty fdecl.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

      let rec expr builder ((_, e) : sexpr) = match e with
    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.Mul     -> 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.Mod | A.And | A.Or ->
          raise (Failure "internal error: semant should have rejected
          ↪  mod/and/or on float")
      ) e1' e2' "tmp" builder
  | 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.Mul     -> L.build_mul
      | A.Div     -> L.build_sdiv
      | A.Mod     -> L.build_srem
      | 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, e) ->
            let e' = expr builder e in
            (match op with
              A.Neg -> L.build_neg
            | A.Not -> L.build_not) e' "tmp" builder

        | SAssign (v, e) ->
      (match e with
        (A.VoidPtr, _) ->
          let e' = expr builder e in
          let v_val = lookup v in
```

```
        let cast_typ = L.element_type (L.type_of v_val) in
        let cast_val = L.build_bitcast e' cast_typ "cast" builder in
          ignore(L.build_store cast_val v_val builder); cast_val
    | _ ->
      let e' = expr builder e in
        ignore(L.build_store e' (lookup v) builder); e')

      (* Function call *)
| SCall ("printi", [e]) | SCall ("printb", [e]) ->
    L.build_call printf_func [| int_format_str ; (expr builder e) |]
      "printf" builder
| SCall ("printf", [e]) ->
    L.build_call printf_func [| float_format_str ; (expr builder e) |]
      "printf" builder
| SCall ("printc", [e]) ->
    L.build_call printf_func [| char_format_str ; (expr builder e) |]
      "printf" builder
| SCall ("prints", [e]) ->
    L.build_call printf_func [| string_format_str ; (expr builder e) |]
      "printf" builder
(* TODO void pointer *)
| SCall ("create_node", [x; y; radius; r; g; b; data]) ->
  let x'      = (expr builder x)
  and y'      = (expr builder y)
  and radius' = (expr builder radius)
  and r'      = (expr builder r)
  and g'      = (expr builder g)
  and b'      = (expr builder b) in
  L.build_call create_node_func [| x'; y'; radius'; r'; g'; b';
  ↪  (L.const_pointer_null void_ptr_t) |]
    "create_node" builder
| SCall ("insert_front" as f, [l; data]) | SCall ("insert_back" as f, [l;
↪  data]) ->
    let cast_data = L.build_bitcast (expr builder data) void_ptr_t
    ↪  "cast_void" builder in
    let fdef = StringMap.find f function_decls in
    L.build_call fdef [| (expr builder l) ; cast_data |] (f ^ "_result")
    ↪  builder
| SCall (f, args) ->
    (* let (fdef, fdecl) = StringMap.find f function_decls in *)
    let fdef = StringMap.find f function_decls in
    let llargs = List.rev (List.map (expr builder) (List.rev args)) in
    L.build_call fdef (Array.of_list llargs) (f ^ "_result") builder
(* Id *)
      | SId s -> L.build_load (lookup s) s builder
(* Lit *)
      | SIntLit i   -> L.const_int i32_t i
| SBoolLit b  -> L.const_int i1_t (if b then 1 else 0)
| SFloatLit l -> L.const_float_of_string float_t l
| SCharLit c  -> L.const_int i8_t (Char.code c)
| SStrLit s   -> L.build_global_stringptr s "fmt" builder
| SNoexpr     -> L.const_int i32_t 0
    in
```

```
let add_terminal builder instr =
  match L.block_terminator (L.insertion_block builder) with
                            Some _ -> ()
  | None -> ignore (instr builder) in

let rec stmt builder = function
                            SBlock sl ->  List.fold_left stmt builder sl
  | SExpr e ->    ignore(expr builder e); builder
  | SVdecl b -> builder (* TODO *)
  | SReturn e ->  ignore (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

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

add_terminal builder (L.build_ret (L.const_int (ltype_of_typ fdecl.styp) 0))
    (* ... *)
```

```
        in
        List.iter build_function_body functions;
        the_module
```

src/gvl.ml (Yaxin Chen, Minhe Zhang)

```
(* Top-level of the MicroC 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: ./gvl.native [-a|-s|-l|-c] [file.gvl]" 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.tokenize 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)
```

src/lib/list.h (Yaxin Chen)

```
//
// Created by cyx on 2021/11/29.
//

#ifndef GRAPH_LIST_H
#define GRAPH_LIST_H

#include <stdlib.h>

typedef struct list_node_t {
    void *data;
    struct list_node_t *next;
```

```
    struct list_node_t *prev;
} list_node;

typedef list_node * list_iterator;

typedef struct list_t {
    list_node *first;
    list_node *last;
    // size_t data_size;
} list;

list *create_list();
int is_empty(list *l);
int insert_front(list *l, void *data);
int insert_back(list *l, void *data);
void *remove_front(list *l);
void *remove_back(list *l);
void *find(int (*cmp)(const void *, const void *), void *ref, list *l);
void *remove_list_node(int (*cmp)(const void *, const void *), void *ref, list
↪  *l);
void remove_all(list *l);
int destroy_list(list *l);
// list copy_all(list *l);

/****************************** iterator ******************************/
list_iterator list_begin(list *l);
list_iterator list_end();
list_iterator list_iter_next(list_iterator iter);
void *list_iter_data(list_iterator iter);

#endif //GRAPH_LIST_H
```

src/lib/list.c (Yaxin Chen)

```
//
// Created by cyx on 2021/11/29.
//

#include <memory.h>
#include "list.h"

list *create_list() {
    list *l = malloc(sizeof(list));
    l->first = NULL;
    l->last = NULL;
    // l->data_size = (size_t)data_size;
    return l;
}

int is_empty(list *l) {
    return (l->first == NULL);
}

int insert_front(list *l, void *data) {
```

```
    list_node *ln = malloc(sizeof(list_node));
    ln->data = data;
    ln->prev = NULL;
    ln->next = l->first;

    if (l->first != NULL) {
        l->first->prev = ln;
    }
    else {
        l->last = ln;
    }
    l->first = ln;

    return 0;
}

int insert_back(list *l, void *data) {
    list_node *ln = malloc(sizeof(list_node));
    ln->data = data;
    ln->next = NULL;
    ln->prev = l->last;

    if (l->last != NULL) {
        l->last->next = ln;
    }
    else {
        l->first = ln;
    }
    l->last = ln;

    return 0;
}

list_node *find_node(int (*cmp)(const void *, const void *), void *ref, list *l) {
    list_node *iter = l->first;
    while (iter != NULL) {
        if (cmp(iter->data, ref)) {
            return iter;
        }
        iter = iter->next;
    }
    return NULL;
}

void *find(int (*cmp)(const void *, const void *), void *ref, list *l) {
    list_node *n = find_node(cmp, ref, l);
    if (n == NULL) {
        return NULL;
    } else {
        return n->data;
    }
}
```

```
void *remove_list_node(int (*cmp)(const void *, const void *), void *ref, list *l)
↪  {
    list_node *n = find_node(cmp, ref, l);
    if (n == NULL) {
        return NULL;
    }

    if (n == l->first) {
        return remove_front(l);
    }
    else if (n == l->last) {
        return remove_back(l);
    }
    else {
        n->next->prev = n->prev;
        n->prev->next = n->next;
        void *data = n->data;
        free(n);
        return data;
    }
}

void *remove_front(list *l) {
    if (is_empty(l)) {
        return NULL;
    }

    list_node *victim = l->first;
    void *data = victim->data;

    l->first = victim->next;
    free(victim);

    if (l->first == NULL) {
        l->last = NULL;
    } else {
        l->first->prev = NULL;
    }

    return data;
}

void *remove_back(list *l) {
    if (is_empty(l)) {
        return NULL;
    }

    list_node *victim = l->last;
    void *data = victim->data;

    l->last = victim->prev;
    free(victim);

    if (l->last == NULL) {
```

```
        l->first = NULL;
    } else {
        l->last->next = NULL;
    }

    return data;
}

void remove_all(list *l) {
    while (!is_empty(l)) {
        void *data = remove_front(l);
        free(data);
    }
    free(l);
    l = NULL;
}

int destroy_list(list *l) {
    remove_all(l);
    return 0;
}

/*
list copy_all(list *l) {
    list new_list = {NULL, NULL, l->data_size};
    list_node *n = l->first;
    while (n) {
        void *data = malloc(l->data_size);
        memcpy(data, n->data, l->data_size);
        insert_back(&new_list, data);
        n = n->next;
    }
    return new_list;
}
*/

/****************************** iterator ******************************/
list_iterator list_begin(list *l) {
    return l->first;
}

list_iterator list_end() {
    return NULL;
}

list_iterator list_iter_next(list_iterator iter) {
    return iter->next;
}

void *list_iter_data(list_iterator iter) {
    return iter->data;
}
```

src/lib/graph.h (Yaxin Chen, Minhe Zhang)

```c
//
// Created by cyx on 2021/11/29.
//

#ifndef GRAPH_GRAPH_H
#define GRAPH_GRAPH_H

#include "list.h"

typedef struct node_t {
    double x;
    double y;
    double radius;
    int r;
    int g;
    int b;
    void *extra;
    // TODO in codegen, turn all added attributes (extra) into pointer_type of
    ↪  struct?
    // or use the change of color to indicate whether it is visited
} node;

typedef struct edge_t {
    node *start;
    node *end;
    int bold;
    int r;
    int g;
    int b;
} edge;

typedef struct graph_node_t {
    node *n;
    list *edges;  // list of edge (data is pointer to edge)
} graph_node;

typedef list graph;  // graph: list of graph_node
// the node pointer in edge points to the address of node in graph_node

/****************************** Node *******************************/

node *create_node(double x, double y, double radius, int r, int g, int b, void
↪  *data);

// Get node attributes.
double get_node_x(node *n);
double get_node_y(node *n);
double get_node_radius(node *n);
int get_node_r(node *n);
int get_node_g(node *n);
int get_node_b(node *n);
void *get_node_extra(node *n);
```

```
// Set node attributes.
int set_node_x(node *n, double x);
int set_node_y(node *n, double y);
int set_node_radius(node *n, double radius);
int set_node_r(node *n, int r);
int set_node_g(node *n, int g);
int set_node_b(node *n, int b);
int set_node_extra(node *n, void *extra);
int set_node_color(node *n, int r, int g, int b);

/****************************** Edge ******************************/

edge *create_edge(node *start, node *end, int bold, int r, int g, int b);
int edge_change_color(edge *e, int r, int g, int b);

// Get edge attributes.
node *get_edge_start(edge *e);
node *get_edge_end(edge *e);
int get_edge_bold(edge *e);
int get_edge_r(edge *e);
int get_edge_g(edge *e);
int get_edge_b(edge *e);
int destroy_node(node *n);

// Set edge attributes.
int set_edge_start(edge *e, node *start);
int set_edge_end(edge *e, node *end);
int set_edge_bold(edge *e, int bold);
int set_edge_r(edge *e, int r);
int set_edge_g(edge *e, int g);
int set_edge_b(edge *e, int b);
int destroy_edge(edge *e);

/****************************** Graph ******************************/

graph *create_graph();
int add_node(graph *g, node *n);
int remove_node(graph *g, node *n);
int add_edge(graph *g, edge *e);
int remove_edge(graph *g, edge *e);
int destroy_graph(graph *g);

list *get_edges(graph *g, node *n);

void print_graph(graph *g, int simple);

/****************************** Visualization ******************************/
int show_graph(graph *g);

#endif //GRAPH_GRAPH_H
```

src/lib/graph.c (Yaxin Chen, Minhe Zhang)

```c
//
// Created by cyx on 2021/11/29.
//

#include <memory.h>
#include <stdio.h>
#include "graph.h"

/****************************** utils ******************************/

int cmp_node(const void *n, const void *ref) {
    graph_node *gn = (graph_node *)n;
    graph_node *gref = (graph_node *)ref;
    // return (gn->n == gref->n);
    return (gn->n->x == gref->n->x && gn->n->y == gref->n->y && gn->n->radius ==
    ↪  gref->n->radius &&
            gn->n->r == gref->n->r && gn->n->g == gref->n->g && gn->n->b ==
            ↪  gref->n->b
            && gn->n->extra == gref->n->extra);
}

int cmp_edge(const void *e, const void *ref) {
    edge *ee = (edge *)e;
    edge *eref = (edge *)ref;
    return (ee->start == eref->start && ee->end == eref->end && ee->bold ==
    ↪  eref->bold &&
            ee->r == eref->r && ee->g == eref->g && ee->b == eref->b);
}

int cmp_edge_end_node(const void *e, const void *ref) {
    edge *ee = (edge *)e;
    edge *eref = (edge *)ref;
    return (ee->end == eref->end);
}

void print_graph(graph *g, int simple) {
    list_node *iter = g->first;
    while(iter) {
        graph_node *gn = (graph_node *)iter->data;
        list_node *edge_iter = gn->edges->first;

        if (simple) {
            printf("(%g, %g)\n", gn->n->x, gn->n->y);
        } else {
            printf("(x:%g, y%g, radius:%g, r:%d, g:%d, b:%d)\n",
                    gn->n->x, gn->n->y, gn->n->radius, gn->n->r, gn->n->g,
                    ↪  gn->n->b);
        }

        while (edge_iter) {
            edge *e = (edge *)edge_iter->data;
            printf("(%g, %g)", e->end->x, e->end->y);
            edge_iter = edge_iter->next;
        }
```

```
        printf("\n");
        iter = iter->next;
    }
}

/****************************** node ******************************/

node *create_node(double x, double y, double radius, int r, int g, int b, void
↪  *extra) {
    node *n = malloc(sizeof(node));
    n->x = x;
    n->y = y;
    n->radius = radius;
    n->r = r;
    n->g = g;
    n->b = b;
    n->extra = extra;
    return n;
}

int set_node_color(node *n, int r, int g, int b) {
    n->r = r;
    n->g = g;
    n->b = b;
    return 0;
}

double get_node_x(node *n) {
    return n->x;
}

double get_node_y(node *n) {
    return n->y;
}

double get_node_radius(node *n) {
    return n->radius;
}

int get_node_r(node *n) {
    return n->r;
}

int get_node_g(node *n) {
    return n->g;
}

int get_node_b(node *n) {
    return n->b;
}

void *get_node_extra(node *n) {
    return n->extra;
}
```

```c
int set_node_x(node *n, double x) {
    n->x = x;
    return 0;
}

int set_node_y(node *n, double y) {
    n->y = y;
    return 0;
}

int set_node_radius(node *n, double radius) {
    n->radius = radius;
    return 0;
}

int set_node_r(node *n, int r) {
    n->r = r;
    return 0;
}

int set_node_g(node *n, int g) {
    n->g = g;
    return 0;
}

int set_node_b(node *n, int b) {
    n->b = b;
    return 0;
}

int set_node_extra(node *n, void *extra) {
    n->extra = extra;
    return 0;
}

int destroy_node(node *n) {
    free(n);
    return 0;
}

/****************************** edge ******************************/

edge *create_edge(node *start, node *end, int bold, int r, int g, int b) {
    edge *e = malloc(sizeof(edge));
    e->start = start;
    e->end = end;
    e->bold = bold;
    e->r = r;
    e->g = g;
    e->b = b;
    return e;
}
```

```
int edge_change_color(edge *e, int r, int g, int b) {
    e->r = r;
    e->g = g;
    e->b = b;
    return 0;
}

// Get edge attributes.
node *get_edge_start(edge *e) {
    return e->start;
}

node *get_edge_end(edge *e) {
    return e->end;
}

int get_edge_bold(edge *e) {
    return e->bold;
}

int get_edge_r(edge *e) {
    return e->r;
}

int get_edge_g(edge *e) {
    return e->g;
}

int get_edge_b(edge *e) {
    return e->b;
}

// Set edge attributes.
int set_edge_start(edge *e, node *start) {
    e->start = start;
    return 0;
}

int set_edge_end(edge *e, node *end) {
    e->end = end;
    return 0;
}

int set_edge_bold(edge *e, int bold) {
    e->bold = bold;
    return 0;
}

int set_edge_r(edge *e, int r) {
    e->r = r;
    return 0;
}

int set_edge_g(edge *e, int g) {
```

```
    e->g = g;
    return 0;
}

int set_edge_b(edge *e, int b) {
    e->b = b;
    return 0;
}

int destroy_edge(edge *e) {
    free(e);
    return 0;
}

/****************************** graph ******************************/

graph *create_graph() {
    graph *g;
    g = create_list();
    return g;
}

int add_node(graph *g, node *n) {
    graph_node *gn = malloc(sizeof(graph_node));
    gn->n = n;
    gn->edges = create_list();
    insert_front(g, gn);
    return 0;
}

int remove_node(graph *g, node *n) {
    graph_node gn_ref;
    gn_ref.n = n;

    // remove all edges containing this node
    list_node *iter = g->first;
    while(iter) {
        graph_node *gn = (graph_node *)iter->data;

        edge e_ref;
        e_ref.end = n;
        edge *removed_e = remove_list_node(cmp_edge_end_node, &e_ref, gn->edges);
        free(removed_e);

        iter = iter->next;
    }

    // remove node from graph
    graph_node *data = (graph_node *)remove_list_node(cmp_node, &gn_ref, g);
    free(data->n->extra); // free extra
    free(data->n);
    remove_all(data->edges);
    free(data);
```

```
    return 0;
}

int add_edge(graph *g, edge *e) {
    graph_node start_ref;
    start_ref.n = e->start;
    graph_node *start_gn = find(cmp_node, &start_ref, g);
    if (start_gn == NULL) {
        return 1;
    }
    insert_back(start_gn->edges, e);
    return 0;
}

int remove_edge(graph *g, edge *e) {
    graph_node start_ref;
    start_ref.n = e->start;
    graph_node *start_gn = find(cmp_node, &start_ref, g);
    edge *removed_e = remove_list_node(cmp_edge, e, start_gn->edges);
    free(removed_e);
    return 0;
}

int destroy_graph(graph *g) {
    while (!is_empty(g)) {
        graph_node *gn = (graph_node *)remove_front(g);
        remove_all(gn->edges);
        free(gn->n->extra); // free extra
        free(gn->n);
        free(gn);
    }
    free(g);
    return 0;
}

list *get_edges(graph *g, node *n) {
    graph_node ref;
    ref.n = n;
    graph_node *gn = find(cmp_node, &ref, g);

    return gn->edges;
}
```

src/lib/graph$_v$*isualization.c* (Yaxin Chen)

```
//
// Created by cyx on 2021/12/18.
//

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <glad/glad.h>
#include <GLFW/glfw3.h>
```

```c
#include "graph.h"

#define ARROWLENGTH 10

static void key_callback(GLFWwindow* window, int key, int scancode, int action,
↪  int mods)
{
    if ((key == GLFW_KEY_ESCAPE || key == GLFW_KEY_ENTER) && action == GLFW_PRESS)
        glfwSetWindowShouldClose(window, GLFW_TRUE);
}

GLFWwindow *create_window() {
    const char* description;
    if (!glfwInit())
    {
        glfwGetError(&description);
        printf("Error: %s\n", description);
        exit(EXIT_FAILURE);
    }

    glfwWindowHint(GLFW_VISIBLE, GLFW_FALSE);
    glfwWindowHint(GLFW_DECORATED, GLFW_FALSE);

    int xpos, ypos, height;
    glfwGetMonitorWorkarea(glfwGetPrimaryMonitor(), &xpos, &ypos, NULL, &height);

    const int size = height / 2;
    GLFWwindow *window = glfwCreateWindow(size, size, "GVL", NULL, NULL);
    if (!window)
    {
        glfwGetError(&description);
        printf("Error: %s\n", description);
        glfwTerminate();
        exit(EXIT_FAILURE);
    }

    glfwSetKeyCallback(window, key_callback);

    glfwSetWindowPos(window, xpos, ypos);
    glfwSetInputMode(window, GLFW_STICKY_KEYS, GLFW_TRUE);

    // set background color
    glfwMakeContextCurrent(window);
    gladLoadGL();
    glClearColor(1.f, 1.f, 1.f, 1.f);

    // set up view
    glViewport(0, 0, size, size);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // creates a canvas for drawing
    glOrtho(0, size, 0, size, 0.0, 1.0);
```

```
    glfwShowWindow(window);

    return window;
}

void draw_line(double v1_x, double v1_y, double v2_x, double v2_y, float
↪  line_width, int r, int g, int b) {
    glLineWidth(line_width);
    glColor3f((float)r / 255, (float)g / 255, (float)b / 255);   // TODO
    glBegin(GL_LINES);
    glVertex3f(v1_x, v1_y, 0.0);
    glVertex3f(v2_x, v2_y, 0.0);
    glEnd();
}

void draw_arrow(double v1_x, double v1_y, double v2_x, double v2_y, int
↪  line_width, int r, int g, int b) {
    draw_line(v1_x, v1_y, v2_x, v2_y, (float)line_width, r, g, b);

    double length = sqrt((v2_x - v1_x) * (v2_x - v1_x) + (v2_y - v1_y) * (v2_y -
    ↪  v1_y));
    draw_line(v2_x + ((v1_x - (v2_y - v1_y)) - v2_x) * ARROWLENGTH / length,
              v2_y + ((v1_y + (v2_x - v1_x)) - v2_y) * ARROWLENGTH / length,
              v2_x, v2_y, (float)line_width / 2, r, g, b);
    draw_line(v2_x + ((v1_x + (v2_y - v1_y)) - v2_x) * ARROWLENGTH / length,
              v2_y + ((v1_y - (v2_x - v1_x)) - v2_y) * ARROWLENGTH / length,
              v2_x, v2_y, (float)line_width / 2, r, g, b);
}

void draw_circle(double c_x, double c_y, double radius, int r, int g, int b) {
    glColor3f((float)r / 255, (float)g / 255, (float)b / 255);   // TODO
    int num_segments = (int)radius * 360;
    float theta = M_PI * 2 / (float)num_segments;

    glBegin(GL_TRIANGLE_FAN);
    float angle = 0;
    for (int i = 0; i < num_segments; i++)
    {
        double x = radius * cosf(angle);
        double y = radius * sinf(angle);
        glVertex2f(c_x + x, c_y + y);
        angle += theta;
    }
    glEnd();

}

void draw_graph(graph *g) {
    list_node *iter = g->first;
    while(iter) {
        graph_node *gn = (graph_node *)iter->data;
        list_node *edge_iter = gn->edges->first;
```

```
        draw_circle(gn->n->x, gn->n->y, gn->n->radius, gn->n->r, gn->n->g,
        ↪  gn->n->b);

        while (edge_iter) {
            edge *e = (edge *)edge_iter->data;
            double length = sqrt((e->end->y - gn->n->y) * (e->end->y - gn->n->y) +
                    (e->end->x - gn->n->x) * (e->end->x - gn->n->x));
            draw_arrow(gn->n->x + (e->end->x - gn->n->x) * gn->n->radius / length,
                    gn->n->y + (e->end->y - gn->n->y) * gn->n->radius / length,
                    e->end->x + (gn->n->x - e->end->x) * gn->n->radius / length,
                    e->end->y + (gn->n->y - e->end->y) * gn->n->radius / length,
                    e->bold, e->r, e->g, e->b);

            edge_iter = edge_iter->next;
        }

        iter = iter->next;
    }
}

int show_graph(graph *g) {
    GLFWwindow *window = create_window();
    while (!glfwWindowShouldClose(window)) {
        glfwMakeContextCurrent(window);
        glClear(GL_COLOR_BUFFER_BIT);

        draw_graph(g);

        glfwSwapBuffers(window);
        glfwWaitEvents();
    }
    glfwDestroyWindow(window);
    glfwTerminate();

    return 0;
}
```

8.2 Project Log

```
commit cb6e164114f9298bfc3deb949b153a15e1ac3cb2 (HEAD -> main, origin/main,
↪  origin/HEAD)
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Wed Dec 22 14:32:40 2021 -0500

    Syntactic sugar (#38)

    * add syntactic sugar for add node and edge

    * add demo

 src/demo/bfs.gvl                                | 90
 ↪  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/demo/dfs.gvl                                | 78
 ↪  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
 src/parser.mly                              |  6 +++--
 src/scanner.mll                             |  2 ++
 src/tests/syntax_sugar/test-syn-graph1.gvl | 39 ++++++++++++++++++++++++++++++++++++++
 src/tests/syntax_sugar/test-syn-graph1.out |  2 ++
 6 files changed, 215 insertions(+), 2 deletions(-)

commit 768386bb28a5454da433866e325b81521c8c350c
Author: AsterWei <53880314+AsterWei@users.noreply.github.com>
Date:   Wed Dec 22 14:11:38 2021 -0500

    Add test cases for variables (#37)

    * add variable tests

    * remove log

    * remove unnecessary test cases

 src/tests/variables/fail-and1.err    |  1 +
 src/tests/variables/fail-and1.gvl    |  7 +++++++
 src/tests/variables/fail-and2.err    |  1 +
 src/tests/variables/fail-and2.gvl    |  7 +++++++
 src/tests/variables/fail-assign4.err |  1 +
 src/tests/variables/fail-assign4.gvl |  6 ++++++
 src/tests/variables/fail-div1.err    |  1 +
 src/tests/variables/fail-div1.gvl    |  9 +++++++++
 src/tests/variables/fail-div2.err    |  1 +
 src/tests/variables/fail-div2.gvl    |  9 +++++++++
 src/tests/variables/fail-leq1.err    |  1 +
 src/tests/variables/fail-leq1.gvl    |  5 +++++
 src/tests/variables/fail-minus1.err  |  1 +
 src/tests/variables/fail-minus1.gvl  |  9 +++++++++
 src/tests/variables/fail-minus2.err  |  1 +
 src/tests/variables/fail-minus2.gvl  |  8 ++++++++
 src/tests/variables/fail-minus3.err  |  1 +
 src/tests/variables/fail-minus3.gvl  |  9 +++++++++
 src/tests/variables/fail-mul1.err    |  1 +
 src/tests/variables/fail-mul1.gvl    |  8 ++++++++
 src/tests/variables/fail-not1.err    |  1 +
 src/tests/variables/fail-not1.gvl    |  5 +++++
 src/tests/variables/fail-not2.err    |  1 +
 src/tests/variables/fail-not2.gvl    |  6 ++++++
 src/tests/variables/fail-plus3.err   |  1 +
 src/tests/variables/fail-plus3.gvl   |  8 ++++++++
 src/tests/variables/test-assign1.gvl |  6 ++++++
 src/tests/variables/test-assign1.out |  1 +
 src/tests/variables/test-assign2.gvl |  6 ++++++
 src/tests/variables/test-assign2.out |  1 +
 src/tests/variables/test-assign3.gvl |  2 --
 src/tests/variables/test-div1.gvl    | 10 ++++++++++
 src/tests/variables/test-div1.out    |  1 +
 src/tests/variables/test-div2.gvl    | 10 ++++++++++
 src/tests/variables/test-div2.out    |  1 +
 src/tests/variables/test-leq1.gvl    |  6 ++++++
```

```
 src/tests/variables/test-leq1.out    |  1 +
 src/tests/variables/test-minus1.gvl  | 10 ++++++++++
 src/tests/variables/test-minus1.out  |  1 +
 src/tests/variables/test-minus2.gvl  |  9 +++++++++
 src/tests/variables/test-minus2.out  |  1 +
 src/tests/variables/test-not1.gvl    |  7 +++++++
 src/tests/variables/test-not1.out    |  0
 src/tests/variables/test-or1.gvl     |  9 +++++++++
 src/tests/variables/test-or1.out     |  1 +
 src/tests/variables/test-plus1.gvl   | 10 ++++++++++
 src/tests/variables/test-plus1.out   |  1 +
 src/tests/variables/test-plus2.gvl   |  9 +++++++++
 src/tests/variables/test-plus2.out   |  1 +
 49 files changed, 211 insertions(+), 2 deletions(-)

commit 9d006d904c412b03b9e3057ac217fcda56cbf4db (origin/aster, aster)
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 13:59:10 2021 -0500

    remove unnecessary test cases

 src/tests/variables/fail-assign3.err | 1 -
 src/tests/variables/fail-assign3.gvl | 6 ------
 src/tests/variables/fail-mod1.err    | 1 -
 src/tests/variables/fail-mod1.gvl    | 9 ---------
 4 files changed, 17 deletions(-)

commit ff23788a9f50b249e99ace461ea34c836ad0a0fa
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 13:15:48 2021 -0500

    remove log

 src/tests/variables/testall.log | 1 -
 1 file changed, 1 deletion(-)

commit 95816ce9cfea0d53d05fe39e42a37f7194c5967a
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 13:14:29 2021 -0500

    add variable tests

 src/tests/variables/fail-and1.err    |  1 +
 src/tests/variables/fail-and1.gvl    |  7 +++++++
 src/tests/variables/fail-and2.err    |  1 +
 src/tests/variables/fail-and2.gvl    |  7 +++++++
 src/tests/variables/fail-assign3.err |  1 +
 src/tests/variables/fail-assign3.gvl |  6 ++++++
 src/tests/variables/fail-assign4.err |  1 +
 src/tests/variables/fail-assign4.gvl |  6 ++++++
 src/tests/variables/fail-div1.err    |  1 +
 src/tests/variables/fail-div1.gvl    |  9 +++++++++
 src/tests/variables/fail-div2.err    |  1 +
 src/tests/variables/fail-div2.gvl    |  9 +++++++++
```

```
 src/tests/variables/fail-leq1.err    |  1 +
 src/tests/variables/fail-leq1.gvl    |  5 +++++
 src/tests/variables/fail-minus1.err  |  1 +
 src/tests/variables/fail-minus1.gvl  |  9 +++++++++
 src/tests/variables/fail-minus2.err  |  1 +
 src/tests/variables/fail-minus2.gvl  |  8 ++++++++
 src/tests/variables/fail-minus3.err  |  1 +
 src/tests/variables/fail-minus3.gvl  |  9 +++++++++
 src/tests/variables/fail-mod1.err    |  1 +
 src/tests/variables/fail-mod1.gvl    |  9 +++++++++
 src/tests/variables/fail-mul1.err    |  1 +
 src/tests/variables/fail-mul1.gvl    |  8 ++++++++
 src/tests/variables/fail-not1.err    |  1 +
 src/tests/variables/fail-not1.gvl    |  5 +++++
 src/tests/variables/fail-not2.err    |  1 +
 src/tests/variables/fail-not2.gvl    |  6 ++++++
 src/tests/variables/fail-plus3.err   |  1 +
 src/tests/variables/fail-plus3.gvl   |  8 ++++++++
 src/tests/variables/test-assign1.gvl |  6 ++++++
 src/tests/variables/test-assign1.out |  1 +
 src/tests/variables/test-assign2.gvl |  6 ++++++
 src/tests/variables/test-assign2.out |  1 +
 src/tests/variables/test-assign3.gvl |  2 --
 src/tests/variables/test-div1.gvl    | 10 ++++++++++
 src/tests/variables/test-div1.out    |  1 +
 src/tests/variables/test-div2.gvl    | 10 ++++++++++
 src/tests/variables/test-div2.out    |  1 +
 src/tests/variables/test-leq1.gvl    |  6 ++++++
 src/tests/variables/test-leq1.out    |  1 +
 src/tests/variables/test-minus1.gvl  | 10 ++++++++++
 src/tests/variables/test-minus1.out  |  1 +
 src/tests/variables/test-minus2.gvl  |  9 +++++++++
 src/tests/variables/test-minus2.out  |  1 +
 src/tests/variables/test-not1.gvl    |  7 +++++++
 src/tests/variables/test-not1.out    |  0
 src/tests/variables/test-or1.gvl     |  9 +++++++++
 src/tests/variables/test-or1.out     |  1 +
 src/tests/variables/test-plus1.gvl   | 10 ++++++++++
 src/tests/variables/test-plus1.out   |  1 +
 src/tests/variables/test-plus2.gvl   |  9 +++++++++
 src/tests/variables/test-plus2.out   |  1 +
 src/tests/variables/testall.log      |  1 +
 54 files changed, 229 insertions(+), 2 deletions(-)

commit eaf29896d2fbf4286dafaae277a0c776e668d0fb
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Wed Dec 22 09:05:23 2021 -0800

    Added statement tests. (#36)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md
```

```
    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.

    * Automatical testing, credits to Prof. Stephen Edwards.

    * Added subdir in tests/ and some basic array.

    * Linked c files.

    * Added docker commands.

    * Added get and set functions for edge and node.

    * Statement and printstr tests.

    * Resolved redefined functions.

    * Added built-in and tests. Still have bug on get_edges.

    * Changed list_t type to non-pointer.

    * Finished iteration 2.

    * Changed finished iteration.

    * Finished README.md

    * Added some statement tests.

    * Added built-in and tests. Still have bug on get_edges.

    * Changed list_t type to non-pointer.

    * Finished iteration 2.

    * Changed finished iteration.

    * Finished README.md

    * Added some statement tests.

    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-116.dyn.columbia.edu>

src/tests/functions/test-func4.gvl    | 12 ++++++++++
src/tests/functions/test-func4.out    |  6 ++++++
src/tests/statements/fail-for1.err    |  1 +
```

```
 src/tests/statements/fail-for1.gvl    |  6 ++++++
 src/tests/statements/fail-for2.err    |  1 +
 src/tests/statements/fail-for2.gvl    |  5 +++++
 src/tests/statements/fail-for3.err    |  1 +
 src/tests/statements/fail-for3.gvl    |  7 +++++++
 src/tests/statements/fail-for4.err    |  1 +
 src/tests/statements/fail-for4.gvl    |  7 +++++++
 src/tests/statements/fail-if2.err     |  2 +-
 src/tests/statements/fail-if2.gvl     | 10 ++++++++++
 src/tests/statements/fail-if3.err     |  1 +
 src/tests/statements/fail-if3.gvl     | 10 ++++++++++
 src/tests/statements/fail-while1.err |  1 +
 src/tests/statements/fail-while1.gvl |  5 +++++
 src/tests/statements/fail-while2.err |  1 +
 src/tests/statements/fail-while2.gvl |  7 +++++++
 src/tests/statements/fail-while3.err |  1 +
 src/tests/statements/fail-while3.gvl |  5 +++++
 20 files changed, 89 insertions(+), 1 deletion(-)

commit cb1efd117feedd05ab43d9b06721ccefbb82f4bf
Merge: 74520d4 8a75ede
Author: AsterWei <53880314+AsterWei@users.noreply.github.com>
Date:   Wed Dec 22 11:50:19 2021 -0500

    Merge pull request #35 from MinheZhang/aster

    fixed pr for visualization, function, statement tests

commit 8a75ede370c5e1d3c8f3f5aaea08a3705d973592
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 11:22:21 2021 -0500

    fixed fail-var1

 src/tests/variables/fail-var1.gvl | 1 -
 1 file changed, 1 deletion(-)

commit 7dc6c370da5fd448caf7d17936f29210cf32bd20
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 11:09:59 2021 -0500

    fixed fail-func4

 src/tests/functions/fail-func4.gvl | 4 ----
 1 file changed, 4 deletions(-)

commit 27365414e4116fb0568a638677576b4148413c6f
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 10:55:49 2021 -0500

    fixed prev test names

 src/tests/{expressions/fail-expr3.err => variables/fail-and3.err}  |  0
 src/tests/{expressions/fail-expr3.gvl => variables/fail-and3.gvl}  |  0
```

```
 src/tests/{assignments => variables}/fail-assign1.err              |  0
 src/tests/{assignments => variables}/fail-assign1.gvl              |  0
 src/tests/{assignments => variables}/fail-assign2.err              |  0
 src/tests/{assignments => variables}/fail-assign2.gvl              |  0
 src/tests/{expressions/fail-expr1.err => variables/fail-plus1.err} |  0
 src/tests/{expressions/fail-expr1.gvl => variables/fail-plus1.gvl} |  0
 src/tests/{expressions/fail-expr2.err => variables/fail-plus2.err} |  0
 src/tests/{expressions/fail-expr2.gvl => variables/fail-plus2.gvl} |  9 ++----
 src/tests/{assignments => variables}/test-assign3.gvl              |  0
 src/tests/{assignments => variables}/test-assign3.out              |  0
 src/visual_tests/test-dfs2.gvl                                     | 68
 ↪  ------------------------------------------------
 src/visual_tests/test-dfs2.out                                     |  0
 src/visual_tests/test-graph1.gvl                                   | 33
 ↪  ----------------------
 src/visual_tests/test-graph1.out                                   |  0
 src/visual_tests/test-graph2.gvl                                   | 24
 ↪  ----------------
 src/visual_tests/test-graph2.out                                   |  0
 src/visual_tests/test-graph3.gvl                                   | 30
 ↪  --------------------
 src/visual_tests/test-graph3.out                                   |  0
 20 files changed, 2 insertions(+), 162 deletions(-)

commit 29fa1c2e32e875ba78829331763ef9fbd3834d76
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 03:02:53 2021 -0500

    delete new line at eof

 src/tests/assignments/fail-assign1.gvl                  | 2 +-
 src/tests/assignments/fail-assign2.gvl                  | 2 +-
 src/tests/expressions/fail-expr2.gvl                    | 2 +-
 src/tests/functions/{fail-builtin.err => fail-func6.err} | 0
 src/tests/functions/{fail-builtin.gvl => fail-func6.gvl} | 2 +-
 src/tests/others/fail-comment2.gvl                      | 2 +-
 src/tests/variables/fail-var1.gvl                       | 2 +-
 7 files changed, 6 insertions(+), 6 deletions(-)

commit c565081f51f5841ef7a548b817f8a2a1b823c497
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:57:26 2021 -0500

    correct testall.sh

 src/testall.sh | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit b6aa0d280c7f55d7f437bd1de4bdb432c9ce3c17
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:56:00 2021 -0500

    pull from latest main
```

```
 src/builtin.ml                 | 10 ++++++++++
 src/codegen.ml                 | 12 +++++-------
 src/lib/graph.c                | 30 ++++++++++++++++++++----------
 src/lib/graph.h                |  4 +++-
 src/lib/graph_visualization.c |  2 +-
 src/lib/list.c                 | 24 ++++++++++++++++++++----
 src/lib/list.h                 | 11 ++++++-----
 src/lib/main.c                 |  8 ++++----
 8 files changed, 69 insertions(+), 32 deletions(-)

commit a758ca44ea97e598d8ec92ca7152baa92fe70443
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:51:17 2021 -0500

    pull file from main

 src/ast.ml                     | 119 +++++++++++++++++++++++++++++++++
 src/builtin.ml                 |  91 +++++++++++++++++++++++++
 src/codegen.ml                 | 298
 ↪   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/lib/graph.c                | 297
 ↪   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/lib/graph.h                |  99 +++++++++++++++++++++++++++
 src/lib/graph_visualization.c | 148 +++++++++++++++++++++++++++++++++++++++++
 src/lib/list.c                 | 162 +++++++++++++++++++++++++++++++++++++++++++++
 src/lib/list.h                 |  41 ++++++++++++
 src/lib/main.c                 |  92 ++++++++++++++++++++++++++
 9 files changed, 1347 insertions(+)

commit 5ecc8eac1d97284b40d6e4f6c68adf6401a6e00d
Merge: 1d8644a 74520d4
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:47:22 2021 -0500

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL into aster

commit 1d8644a8b15414178ec83f7260dc42b8532b1df2
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:46:05 2021 -0500

    delete file from remote

 src/ast.ml                     | 119 ---------------------------------
 src/builtin.ml                 |  91 -------------------------
 src/codegen.ml                 | 298
 ↪   ----------------------------------------------------------------------------------
 src/lib/graph.c                | 297
 ↪   ---------------------------------------------------------------------------------
 src/lib/graph.h                |  99 ---------------------------
 src/lib/graph_visualization.c | 148 -----------------------------------------
 src/lib/list.c                 | 162 ---------------------------------------------
 src/lib/list.h                 |  41 ------------
 src/lib/main.c                 |  92 --------------------------
 9 files changed, 1347 deletions(-)
```

```
commit 93a210679949ebbb8ab5acb799c00bc49633ca1d
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:42:18 2021 -0500

    remove local file from branch

 src/graph       | Bin 12048 -> 0 bytes
 src/list        | Bin 5780 -> 0 bytes
 src/test-bfs.ll |   0
 3 files changed, 0 insertions(+), 0 deletions(-)

commit bdfd0758d1cedb9228bb7229a29eb6d295df132a
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:33:34 2021 -0500

    fixed previous file on branch aster

 src/ast.ml                    |  7 +++----
 src/builtin.ml                | 10 ----------
 src/codegen.ml                | 18 ++++++++++--------
 src/lib/graph.c               | 30 ++++++++++++++++++++----------
 src/lib/graph.h               |  4 +---
 src/lib/graph_visualization.c |  4 ++--
 src/lib/list.c                | 24 ++++++++++++++++++++----
 src/lib/list.h                | 11 +++++++++++
 src/lib/main.c                |  8 ++++----
 9 files changed, 39 insertions(+), 77 deletions(-)

commit 91382ef1a0c7ec0f50eceeb347456a10c42f560c
Author: AsterWei <aw3389@columbia.edu>
Date:   Wed Dec 22 02:22:48 2021 -0500

    add checked visualization function etc. tests

 src/tests/functions/{fail-undefined.err => fail-func4.err} |   0
 src/tests/functions/{fail-undefined.gvl => fail-func4.gvl} |   0
 src/tests/functions/{fail-dead1.err => fail-func5.err}     |   0
 src/tests/functions/{fail-dead1.gvl => fail-func5.gvl}     |   0
 src/tests/others/fail-comment2.err                         |   1 +
 src/tests/others/fail-comment2.gvl                         |   9 +++++
 src/visual_tests/test-bfs3.gvl                             | 103
 ↪  ---------------------------------------------------
 src/visual_tests/test-dfs2.gvl                             |  68
 ↪  ++++++++++++++++++++++++++++++++++
 src/visual_tests/{test-bfs3.out => test-dfs2.out}          |   0
 src/visual_tests/{test-color.gvl => test-graph1.gvl}       |   0
 src/visual_tests/{test-color.out => test-graph1.out}       |   0
 src/visual_tests/{test-twowayedge.gvl => test-graph2.gvl}  |   0
 src/visual_tests/{test-twoway.out => test-graph2.out}      |   0
 src/visual_tests/test-graph3.gvl                           |  30 ++++++++++++++++++
 src/visual_tests/test-graph3.out                           |   0
 15 files changed, 108 insertions(+), 103 deletions(-)
```

```
commit 74520d41a1ae56c5105abe40e6a1c66caaff47e9
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Wed Dec 22 02:20:49 2021 -0500

    add tests for list (#34)

 src/tests/list/test-list3.gvl | 16 ++++++++++++++++
 src/tests/list/test-list3.out |  0
 src/tests/list/test-list4.gvl | 36 ++++++++++++++++++++++++++++++++++++
 src/tests/list/test-list4.out |  4 ++++
 src/tests/list/test-list5.gvl | 32 ++++++++++++++++++++++++++++++++
 src/tests/list/test-list5.out |  5 +++++
 src/tests/list/test-list6.gvl | 37 +++++++++++++++++++++++++++++++++++++
 src/tests/list/test-list6.out |  6 ++++++
 src/tests/list/test-list7.gvl | 33 +++++++++++++++++++++++++++++++++
 src/tests/list/test-list7.out |  7 +++++++
 10 files changed, 176 insertions(+)

commit d2c0329ba4286f81ab6a23ef655f0c6f8c99b0a0
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Tue Dec 21 20:35:26 2021 -0800

    Finished README.md file and changed incorrect comment from microc (#33)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.

    * Automatical testing, credits to Prof. Stephen Edwards.

    * Added subdir in tests/ and some basic array.

    * Linked c files.

    * Added docker commands.

    * Added get and set functions for edge and node.

    * Statement and printstr tests.

    * Resolved redefined functions.

    * Added built-in and tests. Still have bug on get_edges.

    * Changed list_t type to non-pointer.
```

```
    * Finished iteration 2.

    * Changed finished iteration.

    * Finished README.md

    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-116.dyn.columbia.edu>

 README.md    | 94
 ↪  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-----
 src/Makefile |  2 +-
 2 files changed, 89 insertions(+), 7 deletions(-)

commit f65151e4ea6ce6b065be8fed8f21021bddae42d6
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Tue Dec 21 23:30:16 2021 -0500

    fix run.sh to print (#32)

 src/run.sh | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)

commit 0dab533bd719fa31987dcc4354e033a568315169
Merge: f5b1b33 0a7541f
Author: AsterWei <aw3389@columbia.edu>
Date:   Tue Dec 21 23:19:14 2021 -0500

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL into aster

commit f5b1b33692774a83f59f4f974409ac05780da0a5
Merge: f2fd402 e4d85ae
Author: AsterWei <aw3389@columbia.edu>
Date:   Tue Dec 21 23:18:55 2021 -0500

    Merge branch 'aster' of https://github.com/MinheZhang/COMSW4115-GVL into aster

commit f2fd40216ac0077b4baff34e1dac6fe25c5a96ba
Author: AsterWei <aw3389@columbia.edu>
Date:   Tue Dec 21 23:18:35 2021 -0500

    sample test for report

 src/lib/graph_visualization.c                | 2 +-
 src/tests/assignments/test-assign3.gvl       | 7 +++++++
 src/tests/assignments/test-assign3.out       | 0
 src/tests/{dead => functions}/fail-dead1.err | 0
 src/tests/{dead => functions}/fail-dead1.gvl | 0
 src/tests/functions/test-func3.gvl           | 2 +-
 6 files changed, 9 insertions(+), 2 deletions(-)

commit e4d85ae03e63c87e2b33e6f32c1db8738f13eb1c
```

```
Author: AsterWei <53880314+AsterWei@users.noreply.github.com>
Date:   Tue Dec 21 15:56:42 2021 -0500

    Update graph_visualization.c

 src/lib/graph_visualization.c | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)

commit 0a7541f4308cd9a805918b06f8b7cc63e3b2c70c
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Mon Dec 20 23:44:46 2021 -0500

    add script to compile & run; credits to Prof. Stephen Edwards (#31)

 src/run.sh | 53 +++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 53 insertions(+)

commit 7f4045e34381940877c919b4734a753e96b4b3fb
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Mon Dec 20 18:21:59 2021 -0500

    add bfs demo (#30)

    * adapt lib list to list*

    * adapt list

    * add list to gvl; fix bug of cast void ptr in codegen

    * bfs demo

 src/visual_tests/test-bfs.gvl  | 68 ++++++++++++++++++++++++++++++++++++++++
 src/visual_tests/test-bfs.out  |  0
 src/visual_tests/test-bfs2.gvl | 84 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/visual_tests/test-bfs2.out |  0
 4 files changed, 152 insertions(+)

commit 1b0291d57dfbb0fce3b7b7134af78efc83dbf405
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Sat Dec 18 20:42:39 2021 -0500

    add list to gvl (#27)

    * adapt lib list to list*

    * adapt list

    * add list to gvl; fix bug of cast void ptr in codegen

 src/builtin.ml                | 10 ++++++++++
 src/codegen.ml                | 12 +++++++++++-
 src/lib/graph.c               | 30 ++++++++++++++++++++++++++++++----------
 src/lib/graph.h               |  4 +++-
 src/lib/graph_visualization.c |  2 +-
```

```
 src/lib/list.c                | 24 ++++++++++++++++++++----
 src/lib/list.h                | 11 ++++++-----
 src/lib/main.c                |  8 ++++----
 src/semant.ml                 |  2 +-
 src/tests/list/test-list1.gvl | 10 ++++++++++
 src/tests/list/test-list1.out |  0
 src/tests/list/test-list2.gvl | 19 +++++++++++++++++++
 src/tests/list/test-list2.out |  1 +
 13 files changed, 100 insertions(+), 33 deletions(-)

commit a45a04619fdd12d4a5e5e388983f782b7dec5a85
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Fri Dec 17 21:47:30 2021 -0500

    graph visualization (#26)

    * add graph visualization

    * export show_graph to gvl

 src/Makefile                    |   2 +-
 src/builtin.ml                  |   2 +-
 src/lib/graph.h                 |   3 +
 src/lib/graph_visualization.c   | 148 +++++++++++++++++++++++++++++++++++++++++++
 src/lib/main.c                  |  20 +++---
 src/testall.sh                  |   2 +-
 src/visual_tests/test-dfs.gvl   |  52 +++++++++++++++
 src/visual_tests/test-dfs.out   |   0
 8 files changed, 217 insertions(+), 12 deletions(-)

commit 3e2f36a84cde4bf0e0e0c1ad73d8fd0ab723833e
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Tue Dec 14 22:50:28 2021 -0500

    add get_edges and list_iterator (#25)

    * change return type of get_edges

    * add list iterator

    * add list iterator to codegen; dfs

    * remove useless comment

 src/ast.ml                      |  3 ++-
 src/builtin.ml                  |  7 +++++-
 src/codegen.ml                  | 40 +++++++++++++++++++++-----------
 src/lib/graph.c                 |  4 ++--
 src/lib/graph.h                 |  2 +-
 src/lib/list.c                  |  4 ++--
 src/lib/list.h                  |  2 +-
 src/lib/main.c                  | 10 ++++----
 src/parser.mly                  |  3 ++-
 src/scanner.mll                 |  1 +
```

```
 src/semant.ml                    |  7 ++++--
 src/tests/graph/test-dfs.gvl     | 51 ++++++++++++++++++++++++++++++++++++++++
 src/tests/graph/test-dfs.out     |  3 +++
 src/tests/graph/test-graph4.gvl | 35 +++++++++++++++++++++++++++
 src/tests/graph/test-graph4.out |  2 ++
 src/tests/graph/test-graph5.gvl | 37 ++++++++++++++++++++++++++++
 src/tests/graph/test-graph5.out |  0
 src/tests/graph/test-graph6.gvl | 39 ++++++++++++++++++++++++++++++
 src/tests/graph/test-graph6.out |  2 ++
 src/tests/graph/test-graph7.gvl | 41 ++++++++++++++++++++++++++++++++
 src/tests/graph/test-graph7.out |  2 ++
 21 files changed, 266 insertions(+), 29 deletions(-)

commit 0f3a0ed5b650441724843c14098b62b774bc8f11
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sun Dec 12 18:58:55 2021 -0800

    Added graph built-in and tests. Still have bug on get_edges. (#24)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.

    * Automatical testing, credits to Prof. Stephen Edwards.

    * Added subdir in tests/ and some basic array.

    * Linked c files.

    * Added docker commands.

    * Added get and set functions for edge and node.

    * Statement and printstr tests.

    * Resolved redefined functions.

    * Added built-in and tests. Still have bug on get_edges.

    * Changed list_t type to non-pointer.

    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-116.dyn.columbia.edu>
```

```
 README.md                                  |  4 +--
 src/Makefile                               |  2 ++
 src/ast.ml                                 |  3 +-
 src/builtin.ml                             | 55 ++++++++++++++++++++++++++----
 src/codegen.ml                             | 16 ++++++++++
 src/lib/graph.c                            |  2 +-
 src/parser.mly                             |  3 +-
 src/scanner.mll                            |  1 +
 src/tests/functions/test-printstr1.gvl     |  4 +++
 src/tests/functions/test-printstr1.out     |  1 +
 src/tests/graph/test-edge1.gvl             |  1 +
 src/tests/graph/test-edge2.gvl             | 15 ++++++++++
 src/tests/graph/test-edge2.out             |  4 +++
 src/tests/graph/test-edge3.gvl             | 35 ++++++++++++++++++++++
 src/tests/graph/test-edge3.out             | 16 ++++++++++
 src/tests/graph/test-edge4.gvl             | 23 ++++++++++++++++
 src/tests/graph/test-edge4.out             |  4 +++
 src/tests/graph/test-edge5.gvl             | 19 ++++++++++++
 src/tests/graph/test-edge5.out             |  3 ++
 src/tests/graph/test-graph1.gvl            |  9 ++++++
 src/tests/graph/test-graph1.out            |  0
 src/tests/graph/test-graph2.gvl            | 15 ++++++++++
 src/tests/graph/test-graph2.out            |  0
 src/tests/graph/test-graph3.gvl            | 26 ++++++++++++++++
 src/tests/graph/test-graph3.out            |  0
 src/tests/graph/test-node3.gvl             | 20 ++++++++++++
 src/tests/graph/test-node3.out             |  6 ++++
 src/tests/graph/test-node4.gvl             | 19 ++++++++++++
 src/tests/graph/test-node4.out             |  1 +
 src/tests/graph/test-node5.gvl             | 21 ++++++++++++++
 src/tests/graph/test-node5.out             |  6 ++++
 src/tests/graph/test-node6.gvl             | 13 ++++++++
 src/tests/graph/test-node6.out             |  3 ++
 src/tests/statements/test-for2.gvl         |  9 ++++++
 src/tests/statements/test-for2.out         | 10 +++++++
 35 files changed, 358 insertions(+), 11 deletions(-)

commit aecd89283d56c0a2bb8de1e52bb10146b775e0d5
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Sun Dec 5 11:46:59 2021 -0500

    move built in funcs to builtin.ml (#23)

    * move built in funcs to builtin.ml

    * modify set/get attributes in graph lib

    * modify float type in graph lib to double type

    * add get node attr built-in functions in builtin.ml

 src/builtin.ml                 | 43 ++++++++++++++++++++++++++++++++++++++++++
 src/codegen.ml                 | 11 +++++++++++-------
 src/lib/graph.c                | 26 +++++++++++++++++++++++++-
```

```
 src/lib/graph.h                | 26 ++++++++++++-------------
 src/semant.ml                  | 32 ++++++++++++++++----------------
 src/tests/graph/test-node2.gvl | 11 +++++++++++
 src/tests/graph/test-node2.out |  6 ++++++
 7 files changed, 87 insertions(+), 68 deletions(-)

commit a4be3aeea143a6e4ebe8f6fccea061404959fa78
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Dec 4 17:29:14 2021 -0800

    Added get and set functions for edge and node. (#22)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.

    * Automatical testing, credits to Prof. Stephen Edwards.

    * Added subdir in tests/ and some basic array.

    * Linked c files.

    * Added docker commands.

    * Added get and set functions for edge and node.

    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-116.dyn.columbia.edu>

 src/lib/graph.c | 119 ++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 src/lib/graph.h |  41 ++++++++++++++++++++-
 2 files changed, 157 insertions(+), 3 deletions(-)

commit 1a40a84489f31e43c015aebc2378fe47f722d825
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Sat Dec 4 20:23:19 2021 -0500

    add built-in functions in codegen (#21)

 src/codegen.ml                 | 31 +++++++++++++++++++++++++++++--
 src/semant.ml                  | 10 ++++++++--
 src/tests/graph/test-edge1.gvl |  9 +++++++++
 src/tests/graph/test-edge1.out |  0
 4 files changed, 46 insertions(+), 4 deletions(-)
```

```
commit 5b62aae9518270fcb3cd9ade9044b50c4ecf66b9
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Dec 4 16:12:19 2021 -0800

    Linked c files and added docker commands. (#20)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.

    * Automatical testing, credits to Prof. Stephen Edwards.

    * Added subdir in tests/ and some basic array.

    * Linked c files.

    * Added docker commands.

    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-116.dyn.columbia.edu>

 README.md                          | 16 +++++++++++++--
 src/Dockerfile                     |  2 ++
 src/Makefile                       |  8 ++++++--
 src/ast.ml                         |  3 ++-
 src/codegen.ml                     | 28 ++++++++++++++++++++++++++-
 src/semant.ml                      | 25 ++++++++++++++++++++++++---
 src/testall.sh                     |  2 +-
 src/tests/graph/fail-node1.err     |  1 +
 src/tests/graph/fail-node1.gvl     |  6 ++++++
 src/tests/graph/test-node1.gvl     |  5 +++++
 src/tests/graph/test-node1.out     |  0
 src/tests/statements/fail-if1.err |  2 +-
 12 files changed, 87 insertions(+), 11 deletions(-)

commit 605ad55fc120c201ddaf8b468c91f92c188e75ce
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Mon Nov 29 15:47:10 2021 -0500

    fix comment (#19)

    * fix comment
```

```
    * add test for comment

    * add test for comment

 .gitignore                          | 10 ++++++++++
 src/scanner.mll                     |  4 ++--
 src/tests/others/fail-comment1.err  |  1 +
 src/tests/others/fail-comment1.gvl  |  4 ++++
 src/tests/others/test-comment1.gvl  |  9 +++++++++
 src/tests/others/test-comment1.out  |  1 +
 6 files changed, 27 insertions(+), 2 deletions(-)

commit 0ae80f1c31487a0cbc693323a726bb59525a13c2
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Mon Nov 29 10:33:55 2021 -0500

    graph lib in c (#18)

    * add lib list

    * add graph library

 src/lib/graph.c | 190 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/lib/graph.h |  59 ++++++++++++++++++
 src/lib/list.c  | 162 ++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/lib/list.h  |  41 ++++++++++++
 src/lib/main.c  |  90 ++++++++++++++++++++++++++++
 5 files changed, 542 insertions(+)

commit 6f5ab0824d7c2ab651c63bb3a30ffaed9263e4f5
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sun Nov 28 22:30:40 2021 -0800

    Automatical testing, credits to Prof. Stephen Edwards. (#17)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.

    * Automatical testing, credits to Prof. Stephen Edwards.

    * Added subdir in tests/ and some basic array.

    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>
```

```
 README                                 |   1 -
 README.md                              |  32 +++---
 src/Makefile                           |  55 ++++++++++
 src/_tags                              |   8 ++
 src/printbig.c                         |  75 ++++++++++++++
 src/testall.sh                         | 198 ++++++++++++++++++++++++++++++++++++++
 src/tests/functions/fail-func1.err     |   1 +
 src/tests/functions/fail-func1.gvl     |   3 +
 src/tests/functions/fail-func2.err     |   1 +
 src/tests/functions/fail-func2.gvl     |   7 ++
 src/tests/functions/fail-func3.err     |   1 +
 src/tests/functions/fail-func3.gvl     |   7 ++
 src/tests/functions/test-func1.gvl     |   4 +
 src/tests/functions/test-func1.out     |   1 +
 src/tests/functions/test-func2.gvl     |   8 ++
 src/tests/functions/test-func2.out     |   1 +
 src/tests/functions/test-func3.gvl     |   8 ++
 src/tests/functions/test-func3.out     |   1 +
 src/tests/statements/fail-if1.err      |   1 +
 src/tests/statements/fail-if1.gvl      |   8 ++
 src/tests/statements/test-for1.gvl     |   7 ++
 src/tests/statements/test-for1.out     |  10 ++
 src/tests/statements/test-if1.gvl      |   8 ++
 src/tests/statements/test-if1.out      |   1 +
 src/tests/statements/test-if2.gvl      |   8 ++
 src/tests/statements/test-if2.out      |   1 +
 src/tests/statements/test-if3.gvl      |  16 +++
 src/tests/statements/test-if3.out      |   1 +
 src/tests/statements/test-while1.gvl   |   9 ++
 src/tests/statements/test-while1.out   |  10 ++
 src/tests/variables/test-var1.gvl      |   6 ++
 src/tests/variables/test-var1.out      |   1 +
 src/tests/variables/test-var2.gvl      |   6 ++
 src/tests/variables/test-var2.out      |   1 +
 34 files changed, 490 insertions(+), 16 deletions(-)

commit d557e8b85fc2edcf2868b8065e8dd7635f17d3fa
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sun Nov 28 13:01:44 2021 -0800

     if, while, and for statement. Different scope for varaible declaration. (#16)

    * Updated to hello world scanner and parser.

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    * Draw some bings.

    * Modified indentation.

    * if, while, and for statement. Different scope for varaible declaration.
```

```
    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>
    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-159.dyn.columbia.edu>

 README.md       | 18 +++++++++++++++-
 src/ast.ml      | 23 ++++++++-----------
 src/codegen.ml  | 47 ++++++++++++++++++++++++++++++++++----
 src/parser.mly  |  4 ++--
 src/sast.ml     | 17 ++++++++++++++-
 src/semant.ml   | 72 +++++++++++++++++++++++++++++++++++++++++++++++++++-------
 6 files changed, 151 insertions(+), 30 deletions(-)

commit 9488abe95df29582e48d1843e0b3cc28abdc091f
Merge: ad1e93a acc03c7
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Nov 27 14:34:17 2021 -0800

    Merge pull request #15 from yaxinchen666/iter1-funccall

    add function calls and lits

commit acc03c718b22c2665ad797ea8d6e52ebd1b7a8ff
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sat Nov 27 15:41:30 2021 +0800

    format

 src/codegen.ml |  6 +++---
 src/semant.ml  | 14 +++++++-------
 2 files changed, 10 insertions(+), 10 deletions(-)

commit f716f702aaff301845cd3175294eb6970e1918db
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sat Nov 27 15:33:38 2021 +0800

    format

 src/semant.ml | 42 +++++++++++++++++++++---------------------
 1 file changed, 21 insertions(+), 21 deletions(-)

commit e15c00da3f006e10303abe784101536b2c1b4c68
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sat Nov 27 15:22:56 2021 +0800

    format

 src/semant.ml | 30 +++++++++++++++---------------
 1 file changed, 15 insertions(+), 15 deletions(-)

commit eb0bead49638a9b751c78ac2a3dd05ee5f68e958
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sat Nov 27 15:16:09 2021 +0800
```

```
    format

 src/semant.ml | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)

commit 0a71f8d87eb66ae9b9b9751401f87f8cc1cf3f40
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sat Nov 27 15:08:41 2021 +0800

    add floatlit operation

 src/codegen.ml | 53 +++++++++++++++++++++++++++++++++++------------------
 src/semant.ml  |  7 ++++---
 2 files changed, 39 insertions(+), 21 deletions(-)

commit 3849ecd5db31415c8711bcb50bcc079ef343d66b
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sat Nov 27 14:53:24 2021 +0800

    add function calls and lits

 src/ast.ml      | 12 ++++++++++-
 src/codegen.ml  | 36 +++++++++++++++++++++++++++---
 src/parser.mly  |  6 ++---
 src/sast.ml     | 12 ++++++++++
 src/scanner.mll |  2 +-
 src/semant.ml   | 69 +++++++++++++++++++++++++++++++++++++++++++++++++++++----
 6 files changed, 125 insertions(+), 12 deletions(-)

commit ad1e93a79c368630294f63c26f0104e597624d48
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sun Nov 21 16:46:11 2021 -0800

    Implemented variable declaration functionality. (#14)

    * Added compile commands to README.md

    * Implemented variable declaration functionality.

    Co-authored-by: Mark Zhang <markzhang@dyn-129-236-228-46.dyn.columbia.edu>

 README.md      | 23 +++++++++++++++++++++
 src/Dockerfile | 32 ++++++++++++++++++++++++++++++
 src/codegen.ml | 65 +++++++++++++++++++++++++++++++++++++++++++++++++++++-------
 src/sast.ml    |  3 ++-
 src/semant.ml  | 62 ++++++++++++++++++++++++++++++++++++++++------------------
 5 files changed, 157 insertions(+), 28 deletions(-)

commit 05c28003b5503e784a81fa782c47bd7267eae566
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Wed Nov 10 22:33:42 2021 -0500

    expr for hello world (#12)
```

```
    * add node/edge extension

    * remove repetitive MINUS & NOT

    * adapt parser, ast, scanner for helloworld

    * hello world expr with test

    * remove test

 src/ast.ml     |  2 ++
 src/codegen.ml | 83 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/gvl.ml     | 32 ++++++++++++++++++++++
 src/sast.ml    | 51 ++++++++++++++++++++++++++++++++++++
 src/semant.ml  | 78 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 5 files changed, 246 insertions(+)

commit a0364573c98b6aed00a25d805580b3e68985a6bf
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Tue Nov 9 11:15:27 2021 -0500

    adapt parser, ast, scanner for helloworld (#11)

    * add node/edge extension

    * remove repetitive MINUS & NOT

    * adapt parser, ast, scanner for helloworld

 src/ast.ml      | 38 ++++++++++++++++------
 src/parser.mly  | 97 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/scanner.mll |  8 ++---
 3 files changed, 87 insertions(+), 56 deletions(-)

commit 4e7dfbd7dea3e1234a488e25d6f46e2d1e7d45fa
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Mon Nov 8 17:42:08 2021 -0500

    remove repetitive MINUS & NOT in parser (#10)

    * remove repetitive MINUS & NOT

 src/parser.mly | 4 +---
 1 file changed, 1 insertion(+), 3 deletions(-)

commit 26c11d818ef5f641319056be4db742c6c6d839b5
Author: jiawenyan3088 <71580036+jiawenyan3088@users.noreply.github.com>
Date:   Sun Nov 7 22:52:52 2021 -0500

    Add ast.ml & modify parser (#8)

    * add ast

    * modify parser
```

```
    * Update parser.mly

    * Update parser.mly

 src/ast.ml     | 87 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/parser.mly | 25 ++++++++++-------
 2 files changed, 102 insertions(+), 10 deletions(-)

commit f30f3ebe1cde4519b55977fd266eb59ada77cbb6
Merge: fd3174b 139683e
Author: AsterWei <53880314+AsterWei@users.noreply.github.com>
Date:   Sun Nov 7 22:08:35 2021 -0500

    Merge pull request #9 from MinheZhang/minhe

    Minhe good job!

commit 139683efa2abfa9728c89e1b35052d118a16ca8c
Merge: 03c97f7 fd3174b
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sun Nov 7 22:07:43 2021 -0500

    Merge branch 'main' into minhe

commit 03c97f744eeda59a13e19bd8020eaab820a60bd5
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sun Nov 7 21:33:40 2021 -0500

    helloworld-1.0 version scanner and parser.

 src/parser.mly  | 26 ++++++++++++++++++++--------
 src/scanner.mll |  7 ++-----
 2 files changed, 20 insertions(+), 13 deletions(-)

commit fd3174b92f67ae778b54e4bf8e4f6f43ad669bf5
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Oct 30 22:35:27 2021 -0400

    Added syntax sugar of graph manipulation.

    * Removed invalid type.

    * Added syntax sugar of graph manipulation.

    Co-authored-by: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
    Co-authored-by: Mark Zhang <markzhang@Marks-MBP.lan>

 src/parser.mly  | 5 +++++
 src/scanner.mll | 2 ++
 2 files changed, 7 insertions(+)

commit bb35c702001a254b8d1327b0ade3b022a39b94a1
Author: Mark Zhang <markzhang@Marks-MBP.lan>
```

```
Date:   Sat Oct 30 22:33:04 2021 -0400

    Added syntax sugar of graph manipulation.

 src/parser.mly  | 5 +++++
 src/scanner.mll | 2 ++
 2 files changed, 7 insertions(+)

commit e710027155089ed817745160ea3f9623b3b458c3
Merge: 04bdbbf 864055c
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sat Oct 30 22:28:16 2021 -0400

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL into minhe

commit 864055cf34df65339b0fbf3f6759a8a8211d2988
Author: yaxinchen666 <51110939+yaxinchen666@users.noreply.github.com>
Date:   Sat Oct 30 22:25:35 2021 -0400

    Add node/edge extension (#6)

    * add node/edge extension

    * add colon to scanner

 src/parser.mly  | 4 ++++
 src/scanner.mll | 1 +
 2 files changed, 5 insertions(+)

commit 04bdbbf775c86c223c06844852a321ff224e35a3
Merge: e3a41e1 4bbbbfa
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sat Oct 30 21:51:50 2021 -0400

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL into minhe

commit 4bbbbfa8710ca4ec04fbd5588e9ef498b1ebd9f0
Merge: 33384de 5cd0dcf
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sat Oct 30 21:49:53 2021 -0400

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL

commit 33384deedf815330a84951d095eef7fcde5c0be3
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sat Oct 30 21:49:34 2021 -0400

    Added assignment, negation, and mod expression.

 src/parser.mly | 16 ++++++++++++---
 1 file changed, 13 insertions(+), 3 deletions(-)

commit 5cd0dcf84fa0e90c07ad7bd113ebf5efdd2c3e5b
Merge: 746ff6d 64c0bd6
```

```
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Oct 30 21:34:42 2021 -0400

    Merge pull request #5 from MinheZhang/cyx

    Modified array declaration and initialization in parser.

commit 64c0bd63e00e49c85caf4673b8aa524069c7bf6a
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sun Oct 31 09:25:00 2021 +0800

    add semi in array decl

 src/parser.mly | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 9524e94cfe740a9f95e50153ff41acb66b227d64
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sun Oct 31 06:30:34 2021 +0800

    modify array declaration and initialization

 src/parser.mly | 14 ++++++++++++-
 1 file changed, 13 insertions(+), 1 deletion(-)

commit e3a41e1d0d9d79730ba95a36ef7737d0e688681a
Merge: b33b6b1 746ff6d
Author: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
Date:   Sat Oct 30 17:29:44 2021 -0400

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL into minhe

commit 746ff6dbfefb1ff6722c81f14092d9c596036c21
Author: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
Date:   Sat Oct 30 17:29:06 2021 -0400

    Removed emtpy return and semantic typo.

 src/parser.mly | 7 +++----
 1 file changed, 3 insertions(+), 4 deletions(-)

commit 13707560e0ab2d41d0f2fad8c67a0f8c6dd18f3d
Merge: 369c377 13c8b23
Author: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
Date:   Sat Oct 30 17:26:50 2021 -0400

    Merge branch 'main' of https://github.com/MinheZhang/COMSW4115-GVL

commit 369c377b4bc66885d9e54af06e9374bbb0701a37
Author: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
Date:   Sat Oct 30 17:26:41 2021 -0400

    Fixed semantic typo.
```

```
 src/parser.mly | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 13c8b234775b23b231198f06f79fc787c456bc2c
Merge: 8b37094 f4648e6
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Oct 30 17:17:50 2021 -0400

    Merge pull request #4 from jiawenyan3088/main

    Added return, break, and continue statement in parser.

commit f4648e686cddfd94c74de1569ab6ce002a3d1cde
Author: jiawenyan3088 <jy3088@columbia.edu>
Date:   Sat Oct 30 17:15:48 2021 -0400

    add statement

 src/parser.mly | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 347590b6bf8706e655bb725950e69fe0ea86a1ac
Author: jiawenyan3088 <jy3088@columbia.edu>
Date:   Sat Oct 30 16:49:41 2021 -0400

    add statment

 src/parser.mly | 16 ++++++++--------
 1 file changed, 8 insertions(+), 8 deletions(-)

commit 9e29bf0fc4ebf032d67b706693bdc1cdce01ac3b
Author: jiawenyan3088 <jy3088@columbia.edu>
Date:   Sat Oct 30 16:46:26 2021 -0400

    add statement

 src/parser.mly | 17 ++++++++++-------
 1 file changed, 10 insertions(+), 7 deletions(-)

commit b33b6b1034f2d955c3886fbe8aedc3152e3a85ad
Merge: bbf47e2 8b37094
Author: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
Date:   Sat Oct 30 15:47:04 2021 -0400

    Pulled main.

commit bbf47e2f6caa68dde057f2c73a937381b118720c
Author: Mark Zhang <markzhang@Marks-MacBook-Pro.local>
Date:   Sat Oct 30 15:42:12 2021 -0400

    Removed invalid type.

 src/parser.mly | 1 -
 1 file changed, 1 deletion(-)
```

```
commit 8b37094506a33082f3757f3554ac23bf02127068
Merge: 16e044a 8e16f3e
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Oct 30 15:40:41 2021 -0400

    Merge pull request #3 from MinheZhang/cyx

    add struct and array in parser

commit 8e16f3ef2acc7182cc07a4bc303e5809d487d5c4
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sun Oct 31 03:31:42 2021 +0800

    modify gitigore

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

commit 4d835543fb7bbfd4445398d1a5a5c8deb72569c1
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sun Oct 31 02:45:44 2021 +0800

    add struct and array in parser

    add struct

    remove parser.ml parser.mli

 .gitignore     |   5 +
 src/parser.ml  | 788 ----------------------------------------------------------
 src/parser.mli |  59 -----
 src/parser.mly |  16 +-
 4 files changed, 18 insertions(+), 850 deletions(-)

commit 16e044ab667fa71bc9cf6e6d8ec3b5efa5c54617
Merge: 4ffb9c4 6b12fef
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sat Oct 30 14:38:43 2021 -0400

    Merge pull request #2 from MinheZhang/aster

    Aster

commit 6b12fef8dcdfb62e97b50418d3ad712b3d6ce976
Author: AsterWei <aw3389@columbia.edu>
Date:   Sat Oct 30 14:36:37 2021 -0400

    EQ LEQ NEQ...

 src/parser.ml  | 788 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/parser.mli |  59 +++++
 src/parser.mly |  14 +
 3 files changed, 861 insertions(+)
```

```
commit 1c8eff14151f4bee9730d17b3433f86018ee5489
Merge: a32e8fe 68a1262
Author: AsterWei <aw3389@columbia.edu>
Date:   Sat Oct 30 12:39:04 2021 -0400

    Merge branch 'minhe' of github.com:MinheZhang/COMSW4115-GVL into aster

commit 68a1262b7e44d11733eaf5b0577abd132663058f
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sat Oct 30 12:08:16 2021 -0400

    MicroC style parser and changed SEMMI to SEMI.

 src/parser.mly  | 87 +++++++++++++++++++++++++++++++-----------------------
 src/scanner.mll |  3 +-
 2 files changed, 53 insertions(+), 37 deletions(-)

commit a32e8fe8aa654a041b5699b9ffd55a6269f59fd3
Author: AsterWei <aw3389@columbia.edu>
Date:   Sat Oct 30 11:25:55 2021 -0400

    test

 README | 1 +
 1 file changed, 1 insertion(+)

commit 38464f797fbcb482239ee47f323d6bebeebd7fd0
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Sat Oct 30 10:58:54 2021 -0400

    Added program entrance.

 src/parser.mly | 6 ++++--
 1 file changed, 4 insertions(+), 2 deletions(-)

commit 4ffb9c465dbee59eda4223dfdadf48dd8ac00d60
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Fri Oct 29 21:55:09 2021 -0400

    Added compile commmand in README.md.

 README.md | 14 ++++++++++++++
 1 file changed, 14 insertions(+)

commit e7e1bb8d5222a33e04df5dbc2651b6ebdbd1487b
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Fri Oct 29 21:12:23 2021 -0400

    Parser draft 2.0.

 src/parser.mly | 79 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 62 insertions(+), 17 deletions(-)
```

```
commit 8dcf3b10fc48257252a0172be1426214c8a7031f
Author: Mark Zhang <markzhang@Marks-MBP.lan>
Date:   Fri Oct 29 11:14:33 2021 -0400

    Added mod op into scanner. Trivial version parser.

 src/parser.mly  | 52 ++++++++++++++++++++++++++++++++++++++++++++++++++++
 src/scanner.mll |  1 +
 2 files changed, 53 insertions(+)

commit 1c1c832b452122d8dc62e87749bdaa734a4fd5af
Merge: 6ff4174 71fa791
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Sun Oct 24 11:54:58 2021 -0400

    Merge pull request #1 from MinheZhang/init_scanner

    scanner

commit 71fa7916802394cf97a221d27505f2d0866728f3
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sun Oct 24 23:53:30 2021 +0800

    add some tokens

 src/scanner.mll | 11 ++++++++++-
 1 file changed, 10 insertions(+), 1 deletion(-)

commit 4756b160199472a389fe896faec95fa0bcbf7c0e
Author: yaxin <leepace666666@sjtu.edu.cn>
Date:   Sun Oct 24 16:26:38 2021 +0800

    scanner

 src/scanner.mll | 96 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 96 insertions(+)

commit 6ff41748838d7a13b91c6adb0f05880a340719b1
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Mon Oct 18 23:34:32 2021 -0400

    Changed repo name.

 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 9c4e69f1b3364b8481e178877a1a7345ce938a10
Author: MinheZhang <30498459+MinheZhang@users.noreply.github.com>
Date:   Mon Oct 18 23:33:30 2021 -0400

    Initial commit

 .gitignore | 29 +++++++++++++++++++++++++++++
 README.md  |  2 ++
```

```
 2 files changed, 31 insertions(+)
```

8.3 Testing Script

```
#!/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.
GVL="./gvl.native"
#MICROC="_build/microc.native"

# Set time limit for all operations
ulimit -t 30

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

keep=0

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

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

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

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

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

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

Check() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                             s/.gvl//'`
    reffile=`echo $1 | sed 's/.gvl$//'`
    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
    ↪  ${basename}.out" &&
    Run "$GVL" "$1" ">" "${basename}.ll" &&
    Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s" &&
    Run "$CC" "-o" "${basename}.exe" "${basename}.s" "printbig.o" "list.o"
    ↪  "graph.o" "graph_visualization.o" " -lGL -lGLU -lglfw -lX11 -lXxf86vm
    ↪  -lXrandr -lpthread -lXi -lglad -ldl -lm" &&
    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/.gvl//'`
    reffile=`echo $1 | sed 's/.gvl$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

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

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$GVL" "<" $1 "2>" "${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 [ ! -f printbig.o ]
then
    echo "Could not find printbig.o"
    echo "Try \"make printbig.o\""
    exit 1
fi

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